UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXXVI Ciclo

Advanced Synchronization and Estimation Techniques for Modern Satellite Communications

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Alessandro Ugolini

Chiar.mo Prof. Tommaso Foggi

Dottorando: *Marco Morini*

Anni 2020 - 2023

Do not go where the path may lead,
go instead where there is no path
and leave a trail

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACM adaptive coding and modulation

ADC analog-to-digital

AIR achievable information rate

APSK amplitude phase-shift keying

ASIC application specific integrated circuit

AWGN additive white Gaussian noise

BCH Bose-Chaudhuri-Hocquenghem

BER bit error rate

BPSK binary phase-shift keying

CRC cyclic redundancy check

CRLB Cramér-Rao lower bound

DA data aided

DD decision directed

DER decision error rate

DTH direct-to-home

DVB digital video broadcasting

DVB-S digital video broadcasting for satellite

DVB-S2 digital video broadcasting for satellite, 2nd generation

DVB-S2X	digital video broadcasting for satellite, 2nd generation extensions
FEC	forward error correction
FPGA	field programmable gate array
FWL	fractional word length
HPA	high power amplifiers
IBO	input back-off
IMUX	input multiplexer
IR	information rate
IWL	integer word length
LDPC	low density parity check
LUT	look-up table
MF	matched filter
ModCod	modulation and coding
MSE	mean squared error
NCO	numerically controlled oscillator
NCPDI	non-coherent post-detection integration
NDA	non data aided
NTN	non-terrestrial network
OMUX	output multiplexer
PDF	probability density function
PDI	post-detection integration
PER	packet error rate
PLH	physical layer header
PLL	phase locked loop
PN	phase noise

PPM	part-per-million
PSD	power spectral density
PSK	phase shift keying
QAM	quadrature amplitude modulation
QEF	quasi error free
QPSK	quadrature phase-shift keying
RRC	root-raised cosine
SCO	sampling clock offset
SNR	signal-to-noise ratio
SoF	start of frame
STR	symbol timing recovery
TED	timing error detector
VL-SNR	very-low signal-to-noise ratio
WL	word length

Introduction

Satellite communications have become an integral part of our modern world, revolutionizing the way we connect, share information, and access services on a global scale. The ubiquity of satellite in contemporary society is evident in their multifaceted applications, including telecommunications, remote sensing, localization, navigation and beyond. Satellite networks have also been fully integrated in the 5G infrastructure, to complement and extend the terrestrial network. The introduction of non-terrestrial networks (NTNs), of which satellite communications is an essential part, into 5G, will ensure higher connectivity and provide internet access also to parts of the planet that are unserved or underserved by the terrestrial segment of the global communication system, such as islands or polar regions, as well as to ships and planes located far from the coastlines. The work presented in this thesis is focused on the design of novel algorithms for synchronization and parameter estimation in satellite communications. In each chapter, we will focus the attention on different aspects and channel impairments that can affect the received signal, proposing new solutions that can be easily adopted in different communication standards.

In Chapter 1, we first introduce the architecture of the second generation of the digital video broadcasting for satellite (DVB-S2X), as it will be used as reference scenario in this thesis. This chapter serves as a background to describe the general case study in which we will apply our solutions in the next chapters. However, the solutions and results presented in each chapter can be generalized and applied to other communication standards and applications.

In Chapter 2, we address the problem of the decoding of the physical layer header

(PLH), proposing and comparing novel strategies that can be easily applied into the reference scenario of the DVB-S2X standard. In particular, the study is focused on the design of algorithms to detect variable length PLHs, as such PLHs represent a more flexible solutions with respect to that proposed by the standard, which assumes a fixed length PLH.

In Chapter 3, we first introduce the problem of symbol timing synchronization, then we propose a novel low-complexity parallel architecture, which is suitable for the practical implementation in hardware of high data rates digital receivers. The performance of the proposed scheme is compared with that achievable employing classical methods for timing recovery, in order to evaluate the feasibility of the proposed algorithm.

Chapter 4 focuses on the analysis of phase noise (PN) effects in high order modulations. In fact, phase noise is one of the most critical impairments of communication systems, especially when the modulation size increases. As we did for the timing recovery, we propose a parallel algorithm, which has a simple structure, that can be employed after the timing synchronization block presented in Chapter 3 in high speed receivers.

Finally, in Chapter 5, we consider a typical nonlinear satellite channel. We propose a method to construct new constellations that are robust to nonlinear effects. The proposed constellations can be described by a simple analytical model, and they depend on a single parameters, unlike other solutions in the literature. Finally, we evaluate the performance of the proposed constellations with different metrics, and we compare it with other possible approaches.

Chapter 1

Background

1.1 DVB-S2 and its extension DVB-S2X

The second generation specification of the digital video broadcasting for satellite (DVB-S2) [1] was developed in 2003 with the aim of improving the system performance of existent DVB-S systems [2], while keeping a reasonable receiver complexity. DVB-S2 provides a near Shannon limit performance adopting forward error correction (FEC) techniques based on low-density parity-check (LDPC) codes in combination with Bose-Chaudhuri-Hocquenghem (BCH) codes and high order phase-shift keying (PSK) and amplitude phase-shift keying (APSK) modulations (8PSK, 16APSK and 32APSK) [1]. This results in a 30% capacity increase over the previous standard DVB-S under the same transmission conditions.

One of the main features in DVB-S2 is the adoption of adaptive coding and modulation (ACM), that allows an optimization of the transmission parameters for each individual user, depending on the path conditions [3]. Different combinations of modulation and coding (ModCod) can be used, going from the most robust, offering low spectral efficiency but robustness to fading, to the most efficient, offering very high spectral efficiency but poor protection against fading transmission errors.

The satellite world has changed significantly since the DVB-S2 standard was developed in 2003. Higher speeds, more efficient satellite communications technology, and wider transponders are required to support the exchange of large and increasing volumes of data, video, and voice over satellite. The highest demand for the extension of the DVB-S2 standard comes from video contribution and high speed internet applications, as these services are affected the most by the increased data rates. The need to have more resources in order to improve the existent services and to support new technologies, led to the development of a new standard. In March 2014, the Digital Video Broadcasting (DVB) group released the specifications of DVB-S2 eXtension, called DVB-S2X [4].

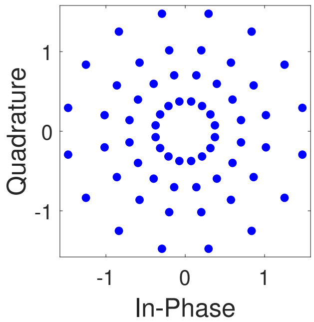
The main technologies involved in this new standard are manifold: smaller roll off values of the transmission root-raised cosine (RRC) shaping pulses (5%, 10%, 15%), smaller carrier spacing, and advanced filtering solutions can be adopted to exploit the bandwidth in a more efficient manner. New modulation and coding schemes have been introduced, as the ones proposed in [5], which improve the signal-to-noise ratio (SNR) granularity, achieving a higher spectral efficiency with respect to the DVB-S2 standard [6]. The large number of available ModCods makes the use of ACM even more effective than DVB-S2.

These new constellations have an increased number of points up to 256 (see Fig. 1.1), and they are particularly suitable for professional applications that work with improved link budgets. The combination of technologies incorporated in the new standard results in a gain of up to 20% for direct-to-home (DTH) networks and 51% for other professionals applications compared to DVB-S2 [7].

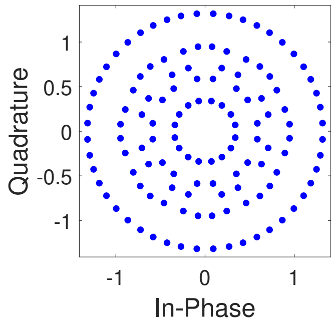
Finally, new ModCods for very-low signal-to-noise ratio (VL-SNR) values have been added in order for satellite networks to deal with heavy atmospheric fading and to enable the usage of smaller antennas for mobile applications. The header of these ModCods has been modified with an extended physical layer header with better error correcting capabilities for signal-to-noise ratio values as low as -10 dB.

1.2 DVB-S2X Architecture

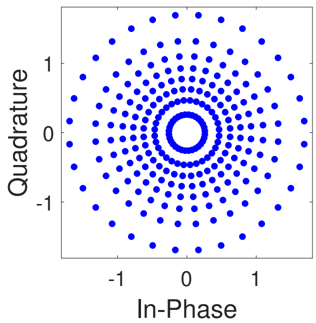
Due to the worldwide adoption of the DVB-S2, DVB-S2X maintains the same architecture as DVB-S2 to the extent possible. The DVB-S2X transmission system



(a) 64APSK, code rate 128/180.



(b) 128APSK, code rate 135/180.



(c) 256APSK, code rate 116/180.

Figure 1.1: Examples of DVB-S2X constellations.

is illustrated in Fig. 1.2 [4]. The block identified as mode adaptation is application dependent. It should provide input stream interfacing, synchronization, null packet deletion, cyclic redundancy check (CRC) coding for error detection at the receiver at packet level and, in case of multiple streams, it must merge them into a single transmission signal.

The formation of the physical layer frame is reported in Fig. 1.3 [4]. The baseband frame is built by appending the baseband header in front of the data block, to notify the receiver the input stream format and mode adaption type. In case the user

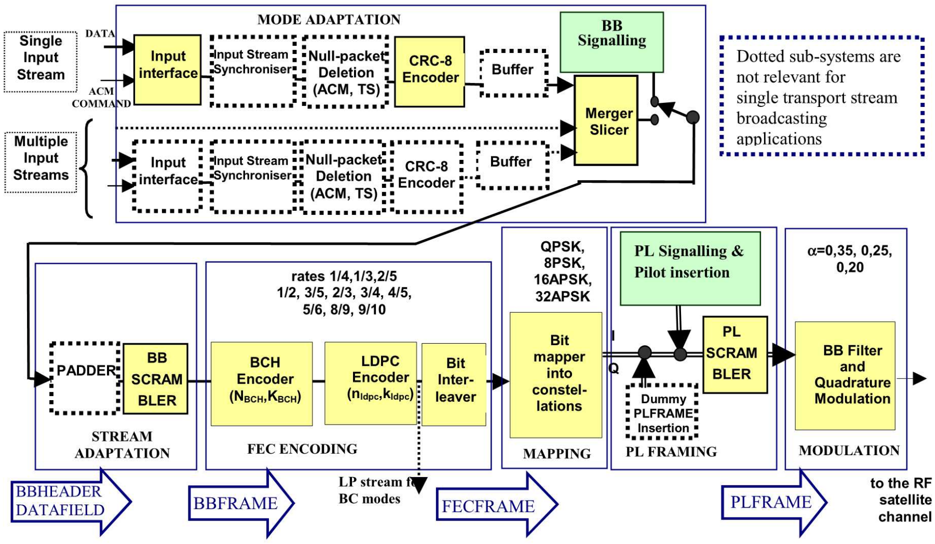


Figure 1.2: DVB-S2X transmission scheme.

data available for transmission are not sufficient to completely fill a baseband frame, padding bits are inserted to complete it. Baseband scrambling is also provided.

FEC encoding is then applied to the baseband frame to obtain the FEC frame. The forward error correction, together with the modulation, is the key subsystem to achieve excellent performance by satellite in the presence of high levels of noise and interference. FEC encoding carries out the concatenation of BCH outer code and LDPC inner codes. Depending on the application area, the FEC frames can have length 64 800 (normal frame) or 16 200 (short frame) bits. For VL-SNR applications, an additional type of packet with length 32 400 bits is available.

Then, the coded bits are mapped into DVB-S2X constellations to get a complex frame. The adopted constellation depends on the channel conditions and on the type of service. By selecting the constellation and code rates, different spectral efficiencies can be achieved depending on the application area. For example, quadrature phase-shift keying (QPSK) could be used in VL-SNR applications, 8APSK for broadcasting applications, while 256APSK for professional applications with very good channel

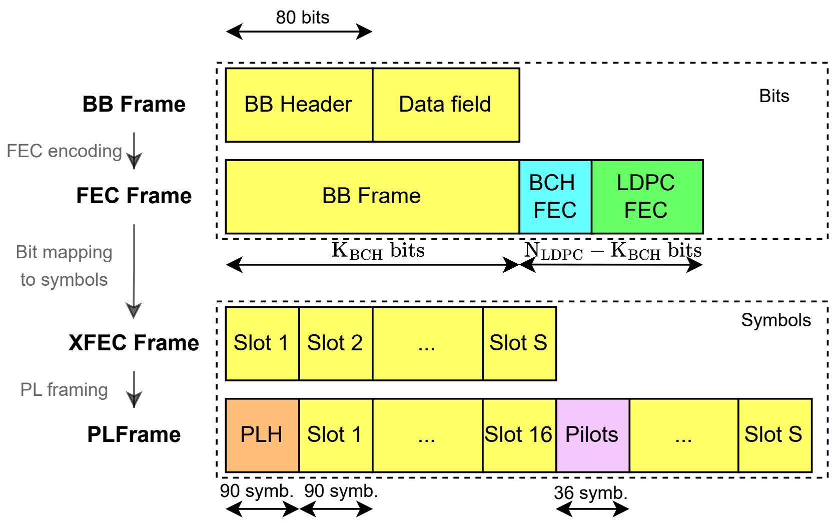


Figure 1.3: DVB-S2X physical layer frame.

conditions.

The physical layer signal is composed of a regular sequence of frames. Every frame is composed of a payload of 64 800 (normal FEC frame), 32 400 (medium FEC frame) or 16 200 (short FEC frame) bits, corresponding to a FEC code block. In order to allow the receiver to perform synchronization and detect the ModCod parameters before demodulation and FEC decoding, a physical layer header of 90 binary modulated symbols is placed before the payload, as can be seen from Fig. 1.3. Since the PLH is the first entity to be decoded by the receiver, it could not be protected by the FEC scheme. On the other hand, it has to be perfectly decoded under the worst case link conditions.

The first 26 symbols of the physical layer header identify the start of the physical layer frame (SoF), while the remaining 64 identify the PLH code and contain information on frame length, presence of pilot symbols and on the adopted ModCod scheme. A representation of the DVB-S2X frame structure is reported in Fig. 1.4, where the division of the header in SoF and PLH code is highlighted. After decoding the PLH, the receiver can derive, through the knowledge of the transmissions param-

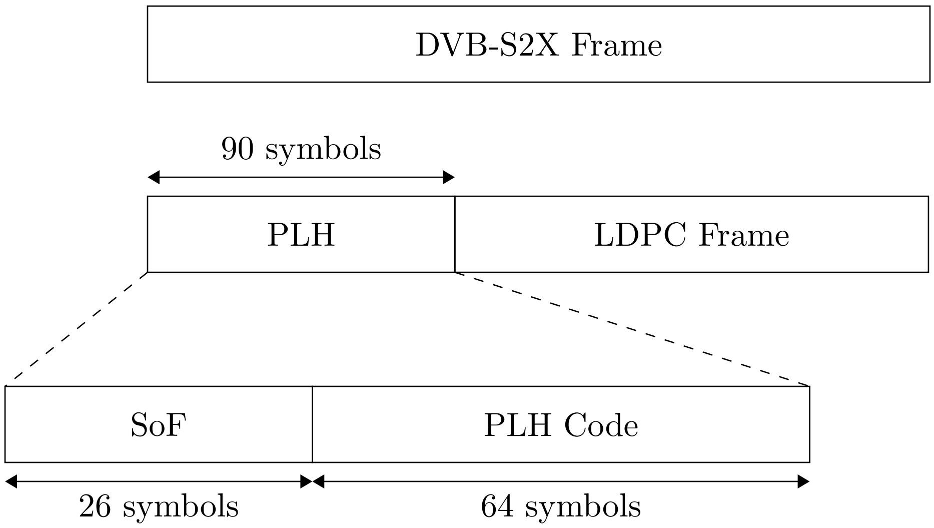


Figure 1.4: Frame structure for the DVB-S2X standard.

eters, the current frame length and thus the start of the following frame, even if the channel conditions do not allow for successful data decoding in the current frame.

1.3 Synchronization in DVB-S2X

Frame synchronization refers to the process of aligning and synchronizing the receiver with the transmitted signal, so that it can correctly decode and demodulate the data contained within the signal frames. As we have explained in the previous section, DVB-S2X frames begin with a known preamble and a frame header. The preamble is a specific bit sequence inserted at the beginning of each frame, which is used by the receiver to detect the start of a new frame.

The adoption of ACM in DVB-S2X results in a scenario where the effective SNR may become very low [8]. In fact, the standard supports various modulation schemes, ranging from QPSK to 256APSK. The receiver does not know which modulation scheme has been used at the transmitter. This information can only be accessed after successful frame synchronization, by analyzing the PLH signaling information.

This emphasizes the critical importance of synchronization and parameters estimation, which is a fundamental operation to guarantee a successful communication. It turns out that frame synchronization is a critical step, especially if it has to be performed at low SNR values and under carrier frequency offsets, which can be even up to 20% of the symbol rate [9]. DVB-S2X frames include pilot fields that can be exploited for synchronization purposes [4]. Therefore, an accurate detection of the start of the frame, along with the pilot fields, is crucial for all the subsequent synchronization blocks. Once the frame synchronization is achieved, the position of the PLH can be correctly determined, and the receiver can access the information contained in the frame header.

One of the most popular strategies to achieve frame synchronization is post-detection integration (PDI) [10, 11, 12, 13]. In case frame synchronization is performed prior the complete recovery of the carrier frequency, i.e. under a residual carrier frequency offset, noncoherent post-detection integration (NCPDI) shall be considered [14, 15, 16].

Once the frame is synchronized, the receiver is aligned with the transmitted data frames. In fact, the primary objective of frame synchronization is to ensure that the receiver knows the start of each frame, allowing it to correctly decode and demodulate the received signal. However, some timing misalignments may still exist even after frame synchronization. This residual timing error is, typically, smaller than the symbol time. However, especially for high symbol rates, the symbol time is relatively short, therefore even a small timing error represents a significant fraction of the symbol time.

In the next chapters, we will investigate two important synchronization aspects that take place after frame synchronization has been performed. In Chapter 2 we will address the problem of PLH detection, while in Chapter 3 the problem of symbol timing synchronization will be analyzed.

Chapter 2

Frame Header Detection in Satellite Systems

In the DVB-S2X standard [4], the accurate detection of the physical layer header is a critical process to enable the successful demodulation and decoding of the received signal. In fact, the PLH carries essential information about the transmission, such as the modulation scheme, coding rate, frame length, and other configuration parameters. Detection of the physical layer header ensures that the receiver correctly interprets these parameters, enabling the demodulation and the decoding of the received payload. Typically, PLH codes have short length, as, for example, the 64-symbols PLH codewords included in the DVB-S2X standard [4]. The standard foresees the use of a constant length PLH, but some advantages can be obtained by adopting variable length codes for the PLH [17, 18, 19, 20]. This, however, makes the decoding of the PLH field especially challenging because the receiver needs to also estimate the length of the received PLH.

We will focus our analysis on the decoding of short codes robust to noncoherent detection. The design of such codes has been object of several research works, see for example [21, 22, 23, 24]. We focus on noncoherent detection since the received signal can be affected by unknown phase offsets. The unknown phase can be estimated by means of pilot symbols, which are distributed in blocks at regular intervals in the frame [4]. However, a variation of this effect over time can result in an imprecise pilot-based estimation, which would negatively affect the PLH decoding performance.

In this chapter, we will analyze the problem of PLH decoding in the DVB-S2X scenario. We first describe an alternative approach to design variable length codes robust to noncoherent detection, based on the work proposed in [24], and we apply it to the DVB-S2X case study. Then, we develop some novel techniques that can be applied in the decoding of the PLH field with unknown length, which can be used as alternatives to those provided in the standard.

The chapter is structured as follows. In Section 2.1, we define the model for the decoding of variable length PLH codes. In Sections 2.2 and 2.3, we give some details on the PLH code specified in the DVB-S2X standard and on the adopted PLH code design, respectively. In Section 2.4, we derive the novel decoding strategies, while in Section 2.5, we present the corresponding numerical results.

2.1 System Model

In this section, we address the problem of the decoding of a variable length PLH. The use of a variable length PLH perfectly fits into the scenario of the DVB-S2X standard, where adaptive coding and modulation is employed. As previously mentioned, the PLH in DVB-S2X standard has constant length. However, a more flexible solution could be to use a variable length PLH depending on the adopted modulation. In fact, when using ModCods operating at high SNR, the PLH length may be reduced with respect to the length foreseen by the standard. In the following, we introduce the system model that we considered in our analysis.

The system considers the transmission of a ModCod with associated a PLH field of length L, belonging to a finite set of lengths, as specified in Table 2.1, where the first column represents the SNR convergence threshold of the LDPC code for each ModCod, as specified in the DVB-S2X standard [4] (see Appendix A for details). The details of the selected code design will be reported in the next section, while the meaning of the third and fourth columns of Table 2.1 will be clear at the end of this

Table 2.1: PLH code details.

SNR convergence threshold [dB]	PLH Length L	N_L	$P(L)$
-2.85	64	1	0.0256
-2.03	58	1	0.0256
[0.22, 4.73]	48	3	0.0769
[5.13, 19.57]	26	34	0.8718

section.

Let us define by $L_{\max}$ the maximum value that L can assume. It is always possible to consider the transmission of a vector of symbols with length $L_{\max}$, regardless of the true value of L. The vector can thus be represented as divided in two distinct parts: the first one, indicated with subscript 1, with length L, contains the PLH symbols, while the second one, indicated with subscript 2, with length $\Delta L = L_{\max} - L$, contains unknown information symbols, which are drawn from a constellation with cardinality M, which depends on the chosen ModCod.

Let us assume to transmit a vector of symbols $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2]^T$ with length $L_{\max}$, such that $\boldsymbol{c}_1$ contains the PLH symbols and $\boldsymbol{c}_2$ contains randomly generated symbols of an unknown constellation. We can define the received vector as

$$\boldsymbol{r} = A\boldsymbol{c}e^{j\theta} + \boldsymbol{n}\,, \tag{2.1}$$

where θ is an unknown channel phase, assumed constant and uniformly distributed in $[0, 2\pi)$, A is the unknown amplitude and $\boldsymbol{n}$ is a vector of independent and identically distributed complex Gaussian noise samples, with variance σ^2 per component. This is the scenario specified in the DVB-S2X implementation guidelines for broadcast applications [4].

All LDPC ModCods are considered equally likely. Since the number of ModCods associated to an arbitrary length L is, in general, different, the probability to choose a PLH with length L can take different values. Denoting by N the number of ModCods, the probability to select a PLH of length L is

$$P(L) = \frac{N_L}{N}, \tag{2.2}$$

where N_L is the number of possible ModCods having length L. For the DVB-S2X standard and the adopted PLH code, the values of N_L and $P(L)$ are reported in Table 2.1, while $N = 39$.

2.2 Standard Fixed Length PLH Code

In the DVB-S2X standard, a new physical layer header has been designed extending the signaling from 7 bits to 8 bits. This has enabled the definition of more modulation and coding schemes, allowing a finer SNR granularity as well as the introduction of additional ModCods optimized for the linear channel.

The DVB-S2 PLH code is extended in such a way that the codewords corresponding to the original DVB-S2 ModCods remain unchanged. In other words, indicating with (n,k) a code with k input bits and n output bits, the $(64,7)$ PLH code of DVB-S2 is a subset of the $(64,8)$ PLH code of DVB-S2X. For the case of DVB-S2 ModCods, 64 symbols PLH codewords are transmitted in the same manner as in DVB-S2. For the new ModCods unique to DVB-S2X, on the other hand, a phase jump of $\pi/2$ is introduced between the transmission of the SoF field and PLH codewords. As a result of the aforementioned design, the $(64,8)$ PLH code of DVB-S2X has almost identical performance to the $(64,7)$ PLH code of DVB-S2 while keeping a backward compatible signaling for DVB-S2 ModCods [4].

