Robust Wireless Communications for Low Power Short Message Internet-of-Things Applications

by

Chin-Wei Hsu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan
2023

Doctoral Committee:

Associate Professor Hun-Seok Kim, Chair
Associate Professor Achilleas Anastasopoulos
Professor David Blaauw
Professor Trevor Mudge

Chin-Wei Hsu

chinweih@umich.edu

ORCID iD: 0000-0002-4817-8364

To my family and Elisa.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Professor Hun-Seok Kim for his mentorship throughout my PhD journey. Hun-Seok is a role model for me as a scientist and engineer. He is a creative person with many ideas in his head, and every discussion with him is inspiring. With his guidance, I gradually grew and eventually became an independent researcher. He is also a kind advisor who cares a lot about his students. I am grateful for every opportunity he gave me and extremely fortunate to be his student.

I have the honor to work with two outstanding professors, Professor David Blaauw and Professor Achilleas Anastasopoulos, on different projects. I worked with David in my first year as a fresh new PhD student. As a new participant in the project, I know very little, but eventually I was able to catch up to the work thanks to David and others' help. I thank David for believing in me and giving me an important role in the project to show my ability. I started to work with Achilleas at the beginning of my fourth year. We developed a new idea together that eventually became a critical part of my dissertation. I thank Achilleas for his guidance to deeper knowledge and leading me through many meaningful discussions. David and Achilleas are both smart and passionate people, and I learn a lot from them.

I would like to thank Professor Trever Mudge for serving on my dissertation committee. He is there to help me finish my last step of the journey with kindness.

The years being a member of DDH lab are the most fulfilling moments in my life. I met so many talented people here, and I thank every single one of them for

fighting together with me for the PhD degree. The lab is full of people with different backgrounds, so I have precious opportunities to learn from many experts. Especially, I would like to thank Li-Xuan, Zhen, Chien-Wei, Mingyu, Demba, Changwoo, Winston, Yao-Shan, Chenghong, and Andrea for their inspiring discussion with me throughout the years.

I also want to thank all my smart friends who pursue PhD degrees at the same time as me. It would be much harder if I were not to share my struggling time with them. Some of them already graduated and some are still working hard, but I have no doubt that all of them will be successful in the future. Especially, I would like to thank Yu-Heng, Eric, Jimmy, Wei-Kuan, Chia-Nan, Jie-Fang, James, Chin-Chia, Siang, Bo-Jui, and Yu-Sheng. All of them are my best friends for life.

My family are my biggest supports during these years. I thank my parents and sisters for their unconditional love and for always being there for me. They are the only reason I can be here today.

Last but not least, I want to thank my wife, Elisa. She is the star of my life. Without her, it wouldn't be possible for me to finish my PhD. Although she knows nothing about how to be a PhD, she knows everything about how to support a PhD.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

ACS add-compare-select

ADC analog to digital converter

AMP approximate message passing

AWGN additive white Gaussian noise

BER bit error rate

BLE Bluetooth low energy

BLER block error rate

BPSK binary phase-shift keying

CC convolutional codes

CDF cumulative distribution function

CFO carrier frequency offset

CRC cyclic redundancy check

DNN deep neural network

DSP digital signal processing

eMBB enhanced mobile broadband

FFT fast Fourier transform

FPGA field-programmable gate array

FWHT Fast Walsh–Hadamard transform

HDC hyper-dimensional computing

HDM hyper-dimensional modulation

HMM hidden Markov model

IC integrated circuit

INR interference-to-noise ratio

IoT Internet of Things

ISM industrial, scientific and medical

LDPC low density parity check

LL log-likelihood

LLR log-likelihood ratio

LOS line-of-signt

MA multiple access

MDP Markov decision process

MIMO multiple input multiple output

mMTC massive machine type communication

NAF normalized advantage function

NLOS non-line-of-sight

OSLA opportunistic symbol length adaptation

PAPR peak-to-average ratio

PDF probability density function

PER packet error rate

PHY physical

PLL phase-locked loop

PMF probability mass function

PN pseudo-random

PPM pulse position modulation

PSD power spectrum density

PSK phase-shift keying

PV photovoltaic

QAM quadrature amplitude modulation

OFDM orthogonal frequency-division multiplexing

QPSK quadrature phase-shift keying

RL reinforcement learning

ROVA eliability output Viterbi algorithm

RRC root-raised-cosine

RSC recursive systematic convolutional

RX receiver

SDNR signal-difference-to-noise ratio

SFO smapling frequency offset

SINR signal-to-interference-plus-noise ratio

SIR signal-to-interference ratio

SK Schalkwijk-Kailath

SNR signal-to-noise ratio

SPARC sparse superpotision codes

TBCC tail-biting convolutional codes

TX transmitter

URLLC ultra-reliable low latency communication

USRP Universal Software Radio Peripheral

VA Viterbi algorithm

WAVA wrap-around Viterbi algorithm

ABSTRACT

This dissertation focuses on robust wireless communication system designs for low power short message IoT applications, aiming to enhance the reliability of the transmission. IoT applications possess unique challenges due to stringent constraints on short message size, low latency, and low power. Short messages can no longer rely on capacity-achieving error correction codes, demanding exploration of new reliable transmission schemes to combat noise and interference. Low latency constraint poses challenges to system design when packet retransmission is not a viable means to boost reliability. Low power devices further limit the available resource at the transmitter, and may even result in unstable frequency and phase of the signal, bringing difficulties in detection and synchronization. In this dissertation, these challenges are tackled by proposing novel system and algorithm designs in three works. The first work introduces a novel non-orthogonal modulation scheme suitable for short-message packets in an interference-heavy channel. The second work proposes an instantaneous feedback-based variable symbol length transmission scheme that provides ultra-reliability for short packets with low latency. Lastly, a gateway receiving algorithm for frequency/phase unstable signal including packet detection, carrier/sampling frequency offset synchronization and demodulation for a low-power crystal-less system is presented in the third work. The three works presented in this dissertation provide solutions to IoT challenges and enhance the reliability of low power short message wireless communications, broadening the use cases of IoT applications and improving the robustness of IoT connectivity.

CHAPTER I

Introduction

1.1 IoT in 5G and Beyond

The concept of Internet of Things (IoT) has been around for decades*, however, not until the era of 5G do we start to see the exponential growth in IoT business. The rise in revenue is expected to continue, forecast to reach one trillion US dollars by 2030 [1]. This evergrowing trend relies on technology advancement in many aspects, one being communication. To support this vision, the International Telecommunication Union Radiocommunication Sector (ITU-R) categorizes the 5G and beyond technologies into three classes, as shown in Figure 1.1: enhanced Mobile Broadband (eMBB), massive Machine-Type Communication (mMTC), and Ultra-Reliable and Low Latency Communication (URLLC), while each targets different applications [2]. The eMBB focuses on enhancing the high data rate that has long been the main focus before 5G, whereas the other two classes target new usage scenarios that are closely related to IoT applications.

IoT applications behave largely differently from human-oriented applications that demand fast data rate and huge data amount. Message size in IoT applications is usually small and the data rate requirement is often relaxed. Instead, the applications

*Kevin Ashton, MIT's Executive Director of Auto-ID Labs, coined the phrase "Internet of Things" in 1999.

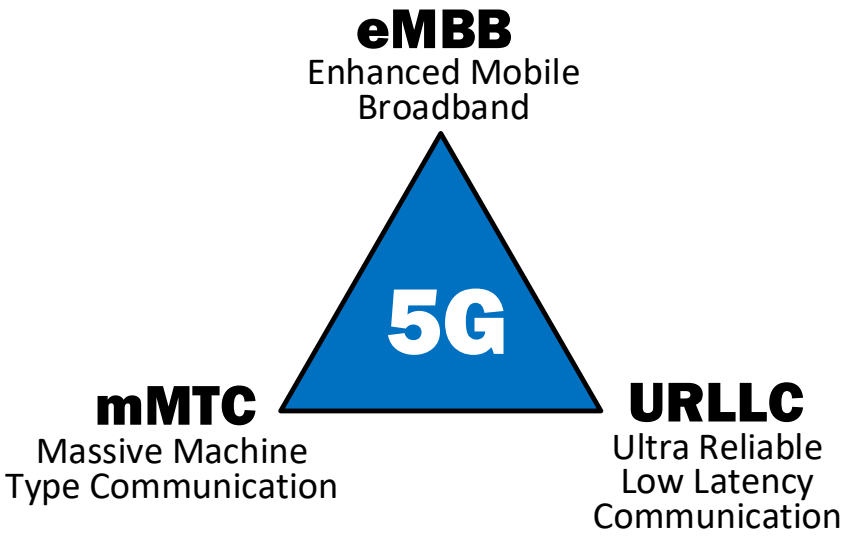


Figure 1.1: Three classes of 5G and beyond technologies.

emphasize other features such as large coverage, huge number of devices supported, or reliable connectivity with very low latency. Each feature comes with new challenges that are unique to specific applications, thus opening an unprecedented wide research area that is urged to be explored. In the subsequent sections, we discuss mMTC and URLLC in detail.

1.1.1 Massive Machine Type Communications

mMTC, as its name suggests, aims at handling the excessive information exchange between a massive number of devices in a network. The exchange message size is typically small such as a few bytes, only serving for machine operation purposes. However, the enormous number of devices existing in the same network can add up to a huge amount of data, making traffic management a challenging task. Reliability requirement for these kinds of applications may not be super high, but failure to provide satisfactory reliability can aggravate the traffic when each device is trying to resend the message, leading to a collapse of the whole network. Thus, how to deliver scalability while maintaining reliability for the network is the main focus for mMTC.

The mMTC use-case examples include logistic, transportation, utilities, health, environment, and security. For instance, tracking merchandise in a warehouse becomes much easier if a gateway can receive status report from every tracker on the merchandise. The status information conveys very little data such as the quantity of the remaining goods, and the information does not need rapid update, but the number of the trackers can be massive, and the coverage may also be large. Furthermore, counting on the battery on the sensor node, the power consumption of the tracker must be kept small to sustain a longer life span.

Existing technologies cannot provide a viable solution to fully satisfy the requirements for mMTC applications. For example, Bluetooth Low Energy (BLE) and Zigbee fail to provide long distance. LoRa and Sigfox can achieve large coverage but the scalability is questionable as they are vulnerable to packet collision. To realize emerging mMTC applications, novel system designs must be explored.