By introducing an increased granularity, the highest resolution for optimal modulation in all circumstances can be ensured, and the service provider can further optimize the satellite link depending on the application. In combination with ACM and the introduction of high order modulations (e.g. 64, 128 and 256APSK), this translates in better performance with respect to DVB-S2 standard when specific SNR targets have to be met [7].

As discussed in Section 1.2, each packet carries a payload protected by a forward error correction encoding scheme, that is a concatenation of LDPC and BCH codes.

Information on coding and modulation, contained in the PLH, are encoded and sent before each frame. PLH also contains synchronization and signaling information, to allow the receiver to synchronize and detect the modulation and coding parameters before demodulation and FEC decoding. Hence, the PLH has to be perfectly decodable under the worst case link conditions, in order to reliably decode the transmitted information.

Each of the transmitted frames includes a 90-symbols PLH field which comprises a 26-symbols SoF sequence. The entire PLH (including SoF) is modulated into 90 $\pi/2$ binary phase-shift keying (BPSK) symbols. PLH code carries 1+7 signaling bits denoted as $(b_0, b_1 \cdots b_7)$, where b_0 is the most significant bit and b_7 is the least significant bit. The most significant bit indicates whether the PLH refers to regular DVB-S2 ModCods ($b_0 = 0$) or to ModCods defined in DVB-S2X standard ($b_0 = 1$). In the latter case, $(b_0, b_1 \cdots b_6)$ shall represent the additional DVB-S2X ModCods and the corresponding FEC length (normal, short or medium), while b_7 shall indicate the presence or absence of pilot symbols.

The 8-bit header field shall be encoded with a $(64, 8)$ code. The 7 most significant bits $(b_0, b_1 \cdots b_6)$ of the header field shall be encoded by a linear block code of length 32 with the following generator matrix.

$$\boldsymbol{G} = \begin{bmatrix} 10010000101011000010110111011101 \\ 01010101010101010101010101010101 \\ 00110011001100110011001100110011 \\ 00001111000011110000111100001111 \\ 00000000111111110000000011111111 \\ 00000000000000001111111111111111 \\ 11111111111111111111111111111111 \end{bmatrix}$$

The most significant bit of the 8-bit header field is multiplied with the first row of the matrix, the following with the second row, and so on. The 32 coded bits are denoted as $(y_1, y_2, \ldots y_{32})$. When $b_7 = 0$, the final PLH code is generated as $(y_1 y_1 y_2 y_2 \ldots y_{32} y_{32})$. When $b_7 = 1$, the final PLH code is generated as $(y_1 \bar{y}_1 y_2 \bar{y}_2 \ldots y_{32} \bar{y}_{32})$. The final PLH codeword is then scrambled with a 64 bits sequence to improve its autocorrelation property. Finally, independently of the modulation scheme in the physical layer frame payload, the 90 binary symbols forming the physical layer header are $\pi/2$ BPSK modulated.

2.3 Adopted Variable Length PLH Code

We used the method presented in [24] to generate short codes that are robust to noncoherent detection for BPSK. We generate codes with length L, code size K and minimum distance $d_{\min}$. The main idea behind the proposed algorithm is to add, at each step, a new column to the generator matrix $\boldsymbol{G}$, with dimension $K \times L$, such that the minimum distance $d_{\min}$ of the resulting new code is increased as much as possible. At each step, the column added to the generator matrix is checked to be different from all the existing columns. Note that this condition is not essential, but it guarantees that the optimal code, that is the maximum length code, is obtained when $L = 2^K - 1$. The algorithm stops when a target $d_{\min}$ is achieved.

We associate, to each of the DVB-S2X ModCods, PLH codewords whose lengths are reported in the second column of Table 2.1, with the aim of reducing the length of the PLH field with respect to the standard. We try to reduce as much as possible the code length by selecting the minimal length code achieving the desired decision error rate at a target SNR.

In the design problem for PLH, given the desired error probability and the channel SNR, one needs to construct the code with the shortest possible length. Therefore, a relation between these parameters is needed in order to correctly design the code. It can be demonstrated (see for example [25]) that a conservative upper bound for the error probability P_{E} when using noncoherent detection is given by

$$P_{\mathrm{E}} \leq \left(2^K - 1\right) \frac{1}{2} \mathrm{erfc}\left(\sqrt{A d_{\min} \frac{E_{\mathrm{s}}}{N_0}}\right), \tag{2.3}$$

with $A = 1$ and $A = 1/2$ for BPSK and QPSK, respectively. This bound provides a relation between the desired word error probability, minimum distance and SNR of the channel.

In our considered scenario, we need $K = 7$ bits to distinguish among all the codewords. Assuming a target decision error rate (DER) equal to 10^{-5} and considering

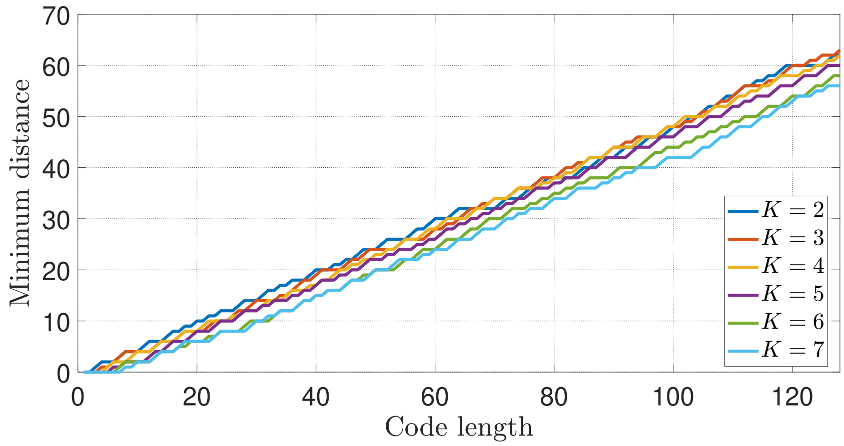


Figure 2.1: Minimum distance growth as a function of code length L for several values of code size K.

that in the worst case condition $\frac{E_s}{N_0} = -2.85$ dB (see Appendix A), one can obtain a lower bound for the minimum distance to be $d_{\min} = 27$. The corresponding PLH code with $K = 7$ and $d_{\min} = 27$ has length $L = 67$.

In Fig. 2.1 we report the minimum distance growth as a function of K and L for the generated codes. Clearly, the figure provides a lower bound on the minimum distance of the codes for a given pair $[L, K]$. Note that the codes with parameters reported in Fig. 2.1 may not be the optimal codes that the algorithm can generate. In fact, each run of the algorithm yields different results. Thus, one can run the algorithm many times and pick the best obtained code for a given pair $[K, d_{\min}]$.

2.4 Proposed PLH Decoding Strategies

In this section, we propose three alternative strategies for the decoding of a variable length PLH code as the one described in Section 2.3. The decoding strategy, when both the PLH and its length are unknown, can be based on the generalized likelihood criterion

$$\begin{aligned}\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}\right) &= \arg\max_{(A,L,\boldsymbol{c})} P(\boldsymbol{c},L,A|\boldsymbol{r}) \\ &= \arg\max_{(A,L,\boldsymbol{c})} \frac{p(\boldsymbol{r}|\boldsymbol{c},L,A)P(\boldsymbol{c}|L)P(L)p(A)}{p(\boldsymbol{r})}\,,\end{aligned} \tag{2.4}$$

Equation (2.4) gives an estimation of both the PLH and, implicitly, its length L. It also provides an estimation of the amplitude A. Clearly, it is

$$P(\boldsymbol{c}|L) = \begin{cases} \frac{1}{N_L} & \text{if } \boldsymbol{c} \text{ is a codeword of length } L \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.5}$$

Hence, with $\boldsymbol{c}$ a codeword of length L,

$$P(\boldsymbol{c}|L)P(L) = \frac{1}{N_L}\frac{N_L}{N} = \frac{1}{N}\,, \tag{2.6}$$

and, because N does not depend on L, assuming also that A is uniformly distributed, the decoding strategy in (2.4) simplifies to

$$\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}\right) = \arg\max_{(A,L,\boldsymbol{c})} \frac{p(\boldsymbol{r}|\boldsymbol{c},L,A)}{p(\boldsymbol{r})}\,. \tag{2.7}$$

Since we are considering vectors of equal length $L_{\max}$, $p(\boldsymbol{r})$ is irrelevant in the maximization and it can be ignored. Moreover, considering that PLH symbols only appear in the first part of vector $\boldsymbol{c}$, which we denoted by $\boldsymbol{c}_1$, the generalized likelihood function can be rewritten as

$$\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}_1\right) = \arg\max_{(A,L,\boldsymbol{c}_1)} p(\boldsymbol{r}|\boldsymbol{c}_1,L,A)\,. \tag{2.8}$$

By exploiting the fact that the vector $\boldsymbol{r}$ can be divided in two parts, the probability density function (PDF) in (2.8) can be expressed as

$$\begin{aligned}p(\boldsymbol{r}|\boldsymbol{c}_1,L,A) &= p(\boldsymbol{r}_1,\boldsymbol{r}_2|\boldsymbol{c}_1,L,A) \\ &= \left(\frac{1}{M}\right)^{\Delta L} \sum_{\boldsymbol{c}_2} p(\boldsymbol{r}_1,\boldsymbol{r}_2|\boldsymbol{c}_1,\boldsymbol{c}_2,L,A) \\ &= \left(\frac{1}{M}\right)^{\Delta L} \frac{1}{2\pi} \int \left[\sum_{\boldsymbol{c}_2} p(\boldsymbol{r}_1,\boldsymbol{r}_2|\boldsymbol{c}_1,\boldsymbol{c}_2,L,A,\theta)\right] \mathrm{d}\theta\,,\end{aligned} \tag{2.9}$$

where we have exploited the average with respect to $\boldsymbol{c}_2$, that is necessary to obtain the marginal PDF in (2.8), and the average on the unknown phase θ. Given the fact that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are uncorrelated, it is possible to write

$$p(\boldsymbol{r}_1,\boldsymbol{r}_2|\boldsymbol{c}_1,\boldsymbol{c}_2,L,A,\theta) = p(\boldsymbol{r}_1|\boldsymbol{c}_1,L,A,\theta)p(\boldsymbol{r}_2|\boldsymbol{c}_2,L,A,\theta)\,. \tag{2.10}$$

Hence, the two PDFs in (2.10) can be computed separately. Since

$$p\left(r_k \mid c_k,A,\theta\right) = \left(\frac{1}{2\pi\sigma^2}\right)\exp\left\{-\frac{1}{2\sigma^2}\left|r_k\right|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\left|c_k\right|^2+\frac{A}{\sigma^2}\Re\left[r_k c_k^* e^{j\theta}\right]\right\}, \tag{2.11}$$

the first PDF can be computed as

$$\begin{aligned} p(\boldsymbol{r}_1|\boldsymbol{c}_1,L,A,\theta) &= \prod_{k=1}^{L} p(r_k|c_k,A,\theta) \\ &= \prod_{k=1}^{L}\left(\frac{1}{2\pi\sigma^2}\right)\exp\left\{-\frac{1}{2\sigma^2}|r_k|^2\right\} \\ &\quad \exp\left\{-\frac{A^2}{2\sigma^2}|c_k|^2\right\}\exp\left\{\frac{A}{\sigma^2}\Re\left[r_k c_k^* e^{j\theta}\right]\right\}. \end{aligned} \tag{2.12}$$

As we will demonstrate in the following, because of some of the approximations involved in the derivation of the algorithms, the dependence on the channel phase θ will only appear in the first part of the right hand side of (2.10). Hence, we can average (2.12) over θ to obtain

$$\begin{aligned} p(\boldsymbol{r}_1|\boldsymbol{c}_1,L,A) &= \frac{1}{2\pi}\int_0^{2\pi} p(\boldsymbol{r}_1|\boldsymbol{c}_1,L,A,\theta)\mathrm{d}\theta \\ &= \left(\frac{1}{2\pi\sigma^2}\right)^L \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{L}|r_k|^2\right\} \\ &\quad \exp\left\{-\frac{A^2}{2\sigma^2}\sum_{k=1}^{L}|c_k|^2\right\}\mathrm{I}_0\left(\frac{A}{\sigma^2}\left|\sum_{k=1}^{L} r_k c_k^*\right|\right), \end{aligned} \tag{2.13}$$

in which $\mathrm{I}_0(\cdot)$ is the modified Bessel function of the first kind of order zero.

For the second term in (2.10), instead, we can write

$$p(\boldsymbol{r}_2|L,A,\theta) = \prod_{k=L+1}^{L_{\max}}\left[\frac{1}{M}\sum_{c_k} p(r_k|c_k,A,\theta)\right]. \tag{2.14}$$

The term in square brackets in (2.14) cannot be computed in closed form, so some approximations are needed. In the following two subsections, we derive two alternative decoding rules based on different approximations.

2.4.1 Strategy 1

Let us consider one term of the summation in (2.14). By exploiting (2.11), it can be written as

$$p(r_k|c_k,A,\theta) = \left(\frac{1}{2\pi\sigma^2}\right)\exp\left\{-\frac{1}{2\sigma^2}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}|c_k|^2+\frac{A}{\sigma^2}\Re\left[r_k c_k^* e^{j\theta}\right]\right\}. \tag{2.15}$$

By using a first order Taylor approximation for the last exponential, we get

$$p(r_k|c_k,A,\theta) \simeq \left(\frac{1}{2\pi\sigma^2}\right)\exp\left\{-\frac{1}{2\sigma^2}|r_k|^2\right\}\left[1-\frac{A^2}{2\sigma^2}|c_k|^2+\frac{A}{\sigma^2}\Re\left[r_k c_k^* e^{j\theta}\right]\right]. \tag{2.16}$$

Moreover, by averaging out terms with zero mean, we obtain that

$$\frac{1}{M}\sum_{c_k} p(r_k|c_k,A,\theta) \simeq \left(\frac{1}{2\pi\sigma^2}\right)\exp\left\{-\frac{1}{2\sigma^2}|r_k|^2\right\}\left[1-\frac{A^2}{2\sigma^2}\right], \tag{2.17}$$

hence the second PDF of (2.10) becomes

$$p(\boldsymbol{r}_2|\boldsymbol{c}_2,L,A,\theta) \simeq \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L}\exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\left[1-\frac{A^2}{2\sigma^2}\right]^{\Delta L}. \tag{2.18}$$

Hence, combining (2.13) and (2.18), (2.9) becomes

$$\begin{aligned} p(\boldsymbol{r}|\boldsymbol{c}_1,L,A) = &\left(\frac{1}{2\pi\sigma^2}\right)^{L_{\max}}\exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\sum_{k=1}^{L}|c_k|^2\right\} \\ &I_0\left(\frac{A}{\sigma^2}\left|\sum_{k=1}^{L} r_k c_k^*\right|\right)\left[1-\frac{A^2}{2\sigma^2}\right]^{\Delta L}. \end{aligned} \tag{2.19}$$

Being $L_{\max}$ constant, the first two terms are irrelevant. For the last one we assume to make the following approximation, which is good if $\sigma^2 \gg 1$:

$$\left[1-\frac{A^2}{2\sigma^2}\right] \simeq 1. \tag{2.20}$$

Hence, by exploiting the approximation $\mathrm{I}_0(x) \simeq e^x$ and taking the logarithm, the metric becomes

$$\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(A,L,\boldsymbol{c}_1)}{\arg\max} \left(A \left| \sum_{k=1}^{L} r_k c_k^* \right| - \frac{A^2 L}{2} \right) . \tag{2.21}$$

By performing the maximization with respect to A first, we obtain that the amplitude estimate is

$$\hat{A} = \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|}{L} . \tag{2.22}$$

Substituting (2.22) in (2.21), we have

$$\left(\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max} \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^2}{L} . \tag{2.23}$$

Considering that we used several approximations in the derivation of this decision rule, some of them involving the PLH length, it can be convenient to correct the metric as follows

$$\left(\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max} \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^2}{L^\alpha} , \tag{2.24}$$

where α is a coefficient to be optimized.

2.4.2 Strategy 2

In this section we propose an alternative decoding strategy based on an approximation of the PDF (2.14). In order to simplify the computation of the average with respect to c_k and θ, we make the following approximation:

$$c_k \simeq E\left\{|c_k|\right\} e^{j\psi_k} = e^{j\psi_k} , \tag{2.25}$$

where $\{\psi_k\}$ are independent random variables with uniform distribution in $[0,2\pi)$. By averaging with respect to ψ_k, we can rewrite (2.14) as

$$p(\boldsymbol{r}_2|L,A,\theta) = \prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi} \int_0^{2\pi} \left[\frac{1}{M} \sum_{c_k} p(r_k|c_k,\theta,A,\psi_k) \right] \mathrm{d}\psi_k . \tag{2.26}$$

Thus, by exploiting (2.11)

$$
\begin{aligned}
p(\boldsymbol{r}_2|L,A,\theta) &= \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\Delta L\right\} \\
&\prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi}\int_0^{2\pi}\exp\left\{\frac{A}{\sigma^2}\Re\left[r_k e^{-j\psi_k}e^{j\theta}\right]\right\}\mathrm{d}\psi_k \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\Delta L\right\} \\
&\prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi}\int_0^{2\pi}\exp\left\{\frac{A}{\sigma^2}|r_k|\cos\left(-\psi_k+\theta+\angle r_k\right)\right\}\mathrm{d}\psi_k\,.
\end{aligned}
$$

Using second order Taylor approximation for the last exponential, we obtain

$$
\begin{aligned}
p(\boldsymbol{r}_2|L,A,\theta) &\simeq \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\Delta L\right\} \\
&\prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi}\int_0^{2\pi}\left[1+\frac{1}{\sigma^2}|r_k|\cos\left(-\psi_k+\theta+\angle r_k\right)\right. \\
&\left.+\frac{1}{2\sigma^4}|r_k|^2\cos^2\left(-\psi_k+\theta+\angle r_k\right)\right]^A \mathrm{d}\psi_k\,.
\end{aligned}
$$

Adopting now the approximation $(1+x)^r \simeq 1+rx$ for $|x| \ll 1$, we get

$$
\begin{aligned}
p(\boldsymbol{r}_2|L,A,\theta) &\simeq \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\Delta L\right\} \\
&\prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi}\int_0^{2\pi}\left[1+\frac{A}{\sigma^2}|r_k|\cos\left(-\psi_k+\theta+\angle r_k\right)\right. \\
&\left.+\frac{A}{2\sigma^4}|r_k|^2\cos^2\left(-\psi_k+\theta+\angle r_k\right)\right]\mathrm{d}\psi_k \qquad (2.27) \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\Delta L} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right\}\exp\left\{-\frac{A^2}{2\sigma^2}\Delta L\right\} \\
&\prod_{k=L+1}^{L_{\max}} \frac{1}{2\pi}\left(1+\frac{A}{8\pi\sigma^4}|r_k|^2\right)\,.
\end{aligned}
$$

Note that this expression is independent of θ. Hence, we can compute (2.9) by using (2.13) and (2.27) as

$$p(\boldsymbol{r}|\boldsymbol{c}_1,L,A) = \left(\frac{1}{2\pi\sigma^2}\right)^{L_{\max}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{L_{\max}}|r_k|^2\right\} \mathrm{I}_0\left(\frac{A}{\sigma^2}\left|\sum_{k=1}^{L} r_k c_k^*\right|\right)$$
$$\exp\left\{-\frac{A^2}{2\sigma^2}L_{\max}\right\}\prod_{k=L+1}^{L_{\max}}\frac{1}{2\pi}\left(1+\frac{A}{8\pi\sigma^4}|r_k|^2\right).$$

By removing the first two irrelevant terms, applying the approximation $\mathrm{I}_0(x) \simeq e^x$, and taking the logarithm, the likelihood function can be written as

$$\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(A,L,\boldsymbol{c}_1)}{\arg\max}\left(\frac{A}{\sigma^2}\left|\sum_{k=1}^{L} r_k c_k^*\right| + \sum_{k=L+1}^{L_{\max}}\log\left(1+\frac{A}{8\pi\sigma^4}|r_k|^2\right) - \frac{A^2 L_{\max}}{2\sigma^2}\right).$$

Approximating $\log(1+x) \simeq x$ for $|x| \ll 1$, we get

$$\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(A,L,\boldsymbol{c}_1)}{\arg\max}\left(A\left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{A}{8\pi\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2 - \frac{A^2 L_{\max}}{2}\right). \tag{2.28}$$

Also in this case, we perform the maximization with respect to A first to obtain the following amplitude estimate value, obtaining that

$$\hat{A} = \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{1}{8\pi\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2}{L_{\max}}. \tag{2.29}$$

Finally, by substituting (2.29) in (2.28), we have that the final metric is

$$\begin{aligned}\left(\hat{L},\hat{\boldsymbol{c}}_1\right) &= \underset{(L,\boldsymbol{c}_1)}{\arg\max}\left(\left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{1}{8\pi\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2\right)^2 \\ &= \underset{(L,\boldsymbol{c}_1)}{\arg\max}\left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{1}{8\pi\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2.\end{aligned} \tag{2.30}$$

Also in this case, we can apply a correction factor β to the metric, to be optimized, in order to take into account the approximations involved in the derivation. Hence, the final metric is

$$\left(\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max}\left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{\beta}{\sigma^2}\sum_{k=L+1}^{L_{\max}}|r_k|^2. \tag{2.31}$$

We can notice that this strategy requires knowledge of the noise variance σ^2.

2.4.3 Strategy 3

Following a similar approach to the previous section, an alternative decoding strategy is derived in the following paragraph. We can write

$$
\begin{aligned}
p(\boldsymbol{r}_1|L,\theta,A) = \left(\frac{1}{2\pi\sigma^2}\right)^L \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{L}\left[|r_k|^2 + A^2|c_k|^2\right]\right\} \\
\frac{1}{2\pi}\int_0^{2\pi} \exp\left\{\frac{A}{\sigma^2}\left|\sum_{k=1}^{L} r_k c_k^*\right|\cos(\phi+\theta)\right\} \mathrm{d}\theta\,,
\end{aligned}
\tag{2.32}
$$

and

$$
\begin{aligned}
p(\boldsymbol{r}_1|L,\theta,A) = \left(\frac{1}{2\pi\sigma^2}\right)^L \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{L}\left[|r_k|^2 + A^2|c_k|^2\right]\right\} \\
\left(1 + \frac{A}{4\sigma^4}\left|\sum_{k=1}^{L} r_k c_k^*\right|^2\right)\,.
\end{aligned}
\tag{2.33}
$$

Thus

$$
\begin{aligned}
p(\boldsymbol{r}|\boldsymbol{c}_1,L,A) = \left(\frac{1}{2\pi\sigma^2}\right)^{L_{\max}} \exp\left\{-\frac{\sum_{k=1}^{L_{\max}}|r_k|^2}{2\sigma^2}\right\}\exp\left\{-\frac{LA^2}{2\sigma^2}\right\} \\
\left[1 + \frac{A}{4\sigma^4}\left|\sum_{k=1}^{L} r_k c_k^*\right|^2\right]\left[1 - \frac{A^2}{2\sigma^2}\right]^{\Delta L}\,.
\end{aligned}
\tag{2.34}
$$

By removing irrelevant terms and introducing the logarithm, the likelihood function can thus be expressed as

$$
\left(\hat{A},\hat{L},\hat{\boldsymbol{c}}_1\right) = \underset{(A,L,\boldsymbol{c}_1)}{\arg\max}\, \frac{A}{4\sigma^4}\left|\sum_{k=1}^{L} r_k c_k^*\right|^2 - \frac{LA^2}{2\sigma^2}\,. \tag{2.35}
$$

By performing the maximization with respect to A first, we obtain that the amplitude estimate is

$$
\hat{A} = \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^2}{4\sigma^2 L}\,, \tag{2.36}
$$

and the final metric thus becomes

$$\left(\hat{L}, \hat{\boldsymbol{c}}_1\right) = \arg\max_{(L,\boldsymbol{c}_1)} \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^4}{32\sigma^6 L^\gamma}\,, \tag{2.37}$$

where γ is a coefficient to be optimized that takes into account the approximations involved in the derivation.

2.4.4 Noise Variance Estimation

Some of the previous decoding strategies assume the knowledge of the noise variance. In cases where this information is not available at the receiver, one needs to consider the implementation of a noise variance estimator to correctly perform the detection [26], [27]. In this section we present a noise variance estimation method based on the received information vector, i.e. that does not need pilot symbols. From the received vector $\boldsymbol{r}$, the signal-to-noise ratio can be written as

$$\frac{E_\mathrm{s}}{N_0} = \frac{S}{W} = \frac{A^2}{\sigma^2}\,, \tag{2.38}$$

where S and W are the signal power and the noise power, respectively. Let M_2 denote the second order moment of the received vector

$$M_2 = E\left\{r_n r_n^*\right\} = SE\left\{|c_n|^2\right\} + \sqrt{SW}E\left\{c_n w_n^*\right\} + \sqrt{SW}E\left\{a_n^* w_n\right\} + WE\left\{|w_n|^2\right\}\,, \tag{2.39}$$

and M_4 the fourth moment

$$\begin{aligned} M_4 &= E\left\{(r_n r_n^*)^2\right\} \\ &= S^2 E\left\{|c_n|^4\right\} + 2S\sqrt{SW}\cdot\left(E\left\{|c_n|^2 a_n w_n^*\right\} + E\left\{|c_n|^2 c_n^* w_n\right\}\right) \\ &\quad + SW\cdot\left(E\left\{(c_n w_n)^2\right\} + 4E\left\{|c_n|^2|w_n|^2\right\} + E\left\{(c_n^* w_n)^2\right\}\right) \\ &\quad + 2W\sqrt{SW}\left(E\left\{|w_n|^2 c_n w_n^*\right\} + E\left\{|w_n|^2 a_n^* w_n\right\}\right) + W^2 E\left\{|w_n|^4\right\}\,. \end{aligned} \tag{2.40}$$

Considering M-ary PSK signal and assuming that the signal and the noise are zero mean, independent random processes, and the in-phase and quadrature components of the noise are independent, (2.39) and (2.40) reduce to

$$M_2 = S + W\,, \tag{2.41}$$

and

$$M_4 = S^2 + 4SW + 2W^2 . \tag{2.42}$$

Solving for W, we obtain

$$\hat{S} = \sqrt{2M_2^2 - M_4} , \tag{2.43}$$

and the noise variance can be obtained as

$$\hat{W} = M_2 - \hat{S} . \tag{2.44}$$

This estimator is based on the second and fourth order moments of the received signal, and it does not use receiver decisions. Moreover, no pilot symbols are required to perform estimation. In practice, the second and fourth order moments can be estimated as

$$M_2 \simeq \frac{1}{N_{\text{sym}}} \sum_{n=0}^{N_{\text{sym}}} |r_n|^2 \tag{2.45}$$

and

$$M_4 \simeq \frac{1}{N_{\text{sym}}} \sum_{n=0}^{N_{\text{sym}}} |r_n|^4 , \tag{2.46}$$

where N_{sym} is the number of symbols used in the estimation process. In the following section, we will present and discuss the performance of this estimator in the considered scenario.

2.5 Numerical Results

In this section, we show the performance of the proposed decoding schemes derived in Sections 2.4.1, 2.4.2 and 2.4.3 in the considered DVB-S2X scenario. For the sake of completeness, we report herein the three proposed decoding strategies.

$$\left(\hat{L}, \hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max} \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^2}{L^\alpha} . \tag{2.47}$$

$$\left(\hat{L}, \hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max} \left|\sum_{k=1}^{L} r_k c_k^*\right| + \frac{\beta}{\sigma^2} \sum_{k=L+1}^{L_{\max}} |r_k|^2 . \tag{2.48}$$

$$\left(\hat{L}, \hat{\boldsymbol{c}}_1\right) = \arg\max_{(L, \boldsymbol{c}_1)} \frac{\left|\sum_{k=1}^{L} r_k c_k^*\right|^4}{32\sigma^6 L^\gamma} . \tag{2.49}$$

For each possible length of the PLH, we select the ModCod associated with the lowest SNR, in order to test the detector in the worst working conditions. This corresponds to ModCods 1, 2, 3, and 6, as can be observed from Table 2.1. For each value of SNR, a maximum of 10^7 PLH codewords are simulated. Thus, results below $\text{DER} = 10^{-6}$ lack statistical reliability.

The performance is compared with that achievable with a system that is fully compliant with the DVB-S2X standard. This means that the PLH codewords have all a fixed length of $L_{\max} = 64$ symbols, and the receiver operates according to the following decision rule

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \left|\sum_{k=1}^{L_{\max}} r_k c_k^*\right| . \tag{2.50}$$

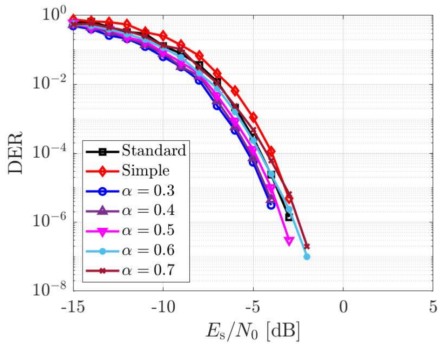
This system is denoted as *standard* in this chapter.

We also compare the proposed decoding strategies with another benchmark strategy, denoted as *simple* in this chapter. This decoder does not know the codeword length and it simply computes the likelihood function of each codeword with the received signal, assuming always the maximum length $L_{\max}$, and it operates according to decoding rule (2.50). This is representative of the performance achievable in a system adopting a variable length PLH, but using a conventional decoding scheme.

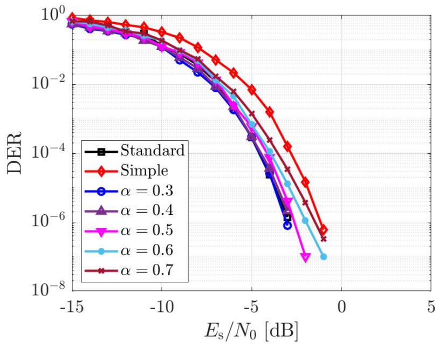
2.5.1 Strategy 1

The results obtained using the first proposed strategy are reported in Fig. 2.2 as a function of the ratio between the average energy per symbol and the noise power spectral density, E_s/N_0, for ModCods 1, 2, 3 and 6. Different values of the parameter α in (2.24) are compared. The gaps from the *standard* detector are reported in Table 2.2 and in Fig. 2.3 (a), while the gaps from the *simple* detector are reported in Table 2.3 and in Fig. 2.3 (b), for several values of α.

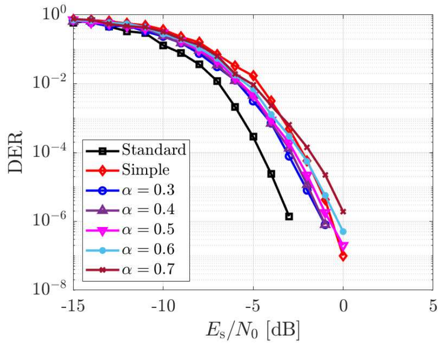
Looking at Fig. 2.2 (a), we can see that for ModCod 1 all tested values of α provide relatively similar performance, and that increasing the value of α the performance tends to degrade. We also notice that, in this case, the *standard* and the *simple*



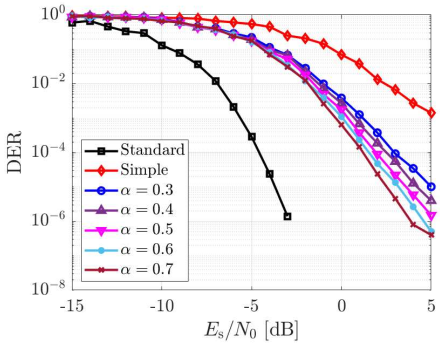
(a) PLH DER for ModCod 1 using Strategy 1.



(b) PLH DER for ModCod 2 using Strategy 1.



(c) PLH DER for ModCod 3 using Strategy 1.

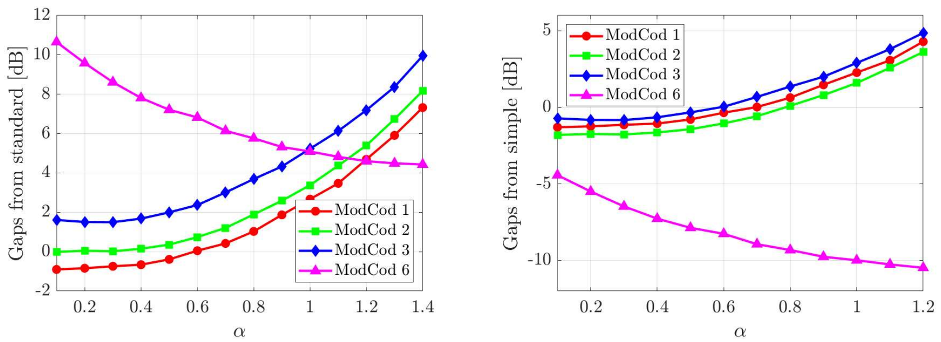


(d) PLH DER for ModCod 6 using Strategy 1.

Figure 2.2: PLH DER for different ModCods using Strategy 1.

Table 2.2: Gaps in dB from *standard* decoding at DER $= 10^{-5}$ using Strategy 1.

ModCod	PLH length	α				
		0.3	0.4	0.5	0.6	0.7
1	64	-0.71	-0.59	-0.3	0.08	0.51
2	58	-0.03	0.15	0.39	0.79	1.24
3	48	1.59	1.74	2.01	2.43	3.02
6	26	8.71	7.9	7.3	6.88	6.22



(a) Gaps from *standard* detector using Strategy 1. (b) Gaps from *simple* detector using Strategy 1.

Figure 2.3: Gaps from *standard* (a) and *simple* (b) detectors for different values of α using Strategy 1.

receivers reach similar performance. This is not surprising, because in our proposed scheme ModCod 1 adopts a 64-symbols PLH codeword, which is the same length considered by both the *standard* and *simple* schemes.

If we look at ModCod 6 (Fig. 2.2 (d)), instead, first of all we notice the poor performance of the *simple* receiver scheme. Also in this case, this was expected, recalling that this scheme attempts decoding on 64 symbols, but the PLH codeword for ModCod 6 has a length of only 26 symbols, so the detector tries to use also part of the information symbols, which are encoded and modulated with the code and the modulation specific to the ModCod. We can also notice the large gap between the *standard* decoder and our proposed solution. This is also unsurprising, considering that the *standard* decoding strategy needs to work properly for all ModCods, including those operating at low SNR, so it has to be robust to the worst case scenario.

Adopting a variable length code, and a corresponding decoding strategy, allows us to relax this robustness constraint for ModCods operating at higher SNR, like ModCod 6 and those above. If we compare the curves for different values of α, we see that the performance improves by increasing α. This is exactly the opposite of the behavior we observed for ModCod 1, so the problem arises of how to select α to

Table 2.3: Gaps in dB from *simple* decoding at DER $= 10^{-5}$ using Strategy 1.

ModCod	PLH length	α 0.3	0.4	0.5	0.6	0.7
1	64	-1.19	-1.06	-0.77	-0.39	0.04
2	58	-1.84	-1.67	-1.43	-1.02	-0.57
3	48	-0.78	-0.61	-0.35	0.08	0.66
6	26	-6.48	-7.29	-7.89	-8.31	-8.97

Table 2.4: Gaps in dB from ModCod convergence threshold at DER $= 10^{-5}$ using Strategy 1.

ModCod	PLH length	α 0.3	0.4	0.5	0.6	0.7
1	64	-1.56	-1.44	-1.15	-0.76	-0.34
2	58	-1.69	-1.52	-1.28	-0.87	-0.42
3	48	-2.32	-2.16	-1.9	-1.47	-0.89
6	26	-0.12	-0.92	-1.53	-1.95	-2.6

guarantee a good trade-off in all cases.

In order to allow for a correct decoding of the transmitted information, it is important that the PLH is reliably decoded at SNR values that are lower than the working conditions of the ModCods. In Table 2.4 and in Fig. 2.4 we report the gaps from the ModCod convergence threshold, as reported in [4], for several values of α, at a target DER $= 10^{-5}$. From Fig. 2.4, we can notice that the maximum gap is minimized for $\alpha = 0.5$. Moreover, we can see that for $0.3 \leq \alpha \leq 0.7$ the gaps are negative, which means that the PLH is decoded with a DER below 10^{-5} for SNR values which are lower than the ModCod convergence threshold. This is fundamental to guarantee the

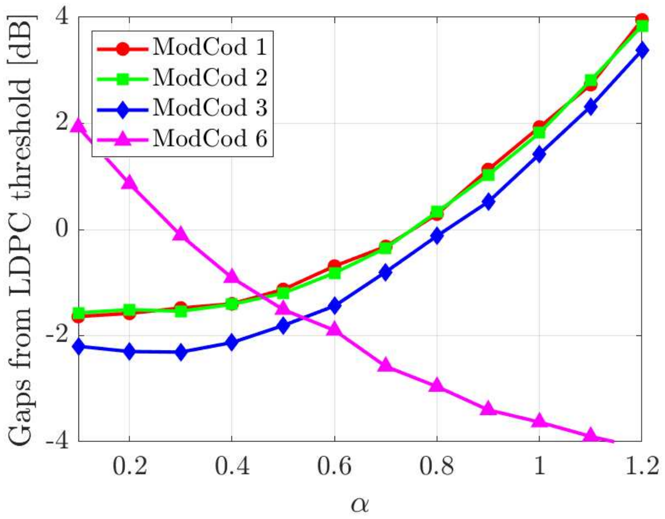


Figure 2.4: Gap in dB from the ModCod convergence threshold for several values of α at DER $= 10^{-5}$, using Strategy 1.

correct decoding of all ModCods.

Table 2.5 reports the average gaps from the ModCod convergence threshold considering the probability of occurrence of the different PLH lengths. We notice that choosing $\alpha = 0.7$ is a value which mostly favors the most probable length, and still keeps the PLH decoding performance below the LDPC convergence threshold for all ModCods. Values of α higher than 0.7 will reduce the gap for ModCod 6, but also increase the gap for the other ModCods, so they should not be considered.

2.5.2 Strategy 2

The results obtained using Strategy 2 in Section 2.4.2 are presented in this section. Through simulations, we verified that the metric shows the best performance for $\beta = 0.2$, for all the tested ModCods. We report the corresponding DER curves in Fig. 2.5. The gaps from the *standard* and *simple* detectors are reported in tables 2.6 and 2.7, respectively, while the gaps from the ModCod convergence threshold are reported in

Table 2.5: Average gaps in dB from ModCod convergence threshold using Strategy 1.

α	0.3	0.4	0.5	0.6	0.7
Avg. gap	-0.09	-0.26	-0.39	-0.46	-0.59

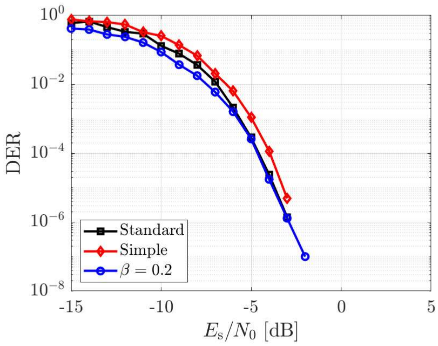
Table 2.6: Gaps in dB from *standard* decoding DER $= 10^{-5}$ using Strategy 2, $\beta = 0.2$.

ModCod	Gap
1	-0.09
2	0.62
3	2.19
6	4.45

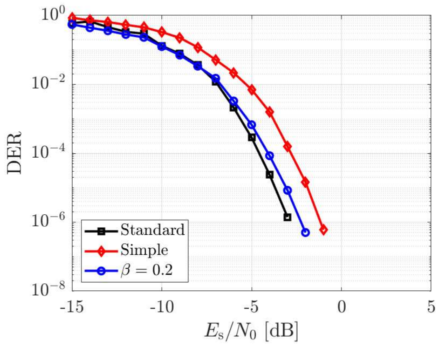
Table 2.8. Finally, Table 2.9 reports the average gap from the ModCod convergence threshold considering the probability of occurrence of the different PLH lengths. We can see that for all the ModCods the PLH is decoded reliably at SNR values that are lower than the ModCods convergence threshold; in particular, for ModCod 6, a very large gap can be achieved.

Considering the *simple* detector, similarly to what has been noted for the first decoding strategy, we note that it has acceptable performance for the longest PLH codewords, while it clearly shows a performance degradation for the shortest codewords. This behavior is consistent with the fact that the detector always operates on the maximum number of symbols $L_{\max}$, even when the actual codeword length is much smaller.

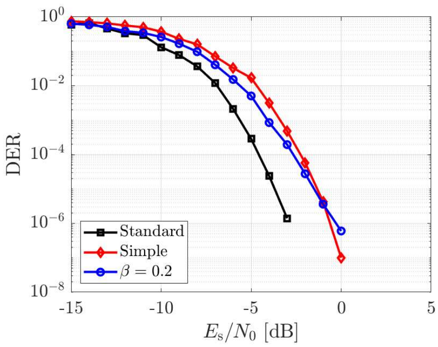
As mentioned, this decoding strategy assumes the knowledge of the noise variance. The results presented in this section assume perfect knowledge of the noise variance. However, this parameter can be estimated [26]. In section 2.5.4 we will



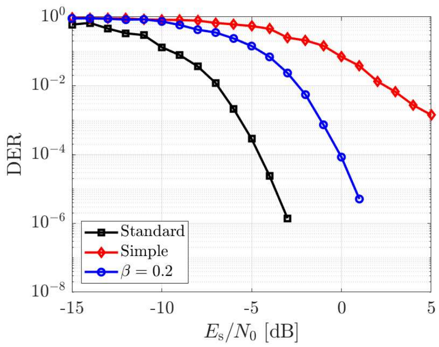
(a) PLH DER for ModCod 1 using Strategy 2.



(b) PLH DER for ModCod 2 using Strategy 2.



(c) PLH DER for ModCod 3 using Strategy 2.



(d) PLH DER for ModCod 6 using Strategy 2.

Figure 2.5: PLH DER for different ModCods using Strategy 2.

refer to the algorithm proposed in [27] to analyze the problem of noise variance estimation.

2.5.3 Strategy 3

In this section, we report the results obtained using the third decoding strategy described in Section 2.4.3. The DER curves are reported in Fig. 2.6, for several values of the parameter γ. The corresponding gaps from the *standard* and *simple* detectors are reported in tables 2.10 and 2.11, respectively, while the gaps from the ModCod

Table 2.7: Gaps in dB from *simple* decoding DER $= 10^{-5}$ using Strategy 2, $\beta = 0.2$.

ModCod	Gap
1	-0.56
2	-1.19
3	-0.17
6	-10.74

Table 2.8: Gaps in dB from ModCod convergence threshold using Strategy 2, $\beta = 0.2$.

ModCod	Gap
1	-0.94
2	-1.05
3	-1.72
6	-4.37

convergence threshold are reported in Table 2.12. Finally, Table 2.13 reports the average gaps from the ModCod convergence threshold considering the probability of occurrence of the different PLH lengths, for different values of γ.

In Fig. 2.7 (a), the gaps from the *standard* detector which knows the PLH length are reported, for several values of γ. In this case, one can see that the maximum gap is minimized for $\gamma = 1.4$. However, if we compute the average gap considering the different probability of occurrence of the PLH lengths, reported in Table 2.13, we see that $\gamma = 1.8$ minimizes this value.

In Fig. 2.7 (b), we report the gaps from the ModCod convergence threshold of each ModCod for several values of γ, at a target DER $= 10^{-5}$. From the figure one can note that, for $0.6 \leq \gamma \leq 1.8$, the PLH is reliably decoded at SNR values that are lower than the ModCod threshold. Thus, considering the probability of occurrence of

Table 2.9: Average gap in dB from ModCod convergence threshold using Strategy 2.

β	0.2
Avg. gap	-0.99

Table 2.10: Gaps in dB from *standard* decoding at DER $= 10^{-5}$, using Strategy 3.

ModCod	PLH length	γ 1.4	1.5	1.6	1.7	1.8
1	64	0.80	0.92	1.03	1.06	1.27
2	58	0.84	0.92	0.98	0.99	1.19
3	48	0.64	0.73	0.73	0.56	0.79
6	26	0.95	0.78	0.55	0.37	0.26

Table 2.11: Gaps in dB from *simple* decoding at DER $= 10^{-5}$, using Strategy 3.

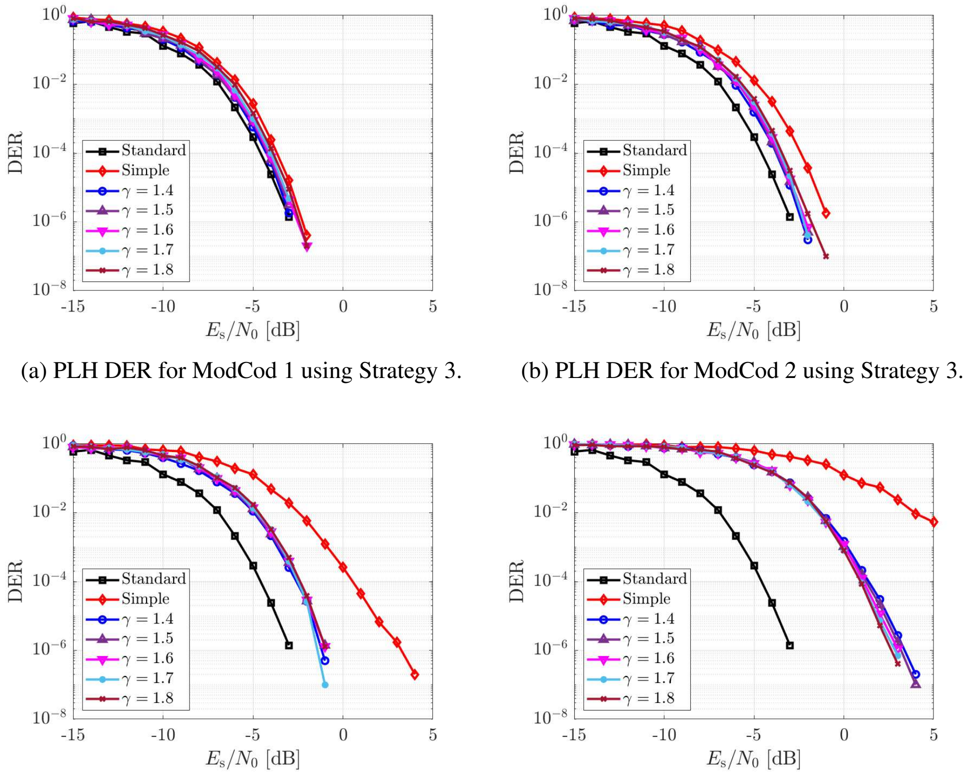
ModCod	PLH length	γ 1.4	1.5	1.6	1.7	1.8
1	64	-0.63	-0.51	-0.41	-0.37	-0.17
2	58	-1.4	-1.32	-1.26	-1.25	-1.05
3	48	-3.54	-3.45	-3.45	-3.62	-3.39
6	26	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

Table 2.12: Gaps in dB from ModCod convergence threshold at DER $= 10^{-5}$, using Strategy 3.

ModCod	PLH length	γ				
		1.4	1.5	1.6	1.7	1.8
1	64	-0.66	-0.54	-0.43	-0.4	-0.19
2	58	-0.93	-0.86	-0.79	-0.78	-0.58
3	48	-1.97	-1.88	-1.88	-2.05	-1.82
6	26	-2.67	-2.84	-3.07	-3.25	-3.36

Table 2.13: Average gaps in dB from ModCod convergence threshold using Strategy 3.