1.1.2 Ultra-Reliable Low Latency Communications

URLLC requires the system to satisfy stringent reliability and latency constraints at the same time. A typical definition that URLLC follows is to achieve at least 99.9999% reliability with end-to-end latency of less than 1 ms for a 32-byte packet. The strict constraints come from the nature of URLLC applications, which are extremely delay-sensitive and demand ultra-reliability as the information is used to make critical decisions. For a long time, it has been taken for granted to trade latency for higher reliability via longer codes or retransmission, as there exists a natural trade-off between two features. Satisfying both reliability and latency at the same time is the main challenge in URLLC.

Examples of URLLC applications include cloud connectivity, industrial automation and coordination between autonomous vehicles. For instance, autonomous vehicles can exchange status or control signals to assist self-driving decisions for more

efficient traffic control. The message only contains a few bytes, but the system cannot afford packet loss or outdated information, which will lead to misguided decisions and irreversiable harmful outcomes. Only when reliability and low latency can be delivered simultaneously, the application becomes useful.

In modern communication systems, high reliability usually comes from capacity-achieving codes that use long block length, together with data scrambling and interleaving to reduce the effect of channel fading and interference. However, when the message is small and the latency needs to be minimized, these techniques are no longer available, making high reliability difficult to attain. URLLC is a relatively new concept, so there are few theoretical frameworks combining latency and reliability together, not to mention practical system design at this time. The most important works are done by Polyanskiy [3], where bounds on block error rates for finite block length codes are derived.

1.2 Challenges in IoT Applications

As more and more IoT applications emerge and the number of IoT devices grows exponentially, the challenges for IoT demand solutions more than ever in the past. In this section, we list three of the most challenging problems that are critical to new IoT applications.

1.2.1 Reliability for Short Packets

The research in the domain of channel coding has never stopped since the seminal work of Claude Shannon [4]. The error correction codes evolve continuously and communication standards adopt new coding schemes every generation, from convolutional codes and turbo codes [5], to low-density parity check codes [6] and polar codes [7]. However, modern codes rely on extremely long block length to approach capacity, and the performance degrades as the block length gets shorter. New theo-

ries have been shown to characterize the performance limit for short length codes in the AWGN channel, in which normal approximation from [8] provides a closed-form expression to quantify the efficiency of codes with finite block length. Motivated by this, good short codes are under investigation [9], and decoding complexity as well as performance are compared.

In addition to noise, interference can cause more severe effects on short packet IoT applications when they can no longer rely on data scrambling/interleaving and diversity to improve robustness to sporadic interference. Evaluating the performance of codes under AWGN assumption may be misleading for interference-heavy channels. Therefore, novel short codes/packets design needs to take both noise and sporadic interference into consideration.

1.2.2 Low Latency Constraint

For delay-sensitive IoT applications, end-to-end latency defines the validity of their usage. At the physical layer, latency is the combination of transmission time, propagation delay and processing delay. While reduction in propagation delay is not easily attainable, it is usually negligible compared to the other two. On the other hand, processing delay such as decoding time can have a huge impact on the whole latency, and is highly dependent on the hardware and the algorithms used. Low complexity or highly parallelizable decoding algorithms are favorable to minimize the processing time.

Shortening the packet length can also reduce the latency. However, this will come with the price of sacrificing reliability because code length has a huge impact on performance. To overcome this challenge, feedback codes can be used as they can achieve the same reliability with a much shorter code length due to better error exponent. Moreover, variable-length feedback codes have been shown to provide more benefits than fixed-length feedback codes, but also is harder to realize because they

often require precise synchronization between the transmitter and the receiver. A robust feedback mechanism is a key to realizing a practical variable-length feedback code.

For some applications, a hard deadline latency constraint is posted. In such cases, variable-length codes may lose their advantages because they cannot guarantee to meet the constraint. How to exploit the benefits of variable-length feedback codes without violating the hard latency constraint is a challenging task.

1.2.3 Low Power Devices

In many IoT applications such as wireless sensors and trackers, the form factor of the devices is essential for their usability. A small device usually has a limited power source, which in many cases is provided by a small-size battery that can hardly be recharged. To span a longer lifetime, the power consumption must be kept very low.

One way to reduce power consumption is to cut off some power-hungry components in the system. For example, high-precision crystals and phase-locked loops (PLL) are critical parts of modern communication systems because modulation and demodulation processes highly rely on the precise and stable frequency and phase. Nevertheless, it may be inevitable to remove them from a device to minimize the power consumption, which leads to unstable transmitted signal with large frequency and phase drift, posing challenges on the receiver side for detection, synchronization and demodulation.

Another commonly used method to save power consumption is duty-cycling, which puts the device into a low power sleep mode when available. Duty-cycling can lead to difficulty in device synchronization for multiple access because of the timing ambiguity and the lack of frequent control signals. As a result, careful planning for packet transmission is unattainable and packet collision becomes unavoidable, making interference handling a more challenging task.

Low power consumption also means lower computation resources. As a result, the system is forced to adopt low complexity algorithms at the IoT device and load the heavy work to the gateway. The cooperation between the IoT devices and the gateway thus becomes crucial.

1.3 Dissertation Outline

This dissertation presents three works that contribute to resolving different challenges described in Section 1.2. These works focus on systems and algorithms design, where end-to-end systems that comprise two sides, a resource-limited node as the transmitter and a resource-abundant gateway as the receiver, and the algorithms the systems adopted are presented. Each work serves as a solution for several challenges that are specific to different requirements of IoT applications. The relation between the three works and the challenges is illustrated in Figure 1.2.

In Chapter II, a novel non-orthogonal modulation scheme, named hyper-dimensional modulation (HDM), is proposed for short packet mMTC. The modulation scheme superimposes multiple non-orthogonal vectors together, where information is spread across the resulting vector to prevent the error from element-wise corruption. A CRC-aided K-best decoding algorithm is proposed to demodulate the packet in AWGN channel, greatly improving the performance of SPARC, a generalization of HDM, to be on par with state-of-the-art short codes. The K-best decoding algorithm is further extended to enhance the robustness under heavy interference. Simulation and real-world measurement results show that HDM provides great reliability under both interference-light and interference-heavy channels, showing great potential for mMTC applications.

In Chapter III, an instantaneous feedback-based variable symbol length transmission scheme, named opportunistic symbol length adaptation (OSLA), for URLLC is presented. OSLA can be viewed as a novel variable length code, where each symbol

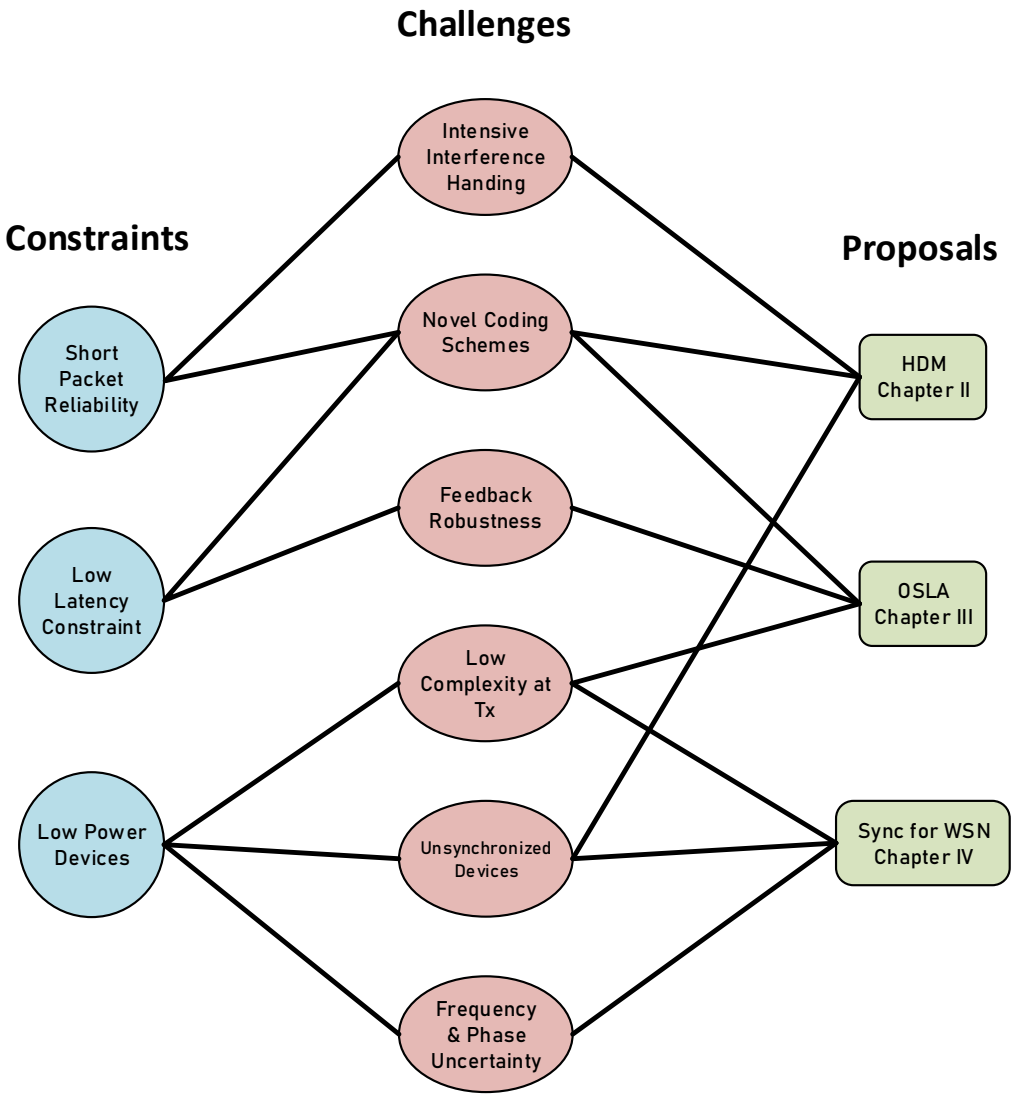


Figure 1.2: Relation between three works in this dissertation and challenges.

in the codeword has varying length thanks to the help of instantaneous feedback. The information destination in the system is responsible for determining the symbol length and informing the information source via binary feedback signal. This strategy makes the feedback more robust and relieves the burden of heavy computation on an IoT device. A modified Viterbi algorithm that is intertwined with OSLA is proposed to transmit convolutional codes. When combined with TBCC or turbo codes, OSLA

shows superior error rate performance and feedback robustness compared to state-of-the-art non-feedback codes and a deep neural network-based feedback code. OSLA can also be extended to handle hard deadline latency constraints, consistently providing gain over non-feedback code even when block length is fixed. For delay-sensitive applications in URLLC, OSLA is considered to be a strong candidate.