γ	1.4	1.5	1.6	1.7	1.8
Avg. gap	-0.63	-0.66	-0.71	-0.76	-0.77



(a) PLH DER for ModCod 1 using Strategy 3.

(b) PLH DER for ModCod 2 using Strategy 3.

(c) PLH DER for ModCod 3 using Strategy 3.

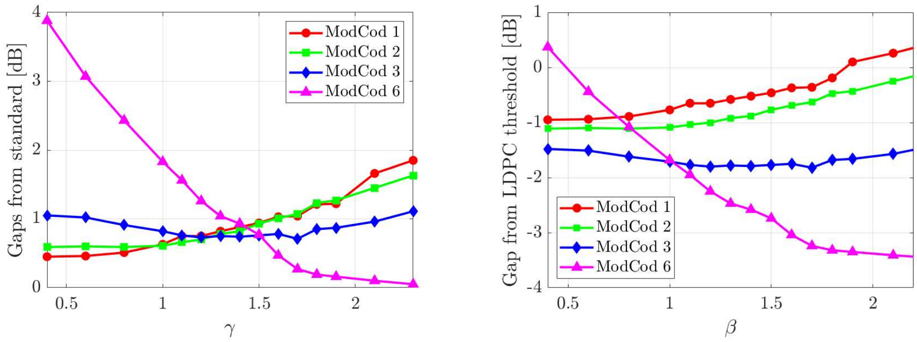
(d) PLH DER for ModCod 6 using Strategy 3.

Figure 2.6: PLH DER for different ModCods using Strategy 3.

the different PLH lengths, the best choice is to select $\gamma = 1.8$, as can be seen from Table 2.13. In fact, this value guarantees that the PLH is reliably detected at SNR values that are below the convergence threshold for all the ModCods, especially for the most probable PLH length.

2.5.4 Noise Variance Estimation

In this section, we present the results obtained using the metric (2.31) derived in Section 2.4.2 adopting the noise variance estimator presented in Section 2.4.4. The



(a) Gaps in dB from *standard* decoding using Strategy 3.

(b) Gaps in dB from the ModCod convergence threshold using Strategy 3.

Figure 2.7: Gaps in dB from *standard* decoding (a) and from the ModCod convergence threshold (b) for several values of γ at DER $= 10^{-5}$, using Strategy 3.

estimator is based on the second and fourth order moments of the received signal, and does not need pilot symbols to perform estimation, so it is independent of the adopted frame structure. Thus, the decoding strategy becomes

$$\left(\hat{L}, \hat{\boldsymbol{c}}_1\right) = \underset{(L,\boldsymbol{c}_1)}{\arg\max} \left| \sum_{k=1}^{L} r_k c_k^* \right| + \frac{\alpha}{\hat{\sigma}^2} \sum_{k=L+1}^{L_{\max}} |r_k|^2 \,. \tag{2.51}$$

The estimation of the noise variance σ^2 is based on the received vector. In particular, the estimation is performed only on the BPSK modulated PLH symbols. Since the length of the PLH is unknown at the receiver, the estimation is performed on 26 BPSK symbols for all the tested ModCods, i.e. using the smallest PLH length among all the ModCods. In order to evaluate the performance of this estimator, we computed the mean squared error (MSE) by simulations from a number of estimates $\hat{\sigma}_i$ according to

$$\mathrm{MSE}\left\{\hat{\sigma}^2\right\} = \frac{1}{N_\mathrm{t}} \sum_{i=1}^{N_\mathrm{t}} (\hat{\sigma}_i^2 - \sigma^2)^2 \,, \tag{2.52}$$

where N_t represents the number of trials and $\hat{\sigma}_i^2$ is the i-th estimates of the noise variance.

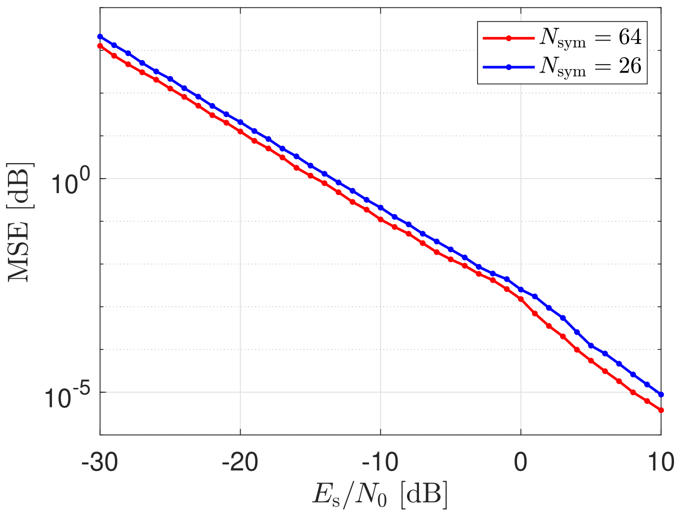


Figure 2.8: MSE for different values of N_{sym} varying signal-to-noise ratio.

Fig. 2.8 reports the MSE computed over $N_{\text{t}} = 10^3$ trials as a function of the channel signal-to-noise ratio. N_{sym} indicates the number of symbols that are used to perform the estimation. As previously discussed, in our simulations we used $N_{\text{sym}} = 26$. However, since the PLH field can have a maximum length of 64 symbols, we also simulated the performance in the case the estimation is performed with $N_{\text{sym}} = 64$. Clearly, in this scenario the estimator performs better since it operates on an higher number of symbols. However, from the figure one can note that using a higher number of symbols yields only a small improvement in terms of MSE, thus making the use of a lower number of symbols acceptable. We also report, in Fig. 2.9, the Cramér Rao lower bound (CRLB) and the variance of the proposed estimator as a function of E_{s}/N_0.

Fig. 2.10 reports the DER curves for ModCods 1 and 6 assuming ideal knowledge of the noise variance (blue curve), and using the variance estimator [26] ($\hat{\sigma}^2$ green curves). We notice that we can achieve the same performance with the variance estimator as in case of perfect knowledge of the variance. We see that, for all the ModCods, the results obtained with this decoding strategy approach the one obtained

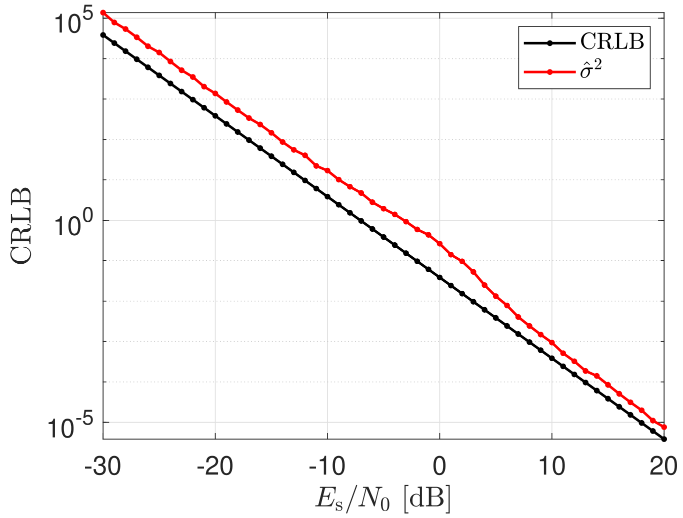


Figure 2.9: CRLB and variance of the estimator varying E_s/N_0, with $N_{sym} = 26$.

with the detector not employing the noise variance estimation and knowing the real value of the noise variance. Thus, we can conclude that the proposed moment-based estimator shows good performance, and it can be successfully adopted in the tested scenario.

2.6 Conclusions

In this chapter, we presented different decoding schemes that can be adopted to decode a PLH code having variable length codewords. Clearly, the best decoding strategy depends on the probability distribution of the PLH length over the ModCods, as well as on the number of ModCods and on the adopted PLH code.

To summarize all the results, we report in Table 2.14 the optimal value of the parameter for each decoding strategy, with associated the corresponding average gap from the ModCod convergence threshold, as it represents the most important performance metric to evaluate the effectiveness of each strategy. From the table, we can see that Strategy 2 shows the best performance in terms of larger gaps from LDPC con-

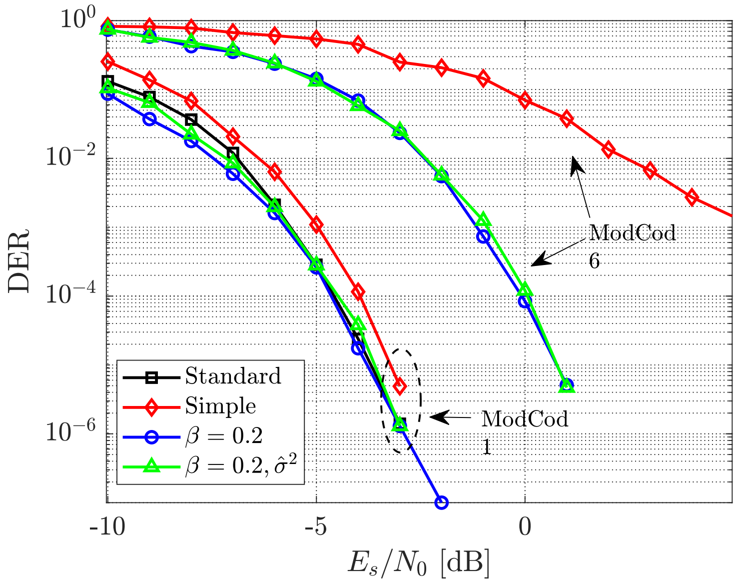


Figure 2.10: PLH DER comparison for Strategy 2 using a noise variance estimator.

vergence threshold, but all the proposed strategies can ensure very good performance, adopting a PLH with length significantly lower than the one used in the DVB-S2X standard. In fact, if we compare the proposed design with the standard, where all 39 ModCods adopt a fixed length 64 symbols code, and taking into account the probability distribution of the codeword length, reported in Table 2.1, we note that the average codeword length is 30 bits, so less than half the value adopted in the standard.

In terms of complexity, Strategy 1 (2.24) and Strategy 3 (2.37) operate only on the first L symbols, while Strategy 2 (2.31) takes all $L_{\max}$ symbols into account. The *standard* fixed length PLH decoding strategy would operate with the metric (2.50) on all $L_{\max}$ symbols, and the same would happen for the described *simple* receiver.

Table 2.15 reports the complexity of the different strategies, in terms of number of real operations. To evaluate the complexity, we have to keep in mind that the PLH uses $\pi/2-$BPSK symbols, so multiplications with preamble symbols only results in a change of the sign of the real or the imaginary part. Hence, multiplications by preamble symbols are not accounted for. Look-up table (LUT) accesses are necessary

Table 2.14: Optimal parameter and average gaps from the ModCod convergence threshold for each decoding strategy.

Strategy	Opt param.	Average gap
1	$\alpha = 0.7$	-0.59
2	$\beta = 0.2$	-0.99
3	$\gamma = 1.8$	-0.77

Table 2.15: Complexity comparison for the proposed decoding strategies

Strategy	# additions	# multiplications	# LUT
Standard (2.50)	$4L_{\text{max}} - 1 = 255$	2	1
1 (2.24)	$4L - 1 = 119$	3	0
2 (2.31)	$3L + L_{\text{max}} - 1 = 153$	$2(L_{\text{max}} - L) + 4 = 72$	1
3 (2.37)	$4L = 120$	13	0

to evaluate square roots to compute absolute values. For strategies 1, 2 and 3 we have used the average codeword length, i.e. 30, to evaluate the complexity. We can conclude that Strategy 1 and Strategy 3 are clearly desirable from an implementation complexity point of view. We have to keep in mind, however, that the performance and complexity may vary depending on the PLH length distribution, number of PLH codewords and the channel conditions. Hence, while the general decoding framework remains valid, performance must be evaluated according to the characteristics of the considered communication system.

The proposed decoding strategies rely on the tuning of a parameter, which takes into account the approximations involved in the derivation. The optimal value of this parameter, clearly, depends on the investigated scenario and on the adopted PLH code, but the general strategy can be easily applied to different cases and applications.

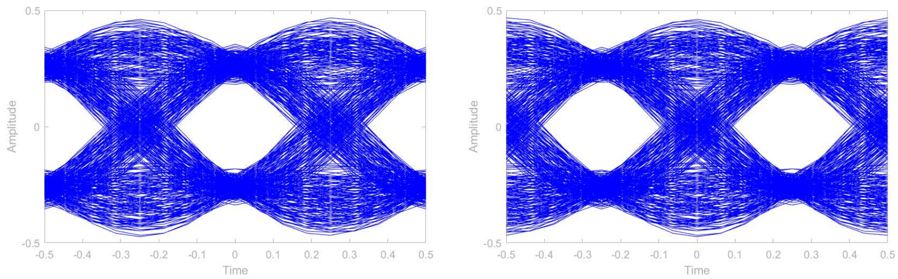
Chapter 3

Timing Synchronization

In this chapter, we investigate the problem of timing synchronization in digital receivers. Timing synchronization is the process of estimating a clock signal that is aligned both in time and in frequency with the clock used to generate the data at the transmitter. Timing synchronization is required at physical layer when the samples of the received signal are misaligned with the data symbols generated by the transmitter. Thus, the received samples are not aligned with the maximum eye opening, as can be seen from the eye diagram in Fig. 3.1.

There are two different approaches to estimate the correct sampling time [28]. The first one is based on *synchronous* sampling, and exploits an error signal which drives a numerically controlled oscillator (NCO) to adjust the sampling time [28]. On the other hand, a more suitable approach for digital receivers is based on *asynchronous* sampling. In *asynchronous* sampling, the sampling clock is independent of data clock used by the transmitter, and the correct sampling time is derived through interpolation of adjacent samples. The chapter will focus on this second approach, as it is the method adopted in real digital receivers [29].

The chapter is structured as follows. We first introduce, in Section 3.1, the system model, while the classical architecture for timing synchronization, with details of each functional block, is presented in Section 3.2. Then, in Section 3.3, we will propose a low-complexity architecture for timing synchronization that is practically



(a) $\tau = 0$ (b) $\tau \neq 0$

Figure 3.1: Eye diagram in presence of a timing offset τ.

suitable for high speed receivers. Such architecture employs only a single NCO, unlike parallel schemes available in the literature [30] [31] [32], allowing a significant complexity reduction. Finally, numerical simulations are carried out to compare the performance of the classical and parallel implementations to evaluate possible losses, and results are reported in Sections 3.4 and 3.5.

3.1 System Model

In this section, we present the model of the symbol timing recovery (STR) scheme. The received signal, prior to matched filtering, can be expressed as

$$r(t) = \sum_i a_i p(t - iT - \tau) e^{j(2\pi i \Delta f T + \theta)} + w(t)\,, \tag{3.1}$$

where $\{a_i\}$ are the transmitted symbols belonging to an M-ary complex constellation, $p(t)$ is the shaping pulse, assumed to have a root-raised cosine (RRC) spectrum with roll-off factor α, T is the symbol interval, θ is a phase offset, and $w(t)$ is a complex additive white Gaussian noise process (AWGN). Parameters τ and $\Delta f T$ identify the time offset and the normalized carrier frequency offset, respectively. τ is assumed to be in the interval $-\frac{T}{2} \leq \tau \leq \frac{T}{2}$, while we assume $\Delta f T = 0.01$, which is assumed to be the maximum uncompensated normalized residual carrier frequency offset after coarse frequency synchronization.

The received signal at the output of the matched filter (MF) is given by

$$y(t) = \sum_i a_i g(t - iT - \tau)e^{j(2\pi i \Delta f T + \theta)} + n(t), \tag{3.2}$$

where $g(t) \triangleq p(t) \otimes p^*(-t)$, and $n(t)$ is a colored noise process. The signal is then sampled at fixed rate $1/T_s$ to obtain N samples per symbol, where the oversampling factor is defined as $N \approx T/T_s$, where the approximation will be clear in the following. The sampling rate $1/T_s$ is *asynchronous* with respect to the symbol rate $1/T$, i.e., in presence of a timing offset $\tau \neq 0$ the received samples are not aligned with the transmitted symbols.

The oversampling factor is assumed to be fractional, equal to $N = 2.25$ samples per symbol. We point out that the choice of N is arbitrary. Since perfect sampling frequency is difficult to achieve because of non ideal oscillators, the clocks used at the transmitter and at the receiver are assumed to be independent. Therefore, a sampling clock offset (SCO) between the transmitter and the receiver is present, i.e. $N \neq T/T_s$.

3.2 Serial Symbol Timing Recovery

The block diagram for digital timing synchronization is reported in Fig. 3.2. After analog-to-digital (ADC) conversion, with rate $1/T_s$, a new sample enters in a shift register with depth L. With the same rate, the entire content of the register is filtered by a polyphase filter, denoted by *Polyphase MF* in Fig. 3.2, which, as we will explain the following, is designed according to the polyphase approach [33] to efficiently combine matched filtering and interpolation. The *Polyphase MF* outputs a new sample with a rate equal to $2/T$, since the Gardner time error detector (TED) [34] requires two samples per symbol. This can be accomplished by using proper control signals, denoted by $\ell_1(mT_s)$ and $\ell_2(mT_s)$, whose dependence on T_s will be omitted in the following for the sake of simplicity. These signals are denoted by ℓ_1 and ℓ_2 in Fig. 3.2, and their role is explained hereafter.

Indeed, the two control signals, ℓ_1 and ℓ_2, are generated by the NCO block, as described below, and their purpose is to label each sample at the output of the *Polyphase*

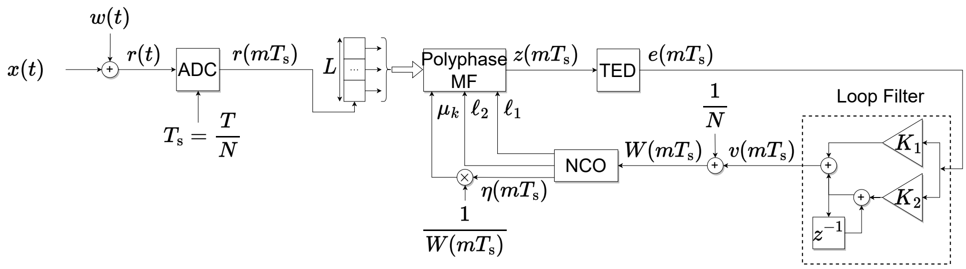


Figure 3.2: Block diagram for symbol timing recovery.

MF either as an *optimum* sample, when $\ell_1 = 0$ and $\ell_2 = 1$, i.e., it is computed at instant $kT + \hat{\tau}$, or as a *valid* sample, when $\ell_1 = \ell_2 = 1$, i.e., it is computed at instant $kT - T/2 + \hat{\tau}$.

Since we assumed a fractional oversampling factor, some samples should be discarded: this occurs when neither the sample at $kT + \hat{\tau}$ nor that at $kT - T/2 + \hat{\tau}$ are present in the interval $[mT_s,\ (m+1)T_s]$ and, as a consequence, the NCO generates the control signal $\ell_2 = 0$ to notify the *Polyphase MF* block that no sample has to be generated at its output. Note that the control signals ℓ_1 and ℓ_2 should also be provided to the TED block, which needs to know if the incoming sample is either *optimum* or *valid*. For the sake of simplicity, this is not reported in Fig. 3.2.

Going into the details of the block diagram in Fig. 3.2, the main features of each block, i.e., the polyphase filter, the TED, the loop filter and the NCO, are described hereafter.

3.2.1 Polyphase filter

The role of the polyphase filter is to produce new samples, that we will call *interpolants* in the following, that are aligned with the clock used at the transmitter. At the core of the polyphase filter subsystem there is a *Polyphase MF*, which in turn consists of a bank of parallel sub filters, each having a different time offset. Such a bank of filters is obtained after polyphase decomposition of an upsampled version of the matched filter, which leads to an architecture able to simultaneously perform matched filtering and interpolation [33]. Each of these filters, with L coefficients, is used to process the whole content of the shift register, which is then updated by adding the new incoming sample.

Timing adjustment is performed by interpolating the asynchronous samples at the output of the ADC, which are denoted by $r(mT_\mathrm{s})$, so as to obtain the samples at the optimal sampling instants, denoted by $y(kT+\hat{\tau})$. The sequence at the output of the *Polyphase MF*, denoted by $z(mT_\mathrm{s})$, can be thus expressed as

$$z(mT_\mathrm{s}) = \ell_2 y\left[\left(k-\frac{\ell_1}{2}\right)T+\hat{\tau}\right]. \tag{3.3}$$

When the k-th *interpolant* is within the interval $[mT_\mathrm{s},\ (m+1)T_\mathrm{s}]$, the sample index m is called the k-th *basepoint index*, and is denoted by m_k. The optimal sampling instant exceeds m_kT_s by some fraction of the symbol time. This fraction is called the k-th *fractional index*, and is denoted by μ_kT_s. Therefore, the *basepoint index* m_k identifies the integer sample closest to the maximum eye opening, and the *fractional index* μ_kT_s is the fraction between the *basepoint index* and the optimal sampling instant, which can thus be expressed as $kT+\hat{\tau}=(m_k+\mu_k)T_\mathrm{s}$. This situation is graphically depicted in Fig. 3.3.

Therefore, the *Polyphase MF* requires only two inputs to compute the *interpolants*, namely the *basepoint index* m_k to identify the correct set of the received samples to use, and the *fractional index* μ_k to properly select the coefficients of the interpolating filter. While the *basepoint index* m_k is not explicitly computed, as it is identified by flagging the correct set of samples, the *fractional index* μ_k is computed from the timing recovery loop, as we will describe in the following.

3.2.2 TED

The adopted timing loop is based on the non-data-aided (NDA) Gardner algorithm, which is a TED requiring only two samples per symbol, corresponding to the rate available at the output of the *Polyphase MF*. The error signal is given by [34]

$$e(mT_\mathrm{s}) = \Re\{y^*\left((k-1/2)\,T+\hat{\tau}\right)\cdot\left[y\left((k-1)\,T+\hat{\tau}\right)-y\left(kT+\hat{\tau}\right)\right]\} \tag{3.4}$$

if $(\ell_1,\ \ell_2)=(0,\ 1)$, while it is $e(mT_\mathrm{s})=0$ otherwise.

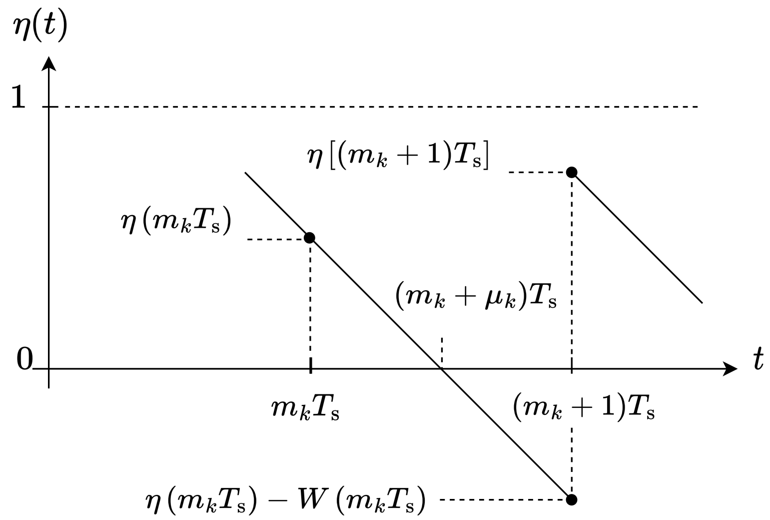


Figure 3.3: Relation between the NCO register and the optimal sampling instant.

3.2.3 Loop Filter

The error signal defined in (3.4) is filtered by a second order proportional-plus-integrator loop filter to track out the symbol clock frequency offset [35]. The noise equivalent bandwidth of the filter is a function of the loop filter constants, whose expression can be found in [29]. The output of the loop filter, denoted by $v(mT_\mathrm{s})$, controls the amount by which the NCO decrements. The loop filter constants are defined as follows.

The detector gain, which we denote by K_p, is obtained from the S-curve of the TED, which is computed by an open-loop measurement of the average error signal conditioned on a fixed value of the timing offset, i.e., $S(\tau) \triangleq E\left[e(k)|\tau_k = \tau\right]$. The value of K_p is defined as the slope of the S-curve at $\tau = 0$. The loop gain K_0, since the NCO is based on a decrementing counter, is equal to $K_0 = -1$.

The acquisition time and the tracking performance of the symbol timing recovery loop depend on the normalized noise equivalent bandwidth B_nT and on the damping factor ζ. From these, we can derive the loop constants, which we denote by K_1 and

K_2, as follows [29]:

$$K_1 = \frac{\frac{4\zeta}{NK_pK_0}\left(\frac{B_nT}{\left(\zeta+\frac{1}{4\zeta}\right)}\right)}{1+\frac{2\zeta}{N}\left(\frac{B_nT}{\left(\zeta+\frac{1}{4\zeta}\right)}\right)+\left(\frac{B_nT}{N\left(\zeta+\frac{1}{4\zeta}\right)}\right)^2} \tag{3.5}$$

$$K_2 = \frac{\frac{4}{N^2K_pK_0}\left(\frac{B_nT}{\left(\zeta+\frac{1}{4\zeta}\right)}\right)^2}{1+\frac{2\zeta}{N}\left(\frac{B_nT}{\left(\zeta+\frac{1}{4\zeta}\right)}\right)+\left(\frac{B_nT}{N\left(\zeta+\frac{1}{4\zeta}\right)}\right)^2}\,. \tag{3.6}$$

3.2.4 NCO

The role of the NCO is to provide the *Polyphase MF* with the k-th *basepoint index* m_k and the k-th *fractional index* $\mu_k T_s$, and label the *interpolants* either as *optimum* or *valid*. The NCO consists in a modulo-1 decrementing counter that is recursively updated as

$$\eta[(m+1)T_s] = \eta(mT_s) - W(mT_s) \quad \mod \quad 1\,, \tag{3.7}$$

where $W(mT_s) = 1/N + v(mT_s)$ is the control word available at the output of the loop filter. The *fractional index* $\mu_k T_s$ is computed each time an underflow of the the NCO occurs. When the NCO underflows, it also generates the control signals $(\ell_1,\ \ell_2) = (0,\ 1)$ to notify the *Polyphase MF* that a new optimum sample should be computed. The sample index is the actual *basepoint index* m_k, while the value of the *fractional index* $\mu_k T_s$ can be computed from the content of the NCO register at instant $m_k T_s$, i.e. the time instant immediately preceding the interpolation instant $(m_k + \mu_k)T_s$, as can be seen from Fig. 3.3. From simple geometrical considerations, it follows that [29]

$$\mu_k T_s = \frac{\eta\,(mT_s)}{W\,(mT_s)}\,. \tag{3.8}$$

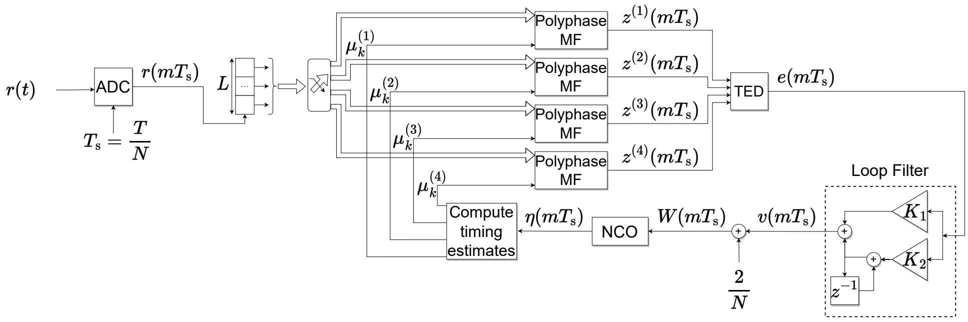


Figure 3.4: Block diagram for parallel symbol timing recovery with $P = 4$.

3.3 Parallel Timing Synchronization

In this section, we propose a novel architecture for the parallel timing recovery suitable for high speed receivers. The problem of high speed parallel timing synchronization has been widely studied in the literature. In [36], the authors implemented a parallel Oerder and Meyr (O&M) TED whose architecture, however, requires an oversampling factor equal to 4, thus demanding considerable hardware resources. Moreover, the performance analysis was limited to 16QAM, without investigating higher order modulations. A parallel architecture, designed for optical coherent receivers, can be found in [31], where the algorithm is tested in the presence of sampling frequency offsets. Therein, only one error signal is exploited by the loop filter, thus increasing the convergence time.

The block diagram of the proposed architecture is reported in Fig. 3.4, for a parallelization factor $P = 4$. After ADC conversion, with rate $1/T_s$, a new sample enters in a shift register with depth L. With the same rate, a selector extracts all the samples from the register, and provides them as input to one of the *Polyphase MF*s. Then, with rate $T_s \cdot P$, the P *Polyphase MF*s are activated to produce the corresponding *interpolants*, which are then labeled either as *optimum* or *valid* by the NCO.

Note that the samples at the *Polyphase MF*s outputs are characterized by a rate equal to $2/T$, since the Gardner TED requires two samples per symbol. This is possible thanks to the introduction of some control signals, similarly to what has been described in Sec. 3.2, denoted by $\ell_1^{(p)}$ and $\ell_2^{(p)}$, and not reported in Fig. 3.4 for the sake of visual clarity, where the superscript $p = 1, \ldots, P$ identifies the p-th *Polyphase MF*.

The control signals are generated by the NCO, as described below, and their purpose is to label each sample at the output of the bank of *Polyphase MF*s. In particular, signal $\ell_1^{(p)}$ can take discrete values in the interval $[-2,\ 3]$. When $\ell_2^{(p)} = 1$ and $\ell_1^{(p)}$ is even, the sample is *optimum*, when $\ell_2^{(p)} = 1$ and $\ell_1^{(p)}$ is odd, the sample is *valid*, while when $\ell_2^{(p)} = 0$ the sample at the output of the p-th *Polyphase MF* is not computed, regardless of the value of $\ell_1^{(p)}$. Note that, when $\ell_1^{(p)}$ is negative, the sample at the output of the p-th *Polyphase MF* will be used in the computation of the error signal at the next clock cycle. Note that, although this is not explicitly shown in Fig. 3.4 for the sake of visual clarity, the control signals $\ell_1^{(p)}$ and $\ell_2^{(p)}$ are produced by the NCO and provided to the p-th *Polyphase MF*.

The functional blocks are conceptually similar to those in Fig. 3.2, while the operational differences are further detailed in the points that follow, except for the loop filter, which is exactly the same as in Sec. 3.2.3.

3.3.1 Polyphase filter

Interpolation is performed by a bank of P parallel *Polyphase MF*s, designed as described in Sec. 3.2.1, which, every $T_\mathrm{s} \cdot P$, output the corresponding *interpolants*, that are then labeled either as *optimum* or *valid* by the NCO. The total number of *interpolants*, and therefore of timing estimates, that are computed every $T_\mathrm{s} \cdot P$, is equal to the number of *optimum* and *valid* samples. The sample at the output of the p-th *Polyphase MF* can be expressed as

$$z^{(p)}(mT_\mathrm{s}) = \ell_2^{(p)} y\left[\left(k - \frac{\ell_1^{(p)}}{2}\right)T + \hat{\tau}\right], \tag{3.9}$$

which is identical to Eq. (3.3) in the case $p = 1$.

We define by $\mathscr{I}_p = [(m + (p-1))T_\mathrm{s},\ (m+p)T_\mathrm{s}]$, with $p = 1, \ldots, P$, the set of P intervals in which a new *interpolant* can be computed. The *interpolant* in the p-th interval will be provided by the p-th *Polyphase MF*.

When the k-th *interpolant* is within the p-th interval of $\mathscr{I}_p$, the sample index $[m+(p-1)]$ is called the k-th *basepoint index*, and is denoted by $m_k^{(p)}$. The optimal sampling instant exceeds $m_k^{(p)}T_\mathrm{s}$ by some fraction of the symbol time. This fraction is called the k-th *fractional index*, and is denoted by $\mu_k^{(p)}T_\mathrm{s}$. Therefore, the *basepoint index* $m_k^{(p)}$ identifies the integer sample closest to the maximum eye opening, and the *fractional index* $\mu_k^{(p)}T_\mathrm{s}$ is the fraction between the *basepoint index* and the optimal sampling instant, which can thus be expressed as $[k+(p-1)]T+\hat{\tau}=(m_k^{(p)}+\mu_k^{(p)})T_\mathrm{s}$.

3.3.2 TED

The expression of the error signal depends on the number of *optimum* samples at the output of the bank of *Polyphase MF*s. It turns out that we can have either 1 or 2 *optimum* samples. In the former case, the error signal is computed as in the serial algorithm, according to Eq. (3.4). In the latter case, instead, the error signal is computed as

$$\begin{aligned} e(mT_\mathrm{s}) = \{&\Re\{y^*((k-1/2)\,T+\hat{\tau})\ \cdot [y((k-1)\,T+\hat{\tau})-y(kT+\hat{\tau})]\}+ \\ &\Re\,\{y^*((k-3/2)\,T+\hat{\tau})\ \cdot [y((k-2)\,T+\hat{\tau})-y((k-1)\,T+\hat{\tau})]\}\}/2\,. \end{aligned} \tag{3.10}$$

3.3.3 NCO

The NCO consists in a modulo-1 decrementing counter that is recursively updated, every $T_\mathrm{s}\cdot P$, according to

$$\eta\,[(m+P)T_\mathrm{s}] = \eta\,(mT_\mathrm{s}) - W(mT_\mathrm{s}) \quad \mod \quad 1\,. \tag{3.11}$$

Its role is to provide the p-th *Polyphase MF* with the k-th *basepoint index* $m_k^{(p)}$ and the k-th *fractional index* $\mu_k^{(p)}$, and label the corresponding *interpolant* either as *optimum* or *valid*. This can be accomplished according to the following steps:

1. Define the following vectors of ordered threshold values: $I_\mathrm{O} \triangleq [0.5, 0, -0.5]$, $I_\mathrm{V} \triangleq [0.75, 0, 25, -0.25, -0.75]$, and $I_t \triangleq 1-0.25\cdot t \quad (t=1,\ldots,7)$. Odd and even values of t refer to elements in I_V and I_O, respectively.

2. Determine the number of intersections between the NCO register and the horizontal thresholds in I_t. Define $I_{t'}$ as the vector containing the elements of I_t that intersect the content of the NCO. If an intersection occurs in the p-th interval of $\mathscr{I}_p$, the NCO generates a control signal $\ell_2^{(p)} = 1$ to notify the p-th *Polyphase MF* that a new sample should be computed. Otherwise, if in the p-th interval there are no intersections, the NCO will generate a control signal $\ell_2^{(p)} = 0$ to notify the p-th filter that the corresponding *interpolant* should not be computed. Intersections with thresholds in I_{O} and I_{V} are associated to *optimum* and *valid* samples, respectively. The last element of $I_{t'}$ belonging to I_{O} is associated to the sample computed at instant $kT + \hat{\tau}$.

3. For each intersection found at the previous step, compute the timing estimates $\mu_k^{(p)}$. From geometrical considerations, it can be seen, for example by looking at Fig. 3.5, that
$$\frac{W(mT_{\mathrm{s}})}{P} = \frac{\eta(mT_{\mathrm{s}}) - I_{t'}}{\mu_k^{(p)}}. \tag{3.12}$$
Thus, the p-th timing estimate can be computed as
$$\mu_k^{(p)} = \frac{\eta(mT_{\mathrm{s}}) - I_{t'}}{W(mT_{\mathrm{s}})} \cdot P. \tag{3.13}$$

In Figs. 3.5 and 3.6 we report a graphic representation on how to compute the timing estimates $\mu_k^{(p)}$ and how to assign labels to the samples, for two different scenarios. In Fig. 3.5, there are 2 intersections between the NCO and the vector I_{O}, i.e. 2 *optimum* samples, and 2 intersections between the NCO and the vector I_{V}, i.e. 2 *valid* samples. Therefore, $I_{t'} = [0.5,\ 0.25,\ 0,\ -0.25]$ and $(\ell_1^{(1)},\ \ell_1^{(2)},\ \ell_1^{(3)},\ \ell_1^{(4)}) = (2,\ 1,\ 0,\ -1)$. In Fig. 3.6, instead, there are 2 intersections with vector I_{O} and 1 intersection with vector I_{V}, i.e. 2 *optimum* samples and 1 *valid* sample. Therefore, $I_{t'} = [0.5,\ 0.25,\ 0]$ and $(\ell_1^{(1)},\ \ell_1^{(3)},\ \ell_1^{(4)}) = (2,\ 1,\ 0)$. We can see that in the interval $[(m+1)T_{\mathrm{s}},\ (m+2)T_{\mathrm{s}}]$ there are no intersections between the NCO and the thresholds in I_t. In this case the NCO will generate a control signal $\ell_2^{(2)} = 0$, so that the the second *interpolant* will not be computed.

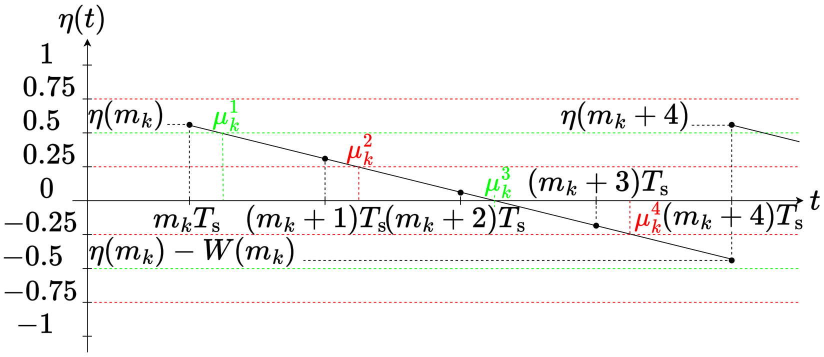


Figure 3.5: Timing estimates computation: 4 intersections corresponding to 2 optimum samples (green) and 2 valid samples (red).

We note that the computation of the timing estimates $\mu_k^{(p)}$ requires a run time division by $W(mT_s)$, as can be seen from (3.13). The control word $W(mT_s)$ is a time varying quantity, since it depends on the output of the loop filter $v(mT_s)$. Since the division by a time varying quantity can create problems in the hardware implementation, we propose a different method to compute the values of $\mu_k^{(p)}$. Since $v(mT_s)$ is the filtered error signal, it will tend to zero once the loop has converged. Therefore, we replace $W(mT_s)$ with the constant value $2/N$, i.e. we assume that $W(mT_s) = 2/N + v(mT_s) \approx 2/N$.

Indicating with $\tilde{\mu}_k^{(p)}$ the new timing estimates, equation (3.13) becomes

$$\tilde{\mu}_k^{(p)} = \frac{\eta(mT_s) - I_{t'}}{2/N} \cdot P. \tag{3.14}$$

The relation between $\mu_k^{(p)}$ and $\tilde{\mu}_k^{(p)}$ can be derived as follows:

$$\begin{aligned}
\mu_k^{(p)} &= \frac{\eta(mT_s) - I_{t'}}{W(mT_s)} \cdot P = \frac{\eta(mT_s) - I_{t'}}{\frac{2}{N} + v(mT_s)} \cdot P \\
&= \frac{\eta(mT_s) - I_{t'}}{\frac{2}{4N}} \cdot \frac{\frac{2}{N}}{\frac{2}{N} + v(mT_s)} \\
&= \tilde{\mu}_k^{(p)} \cdot \frac{1}{1 + v(mT_s)\frac{N}{2}} \approx \tilde{\mu}_k^{(p)} \cdot \left(1 - \frac{N}{2} \cdot v(mT_s)\right),
\end{aligned} \tag{3.15}$$

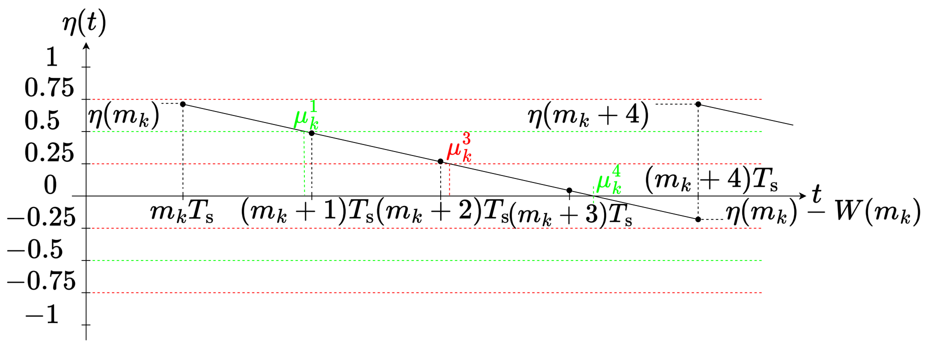


Figure 3.6: Timing estimates computation: 3 intersections corresponding to 2 optimum samples (green) and 1 valid samples (red).

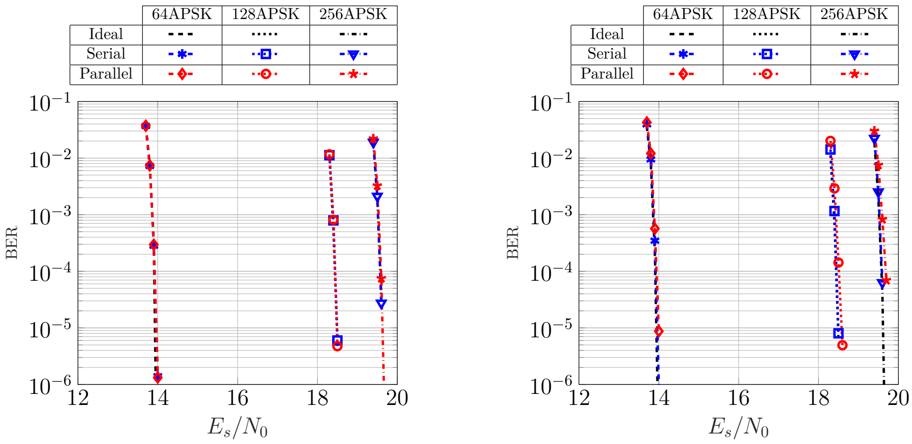
where the last approximation exploits a first order Taylor expansion. In the next section, we compare the performance of the proposed parallel architecture with the classical timing recovery scheme.

3.4 Numerical Results

In this section, we present the numerical results on timing synchronization. We evaluate the performance of different STR architectures, in terms of bit error rate (BER), under different channel conditions. The performance is compared with that achievable by a receiver operating in the absence of timing offset, i.e., $\tau = 0$, denoted as *Ideal* in the figures that follow.

We selected three DVB-S2X ModCods (64APSK with rate $128/180$, 128APSK with rate $140/180$, and 256APSK with rate $135/180$). Other modulation formats with lower cardinality have also been considered in our analysis, but the corresponding results will not be reported in this thesis, as the conclusions we can draw from them are identical to the ones presented here. Moreover, we are interested to focus the analysis on high order modulations, as they are more susceptible to timing offsets.

We adopt the system model introduced in Section 3.1. We assume perfect carrier frequency offset compensation before the LDPC decoder, which can be achieved by



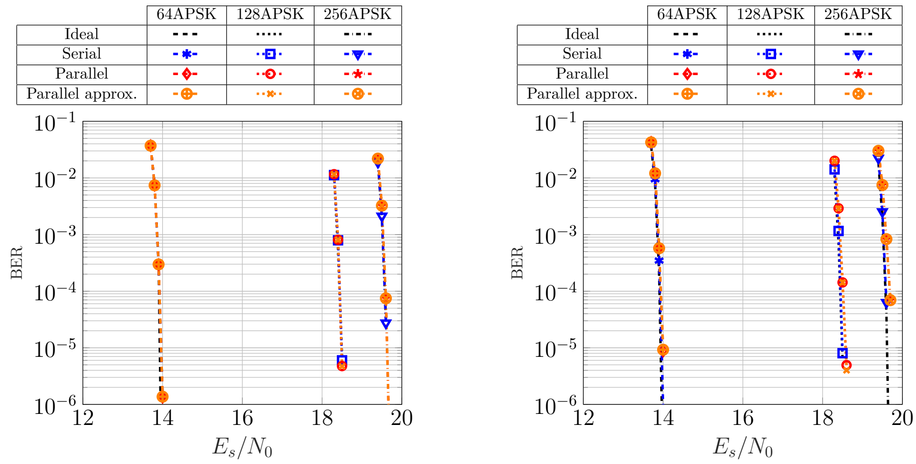
(a) $\Delta f T = 0$, SCO = 0 ppm.

(b) $\Delta f T = 0.01$, SCO = 100 ppm.

Figure 3.7: Comparison between serial and parallel algorithms, floating-point simulations.

fine carrier frequency estimation and compensation. We set the oversampling factor $N = 2.25$. A sampling clock offset between the transmitter and the receiver has also been considered, i.e. $N \neq T/T_s$, to simulate the behavior of non ideal oscillators, as indeed happens in real systems.

Fig. 3.7 shows the BER for serial and parallel timing recovery for different scenarios. In Fig. 3.7 (a) only a random timing offset τ is present, i.e., perfect sampling clock and carrier frequency recovery were assumed. In Fig. 3.7 (b), instead, we consider the presence of a residual carrier frequency offset $\Delta f T = 0.01$, and a non ideal sampling clock frequency, i.e. SCO $= 100$ part-per-million (PPM). For both of these scenarios, we can observe that both serial and parallel algorithms achieve a performance close to the ideal, for all the modulation formats. More importantly, we can see that the parallel algorithm shows no performance losses with respect to the serial architecture. Hence, we can conclude that the proposed parallel algorithm achieves the same performance as the classical serial implementation, for all the considered scenarios.



(a) $\Delta fT = 0$, SCO = 0 ppm.

(b) $\Delta fT = 0.01$, SCO = 100 ppm.

Figure 3.8: Comparison between serial and parallel algorithms, floating-point simulations.

The performance of the parallel algorithm reported in Fig. 3.7 has been obtained using Eq. (3.13) to compute timing estimates μ_k. We also carried out simulations to evaluate the performance when the approximation derived in Eq. (3.15) is employed to compute the values of μ_k. The corresponding results are reported in Fig. 3.8, and they show that the approximation in (3.15) is very good in all the tested cases.

3.5 Fixed-Point Analysis

Fixed-point analysis [37] is the process of evaluating how a digital signal processing algorithm behaves when numerical values are represented with a fixed number bits [38]. Fixed-point analysis is a fundamental step in the design and optimization of an algorithm, especially when working with embedded systems, real-time applications, or resource-constrained environments

The main reason for adopting fixed-point representation in digital signal processing algorithms is the efficient use of hardware resources. Many specialized signal processing chips, application specific integrated circuit (ASICs), and field programmable gate array (FPGAs) utilize fixed-point arithmetic due to its simplicity and resource efficiency. Understanding how the algorithm performs in a fixed-point representation is essential to optimize the hardware implementation. For these reasons, we carried on a fixed-point evaluation of the performance of the proposed algorithms, as it is a necessary step prior practical implementation of such algorithms.