In Chapter IV, a synchronization scheme that is suitable for low power miniaturized wireless sensor nodes is presented. A gateway algorithm is proposed to overcome the difficulties of receiving packets from a PLL-less transmitter node, which produces frequency and phase unstable signal. The packet is consist of a pulse train as the preamble for synchronization purposes, using a custom-designed sparse M-PPM scheme. A 2D-FFT based method is proposed to detect the preamble and estimate the carrier/sampling frequency offset simultaneously for real-time compensation and decoding. Several enhancement strategies are adopted to enhance the robustness of the end-to-end system. Assisted by the proposed synchronization scheme, the system has been applied to a monarch butterfly migration tracking application, where millimeter-scale trackers are carried by butterflies when they travel. The experiment results show that the system can achieve more than 100 m signal receiving range.

Finally, Chapter V concludes the contributions of each work in this dissertation and discusses some future directions for the research.

CHAPTER II

HDM: Hyper-Dimensional Modulation for Robust Short Message for Massive Machine Type Communication

2.1 Introduction

The ITU-R categorizes the 5G and beyond technologies into three classes: enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low Latency Communications (URLLC), while each targets different applications [2]. mMTC use-case examples include transportation, utilities, health, environment, and security [10]. Packets for these mMTC applications usually carry relatively small amount of information such as control commands or sensor readings. However, the number of nodes in mMTC networks can be much greater than that of consumer (non-machine) mobile cellular networks. Thus it poses new challenges in the physical layer (PHY) design for reliable communication of short packets in interference-heavy channels [11].

Up to 4G, the main focus of the development had been to boost the data rate with high spectral efficiency for human-oriented communications. However, novel applications in mMTC usually convey relatively short information as small as a few bytes per packet. The conventional PHY and network design optimized for large

amount of information is not necessarily efficient in those applications. First, the overhead of preamble and pilot symbols is no longer negligible compared to the small number of information bits. Therefore, the frame structure needs to be re-designed with consideration of the overhead [12]. Second, the efficiency of modern codes such as Turbo and LDPC codes greatly relies on the long block-length and when the packet size is small, reliability of these codes significantly degrades. To quantify the efficiency of short codes, Polyanskiy showed that the normal approximation [8] provides a closed-form expression that tightly follows the achievability and converse bounds for short blocklength. Motivated by this, new coding schemes have been recently investigated [13, 14] to approach the limit for short codes.

Another challenge in mMTC is the interference especially when many nodes share the same unlicensed ISM (industrial, scientific and medical) band with heterogeneous PHY and multiple access (MA) protocols. Since grant-based multiple access protocols are often inefficient when the number of nodes is large [11], grant-free non-orthogonal multiple access schemes that allow multiple nodes accessing the channel at the same time have been investigated to support more nodes in a network [15–18]. However, these schemes still require slot-based time synchronization among nodes.

The overhead of control signals for network coordination often offsets the potential benefits of being synchronous. Moreover, precise synchronization in time and frequency is impractical for many narrowband low power mMTC nodes because of accuracy limitations in carrier frequency and sampling frequency generation. When an asynchronous mMTC network without strict synchronization among nodes operates in an unlicensed ISM band shared with heterogeneous networks such as WiFi, Bluetooth, Zigbee, etc., it is inevitable to observe severe intra- and inter-network interference. Therefore, it is a critical task for mMTC to design a novel PHY and multiple access scheme for short packets to mitigate severe interference from both intra- and inter-network traffic.

We propose Hyper-Dimensional Modulation (HDM) as a potential solution to address aforementioned challenges in mMTC. HDM is a non-orthogonal modulation scheme that can provide excellent reliability with short packet lengths, and it is inherently tolerant to interference. HDM is a special case of sparse superposition codes (SPARC) [19, 20]. For the encoding/modulation of SPARC, multiple columns from a dictionary matrix are selected and superimposed together based on multiple sparse vectors that convey information. This modulation process is equivalent to projecting sparse vectors onto a hyper-dimensional space. For HDM, such projection is defined by a fast linear transformation and pseudo-random permutation. This resembles the principles of compressive sensing [21], whereas the unique modulation structure of HDM makes it feasible to apply efficient decoding algorithms. Moreover, its robustness against interference makes HDM appealing to low cost mMTC networks where many low power mMTC nodes transmit short packets in an asynchronous (grant-free) manner to a more resourceful (computation capability and energy) gateway receiver to form a star network with pure ALOHA random access [22].

The main contributions of this chapter are summarized as follows:

1) We propose HDM with a cyclic redundancy check (CRC)-aided K-best decoding algorithm that can achieve very low error rate for short packets in additive white Gaussian noise (AWGN) channels. The proposed algorithm traverses a tree structure with pruning to find a candidate list for transmitted vectors with higher likelihood probabilities. CRC is then used to check all the candidates to find a valid codeword.

2) We propose extended algorithms to further combat both intra- and inter-network interference caused by packet collisions from unsynchronized transmissions. We evaluate different objective metrics in the K-best tree pruning algorithm to make the proposed scheme more robust when the system performance is limited by the interference, not by the channel noise.

3) We evaluate the packet error rate (PER) performance of HDM with exten-

sive simulations and real-world experiments using a software-defined radio platform. Results show that HDM greatly outperforms SPARC and is on par with state-of-the-art short codes in AWGN channels. HDM outperforms TBCC and polar codes in interference-heavy scenarios.

The rest of this chapter is organized as follows. Section 2.2 introduces the motivation and modulation process of HDM. Section 2.3 presents two demodulation algorithms algorithms for HDM, assuming AWGN channels. The advanced decoding algorithm is then extended in 2.4 to enhance the reliability in interference-heavy scenarios. Practical considerations such as decoding complexity, PAPR and type of linear transformation are discussed in 2.5. Section 2.6 shows the performance evaluation with simulation and real-world testing. Some related works are discussed in section 2.7, and finally section 2.8 summarizes the work.

2.2 Hyper-Dimensional Modulation

HDM is inspired by hyper-dimensional computing [23] where hyper-dimensional vectors are used to represent information and perform cognitive computing. The hyper-dimensional presentation is tolerant of component failure, and thus is suitable for communicating message through wireless channels where excessive noise and interference can cause signal corruption. This robustness comes from redundant representation, in which information symbols are spread across many components in the hyper-dimensional vector [23].

The modulation process of HDM utilizes two observations from using hyper-dimensional vectors: *near-orthogonality* and *linearity*. Consider a hyper-dimensional vector space $\mathbb{C}^D$ where D is the dimension of the hyper-dimensional vector. Similarity between two energy-normalized vectors $\mathbf{x}$ and $\mathbf{y}$ can be measured by cross-correlation $\mathbf{x}^H\mathbf{y}$. Here, $\mathbf{x}^H$ stands for transpose conjugate of $\mathbf{x}$. Two identical vectors result in a cross-correlation output that is equal to the vector energy $\mathbf{x}^H\mathbf{x} = ||\mathbf{x}||^2$. The

first observation that motivates HDM is the fact that two hyper-dimensional vectors whose components are i.i.d. zero-mean random variables have *nearly-orthogonal* cross-correlation; $\mathbf{x}^H\mathbf{y} \approx 0$ for a large D (hyper-dimension). Randomly-selected two vectors in the hyper-dimensional space have very small cross-correlation with high probability. The second observation is that the sum of two random vectors have high correlation with both vectors being added together. That is, the vector $\mathbf{x}+\mathbf{y}$ has high cross-correlation with both $\mathbf{x}$ and $\mathbf{y}$ since $\mathbf{x}^H(\mathbf{x}+\mathbf{y}) \approx \|\mathbf{x}\|^2$ because of near-orthogonality between $\mathbf{x}$ and $\mathbf{y}$. In other words, addition/superimposition of multiple independent hyper-dimensional vectors preserves the information that each vector carries without significant interference from each other although they are not strictly orthogonal. Based on these observations, HDM superimposes multiple (near-orthogonal) vectors to transmit numerous information bits using a single D-dimensional vector.

The HDM modulation process is expressed by

$$\mathbf{s} = \sum_{i=1}^{V} \mathbf{A}_i\mathbf{x}_i = \sum_{i=1}^{V} \mathbf{P}_i\mathbf{W}\mathbf{x}_i = \sum_{i=1}^{V} \mathbf{P}_i\mathbf{W}(s_i\mathbf{e}_{p_i}) \tag{2.1}$$

where $\mathbf{s}$ denotes the complex-valued transmitted vector with dimension of $D \times 1$ ($\mathbf{s} \in \mathbb{C}^D$) and V is defined as the number of non-orthogonal vectors $\mathbf{A}_i\mathbf{x}_i$, $i = 1, \cdots, V$, that are transmitted at the same time. Each $\mathbf{A}_i\mathbf{x}_i$ is obtained by projecting an information vector $\mathbf{x}_i$ onto a hyper-dimensional space using a matrix $\mathbf{A}_i \in \mathbb{C}^{D\times D}$.