Fixed-point numbers are represented in terms of word length and fractional length by defining the total number of bits allocated to represent the number, and how those bits are divided between the integer part and the fractional part. The word length (WL) is the total number of bits used to represent the fixed-point number. It includes both the integer and fractional parts, as well as the sign bit if the representation is signed. The integer word length (IWL) is the number of bits allocated for the integer part of the fixed-point number, which represents the whole number portion. It does not include the sign bit if it is a signed representation. The fractional word length (FWL) is the number of bits allocated for the fractional part of the fixed-point number, which represents the fractional or decimal portion. In practice, the word length can be expressed as the sum of the integer word length and the fractional word length, plus a possible bit for the sign. By adjusting the values of IWL and FWL, we can control the precision and range of the fixed-point representation to suit the requirements of a specific application.

We point out that the fine optimization of the number of bits used for the quantization is not the objective of this thesis. Instead, the goal is to verify that, under specific quantization values, the algorithms do not show quantization losses compared to the floating-point versions. The number of bits used for the quantization are reported in Table 3.1, in terms of word length, integer word length and fractional word length, for different system parameters. These values have been selected considering the range of representation of the floating-point parameters, and they are realistic and reasonable for FPGA hardware implementation.

Fig. 3.9 shows the BER curves derived from fixed-point simulations for the identical systems studied in Fig. 3.7. It is evident from these curves that, within the specific quantization parameters considered, there are no discernible quantization losses

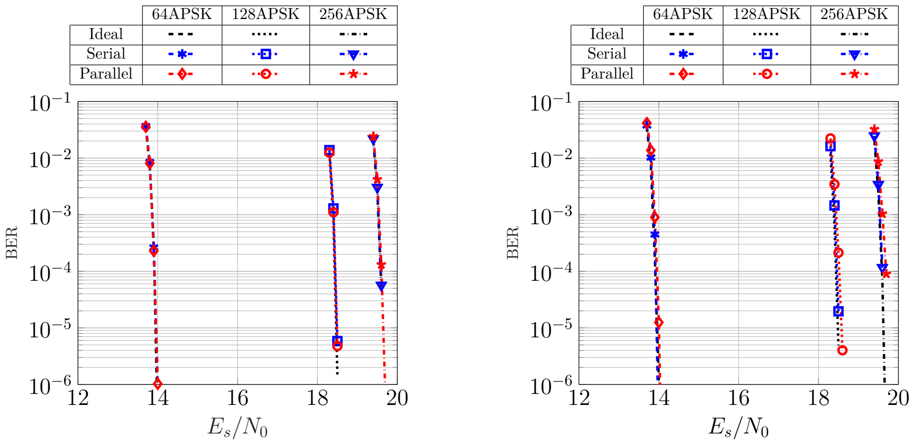
Table 3.1: Number of bits used for quantization in terms of WL, IWL and FWL. For all the parameters, 1 bit for the sign is considered.

Parameter	WL	IWL	FWL
$r(mT_s)$	12	2	9
$z(mT_s)$	13	2	10
$h(mT_s)$	12	2	9
$\eta(mT_s)$	27	1	25
$W(mT_s)$	20	1	18
$e(mT_s)$	14	3	10
μ_k	32	3	28

when compared to the floating-point case. As we did for the floating-point analysis, we also evaluated the fixed-point performance of the approximation in Eq. (3.15). The corresponding results are reported in Fig. 3.10, and also in this case we can conclude that this approximation works well for all the tested cases.

3.6 Conclusions

We proposed a new closed-loop symbol timing synchronization scheme suitable for high data rates digital receivers. We started the design from a serial timing recovery loop, deriving an equivalent low-complexity parallel architecture. The most important feature of the proposed approach is that it utilizes a single NCO, unlike existing parallel schemes in the literature. The algorithms performance was assessed in terms of BER, in the presence of both a residual carrier frequency offset and a sampling clock offset, for different DVB-S2X ModCods. To evaluate the effectiveness of our approach, we performed numerical simulations to compare serial and parallel implementations. Our results indicate that, across all tested scenarios, the proposed parallel algorithm consistently matches the performance of the original serial implementa-

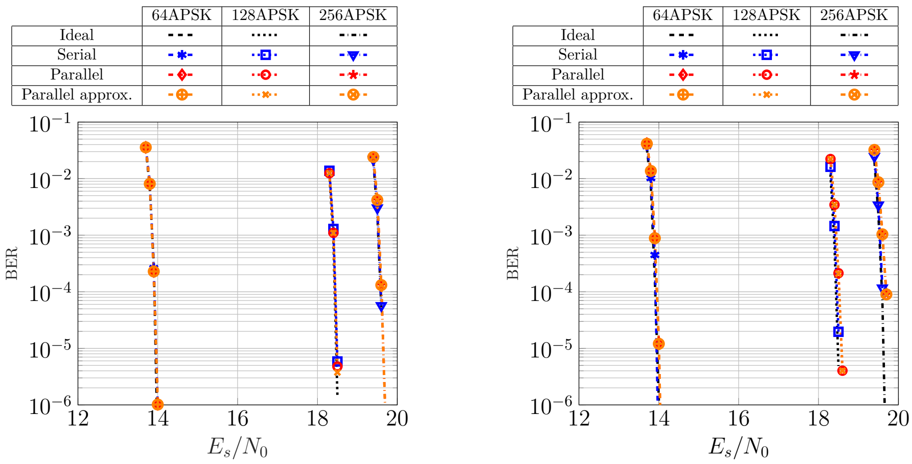


(a) $\Delta fT = 0$, SCO = 0 ppm.

(b) $\Delta fT = 0.01$, SCO = 100 ppm.

Figure 3.9: Comparison between serial and parallel algorithms, fixed-point simulations.

tion. In addition, we also conducted fixed-point simulations to examine the impact of quantization, to assess potential hardware implementation limitations. Our findings demonstrate that, under realistic quantization settings, the performance of the algorithm remains identical to that achieved with floating-point simulations.



(a) $\Delta fT = 0$, SCO = 0 ppm.

(b) $\Delta fT = 0.01$, SCO = 100 ppm.

Figure 3.10: Comparison between serial and parallel algorithms, fixed-point simulations.

Chapter 4

Phase Noise Compensation

Phase noise is one of the major channel impairments in digital communications. It can arise from instabilities of local oscillators [39], either at the transmitter or at the receiver, and can significantly degrade the system performance. For this reason, PN effects have been widely studied in the literature [40, 41, 42, 43, 44]. The effects of phase noise become more and more critical as the modulation order increases. PN causes random phase rotations in the transmitted carrier signal. These phase variations affect the precise positioning of the constellation points. In APSK, the points are distributed both in a circular and radial pattern. Phase noise can cause random phase rotations, leading to symbol misinterpretation, as can be seen from Fig. 4.1. In this chapter, we focus our analysis on phase noise recovery for high order modulations, as the ones proposed in DVB-S2X. We compare the performance achievable by different algorithms, for different channel scenarios. Then, we also propose a simple parallel architecture that can be used in high speed applications, and we compare its performance with that of classical serial schemes.

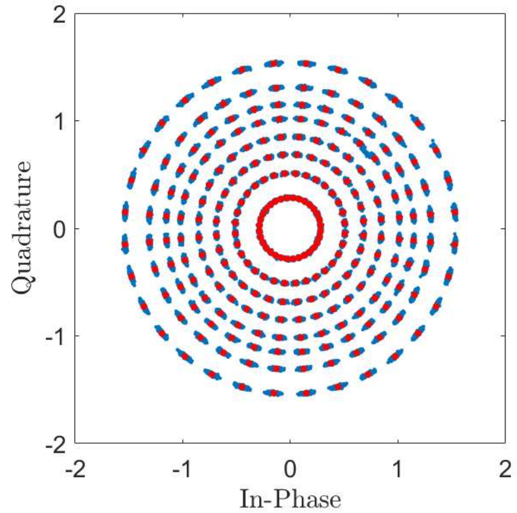


Figure 4.1: Phase noise effect on 256APSK.

4.1 System Model

In this section, we investigate estimation techniques to compensate the effect of phase noise. Let the discrete time received signal be expressed as

$$r_k = a_k e^{j\theta_k} + n_k\,, \tag{4.1}$$

where a_k is the transmitted symbol, θ_k is the phase noise sample, and n_k is a complex additive white Gaussian noise noise sample. We adopt the same system model as in Chapter 3. We assume that the timing offset τ has been perfectly recovered by the STR block, as well as possible effects of SCOs.

We consider a scenario where the phase estimator operates on a sequence where 36 pilots symbols have been periodically inserted every 1440 information symbols, as specified by the DVB-S2X standard [4]. The number of information symbols is set such that an integer number of pilot blocks can be inserted in each transmitted packet. Therefore, each packet starts and terminates with a block of 36 pilots symbols.

First, a Data Aided (DA) coarse phase estimation is performed exploiting the

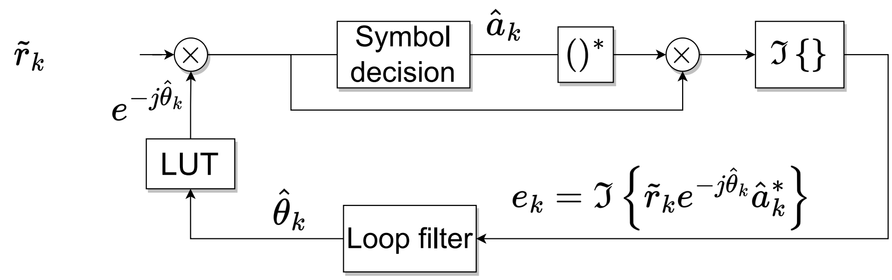


Figure 4.2: Block diagram for DD-PLL phase noise compensation.

pattern of known pilots. The phase estimate is computed as

$$\hat{\theta}_k = \arg\left(\sum_i r_i p_i^*\right) , \tag{4.2}$$

where the index i spans over the entire block of pilots, and p_i identifies the i-th pilot symbol. Once coarse estimates are available, linear interpolation between two consecutive estimated phases is performed. Then, samples r_k are de-rotated to obtain the new sequence $\tilde{r}_k = r_k e^{(-j\hat{\theta}_k)}$. We call this first strategy as *Linear interp.* in the following figures. In order to improve the phase noise compensation, a Decision Directed (DD) Phase Locked Loop (PLL) is then employed over the de-rotated samples $\tilde{r}_k$. Then, an error signal is generated as

$$e_k = \Im\left\{\tilde{r}_k e^{-j\hat{\theta}_k}\hat{a}_k^*\right\} . \tag{4.3}$$

The block diagram for DD-PLL phase recovery is reported in Fig. 4.2 [29]. To cope with a possible residual carrier frequency offset after coarse frequency synchronization, a second order PLL is employed. We denote this second approach as *Linear interp. + PLL* in the following figures.

In Fig. 4.3 we report the block diagram for parallel phase noise recovery, for a parallelization factor $P = 4$. The received samples, after serial to parallel conversion, are processed in parallel at a slower clock rate. At each clock cycle, P samples enter the feedback loop. All the P parallel samples are de-rotated using the same phase estimate $\hat{\theta}_k$. The expression for the p-th error signal can be expressed as

$$e_{k,p} = \Im\left\{\tilde{r}_{k-p} e^{-j\hat{\theta}_k} \hat{a}^*_{k-p}\right\} \quad p = 0\ldots P-1\,, \tag{4.4}$$

where $\tilde{r}_{k-p}$ and $\hat{a}^*_{k-p}$ are the p-th received sample and the p-th symbol decision, respectively. At each clock cycle, P error signals are computed and then combined together to form a single error which controls the feedback loop, defined as

$$E_k = \frac{1}{P}\sum_p e_{k,p}\,. \tag{4.5}$$

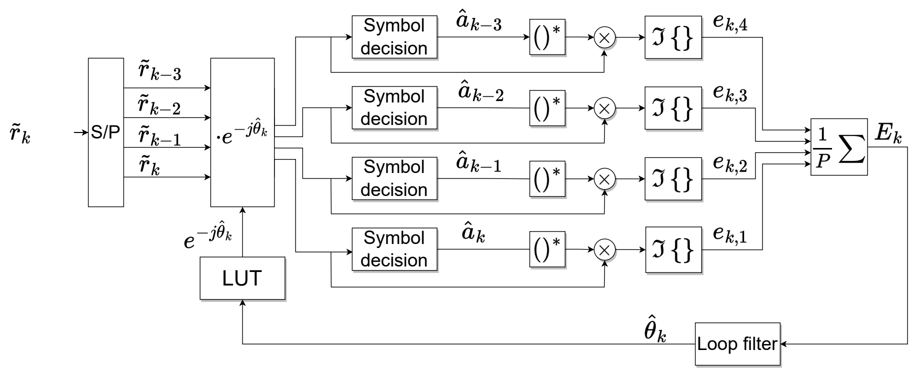


Figure 4.3: Block diagram for DD-PLL parallel phase noise estimation, $P = 4$.

4.2 Synchronization Strategies

Since the effects of phase noise are more critical for high order modulations, in our analysis we considered 128APSK and 256APSK modulations with LDPC codes foreseen by the DVB-S2X standard [4]. In particular, we used LDPC codes with rate equal to $140/180$ and $135/180$ for 128APSK and codes with rate equal to $128/180$, and $135/180$ for 256APSK.

For the sake of completeness, we also considered a scenario where the entire sequence of transmitted symbols is assumed to be fully known at the receiver. Even

Freq.	10 Hz	100 Hz	1 KHz	10 KHz	100 KHz	1 MHz	10 MHz	100 MHz
PN 1	−50	−58	−71	−77	−97	−121	−140	−150
PN 2	−49.5	−57.3	−70	−76.6	−95.1	−118.7	−137.9	−149.1
PN 3	−49.1	−56.5	−69	−76.3	−93.2	−116.5	−135.8	−148.2
PN 4	−48.6	−55.8	−68	−75.9	−91.3	−114.2	−133.7	−147.3
PN 5	−48.2	−55.1	−67	−75.5	−89.4	−111.9	−131.6	−146.3
PN 6	−47.7	−54.4	−66	−75.2	−87.5	−109.6	−129.5	−145.5
PN 7	−47.3	−53.6	−65	−74.8	−85.5	−107.4	−127.5	−144.5
PN 8	−46.8	−52.9	−64	−74.5	−83.6	−105.1	−125.4	−143.6
PN 9	−46.4	−52.2	−63	−74.1	−81.7	−102.8	−123.3	−142.7
PN 10	−45.9	−51.5	−62	−73.7	−79.8	−100.5	−121.2	−141.8
PN 11	−45.5	−50.7	−61	−73.4	−77.9	−98.3	−119.1	−140.9
PN 12	−45	−50	−60	−73	−76	−96	−117	−140

Table 4.1: Tested phase noise power spectral densities.

if this is an ideal scenario not suitable for realistic applications, it gives us a lower bound of the performance achievable by a generic phase noise estimator. We will call this approach *DA*. The performance of all these proposed strategies will be compared with that achievable by a receiver operating in the absence of phase noise, which we denote as *ideal*.

The phase noise power spectral densities that we used for the simulations are reported in Tab. 4.1, and some of them are reported in Fig. 4.4. The phase noise is described by masks, that specify the frequencies and the maximum acceptable PN level at those frequencies, expressed in dBc/Hz, that guarantee the target performance. The phase noise masks that we considered in our analysis are similar to the ones suggested in [4], making them suitable to assess phase noise recovery algorithms in practical scenarios. The best phase noise mask is indicated with index 1, while the worst masks has index 12. For each phase noise mask, we computed the gap in dB at a BER = 10^{-4} between the case under test and the *ideal* receiver.

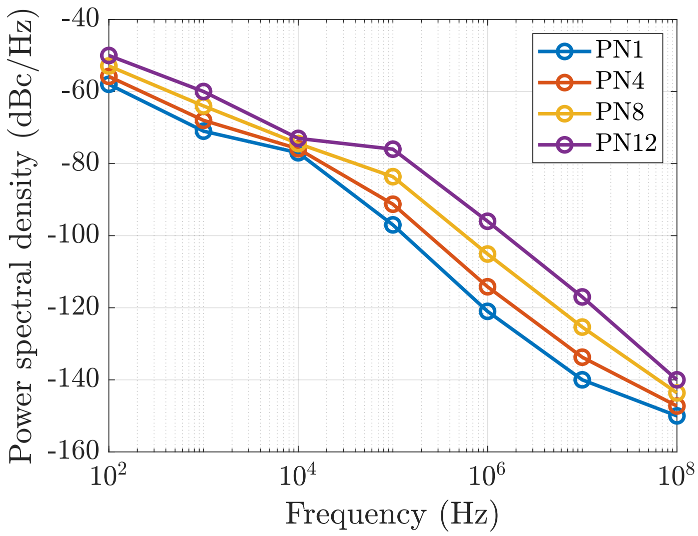


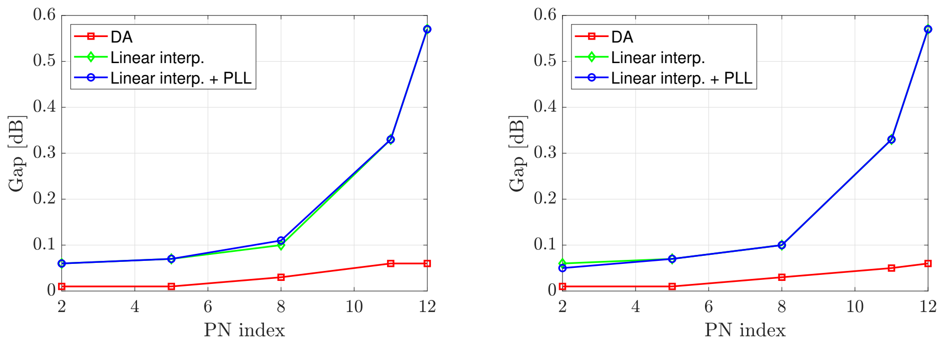
Figure 4.4: Power spectral densities for different phase noise masks.

4.3 Numerical results

We evaluate the performance of each algorithm by computing the gap, at a BER = 10^{-4}, from the *ideal* scenario which assumes perfect phase compensation. The PLL parameters have been properly optimized to minimize the BER.

In Figs. 4.5 and 4.6 we report the results obtained for 256APSK modulation with code rate $128/180$ and $135/180$, respectively, for different phase noise masks. As expected, we can see that the DA estimator achieves the best performance, minimizing the gap from the *ideal* case. For all the tested ModCods, we can observe that the DD PLL does not work properly. In fact, we can see that the same gap is obtained as for the case where only the linear interpolator is used. The same considerations hold for the results obtained for 128APSK modulation with code rate $140/180$ and $135/180$, which are reported in Fig. 4.7 and 4.8, respectively. For all the tested ModCods, the gap from the *ideal* receiver is below 1 dB, for all the tested power spectral densities.

All the results presented so far in this chapter refer to a scenario where phase recovery has been performed assuming perfect timing synchronization. We now eval-



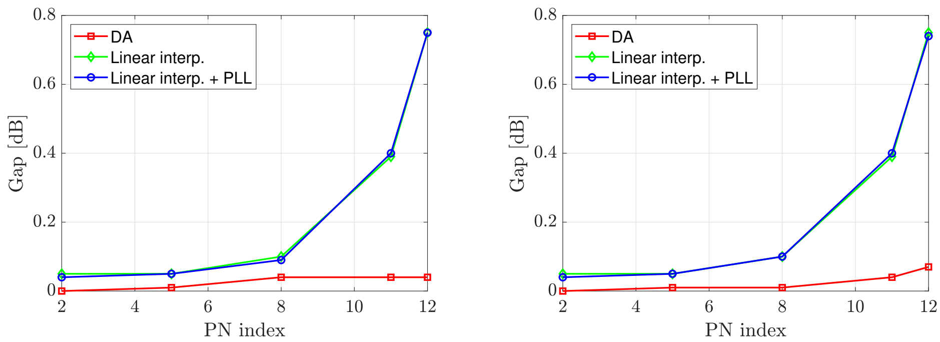
(a) Serial (b) Parallel

Figure 4.5: 256APSK rate 128/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms.

uate the performance of an overall system composed by the concatenation of the parallel STR scheme presented in Chapter 3 and the PN recovery block adopting the different strategies presented in this chapter. This analysis is carried on to evaluate possible implementation losses due to the concatenation of the two blocks.

We recall that in our analysis we assumed that the signal at the input of the STR block has a fractional oversampling factor, equal to $N = 2.25$. A rate conversion to an integer oversampling factor is performed by the STR through the NCO, as described in Section 3.2. For this reason, the results presented so far in this chapter are obtained assuming an integer oversampling factor, since this is the scenario at the output of the timing synchronizer.

The results obtained considering the concatenation of the STR and the phase recovery loop are reported, with dashed lines, in Fig. 4.9, for the case where no channel impairments, except for a timing offset $\tau \neq 0$, are present, and in Fig. 4.10, for the case where both a SCO and frequency offset are present. We point out that in this analysis, where we considered the concatenation of timing recovery and phase estimation, only a coarse optimization of the PLL parameters has been conducted. Observing the figures, we can see that the same gaps are obtained in the two dif-



(a) Serial

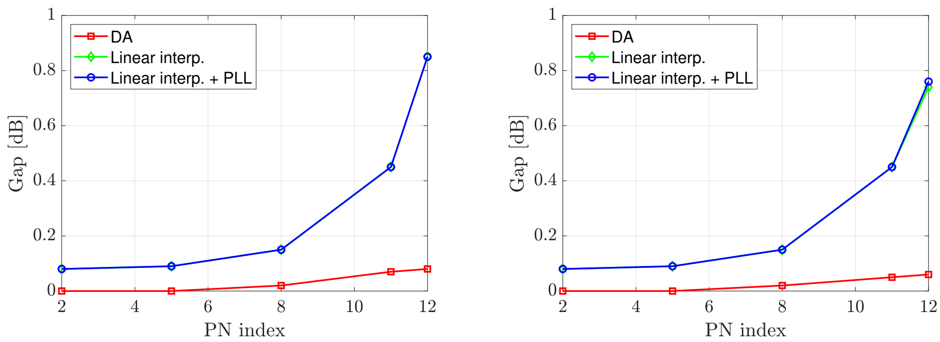
(b) Parallel

Figure 4.6: 256APSK rate 135/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms.

ferent scenarios. This confirms the fact that the proposed parallel STR works well in different channel scenarios, as we proved in Chapter 3 through BER analysis. In Figs. 4.9 and 4.10, for the sake of completeness, we also report, with solid lines, the gaps reported in Fig. 4.5, i.e. that achieved in a scenario where perfect timing synchronization was assumed. We point out that in this analysis, instead, a fine PLL parameters optimization was considered. Looking at the figures, we note that only a small difference, below 0.2 dB, can be seen for the worst phase noise masks, i.e. PN with indexes 11 and 12, in favour of the scenario depicted in Fig. 4.5

4.4 Alternative PN generation

In [45] an analytical model for the power spectral density (PSD) of the phase noise described by few parameters that are explicitly linked to the oscillator measurements is presented. The model has been used to successfully represent the PN typical of modern communication systems, among which phase noise specified in DVB-S2X standard. We performed the same analysis presented in the previous section, generating the PN samples according to the model presented in [45]. The results reported in



(a) Serial (b) Parallel

Figure 4.7: 128APSK rate 140/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms.

Fig. 4.11 show the same performance as those reported in Fig. 4.5, but obtained with a different generation of the phase noise samples. By comparing the two figures, we can conclude that almost identical results are obtained in the two cases. We point out that in Fig. 4.5 a fine optimization of the PLL parameters has been done, while for the results in Fig. 4.11 only a coarse optimization of the PLL parameters has been performed. Despite this, almost the same gaps from the *ideal* receiver can be achieved.