The information vectors $\mathbf{x}_i \in \mathbb{C}^D$ for $i = 1, \cdots, V$ have 'sparse' representations with only one non-zero element, embedding information bits in the position of the non-zero element by $\mathbf{e}_{p_i}$ and its non-zero value (phase) s_i. The sparse vector $\mathbf{e}_{p_i} = [e_0, \cdots, e_{D-1}]^T$ is a $D \times 1$ unit vector with $e_p = 0 \ \forall \ p \neq p_i$ and $e_{p_i} = 1$. The non-zero position p_i is selected based on the information bits. We use a QPSK symbol $s_i \in \{\pm\sqrt{E_i/2}, \pm j\sqrt{E_i/2}\}$ for each non-zero element where E_i is the energy allocated

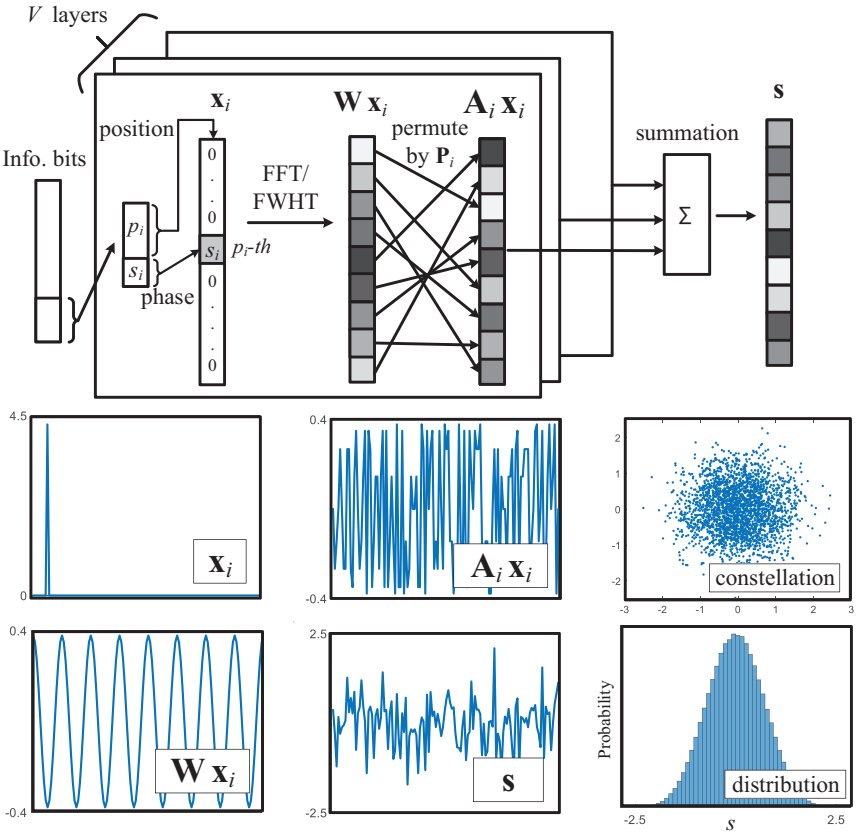


Figure 2.1: HDM modulation process visualization

to $\mathbf{x}_i$. The results in [20] (and our prior work [24]) show that QPSK is more efficient than other M-ary phase shift keying (PSK) schemes for SPARC (and HDM) to attain a lower PER given the same energy, bandwidth, and throughput. $\mathbb{E}\{\|\mathbf{s}\|^2\} = D$ and $E_i = D/V,\ i = 1, \cdots, V$ hold for energy-normalized HDM using equal-energy for each superimposed vector. In SPARC point of view, this modulation process is equivalent to selecting a column from a dictionary $\mathbf{A}_i$ based on the position index p_i, and multiplying it with a QPSK symbol s_i.

In HDM, the projection matrix $\mathbf{A}_i = \mathbf{P}_i\mathbf{W}$ is obtained by a fast linear transformation $\mathbf{W}$ such as fast Fourier transform (FFT) or fast Walsh–Hadamard transform (FWHT) followed by a pseudo-random permutation $\mathbf{P}_i$. Note that $\mathbf{P}_i$ is different for each i but $\mathbf{W}$ is common to all i's. Since HDM uses a fast linear transform whose

complexity is $O(D \log_2 D)$, it can be efficiently implemented in low power mMTC transmitters without explicitly computing costly matrix-vector multiplications with $\mathbf{W}$ that has a large dimension of $D \times D$.

The modulation process along with signal visualization of each step (only showing the real part) is summarized in Figure 2.1. In the modulation process, each independent vector goes through a separate layer/path with a different permutation pattern $\mathbf{P}_i$. Since HDM adds V i.i.d. vectors, elements of the final output vector $\mathbf{s}$ approximately follow a complex Gaussian distribution as shown in Figure 2.1 (bottom right).

One problem of using typical fast linear transform such as FFT or FWHT is that the first column of $\mathbf{W}$ has all ones, which makes the pseudo-random permutation meaningless and therefore violates the near-orthogonal property with other vectors. To avoid this problem, we use a pseudo-random vector whose elements are randomly selected from the set $\{\exp(j2\pi\frac{n}{D}), n = 0, \cdots, D-1\}$ to replace the all-ones column in $\mathbf{W}$ for FFT. Similarly, a pseudo-random vector with random 1 and -1 is used to replace that column for FWHT.

Parameters D and V determine the length and rate of transmission. For a given D, the number of information bits is proportional to V and each vector $\mathbf{x}_i$ conveys $\log_2 D$ information bits by the non-zero element position and additional 2 bits by the phase of the non-zero QPSK symbol. The modulation rate (or coding rate) C_R is the ratio of the number of information bits to the dimension of the transmitted vector, thus it is given by

$$C_R = \frac{V(\log_2 D + 2)}{D}. \tag{2.2}$$

With a unit energy constraint, the energy allocated to a vector $\mathbf{A}_i\mathbf{x}_i$ decreases as V increases and the inter-vector interference also increases at the same time because

of non-orthogonality among vectors. Therefore, there is a fundamental trade-off between the rate C_R and the error probability. Vector dimension D also affects the performance of HDM transmission. Since the near-orthogonality improves as D increases, larger D results in lower error rate for a fixed rate C_R. However, it also leads to higher demodulation/decoding complexity, which poses another trade-off between complexity and performance. It is worth noting that as the information length increases with a larger D, the relative advantage of HDM diminishes compared to other schemes such as LDPC, Turbo, and polar codes which are capacity achieving when the message length is sufficiently long.

2.3 HDM demodulation

In this chapter, we use *decoding* and *demodulation* interchangeably for HDM. Although HDM does not employ an explicit error correction scheme (except for CRC-based codeword selection), it exhibits superior or similar performance compared to conventional error correction codes applied to orthogonal modulation such as B/QPSK for short messages. Using the term *decoding* emphasizes the aspect of HDM imposing redundancy to increase robustness against signal corruption by noise or interference during transmission.

We consider an AWGN channel or narrowband frequency flat fading channel with perfect channel estimation. The received signal $\mathbf{y}$ can be represented by a model in (2.3), assuming the flat fading channel is equalized.

$$\mathbf{y} = \mathbf{s} + \mathbf{n} = \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i + \mathbf{n} \tag{2.3}$$

In (2.3), $\mathbf{n} \sim \mathcal{CN}(0, N_0\mathbf{I})$ is the complex Gaussian noise vector with zero mean and element-wise variance N_0. Note that under this model with an energy-normalized packet, the SNR is defined as $1/N_0$. The decoding process in AWGN can be considered

as finding the optimal solution of the non-convex minimization problem:

$$\text{P1:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_2^2, \tag{2.4}$$

where $\mathcal{X}$ represents the set of all possible sparse information vectors $\mathbf{x}_i$ (i.e., each $\mathbf{x}_i$ contains only one non-zero QPSK symbol encoding $\log_2 D + 2$ bits by the position and phase).

2.3.1 Basic Algorithm: Iterative Interference Cancellation Decoding

An intuiative and efficient way to demodulate the signal is to reverse the process of HDM modulation. During modulation, the superposition in the final stage causes interference between each layers. Therefore, interference cancellation can be applied along with inverse process iteratively after each iteration to increase the SINR for the next round, and thus increase the successful probability.

Esimtation of each superimposed transmission vector $\hat{\mathbf{x}}_i$ can be performed in parallel to enhace overall demodulation throughput. For layer i, the residual received vector $\hat{\mathbf{y}}_i^{(t)}$ for iteration t is obtained by subtracting estimated interference vectors from the original received signal $\mathbf{y}$ as in (2.5), where $\hat{\mathbf{x}}_i$ is the estimated sparse vector for the t-th iteration. For the initial iteration, $\hat{\mathbf{x}}_i = \mathbf{0}$.

$$\hat{\mathbf{y}}_i^{(t)} = \mathbf{y} - \sum_{v \neq i} \mathbf{A}^H \hat{\mathbf{x}}_v \tag{2.5}$$

To obtain $\hat{\mathbf{x}}_i^{(t)}$, the residual vector $\hat{\mathbf{y}}_i^{(t)}$ is permuted (multiplied) by $\mathbf{P}_i^{-1}$, and then inverse Fouruer transform is performed on the permuted vector as in (2.6).

$$\hat{\mathbf{z}}_i^{(t)} = \mathbf{A}_i^{-1}\hat{\mathbf{y}}_i^{(t)} = \mathbf{W}^H\mathbf{P}_i^{-1}\hat{\mathbf{y}}_i^{(t)} = \mathbf{x}_i + \sum_{v\neq i}\mathbf{d}_{(i,v)}^{(t)}. \tag{2.6}$$

The residual distortion vector $\sum_{v\neq i}\mathbf{d}_{i,v}^{(t)}$ satisfies (2.7) where $\mathbf{r}_v$ represents the residual error vector $\mathbf{r}_v = \mathbf{x}_v - \hat{\mathbf{x}}_v$.

$$\mathbf{d}_{i,v}^{(t)} = \mathbf{W}^H\mathbf{P}_i^{-1}\mathbf{P}_v\mathbf{W}\mathbf{r}_v. \tag{2.7}$$

For the next iteration $t+1$, the demodulated vectors $\hat{\mathbf{x}}_i$ is obtained from $\hat{\mathbf{z}}_i^{(t)}$ by (2.8), which can be performed in parallel for each vector index i. The vector with the maximum cross-correlation as in (2.8) provides the estimated sparse vector for the next iteration $t+1$.

$$\hat{\mathbf{x}}_i^{(t+1)} = \underset{\mathbf{x}_i\in\mathcal{X}}{\operatorname{argmax}}\,\Re\{\mathbf{x}_i^H\hat{\mathbf{z}}_i^{(t)}\}. \tag{2.8}$$

Note that the equation in (2.8) is actually a maximum-likelihood estimation of $\mathbf{x}_i$ assuming the residual distrotion vector $\sum_{v\neq i}\mathbf{d}_{i,v}^{(t)}$ behaves as uncorrelated multivariate Gaussian random vector. Furthermore, implementation of (2.8) is efficient since each vector in $\mathcal{X}$ is sparse with only one nonzero element that is picked from $\{\pm\sqrt{E_i/2}, \pm j\sqrt{E_i/2}\}$. This process is therefore simplified to finding the position with largest amplitude in its real or imaginary part.

Iteration stops when $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)}$ for all i. To guarantee a constant demodulation throughput, it is also possible to terminate the iteration after a predetermined number of iterations. As shown in [24], the average number of iterations at a reasonable low BER is < 3 including the initial iteration with $t = 0$.

2.3.2 Advanced Algorithm: K-best Decoding

While the aforementioned is efficient, it does not directly solve the optimization problem of (2.4). A brute-force method to find the minimum of (2.4) by trying all possible combinations of $\mathbf{x}_i, i = 1, \cdots, V$ is practically infeasible due to excessive complexity. Therefore, in this section we propose a tree-based algorithm that finds a suboptimal solution of (2.4) through a K-best breath-first search that is similar to a variant in MIMO decoding [25].

First observe that the objective in (2.4) can be expressed as

$$\|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_2^2 = \|\mathbf{y} - \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i\|_2^2 + \|\mathbf{A}_V \mathbf{x}_V\|_2^2 - 2\Re\{\mathbf{y}^H \mathbf{A}_V \mathbf{x}_V\} + 2\Re\{(\sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i)^H \mathbf{A}_V \mathbf{x}_V)\}, \tag{2.9}$$

where $\Re\{\}$ and $\Im\{\}$ are the operations of taking real and imaginary parts of a complex number or vector, respectively. In (2.9), the right-hand side consists of four terms. The first term $\|\mathbf{y} - \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i\|_2^2$ has the same form as the left-hand side except that the summation is now from 1 to $V-1$. The second term $\|\mathbf{A}_V \mathbf{x}_V\|_2^2$ is constant regardless of $\mathbf{x}_V$ as long as the fast linear transformation matrix $\mathbf{W}$ has equal norm columns (as in FFT and FWHT). The third term is the correlation between $\mathbf{y}$ and $\mathbf{A}_V \mathbf{x}_V$. Finally, the last term is the correlation between $\mathbf{A}_V \mathbf{x}_V$ and the accumulative vector $\sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i$. With recursion, the first term can be further decomposed until only $\mathbf{y}$ remains.

Subtracting constant terms $\|\mathbf{A}_i \mathbf{x}_i\|_2^2$ for $i = 1, \cdots, V$ and division by 2 does not change the solution of (2.4). Hence we define the *score metric* at recursion layer l as $s^{(l)} = \frac{1}{2}(\|\mathbf{y} - \sum_{i=1}^{l} \mathbf{A}_i \mathbf{x}_i\|_2^2 - \sum_{i=1}^{l} \|\mathbf{A}_i \mathbf{x}_i\|_2^2)$, which can be expressed in an iterative form:

$$s^{(l)} = s^{(l-1)} - \Re\{\mathbf{x}_l^H \mathbf{A}_l^H \mathbf{y}\} + \Re\{\mathbf{x}_l^H \mathbf{A}_l^H \mathbf{u}^{(l-1)}\}, \tag{2.10}$$

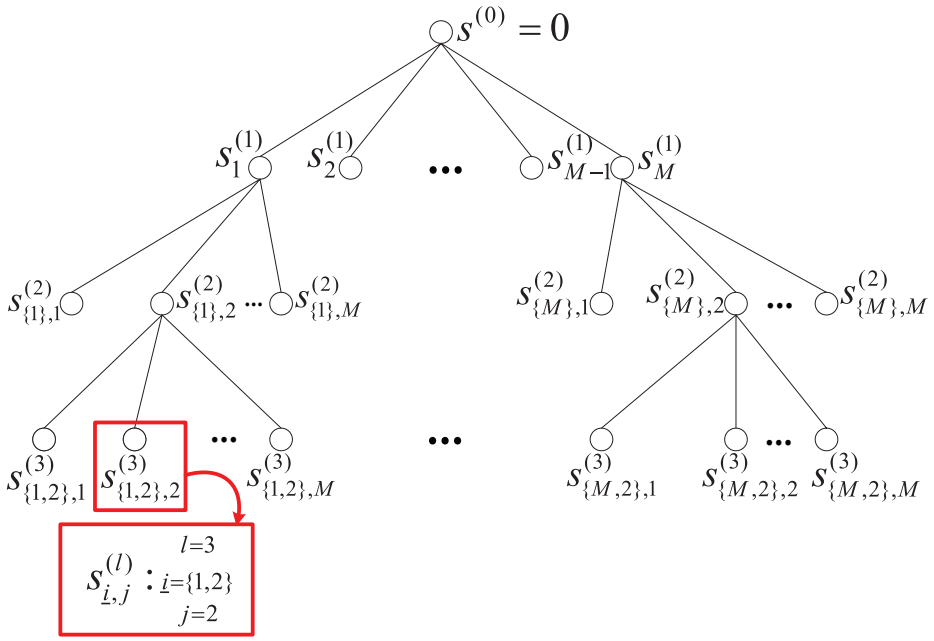


Figure 2.2: The tree structure of K-best algorithm. $M = 4D$ is the total number of candidates $\mathbf{x}_l$ at each layer.

where $\mathbf{u}^{(l)} = \sum_{i=1}^{l} \mathbf{A}_i \mathbf{x}_i$. This score metric depends on the selection of sparse vectors up to layer l, i.e., $\mathbf{x}_i$ for $i = 1, \cdots, l$.

The objective is to minimize (2.10) for the last layer V, $s^{(V)}$. Thus we find the minimum metric through a tree structure by evaluating candidate sparse vectors $\mathbf{x}_l$ for each layer. Note that at each node of the tree, we calculate the metric (2.10) for each candidate of $\mathbf{x}_l$ with given candidates determined by all previous layers from 1 to $l-1$ (i.e., $s^{(l-1)}$ and $\mathbf{u}^{(l-1)}$).

The tree structure is illustrated in Figure 2.2. For each node, we calculate its children's metric based on its parent and ancestor path using an iterative equation given as

$$s_{\underline{i},j}^{(l)} = s_{\underline{i}}^{(l-1)} - \Re\{\mathbf{x}_{l,j}^H \mathbf{A}_l^H \mathbf{y}\} + \Re\{\mathbf{x}_{l,j}^H \mathbf{A}_l^H \mathbf{u}_{\underline{i}}^{(l-1)}\} \tag{2.11}$$

where j is the candidate index of possible $\mathbf{x}_l$, and $\underline{i} = \{i_1, i_2, \cdots, i_{l-1}\}$ is the index list of previously chosen paths/vectors by its ancestor nodes. The accumulative vector

$\mathbf{u}_{\underline{i}}^{(l-1)}$ is the interference term given by candidates chosen by parent/ancestor layers.

Without pruning, the number of nodes and the size of possible x_l candidates grow exponentially as we go deeper into the tree. Since the paths with relatively large metrics are very unlikely to be part of the transmitted vector set that minimizes the metric, they can be pruned without degrading the performance much. Therefore, at the i-th layer we only keep best K_i candidates with lowest metrics and prune all the others. The value K_i is dynamically chosen to include all nodes whose metrics are not greater than the minimum metric of the i-th layer plus a pre-determined threshold. Our algorithm also defines a pre-determined $K_{\max}$ to prevent the dynamic K_i being too large so that $K_i \leq K_{\max}$ when more than $K_{\max}$ nodes satisfy the aforementioned condition. The iteration continues until the last layer, where no pruning happens. We denote the average number of candidates kept at each layer as $\overline{K} = \frac{1}{V-1}\sum_{i=1}^{V-1} K_i$.

There are efficient ways to calculate the metric (2.11). Since only one element of $\mathbf{x}_l$ is a non-zero QPSK symbol, evaluating the last two terms in (2.11) is equivalent to simply choosing a single real or imaginary number multiplied with different signs from the elements of $\mathbf{A}_l^H(\mathbf{y} - \mathbf{u}_{\underline{i}}^{(l-1)})$, which can be computed by a fast linear transform of $(\mathbf{y} - \mathbf{u}_{\underline{i}}^{(l-1)})$ followed by permutation (without matrix–vector multiplication).

One potential issue of the K-best algorithm is that a wrong pruning decision made in an upper layer can not be recovered in lower layers. To mitigate this issue, we propose a strategy to (re-)sort the order of decoding layers based on the score metric along the tree traversal. At each layer, we first evaluate minimum metrics of all remaining layers as the possible next layer based on (2.11) using the up-to-now best candidate. Then the layer with the lowest metric is selected as the next layer to proceed. This per-layer re-sorting approach significantly (up to 2 dB at PER=10^{-3} compared to no sorting) improves the error rate performance of the K-best HDM decoding.

Finally, CRC-assisted error correction is applied to further increase the error rate

performance of the proposed decoding algorithm. Since the K-best algorithm produces a list of candidates at the end, one can try each of them with the order of ascending metric until the candidate passes CRC. The error rate is improved because even in the event that the correct vector does not minimize the metric (2.10), it is still highly probable to be contained in the final candidate list.

2.4 HDM for Massive Machine-Type Communication Networks

2.4.1 mMTC Network Model

In a star topology mMTC network that allows grant-free transmissions, plenty of devices can transmit packets simultaneously, thus causing overwhelming interference at the gateway receiver. In this case, the channel noise may not be the performance limiting factor when the interference is stronger than the noise. An AWGN channel model may not capture the performance of a system in such a scenario, and the decoding algorithm designed for AWGN channels may experience significant performance degradation with interference. In this section, we propose modified versions of the K-best decoding algorithm to make HDM more robust against interference in mMTC networks.

We consider an narrowband uplink star network with multiple transmitters and one receiver. This network topology is widely adopted in mMTC networks because of its simplicity and efficiency [22]. Each device can transmit a packet at any time (i.e., grant-free) without considering other devices. Moreover, carrier frequencies of transmitters are assumed to be uniformly distributed in a pre-defined frequency range [26]. This is because the frequency uncertainty may extend over multiple times of the signal bandwidth of narrowband systems. For example, a low cost crystal with 50 ppm accuracy results in 120kHz carrier frequency offset for the 2.4GHz carrier frequency,

which is much larger than the bandwidth of an (ultra) narrowband scheme that often operates with $<$ 1kHz bandwidth [27]. In the considered scenario, the network adopts a pure unslotted (grant-free) ALOHA scheme in both time and frequency domains, thus multiple packet transmissions can (partly) collide in time and frequency.