4.5 Conclusions

In this chapter, we addressed the problem of phase noise compensation in high order ModCods. As we did for the timing synchronization in Chapter 3, starting from the standard serial architecture [28], we proposed an equivalent parallel scheme suitable for high speed receivers. We evaluated the performance of different algorithms, focusing on high order ModCods foreseen by DVB-S2X standard and comparing different phase noise spectral densities. We showed that the best approach to achieve phase synchronization is obtained by performing linear interpolation between two

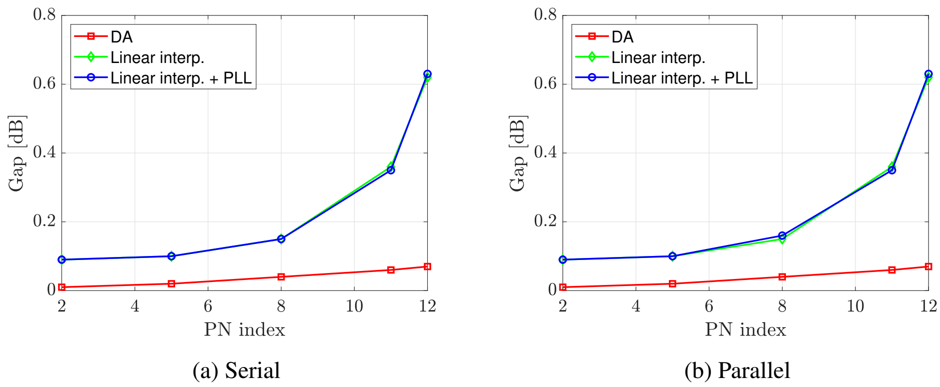
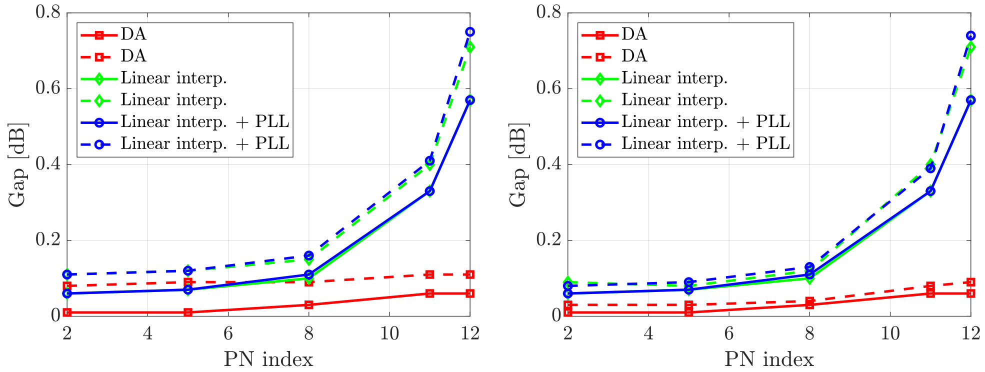


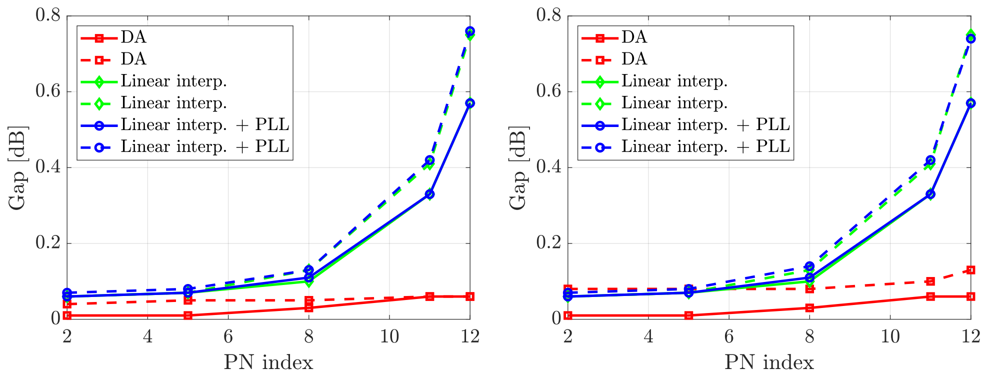
Figure 4.8: 128APSK rate 135/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms.

consecutive phases estimated over pilot fields according to Eq. (4.2). This method achieves very good performance for all the tested scenarios. Then, to verify a possible performance improvement, we also considered the scenario where a DD-PLL is employed in cascade of the linear interpolator. The results show that the PLL does not contribute to enhance the performance with respect to the case where only the linear interpolation is used.



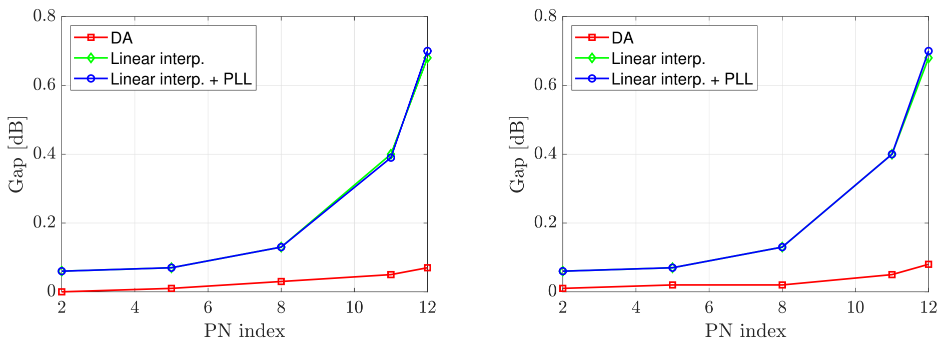
(a) Serial (b) Parallel

Figure 4.9: 256APSK rate 128/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms. $\Delta fT = 0$, SCO = 0 ppm. Perfect timing synchronization is assumed for solid lines, while it is NOT assumed for dashed lines.



(a) Serial (b) Parallel

Figure 4.10: 256APSK rate 128/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms. $\Delta fT = 0.01$, SCO = 100 ppm. Perfect timing synchronization is assumed for solid lines, while it is NOT assumed for dashed lines.



(a) Serial (b) Parallel

Figure 4.11: 256APSK rate 128/180, gaps from ideal receiver for (a) serial and (b) parallel algorithms. Phase noise generated using the alternative method.

Chapter 5

Constellation Design for Nonlinear Channels

High power amplifiers (HPA) are key components of many communication systems to ensure that the signals can keep a sufficient power level to reach their destination. As they represent the most power-intensive components in the entire transceiver chain, it is crucial to maximize their efficiency. This implies driving them as close to saturation as possible, even though this is also the region in which the amplifier characteristics exhibit the highest level of nonlinearity. Therefore, addressing this problem is of great importance, given the widespread adoption of power amplifiers in various contemporary communication scenarios and applications, including wireless and satellite links.

The problem of nonlinear power amplifiers has been widely studied, and multiple solutions have been proposed to mitigate the effects of the nonlinearity. The most commonly used refers to digital predistortion [46], which consists in the computation, by means of a training phase, of a new constellation to be used in transmission, in place of the real one. This solution, for example, is used in several satellite communication scenarios [47]. Signal predistortion is another option, feasible when it can be applied directly before the nonlinear amplifier, which works directly on the samples of continuous-time signals, rather than on the transmitted symbols [48, 49].

The two predistortion approaches can also be combined to further improve the performance [49].

Another approach to mitigate nonlinear effects is to properly design a constellation which, when used on a channel affected by a nonlinearity, is more robust to distortion with respect to conventional constellation formats, such as QAM. Novel constellation designs have been historically proposed often to improve the performance of communication systems [50] or to combat different channel effects, such as, for example, phase noise [51, 52]. Regarding nonlinear channels, satellite communication standards such as the digital video broadcasting, DVB-S2 [1] and its extensions DVB-S2X [4], foresee the use of APSK constellations, which have been designed to achieve satisfactory performance on typical nonlinear satellite channels [53]. In [54], the authors proposed a design, based on a simulated annealing algorithm, to maximize the pragmatic capacity, by jointly optimizing the position of the points and the bits-to-symbols mapping. In [55], a nonlinear mapping method has been proposed to design non equiprobable constellations for nonlinear satellite channels. An interesting APSK design has been proposed in [56], which is not specifically thought for nonlinear channels, but it relies on the choice of a single parameter to build the whole constellation.

Although effective, most of the design methods proposed in the literature suffer from a large complexity to obtain satisfactory performance. For instance, when considering high order APSK modulations, multiple parameters should be jointly optimized, such as the number of points on each circle, the radii of the rings and the phases of points on each ring, in order to achieve good performance, resulting in a complex optimization procedure. Even constraining the constellation to be symmetric, the number of parameters is still large, especially when the size of the constellation increases. Other optimization algorithms, whose complexity is usually high, often produce unstructured or asymmetric constellations, which are not suitable for a reduced complexity detection or a high speed hardware implementation. These drawbacks become more and more significant when the size of the constellation grows large. The DVB-S2X standard proposes APSK constellations with up to 256 points for high throughput applications. Indeed, high order constellations offer a potential

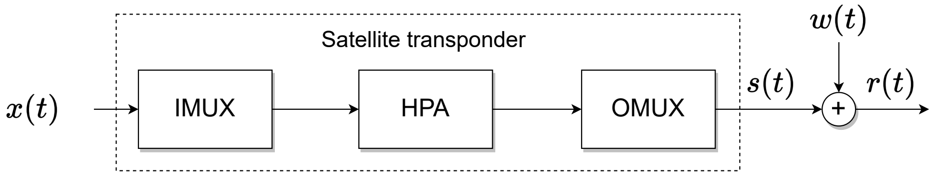


Figure 5.1: Block diagram of the satellite transponder.

solution to meet the growing demand for high data rates and spectral efficiency over bandlimited channels, but they are also the most sensitive to channel impairments. As such, it is of critical importance to develop new high order constellation design methods, ensuring good performance and a low design and detection complexity.

Recently, the design of spiral constellations for channels affected by PN has been considered in [52], where the constellation depends on a single parameter to be optimized, thus leading to a much simpler optimization procedure with respect to other works available in the literature.

Inspired by the design of [52], in this chapter we consider the design of spiral constellations for nonlinear channels. The proposed method relies also in this case on the optimization of a single parameter, as we will see in the next section, thus allowing a significantly lower complexity with respect other approaches.

Fig. 5.1 reports the block diagram of a typical satellite transponder, including an input multiplexer (IMUX) filter, which removes the adjacent channels interference, a HPA, and an output multiplexer (OMUX) filter aimed at reducing the spectral broadening caused by the nonlinear amplifier. For the sake of simplicity, for our proposed design method, we consider a simplified transponder model, not including IMUX and OMUX, and adopting an ideal nonlinearity, whose input-output relationship can be described by

$$s(t) = \begin{cases} x(t) & \text{if} \quad |x(t)| \leq 1 \\ 1 & \text{otherwise.} \end{cases} \tag{5.1}$$

This is clearly an approximation of more complex channel models, but sufficient to understand the behavior of the proposed constellations in nonlinear environments.

Let us consider the following system model. The complex envelope of the signal at the input of the satellite transponder can be expressed as

$$x(t) = \sum_k a_k p(t - kT), \tag{5.2}$$

where $\{a_k\}$ is a symbol belonging to the adopted constellation, $p(t)$ is the shaping pulse, assumed to have a RRC spectrum with roll-off factor $\alpha = 0.25$, T is the symbol interval, and $w(t)$ is a complex white Gaussian noise process. The received signal has expression

$$r(t) = s(t) + w(t), \tag{5.3}$$

where $s(t)$ is the signal at the output of the OMUX filter and $w(t)$ is Gaussian noise with power spectral density N_0.

5.1 Spiral Constellation Design

The proposed constellation construction is based on the observation that the nonlinearity affects more severely symbols with higher magnitude [57]. This effect can be visualized through the constellation scatter diagrams. In Fig. 5.2, the scatter diagram for a 64QAM transmitted through a nonlinear channel with large SNR is reported. The received samples (in blue) form clouds which are stretched in the radial direction and tend to become larger as the magnitude of the transmitted points increases. For this reason, a good constellation for nonlinear channels should be such that adjacent symbols in the radial direction have increasing distance for increasing magnitude of the symbols.

An Archimedean spiral is defined by the following equation

$$s(t) = te^{jt}, \tag{5.4}$$

and it has the property that consecutive spiral laps have the same distance from each other. In order to increase the distance between consecutive laps, as required for a nonlinear channel, we propose to place the constellation points along the following spiral

$$s(t) = t^n e^{jt}, \tag{5.5}$$

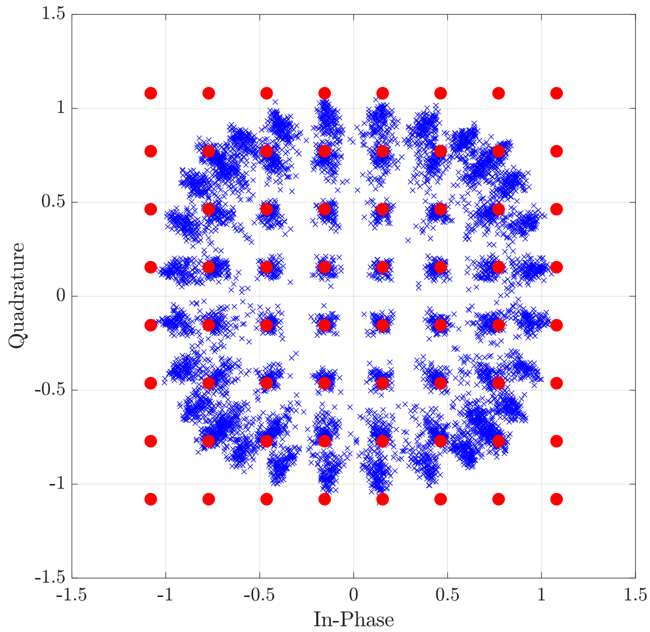


Figure 5.2: Effect of a nonlinearity on a 64QAM.

for $n > 1$. The radial distance between two consecutive laps of the above spiral is in fact an increasing function of the magnitude of the spiral. In particular, for $n = 2$, the distance increases with the square root of the modulus of the spiral. To show this, we define by $D_r(t)$ the radial distance between consecutive laps and compute it in the case $n = 2$

$$D_r(t) = |(t+2\pi)^2 - t^2| = (2\pi)^2 + 4\pi\sqrt{|s(t)|}\,. \tag{5.6}$$

In general, for $n > 1$, the distance increases with $|s(t)|^{\frac{n-1}{n}}$. Notice that, for $n = 1$, we obtain the Archimedean spiral, whose points in consecutive laps have constant radial distance.

We propose to place the constellation points along the spiral according to the following rule

$$c_m = t_m^n e^{jt_m}, \quad m = 1, \dots, M\,, \tag{5.7}$$

where M is the cardinality of the desired constellation and $t_1 < t_2 < \ldots < t_M$. In the following, we describe how to choose the parameters t_m. The idea is to choose an expression for t_m that guarantees that the distance between two consecutive points along the spiral is approximately equal to a constant D, that is

$$D = |c_{m+1} - c_m|\,, \quad m = 1, \ldots, M\,. \tag{5.8}$$

We replace (5.7) into (5.8) and obtain

$$|t_{m+1}^n e^{jt_{m+1}} - t_m^n e^{jt_m}| = D\,, \tag{5.9}$$

$$t_{m+1}^n \left| 1 - \left(\frac{t_m}{t_{m+1}}\right)^n e^{j(t_m - t_{m+1})} \right| = D\,. \tag{5.10}$$

We assume that $|t_m - t_{m+1}| \ll 1$ and use the approximations $e^{j(t_m - t_{m+1})} \simeq 1 + j(t_m - t_{m+1})$ and $t_m/t_{m+1} \simeq 1$ to obtain

$$t_{m+1}^n \left| -j\left(\frac{t_m}{t_{m+1}}\right)^n (t_m - t_{m+1}) \right| = D\,, \tag{5.11}$$

$$t_m^n (t_{m+1} - t_m) = D\,, \tag{5.12}$$

and finally we have the recursive expression for t_{m+1}

$$t_{m+1} = t_m + \frac{D}{t_m^n}\,. \tag{5.13}$$

If we equivalently define D as

$$D = |c_m - c_{m+1}|\,, \quad m = 1, \ldots, M\,, \tag{5.14}$$

following the same steps, we get

$$t_{m+1}^{n+1} - t_{m+1}^n t_m - D = 0\,, \tag{5.15}$$

and from this expression, assuming $t_0 = 0$, we can find a value for t_1

$$t_1 = D^{\frac{1}{n+1}}\,. \tag{5.16}$$

The spiral is defined by the parameter D, that we can choose as the one that maximizes the information rate (IR) or the pragmatic IR, as we will explain in the following section.

5.2 Performance Metrics

This section provides an overview of the figures of merit that will be utilized to assess the proposed constellations. The first performance metric that we will consider is the achievable information rate (AIR), which is defined as

$$I(\boldsymbol{x};\boldsymbol{y}) = \lim_{K\to\infty} \frac{1}{K} \mathrm{E}\left[\log_2 \frac{p(\boldsymbol{y}|\boldsymbol{x})}{\sum_{\boldsymbol{x}'} p(\boldsymbol{y}|\boldsymbol{x}')P(\boldsymbol{x}')}\right] \quad [\text{ bits / ch.use }], \tag{5.17}$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator, $p(\boldsymbol{y}|\boldsymbol{x})$ is the channel PDF and $P(\boldsymbol{x}')$ is the probability distribution of the transmitted symbols, assumed equal to $1/M$. The AIR can be interpreted as the average mutual information per transmitted symbol and thus represents the maximum amount of information bits per each transmitted symbol that can be reliably sent on the channel with the selected input alphabet. To achieve the highest information rate using the proposed spiral constellation, it is essential to appropriately optimize the parameter D. We consider the following optimization

$$D^* = \arg\max_{D} I(D)\,, \tag{5.18}$$

where I is computed as in (5.17), in which we have explicitly shown the dependence on the adopted constellation, defined by the parameter D.

An alternative theoretical measure is given by the pragmatic IR [58, 59], which represents an upper bound to the IR achieved by a practical ModCod. As suggested by the name, in this case the receiver follows a *pragmatic* approach, in the sense that it does not perform iterations between detector and decoder. The pragmatic IR depends on the adopted bit-to-symbol mapping, which determines how transmitted bits are mapped on the complex constellation symbols.

Commonly used constellations, such as PSK and QAM, implement Gray mapping, which guarantees that adjacent symbols differ by a single bit. APSKs usually adopt a quasi Gray mapping [4]. When the constellation does not have a regular structure, instead, the mapping must be determined through optimization, possibly jointly with the constellation points [54]. As an example, in [52] different mapping schemes for spiral constellations have been proposed. For the constellation design presented here, we chose to adopt a mapping which is Gray along the spiral, meaning that each

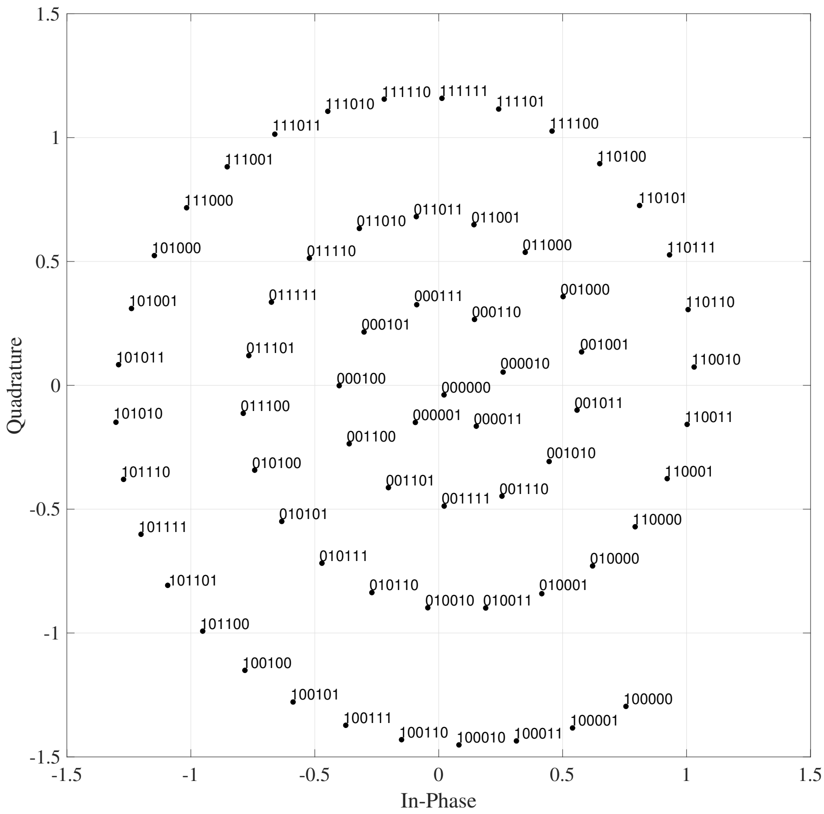


Figure 5.3: Adopted Gray mapping for the proposed spiral constellation with $M = 64$ and $D = 142$.

symbol differs by only one bit with respect to the previous and next symbols along the spiral. The rationale of this choice derives from the already mentioned characteristics that the distance between spiral laps increases with the magnitude. Hence, the most probable mistakes occur between adjacent symbols. An example of the adopted mapping is shown in Fig. 5.3 for a spiral with $M = 64$ and $D = 142$. Also in this case, an optimization of the parameter D is performed as in (5.18), with the function I replaced by the pragmatic IR. As a final performance measure, we will consider the packet error rate (PER) of coded transmission using the LDPC codes proposed in the DVB-S2X standard [4].

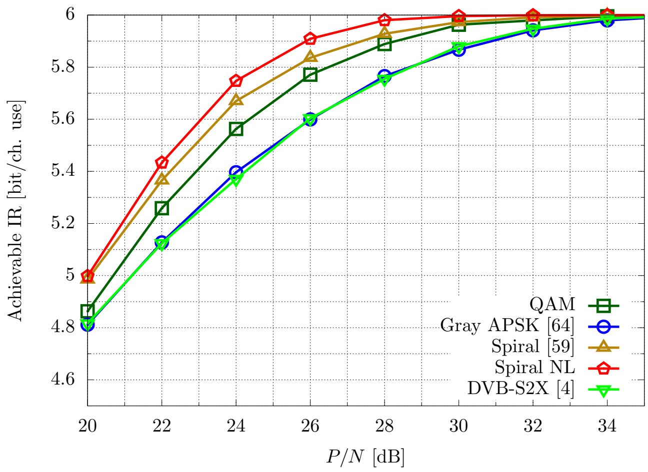


Figure 5.4: Achievable information rate for constellations with $M = 64$.

5.3 Numerical Results

In this section, we report numerical results obtained with the designed spiral constellation. We consider constellations with size $M = 64$ and 256, and compare different modulation formats, namely classical QAMs, APSKs foreseen by the DVB-S2X standard [4], the APSKs with Gray mapping proposed in [56], and the spiral constellations proposed for phase noise channels in [52].

Figs. 5.4 and 5.5 report the achievable IR for constellations with $M = 64$ and 256 points, respectively. For the proposed spirals, the parameter D has been selected to maximize the performance, as described in Section 5.2. For the spiral constellations of [52], the parameter f_s has been optimized as described in [52], and also for Gray APSKs we selected the optimal number of circles. Moreover, for all modulation formats, we optimized the input back-off of the amplifier. All curves in this section are reported as a function of the ratio between the saturation power of the nonlinear amplifier and the noise power, P/N. We notice that the proposed spirals ensure good achievable IR gains with respect to all the considered alternatives. These gains increase as the ratio P/N increases, and become more significant, in the order of 2-3 dB, for larger constellation size.