In practice, not just transmitters in the same network but also transmitters in other heterogeneous networks using the same band can cause interference. A narrowband mMTC network operating in the 2.4GHz ISM band, for example, experiences inter-network interference from other technologies such as WiFi, Bluetooth Low Energy (BLE), etc. Since the WiFi and BLE bandwidth is $\geq$ 20MHz and 1MHz, respectively, with a typical packet duration of $\leq$ 2ms, interference from these networks are wideband and short compared to narrowband mMTC packets as illustrated in Figure 2.3. While the relatively wideband interference can easily overlap with the desired narrowband mMTC signal, it may only affect a few symbols of an mMTC packet because of its wideband (short symbol) nature.

In our scenario depicted in Figure 2.3, we categorize the interference into two types: *intra*-network and *inter*-network interference. Intra-network interference comes from other transmitters in the same mMTC network and the statistics of this interference is known to the gateway. On the other hand, inter-network interference is introduced by other heterogeneous networks and the statistics is unknown although it can be modeled as a random arrival process of wideband short pulses/packets.

The receiver signal $\mathbf{y}$ in this mMTC network model with potential interference can be expressed as

$$\mathbf{y} = \mathbf{s} + \mathbf{n} + \mathbf{w} = \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i + \mathbf{n} + \mathbf{w} \tag{2.12}$$

where $\mathbf{w}$ denotes the sum of all potential interference sources, including intra-network and inter-network interference. Note that elements in $\mathbf{w}$ may not have constant

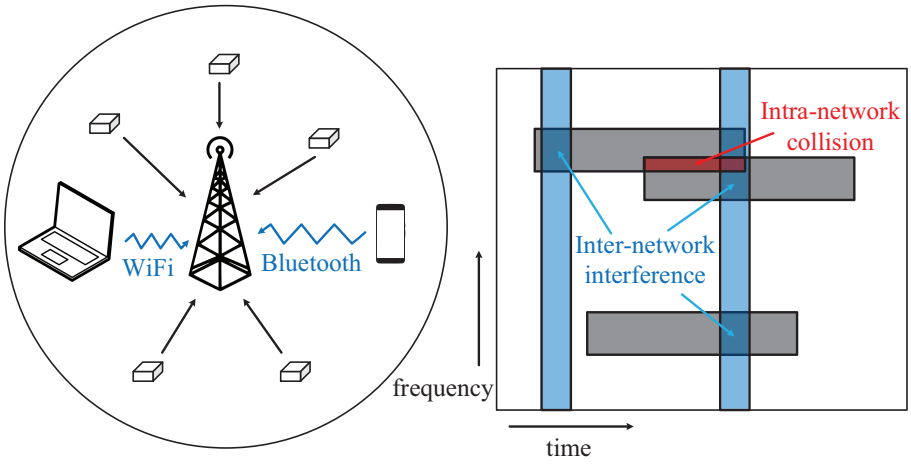


Figure 2.3: mMTC network and interference. Left: Star network topology with pure ALOHA. Right: Grey blocks are HDM packets and blue stripes are wideband interference packets.

power, and do not necessarily behave as i.i.d. Gaussian random variables. We assume that the receiver is aware of its presence, but may or may not know its statistics. The proposed algorithms try to decode $\mathbf{x}_i$'s without jointly decoding the interference source $\mathbf{w}$.

2.4.2 Dealing with Intra-network Interference

The P1 formulation of (2.4) assumes an AWGN channel with constant noise variance given observed signal $\mathbf{y}$. However, when the interference exists, the variance of interference plus noise is no longer constant across $\mathbf{y}$ (i.e., a packet). Under this scenario, the solution of (2.4) is no longer optimal for estimating the transmitted vector.

Considering the case that all devices in the network are transmitting HDM signals, interference can be approximated as a Gaussian random variable as discussed in Section 2.2. And, in a star network where the gateway is listening to all transmitting devices, it is possible that the gateway receiver first identifies the timing and average received power of the signal from each device (via packet detection) before decoding

individual packets that are collided.

Assuming the receiver has the information of timing and power level of the interference caused by each collided transmission, the problem statement in (2.4) can be modified to match this scenario by using weighted L2-norm as in (2.13)

$$\text{P2:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_{\mathbf{C}}^2, \tag{2.13}$$

where $\|\mathbf{x}\|_{\mathbf{C}} = \|\mathbf{C}^{\frac{1}{2}}\mathbf{x}\|_2$ and $\mathbf{C}$ is a diagonal matrix. As the interference plus noise has element-dependent variance, each diagonal element of $\mathbf{C}$ can be found by adding the average interference power level to the noise variance at a corresponding sample index, and then taking the inverse such that (2.14) holds.

$$\mathbf{C}_{j,j} = \frac{1}{\sum_{k \in \mathcal{I}_j} P_k + N_0} \tag{2.14}$$

In (2.14), $\mathcal{I}_j$ is the set containing all packets colliding with the desired signal at time index j, and P_k is the average interference power from interfering device k. A more detailed derivation of obtaining P_k in a star network with a pure ALOHA scheme in both time and frequency domain can be found in [28].

To modify the K-best decoding algorithm for weighted L2-norm, observe that the objective function (2.9) now changes to

$$\begin{aligned}
&\|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_{\mathbf{C}}^2 = \|\mathbf{C}^{\frac{1}{2}}\mathbf{y} - \mathbf{C}^{\frac{1}{2}} \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_2^2 \\
&= \|\mathbf{C}^{\frac{1}{2}}\mathbf{y} - \mathbf{C}^{\frac{1}{2}} \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i\|_2^2 + \|\mathbf{C}^{\frac{1}{2}}\mathbf{A}_V \mathbf{x}_V\|_2^2 \\
&\quad - 2\Re\Big\{(\mathbf{C}^{\frac{1}{2}}\mathbf{y})^H \mathbf{C}^{\frac{1}{2}}\mathbf{A}_V \mathbf{x}_V \ - (\mathbf{C}^{\frac{1}{2}} \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i)^H \mathbf{C}^{\frac{1}{2}}\mathbf{A}_V \mathbf{x}_V)\Big\} \\
&= \|\mathbf{y} - \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i\|_{\mathbf{C}}^2 + \|\mathbf{A}_V \mathbf{x}_V\|_{\mathbf{C}}^2 - 2\Re\{\tilde{\mathbf{y}}^H \mathbf{A}_V \mathbf{x}_V\} + 2\Re\{(\tilde{\mathbf{u}}^{(V-1)})^H \mathbf{A}_V \mathbf{x}_V)\}
\end{aligned} \tag{2.15}$$

where $\tilde{\mathbf{y}} = \mathbf{C}\mathbf{y}$ and $\tilde{\mathbf{u}}^{(V-1)} = \mathbf{C}\mathbf{u}^{(V-1)}$.

Notice that

$$\|\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2 = \mathbf{x}_i^H\mathbf{A}_i^H\mathbf{C}\mathbf{A}_i\mathbf{x}_i = \mathbf{x}_i^H\mathbf{W}^H\mathbf{P}_i^H\mathbf{C}\mathbf{P}_i\mathbf{W}\mathbf{x}_i = \mathbf{x}_i^H\mathbf{W}^H\mathbf{\Lambda}\mathbf{W}\mathbf{x}_i = \mathbf{x}_i^H\mathbf{Q}\mathbf{x}_i = \frac{D}{V}\cdot\mathbf{Q}_{j,j}$$

holds where j denotes the non-zero position of $\mathbf{x}_i$, $\mathbf{\Lambda} = \mathbf{P}_i^H\mathbf{C}\mathbf{P}_i$ is a diagonal matrix, and $\mathbf{Q} = \mathbf{W}^H\mathbf{\Lambda}\mathbf{W}$.

As we choose a fast linear transformation matrix $\mathbf{W}$ with constant magnitude elements such as FFT or FWHT, $\|\mathbf{A}_i\mathbf{x}_i\|_{\mathbf{C}}^2$ is constant regardless of the choice of $\mathbf{x}_i$. Therefore, the metric update rule in (2.10) can be re-written with minor modifications:

$$s^{(l)} = s^{(l-1)} - \Re\{\mathbf{x}_l^H\mathbf{A}_l^H\tilde{\mathbf{y}}\} + \Re\{\mathbf{x}_l^H\mathbf{A}_l^H\tilde{\mathbf{u}}^{(l-1)}\}. \tag{2.16}$$

Moreover, the computation complexity does not increase much since $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{u}}$ can be obtained by element-wise multiplications since $\mathbf{C}$ is diagonal. Except for this updated metric calculation, the remaining K-best algorithm is identical to the AWGN channel case.

2.4.3 Dealing with Inter-network Interference

When the inter-network interference is involved, the optimization problem P1 in (2.4) does not yield the optimal performance. For a narrowband mMTC scenario, we assume the inter-network interference burst length is much shorter than the length of the desired HDM packet, and independent random arrival processes can cause multiple interference sources collide with a single HDM packet. In this scenario, it is reasonable to assume the receiver does not know properties of interference such as the average/instantaneous power level and position of interference bursts within a desired packet. This assumption holds for an example scenario where a narrowband mMTC network using HDM operates in the 2.4GHz ISM band with heavy interference coming from WiFi and BLE. A packet from WiFi or BLE is much shorter ($\leq$ 2 ms) than a

narrowband (e.g., $D = 128$ with 1 kHz symbol rate) mMTC packet, and the HDM receiver does not have the capability of demodulating all WiFi and Bluetooth packets that collide with the HDM packet.

Performance of decoding algorithms under such severe interference can significantly degrade because of sporadic interference causing occasional large deviation (in Euclidean distance) from the transmitted samples. These events can result in large L2-norm during the objective function evaluation. One technique to alleviate this problem is to set a saturation threshold on the received sample to prevent large offsets from the transmitted signal caused by the sporadic strong interference [29]. Another strategy is to use an alternative metric to replace the L2-norm. We propose to use L1-norm as an alternative metric because it is less sensitive than L2-norm to sporadic outlier elements. The optimization in this case changes from L2- to L1-norm objective:

$$\text{P3:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \ \|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_1. \tag{2.17}$$

Note that the solution of P3 is indeed optimal when the noise plus interference follows Laplace distribution, which has a longer tail compared to Gaussian distribution.

The L1-norm in (2.17) can not be decomposed in the same form as the L2-norm in (2.4) with iterative equations. Thus we reformulate (2.17) with real-valued vectors and matrices with a goal to obtain an iterative additive form to replace (2.17).

Consider two real-valued scalars $a, b \in \mathbb{R}$ and observe that $|a + b| = |a| + |b| - 2 \cdot \mathbb{1}(ab < 0) \cdot \min(|a|, |b|)$ holds, where $\mathbb{1}(\cdot)$ is the indicator function. Using this, we decompose the L1-norm of the sum of two real-value vectors as

$$\|\mathbf{a} + \mathbf{b}\|_1 = \|\mathbf{a}\|_1 + \|\mathbf{b}\|_1 - 2 \sum_{i=1}^{D} \mathbb{1}(a_i b_i < 0) \cdot \min(|a_i|, |b_i|) \tag{2.18}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$, and a_i, b_i denotes the i-th element of $\mathbf{a}, \mathbf{b}$.

Now, the objective function (2.17) can be decomposed as

$$\|\mathbf{y} - \sum_{i=1}^{V} \mathbf{A}_i \mathbf{x}_i\|_1 = \|\mathbf{y} - \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i\|_1 + \|\mathbf{A}_V \mathbf{x}_V\|_1 \\ - 2 \cdot \mathbf{1}^T \big(\mathbb{1}(\mathbf{r}_{V-1} \circ \mathbf{A}_V \mathbf{x}_V > 0) \circ \min(|\mathbf{r}_{V-1}|, |\mathbf{A}_V \mathbf{x}_V|) \big) \tag{2.19}$$

where $\mathbf{r}_{V-1} = \mathbf{y} - \sum_{i=1}^{V-1} \mathbf{A}_i \mathbf{x}_i$, $\mathbf{1}$ is a vector with all ones, and $\circ$ denotes element-wise multiplication. The term $\|\mathbf{A}_V \mathbf{x}_V\|_1$ is constant if $\mathbf{A}_V$ has constant L1-norm columns. Hence, the iterative metric calculation for L1-norm has the form:

$$s^{(l)} = s^{(l-1)} - \mathbf{1}^T \big(\mathbb{1}(\mathbf{r}_{l-1} \circ \mathbf{A}_l \mathbf{x}_l > 0) \circ \min(|\mathbf{r}_{l-1}|, |\mathbf{A}_l \mathbf{x}_l|) \big). \tag{2.20}$$

The same K-best algorithm can be used with this recursive L1-norm metric in (2.20) replacing the previous L2-norm metric updating. Note that (2.20) does not involve matrix-vector multiplications since $\mathbf{x}_l$ has only one non-zero element and the other terms are evaluated by just selecting sign and magnitude values from two vectors.

However, the formulation of (2.20) only works for real-valued vectors, and this L1-norm metric cannot be directly applied to HDM that involves complex-valued vectors. Addressing this issue, we relax the problem and minimize an upper bound of the L1-norm.

Observe that for a complex-valued vector $\mathbf{z}$,

$$\|\mathbf{z}\|_1 = \sum_i \sqrt{\Re\{z_i\}^2 + \Im\{z_i\}^2} \leq \sum_i |\Re\{z_i\}| + |\Im\{z_i\}| = \| \begin{bmatrix} \Re\{\mathbf{z}\} \\ \Im\{\mathbf{z}\} \end{bmatrix} \|_1$$

holds. Then by denoting $\mathbf{y}' = [\Re\{\mathbf{y}\}^T \ \Im\{\mathbf{y}\}^T]^T$, $\mathbf{x}'_i = [\Re\{\mathbf{x}_i\}^T \ \Im\{\mathbf{x}_i\}^T]^T$ and $\mathbf{A}'_i = \begin{bmatrix} \Re\{\mathbf{A}\} & -\Im\{\mathbf{A}\} \\ \Im\{\mathbf{A}\} & \Re\{\mathbf{A}\} \end{bmatrix}$, a relaxed problem of (2.17) can be expressed as

$$\text{P3}'\text{:} \quad \underset{\mathbf{x}_i \in \mathcal{X}, i=1,\cdots,V}{\operatorname{argmin}} \|\mathbf{y}' - \sum_{i=1}^{V} \mathbf{A}'_i \mathbf{x}'_i\|_1, \tag{2.21}$$

which is now real-valued and can be solved with the proposed K-best algorithm via L1-norm minimization. Note that, this L1 optimization formulation requires a fast linear transform matrix $\mathbf{W}$ whose real and imaginary parts have constant L1-norm columns. Because of this requirement, we use FWHT instead of FFT when the L1-norm metric is adopted. An alternative method (with worse performance) that separates the I and Q channels and sends two independent real-valued HDM vectors is described in [30].

2.5 Discussion

2.5.1 Decoding Complexity and Latency

Besides the error rate performance, decoding complexity is a crucial factor when choosing a practical scheme. In this section we discuss the complexity of the K-best based HDM decoding algorithm proposed in Section 2.3.

The complexity of the proposed K-best decoding algorithm can be estimated by summing the number of operations of the following four parts: 1) Sorting and selecting the next layer for decoding, 2) Calculating metrics, 3) Selecting survivor nodes, and 4) Calculating cumulative interference vector. Note that these steps are repeated for each layer processing. For simplicity, we assume that all layers has the same $K_i = \overline{K}$ which is the average value. FFT is used for the fast linear transform.

1) Sorting and selecting the next layer: At the beginning of each layer, we evaluate $\Re\{\mathbf{x}_l^H \mathbf{A}_l^H (\mathbf{y} - \mathbf{u})\}$ for all remaining layers as a potential l-th layer and select the one with the highest value for the l-th layer processing. This requires $V_{\text{rem}}(4D \log_2 D - 5D + 8)$ operations, where V_{rem} indicates the number of remaining layers to decode. The value inside the parentheses can be obtained in a similar way in the next step 2).

2) Calculating metrics: Metric calculation is performed for every node to evaluate (2.11). Starting with $\overline{K}$ surviving nodes from the last layer, we first calculate $-\Re\{\mathbf{x}_l^H \mathbf{A}_l^H(\mathbf{y}-\mathbf{u}_k)\}$ in (2.11) for the k-th node among $\overline{K}$. To get the vector $(\mathbf{y}-\mathbf{u}_k)$ we need D additions. Since HDM utilizes a fast linear transformation and $\mathbf{x}_l$ has only one non-zero element, all $\mathbf{x}_l$ candidates can be evaluated by performing FFT (and pseudo-random permutation), which requires $4D\log_2 D - 6D + 8$ operations. Phase rotation due to the QPSK modulation is equivalent to taking real and imaginary parts of the result with different signs without additional operations. Finally, the results are added to the metric of the parent node, which requires another $4D$ additions. Therefore, the total number of operations for this step is $\overline{K}(4D\log_2 D - D + 8)$.

3) Selecting survivor nodes: To select $\overline{K}$ nodes out of $4\overline{K}D$, we use partial QuickSort [31]. This step requires $8\overline{K}D + (\ln \overline{K}D + 1.27)(2\overline{K} - 4) - 6\overline{K} + 6$ comparisons on average.

4) Cumulative interference vector: For each surviving node, we calculate the cumulative interference vector $\mathbf{u}_k = \mathbf{u}_{\text{old},k} + \mathbf{u}_{\text{new},k}$, where the former is the interference from previous layers, and the latter is the newly introduced interference from the current layer. This requires $\overline{K}D$ additions.

In summary, the total number of operations required to process all V layers is

$$\begin{aligned} N_{\text{op}} &= \overline{K}(V-1)\Big(4D\log_2 D + 2\ln \overline{K}D + 8D + 4.54\Big) \\ &+2(V^2+V)D\log_2 D - \frac{D}{2}(5V^2+5V-8) + 2(V-1)(\ln \overline{K}D + 1.27) + 4V^2 + 10V - 6. \end{aligned} \tag{2.22}$$

It has $O(V\overline{K}D\log_2 D)$ complexity as the first term dominates when $\overline{K}$ is large. For a $\{D = 128, V = 8\}$ HDM packet with 64 information bits (excluding CRC), the number of operations per bit is about $505\overline{K} + 1667$. Note, for comparison, that the number of operations required for polar and TBCC decoding for the same rate is on

the order of $10^3 - 10^4$ per information bit depending either on the list size of the successive cancellation list (SCL) polar decoder or the constraint length of the TBCC [32].

For practical systems, the number of operations is not the only complexity indicator and it is not necessarily proportional to the latency (or run-time) in modern parallel computing processors and accelerators. While the HDM decoding complexity scales with $V\overline{K}D\log_2 D$, the decoder can take the advantage of a fully parallelizable structure of K-best decoding to reduce the decoding latency in practical implementations on many-core processors and hardware accelerators. When the gateway has sufficient compute resources, steps 2) and 4) can be calculated in parallel for different candidates, removing the computation latency dependency on $\overline{K}$. Moreover, step 3) can also use a parallelized version of the algorithm [33] to achieve the latency of $O(V\log\log\overline{K}D)$. The resulting decoding time complexity, or latency, becomes $O(VD\log D + V\log\log\overline{K}D)$, which only grows with $\log\log\overline{K}$. This implies that increasing $\overline{K}$ for better PER performance does not significantly increase the latency as long as the receiver has a proportional number of parallel processing elements.

It is worth noting that, although SCL decoding for polar codes can also be parallelized [34], the number of newly generated paths increases exponentially at each time step with the number of parallel decoders. This makes parallel execution of the algorithm practically difficult, unlike the proposed K-best decoding for HDM.

2.5.2 PAPR and Clipping

Peak-to-average power ratio (PAPR) is an important metric for a practical modulation scheme because a high PAPR requires a wide dynamic range power amplifier (PA). A PA typically exhibits the highest power efficiency at its peak output power but a large PAPR forces the PA to mostly operate around the average power where power efficiency is significantly reduced. A constant envelope (baseband PAPR is

1) or low PAPR modulation scheme such as BPSK/QPSK is generally preferred to achieve higher PA efficiency.

However, the use of a fast linear transform and superposition of V vectors results in a relatively high PAPR of the HDM signal. When a normalized HDM vector has the element-wise average power of 1, the worst case peak value may occur when all V superimposed vector $\mathbf{A}_i\mathbf{x}_i$'s have the same phase on the same element after the fast linear transformation and pseudo-random permutation. As each element of superimposed vector $\mathbf{A}_i\mathbf{x}_i$ has the average amplitude of $\sqrt{1/V}$, the worst case PAPR of HDM is $10\log_{10} V$ dB. Note that most practical systems apply an additional pulse shaping filter before the signal goes into a PA to tighten the spectrum. This further (by ≈ 2.5 dB) increases the PAPR regardless of the modulation type.