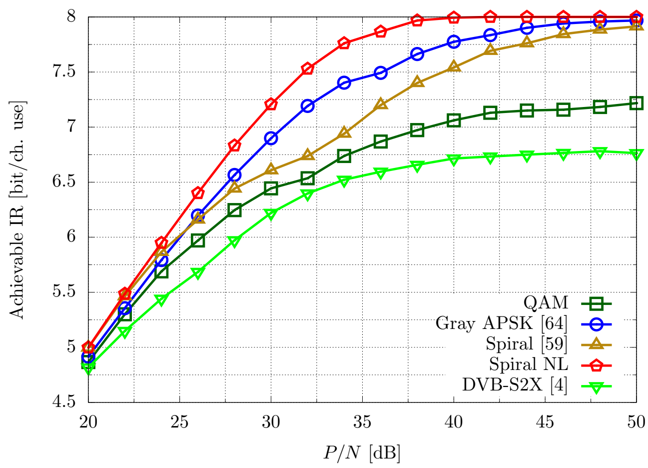


Figure 5.5: Achievable information rate for constellations with $M = 256$.

Figs. 5.6 and 5.7 show the pragmatic IR for the same cases already considered. The conclusions we can draw are similar, however the gains ensured by the proposed spirals are slightly reduced. This is because, as mentioned, the proposed bit-to-symbol mapping is not optimal. However, the gains are still significant, especially for $M = 256$ and medium-high P/N values.

Finally, in Figs. 5.8 and 5.9, we report the PER using three LDPC codes selected from the DVB-S2X standard, namely those with rates $r = 3/4$, $5/6$, and $9/10$ for $M = 64$ and 256, respectively. The parameter D of the spiral and the IBO are selected as those maximizing the pragmatic IR. We notice that these results clearly correspond to those foreseen by Figs. 5.6 and 5.7. For $M = 64$, we notice that the proposed spiral can be convenient only at very high code rates, while more interesting results can be observed for $M = 256$. In fact, for the code with rate $3/4$, which corresponds to an achieved IR equal to $r\log_2(M) = 6$ bit/ch. use, Gray APSK is only slightly better than the proposed spiral. For the higher code rates, instead, corresponding to an achieved IR of 6.67 and 7.2 bit/ch. use, respectively, spirals ensure increasing gains with respect to Gray APSKs. In particular, for code rate $r = 9/10$, we observe a gain of about 1.5 dB. We also note that QAM and DVB-S2X APSK are unable to converge for $r = 9/10$ (the corresponding curves are not shown in the figure). To conclude, in

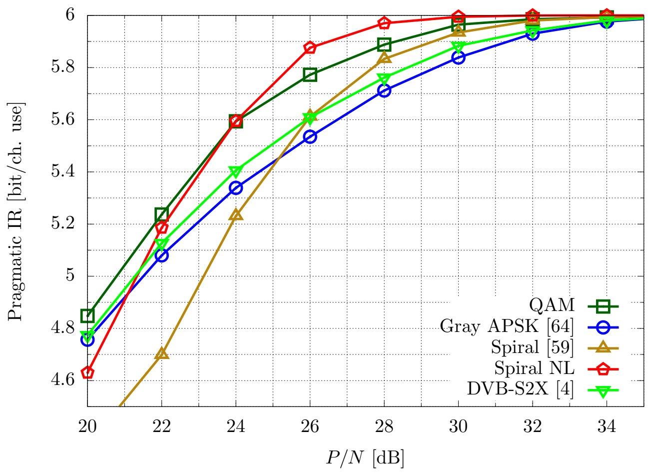


Figure 5.6: Pragmatic information rate for constellations with $M = 64$.

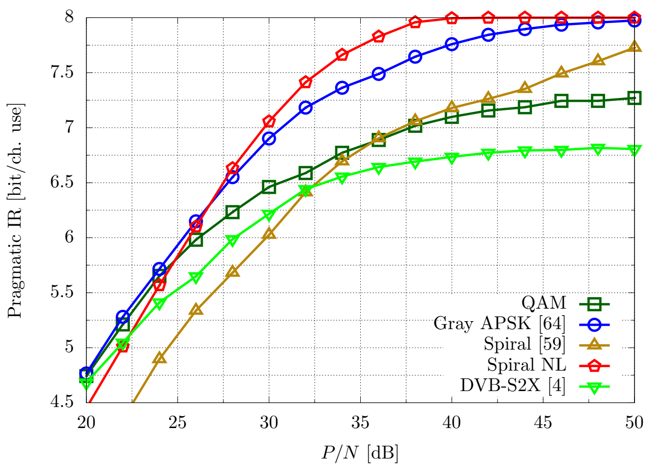


Figure 5.7: Pragmatic information rate for constellations with $M = 256$.

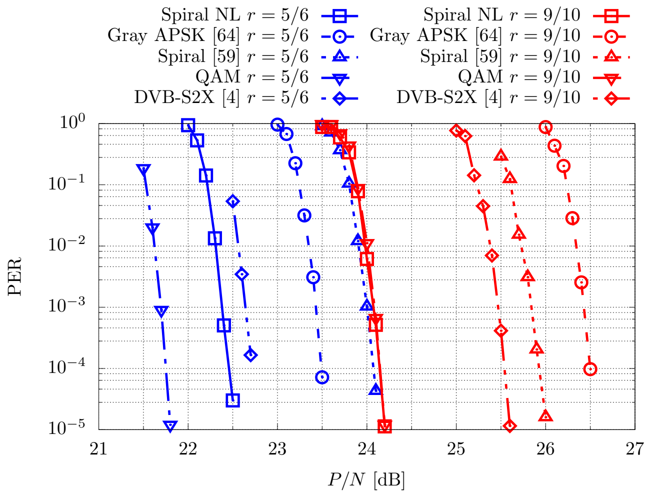


Figure 5.8: Packet error rate for constellations with $M = 64$.

Fig. 5.10 we report the spirals for $M = 256$ corresponding to the optimum values of the parameter D, for rates $r = 5/6$ and $r = 9/10$.

5.4 Conclusions

In this chapter, we address the problem of nonlinear effects, which is typical of communication links where HPAs are used, such as satellite or optical links. We propose the design of spiral constellations suitable for channel impaired by nonlinear effects. Such constellations rely on the optimization of a single parameter, which allows a significant complexity reduction with respect other approaches. We demonstrated that, compared to other modulation formats, they ensure good performance, especially when considering large constellation size and high SNR.

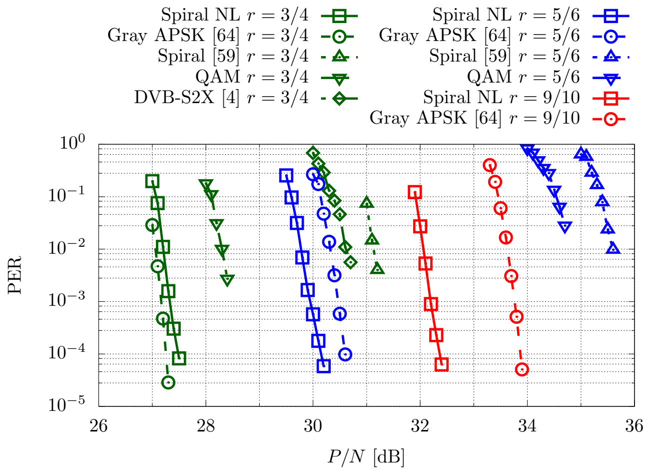
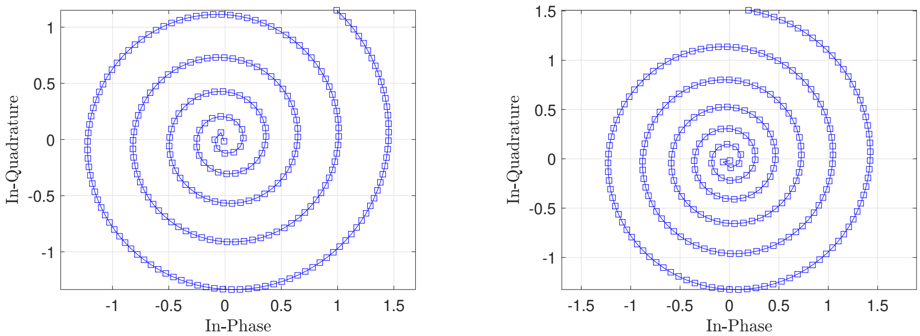


Figure 5.9: Packet error rate for constellations with $M = 256$.



(a) $M = 256$, $r = 5/6$, $D = 74$ (b) $M = 256$, $r = 9/10$, $D = 121$

Figure 5.10: Spirals corresponding to optimal value of parameter D.

Conclusions

In this thesis, we addressed the design of advanced algorithms for synchronization and parameter estimation in spectrally efficient satellite communications. The objective of this work is to propose new solutions to mitigate channel impairments that typically affect digital communication systems, and to compare them with established strategies from the existing literature.

We started by considering the problem of the PLH decoding in the DVB-S2X standard, since its accurate decoding is a crucial step after frame synchronization. We first considered an alternative approach to design variable length PLHs codes, that can be easily integrated in the DVB-S2X scenario, allowing greater flexibility. Then, we presented some novel techniques that can be applied in the decoding of PLH fields with unknown length. In Chapter 2, we demonstrated that such techniques achieve very good performance, and they can be used effectively adopted as valid alternatives to those provided in the standard.

Then, we addressed the problem of symbol timing synchronization, since the signal after frame synchronization could still be affected by a residual timing offset τ. In particular, we proposed a new closed-loop low-complexity parallel symbol timing synchronization scheme suitable for high data rates digital receivers, that can be easily implemented in hardware. In fact, it requires only a single NCO, as we have described in Chapter 3, unlike typical approaches proposed in the literature. We verified by numerical simulations that the proposed architecture allows to achieve the same performance as classical STR schemes, without any losses due to the parallelization, under different channel conditions. In addition, we also conducted a fixed-point performance evaluation, whose details are reported in Section 3.5, in order to validate the algorithm prior to a possible hardware implementation.

A similar analysis on the design of parallel algorithms has been conducted in Chapter 4, where we considered the effects that high order modulations suffer on channels affected by phase noise. We proposed a simple parallel scheme to mitigate phase noise effects which is suitable for high speed receivers. Different strategies have been compared, under different phase noise power spectral densities, for different DVB-S2X ModCods.

Finally, we have addressed channels impaired by nonlinear effects. A design of spiral constellations has been proposed which, unlike the approaches presented in the literature, allows a simpler optimization of the constellation parameters, as detailed in Chapter 5. We have then proved, through numerical simulations, that the proposed constellations can achieve very good results, considering different performance metrics.

Appendix A

ModCods performance on the AWGN channel

In this appendix, we show the Quasi Error Free (QEF) DER $= 10^{-5}$ thresholds for all the simulated ModCods, which are taken from [4].

Table A.1: DVB-S2X ModCods performance over AWGN channel.

ModCod	**Spectral Efficiency [bit/symbols]**	**Ideal $E_s N_0$ [dB]**
QPSK 2/9	0.434841	-2.85
QPSK 13/45	0.567805	-2.03
QPSK 9/20	0.889135	0.22
QPSK 11/20	1.088581	1.45
8APSK 5/9-L	1.647211	4.73
8APSK 26/45-L	1.713601	5.13
8APSK 23/36	1.896173	6.12
8APSK 25/36	2.062148	7.02
8APSK 13/18	2.145136	7.49
16APSK 1/2-L	1.972253	5.97

Continued on next page

Table A.1 – *Continued from previous page*

ModCod	**Spectral Efficiency [bit/symbols]**	**Ideal E_sN_0 [dB]**
16APSK 8/15-L	2.104850	6.55
16APSK 5/9-L	2.193247	6.84
16APSK 26/45	2.281645	7.51
16APSK 3/5	2.370043	7.80
16APSK 3/5-L	2.370043	7.41
16APSK 28/45	2.458441	8.10
16APSK 23/36	2.524739	8.38
16APSK 2/3-L	2.635236	8.43
16APSK 25/36	2.745734	9.27
16APSK 13/18	2.856231	9.71
16APSK 7/9	3.077225	10.65
16APSK 77/90	3.386618	11.99
32APSK 2/3-L	3.289502	11.10
32APSK 32/45	3.510192	11.75
32APSK 11/15	3.620536	12.17
32APSK 7/9	3.841226	13.05
64APSK 32/45-L	4.206428	13.98
64APSK 11/15	4.338659	14.81
64APSK 7/9	4.603122	15.47
64APSK 4/5	4.735354	15.87
64APSK 5/6	4.933701	16.55
128APSK 3/4	5.163248	17.73
128APSK 7/9	5.355556	18.53
256APSK 29/45-L	5.065690	16.98
256APSK 2/3-L	5.241514	17.24
256APSK 31/45-L	5.417338	18.10
256APSK 32/45	5.593162	18.59

Continued on next page

Table A.1 – *Continued from previous page*

ModCod	**Spectral Efficiency [bit/symbols]**	**Ideal E_sN_0 [dB]**
256APSK 11/15-L	5.768987	18.84
256APSK 3/4	5.900855	19.57

Bibliography

[1] ETSI EN 302 307-1 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 I: DVB-S2, available on ETSI web site (http://www.etsi.org).

[2] ETSI EN 300 421 Digital Video Broadcasting (DVB), Framing structure, channel coding and modulation for 11/12 GHz satellite services., available on ETSI web site (http://www.etsi.org).

[3] S. Benedetto, R. Garello, G. Montorsi, C. Berrou, C. Douillard, D. Giancristofaro, A. Ginesi, L. Giugno, and M. Luise, "MHOMS: high-speed ACM modem for satellite applications," *IEEE Wireless Communications*, vol. 12, no. 2, pp. 66–77, Apr. 2005.

[4] ETSI EN 302 307-2 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 II: S2-Extensions (DVB-S2X), available on ETSI web site (http://www.etsi.org).

[5] M. Eroz, L.-N. Lee, N. Loghin, U. De Bie, F. Simoens, and D. Delaruelle, "New DVB-S2X constellations for improved performance on the satellite channel," *International Journal of Satellite Communications and Networking*, vol. 34, no. 3, pp. 351–360, Sep. 2015.

[6] A. Morello and V. Mignone, "DVB-S2X: Extending DVB-S2 flexibility for core markets and new applications," *International Journal of Satellite Communications and Networking*, vol. 34, no. 3, pp. 327–336, Dec. 2015.

[7] K. Willems, "DVB-S2X demystified," Newtec, Tech. Rep., Mar. 2014.

[8] F. W. Sun, Y. Jiang, and L. N. Lee, "Frame synchronization and pilot structure for second generation DVB via satellites," *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 319–339, Jun. 2004.

[9] Q. Li, X. Zeng, C. Wu, Y. Zhang, Y. Deng, and J. Han, "Optimal frame synchronization for DVB-S2," in *2008 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seattle, WA, USA, May 2008, pp. 956–959.

[10] R. Pedone, M. Villanti, A. Vanelli-Coralli, G. E. Corazza, and P. Mathiopoulos, "Frame synchronization in frequency uncertainty," *IEEE Trans. Commun.*, vol. 58, no. 4, pp. 1235 – 1246, May 2010.

[11] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley, 1995.

[12] M. Villanti, P. Salmi, and G. E. Corazza, "Differential post detection integration techniques for robust code acquisition," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2172 – 2184, Nov. 2007.

[13] G. E. Corazza and R. Pedone, "Generalized and average likelihood ratio testing for post detection integration," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2159 – 2171, Nov. 2007.

[14] G. E. Corazza, R. Pedone, and M. Villanti, "Frame acquisition for continuous and discontinuous transmission in the forward link of satellite systems," *International Journal of Satellite Communications and Networking*, vol. 24, no. 2, pp. 185–201, Mar. 2006.

[15] N. Mazzali, G. Stante, S. Bhavani, and B. Ottersten, "Performance analysis of noncoherent frame synchronization in satellite communications with frequency

uncertainty," in *2015 IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Luxembourg, Nov. 2015, pp. 1–6.

[16] Z. Y. Choi and Y. Lee, "Frame synchronization in the presence of frequency offset," *IEEE Trans. Commun.*, vol. 50, no. 7, Jul. 2002.

[17] F. Kayhan and G. Montorsi, "Code design for adaptive coding and modulation indicator in broadcast systems," *International Journal of Satellite Communications and Networking*, vol. 34, no. 6, pp. 771–785, Aug. 2015.

[18] ——, "Code design for Frame-Headers in Adaptive Coding and Modulation systems," in 31*st AIAA International Communications Satellite Systems Conference*, Florence, Italy, Oct. 2013.

[19] ——, "Method and system for generating channel codes, in particular for a frame header," U.S. Patent 9:692.

[20] ——, "Code Design for Non-Coherent Detection of Frame Headers in Precoded Satellite Systems," in *9th Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Berlin, Germany, Sep. 2018, pp. 1–6.

[21] R. Knopp and H. Leib, "M-ary phase coding for the noncoherent AWGN channel," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1968–1984, Nov. 1994.

[22] H. Jin and T. Richardson, "Design of low-density parity-check codes for noncoherent MPSK communication," in *Proc. IEEE International Symposium on Information Theory*, Lausanne, Switzerland, Feb. 2002, p. 169.

[23] R.-Y. Wei, "Noncoherent block-coded MPSK," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 978–986, Jun. 2005.

[24] F. Kayhan and G. Montorsi, "Short-length rate-compatible code design for noncoherent detection," *International Journal of Satellite Communications and Networking*, vol. 40, no. 1, pp. 27–38, Dec. 2021.

[25] G. Colavolpe and R. Raheli, "Theoretical analysis and performance limits of noncoherent sequence detection of coded PSK," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1483–1494, Jul. 2000.

[26] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1681–1691, Nov. 2000.

[27] N. Kamel and V. Jeoti, "A Linear Prediction Based Estimation of Signal-to-Noise Ratio in AWGN Channel," *Etri Journal - ETRI J*, vol. 29, pp. 607–613, Oct. 2007.

[28] U. Mengali and A. D'Andrea, *Synchronization techniques for digital receivers*. Springer, 1997.

[29] M. Rice, *Digital communications: a discrete-time approach*. Pearson Education, 2009.

[30] X. Hao, C. Lin, and Q. Wu, "A Parallel Timing Synchronization Structure in Real-Time High Transmission Capacity Wireless Communication Systems," *Electronics*, vol. 9, no. 4, p. 652, Apr. 2020.

[31] X. Zhou and X. Chen, "Parallel implementation of all-digital timing recovery for high-speed and real-time optical coherent receivers," *Opt. Express*, vol. 19, no. 10, pp. 9282–9295, May 2011.

[32] X. Zhou, X. Chen, W. Zhou, Y. Fan, H. Zhu, and Z. Li, "All-Digital Timing Recovery and Adaptive Equalization for 112 Gbit/s POLMUX-NRZ-DQPSK Optical Coherent Receivers," *Journal of Optical Communications and Networking*, vol. 2, no. 11, pp. 984–990, Nov. 2010.

[33] F. J. Harris and M. Rice, ""Multirate digital filters for symbol timing synchronization in software defined radios"," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 12, pp. 2346–2357, Dec. 2001.

[34] F. M. Gardner, “A BPSK/QPSK Timing-Error Detector for Sampled Receivers,” *IEEE Trans. Commun.*, vol. 34, no. 5, pp. 423–429, May 1986.

[35] ——, “Phaselock techniques,” *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 170–171, Jan-Feb 1984.

[36] C. Lin, J. Zhang, and B. Shao, “A High Speed Parallel Timing Recovery Algorithm and Its FPGA Implementation,” in *2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing*, Wuhan, China, 2011, pp. 63–66.

[37] R. Yates, *Fixed-Point Arithmetic: An Introduction*, Sep. 2020.

[38] K. Cullen, G. Silvestre, and N. Hurley, “Simulation Tools for Fixed Point DSP Algorithms and Architectures,” *International Journal of Signal Processing*, vol. 1, pp. 199–203, Jan. 2005.

[39] A. Hajimiri and T. Lee, “A general theory of phase noise in electrical oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, pp. 179–194, Feb. 1998.

[40] F. Kayhan and G. Montorsi, “Constellation Design for Memoryless Phase Noise Channels,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2874–2883, May 2014.

[41] U. Mengali and M. Morelli, “Data-aided frequency estimation for burst digital transmission,” *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 23–25, Jan. 1997.

[42] G. Colavolpe, A. Barbieri, and G. Caire, “Algorithms for iterative decoding in the presence of strong phase noise,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 9, pp. 1748–1757, 2005.

[43] A. Barbieri and G. Colavolpe, “Soft-Output Decoding of Rotationally Invariant Codes Over Channels with Phase Noise,” in *2006 IEEE International Symposium on Information Theory*, Seattle, WA, USA, Jul. 2006, pp. 2854–2858.

[44] G. Colavolpe, "Communications over phase-noise channels: A tutorial review," *International Journal of Satellite Communications and Networking*, vol. 32, no. 3, pp. 167–185, Jul. 2013.

[45] A. Piemontese, G. Colavolpe, and T. Eriksson, "A New Analytical Model of Phase Noise in Communication Systems," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, TX, USA, Apr. 2022, pp. 926–931.

[46] G. Karam and H. Sari, "Analysis of predistortion, equalization, and ISI cancellation techniques in digital radio systems with nonlinear transmit amplifiers," *IEEE Trans. Commun.*, vol. 37, no. 12, pp. 1245–1253, Dec. 1989.

[47] E. Casini, R. D. Gaudenzi, and A. Ginesi, "DVB-S2 modem algorithms design and performance over typical satellite channels," *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 281–318, Jun. 2004.

[48] L. Ding, G. Zhou, D. Morgan, M. Zhengxiang, J. Kenney, K. Jaehyeong, and C. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. on Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[49] A. Ugolini, G. Montorsi, and G. Colavolpe, "Next generation high-rate telemetry," vol. 36, no. 2, pp. 327–337, Feb. 2018.

[50] G. Foschini, R. Gitlin, and S. Weinstein, "Optimization of two-dimensional signal constellations in the presence of Gaussian noise," vol. 22, no. 1, pp. 28–38, Jan. 1974.

[51] F. Kayhan and G. Montorsi, "Constellation design for memoryless phase noise channels," vol. 13, no. 5, pp. 2874–2883, May 2014.

[52] A. Ugolini, A. Piemontese, and T. Eriksson, "Spiral constellations for phase noise channels," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7799–7810, Nov. 2019.

[53] R. De Gaudenzi, A. Guillén i Fabregas, and A. Martinez, "Performance analysis of turbo-coded APSK modulations over nonlinear satellite channels," vol. 5, pp. 2396–2407, Sep. 2006.

[54] F. Kayhan and G. Montorsi, "Joint signal-labeling optimization under peak power constraint," *International Journal of Satellite Communications and Networking*, vol. 30, no. 6, pp. 251–263, Sep. 2012.

[55] W. Chen, Y. Peng, C. Han, and J. Yang, "Design of nonequiprobable high-order constellations over non-linear satellite channels," *Electronics*, vol. 9, no. 1, Jan. 2020.

[56] Z. Liu, Q. Xie, K. Peng, and Z. Yang, "APSK constellation with Gray Mapping," *IEEE Commun. Letters*, vol. 15, no. 12, pp. 1271–1273, Dec. 2011.

[57] M. Moghaddam, S. Aghdam, N. Mazzali, and T. Eriksson, "Statistical modeling and analysis of power amplifier nonlinearities in communication systems," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 822–835, Feb. 2022.

[58] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "Determining and approaching achievable rates of binary intersymbol interference channels using multistage decoding," *IEEE Trans. Inform. Theory*, vol. 53, pp. 1416–1429, Apr. 2007.

[59] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

Acknowledgements

Here we are at the acknowledgements, i.e., the only part of this thesis which comes from my heart rather than logic. This long journey came to an end, but a new beginning is on the horizon.

The first thanks goes to my parents and to my friends, for their continuous support in every day of my life.

I wish to thank my supervisors, Alessandro and Tommaso, for always being available and ready to help me during these years. I would like to express my sincere gratitude to Giulio, for his support and guidance, not only during the PhD, but from the very beginning.

Thanks to all the friends that I encountered along these years spent here at the university, for sharing with me this long journey, and making it much lighter.

A special thank goes to Alice, who in a short time has become so special. Thank you for being such a unique person, I hope to share with you many new adventures.