One practical method to constrain the PAPR to a lower target level is to apply intentional deliberate clipping to the signal as discussed in [35]. Consider a power normalized HDM vector whose unclipped samples have the form of $s = Ae^{j\phi}$ where A is the sample amplitude, ϕ is the sample phase, and $\mathbb{E}\{A^2\} = 1$ holds. The clipped sample $\tilde{s}$ after the signal clipping at a pre-defined level c is obtained by

$$\tilde{s} = \begin{cases} Ae^{j\phi}, & \text{if } A \leq c \\ ce^{j\phi}, & \text{otherwise.} \end{cases} \tag{2.23}$$

The clipped signal can be regarded as the combination of the desired signal and distortion. With an assumption that HDM samples are approximated by complex Gaussian random variables, the PAPR after clipping (but before pulse shaping) can be calculated [36] by

$$\text{PAPR} = \frac{c^2}{1 - e^{-c^2}}. \tag{2.24}$$

The PAPR after pulse shaping depends on the pulse shaping function itself. Since

it is not straightforward to characterize the impact of pulse shaping on PAPR with deliberate clipping [36], we use numerical analysis in Section VI to quantify the HDM's PAPR with pulse shaping.

Denoting the *after clipping* average signal power E_s and noise variance N_0, the signal-to-noise-plus-distortion ratio (SNDR) after clipping is obtained [36] by

$$\text{SNDR} = \frac{\mathcal{K} E_s/N_0}{(1-\mathcal{K})E_s/N_0 + 1}, \tag{2.25}$$

where $\mathcal{K}$ is the signal attenuation factor given by

$$\mathcal{K} = \frac{\left(1 - e^{-c^2} + \frac{\sqrt{\pi}c}{2}\text{erfc}(c)\right)^2}{1 - e^{-c^2}}. \tag{2.26}$$

The parameter c determines the tradeoff between PAPR reduction and SNDR degradation.

Since HDM is designed to operate in relatively high noise/interference scenarios, it can tolerate moderate clipping distortion as long as it does not dominate the channel noise. One can choose the clipping parameter c such that the corresponding SNDR (2.25) is comparable to the original (pre-clipping) SNR given N_0 without significant PER degradation. The impact of signal clipping with various levels of c on PER is evaluated in Section 2.6.

2.5.3 Linear transforms for HDM

There are multiple options for the (fast) linear transform matrix $\mathbf{W}$ in HDM. In this section, we discuss the impact of choosing different transforms. While there is no strict constraint on the orthogonality among transform matrix columns for HDM, we only consider common discrete linear transforms whose matrices have orthogonal columns. This means that all valid HDM vectors that belong to the same layer are also orthogonal and the HDM performance is governed by the interference between

different layers that use different pseudo-random permutations. This can be seen in (2.9), where the last term is due to the interference from the other layers.

To determine the 'effectiveness' of a certain (fast) linear transform, we quantify its inter-layer interference by analyzing the statistics of the last term in (2.9). Roughly speaking, a linear transform that does not have large amplitude realizations of $\Re\{\mathbf{x}_i^H \mathbf{A}_i^H \mathbf{A}_j \mathbf{x}_j\} = \Re\{\mathbf{x}_i^H \mathbf{W}^H \mathbf{P}_i^H \mathbf{P}_j \mathbf{W} \mathbf{x}_j\}$ leads to a lower error rate for HDM decoding. Note that $\mathbf{x}_i, \mathbf{x}_j, \mathbf{P}_i, \mathbf{P}_j$ are random variables/matrices.

Figure 2.4 plots the empirical CDF of $\Re\{\mathbf{x}_i^H \mathbf{W}^H \mathbf{P}_i^H \mathbf{P}_j \mathbf{W} \mathbf{x}_j\}$ for different linear transform matrices $\mathbf{W}$. We test the following five discrete linear transforms that allow fast/efficient algorithms such as FFT: discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete Walsh-Hadamard transform (DWHT), discrete Slant transform (DST), and discrete Haar transform (DHT) [37]. As shown in the figure, DFT has fewest large amplitude values, while DWHT and DCT are behind it. DST and DHT have much more large amplitude values than the others. Based on this observation, we use DFT (i.e., FFT) for all cases except for the L1-norm minimization algorithm (i.e., P3 formulation in (2.17)). Recall that the L1-norm minimization algorithm cannot use DFT/FFT because the real and imaginary parts of a DFT matrix do not satisfy the constant L1-norm column condition. Hence we use DWHT/FWHT instead as it is more computationally efficient than DCT which exhibits a similar CDF.

2.6 Evaluation

2.6.1 Simulation Results

The proposed HDM schemes are compared with QPSK modulation protected by a 3GPP specified CRC-aided polar code with an SCL decoding algorithm [38, 39] and a tail-biting convolutional code (TBCC) decoded by a wrap-around viterbi algorithm

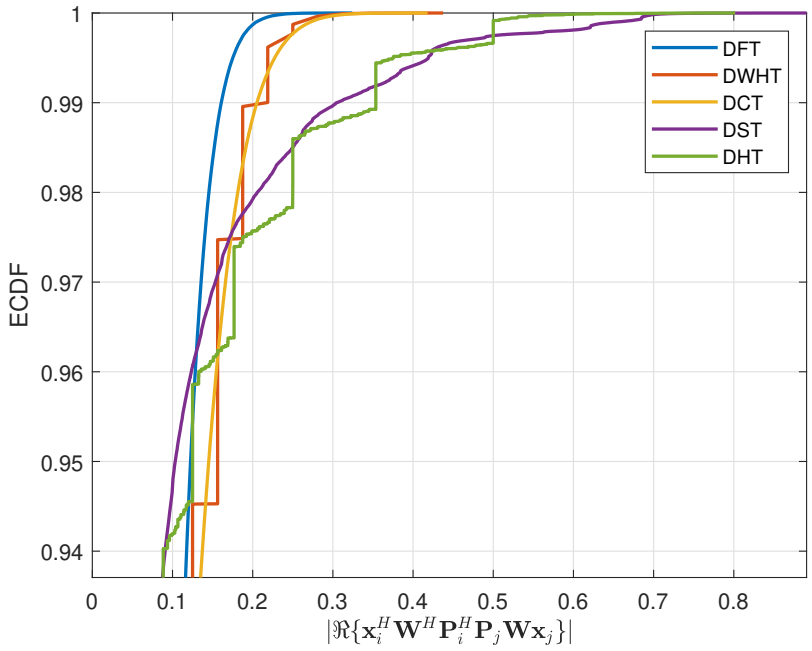


Figure 2.4: CDF of interference magnitude. The size of $\mathbf{W}$ is 128×128.

(WAVA) [40]. Both polar code and TBCC configurations are known to be very robust for short-length codes [41]. For the fast linear transformation in HDM, we use FFT for HDM with (weighted) L2-norm minimization (in AWGN and intra-network interference-heavy scenarios) and FWHT for L1-norm minimization (in inter-network interference-heavy scenarios).

For the narrowband mMTC scenario, we assume a short packet with length $D = 128$. The number of information bits is either 64 for a rate-1/2 packet, or 43 for a rate-1/3 packet. For HDM and polar codes, additional CRC bits are concatenated with the information bits before modulation/encoding. Polar codes use an 11-bit CRC as in the 3GPP uplink setting, while HDM use an 8-bit and 11-bit CRC for rate-1/2 and rate-1/3 packet, respectively. These additional CRC-bits require the information bits to be coded with a higher rate (to keep the effective rate unchanged with and without the CRC) which could potentially lead to a worse error rate. However, the SNR gain of CRC-based valid codeword/vector identification offsets the loss of using a higher rate

for information bits. TBCC does not utilize any CRC bits as the decoding algorithm does not create a list (unlike polar and HDM decoding) and thus CRC is not directly usable to correct decoding errors. Regardless of the CRC-usage, the rate of HDM, polar-QPSK and TBCC-QPSK schemes is identical as we send the same number of information bits (64 or 43) with the same number of complex-valued channel use: $D = 128$ for HDM and 128 QPSK symbols for polar-QPSK and TBCC-QPSK. Note that for the rate-1/3 TBCC-QPSK, 2-bit punctuation is used to change the length of (258,43)-TBCC to 256 bits before QPSK transmission. HDM superimposes $V = 8$ and $V = 6$ layers of vectors for rate-1/2 and rate-1/3 settings, respectively. Constraint lengths for TBCC are 9 and 8 for rate-1/2 and rate-1/3 packet, respectively. We also show the comparison to complex modulated SPARC [20] with the same length and similar rates (0.4922 and 0.3515 to be precise). AMP decoding is adopted for SPARC, but no outer code or CRC is utilized.

Figures 2.5 and 2.6 shows the packet error rate (PER) of two rate settings. One packet corresponds to a single transmit vector in HDM / SPARC, or a single codeword in polar-QPSK / TBCC-QPSK. The list size of polar SCL decoding is set to 8 and the maximum iteration number for WAVA is set to 10. For HDM, we test different values of $\overline{K}$ for the proposed K-best decoding algorithm with L2-norm minimization by setting a proper threshold and $K_{\max}$ for each SNR. As shown in Figures 2.5 and 2.6, the proposed HDM decoding algorithm greatly outperforms AMP decoder for SPARC, and its performance is on par with polar-QPSK and TBCC-QPSK in the AWGN channel. While the HDM performance is moderately worse than polar-QPSK and TBCC-QPSK with the rate-1/2 setting, HDM can slightly outperform them in the AWGN channel with the rate-1/3 setting. Normal approximations [8, 42] for both rate settings are also shown in the figures. Although $\overline{K}$ for HDM is larger than the list size of the SCL decoding, the runtime of HDM decoding is substantially faster (about 5$\times$ on Intel Core i7-7700 CPU for $\overline{K} = 50$) than polar-QPSK SCL decoding

(implementation in a Matlab toolbox) due to the computation-friendly parallel processing nature of the proposed K-best decoding. The runtime of TBCC-QPSK (our own implementation) is similar to that of HDM with $\overline{K} = 50$.

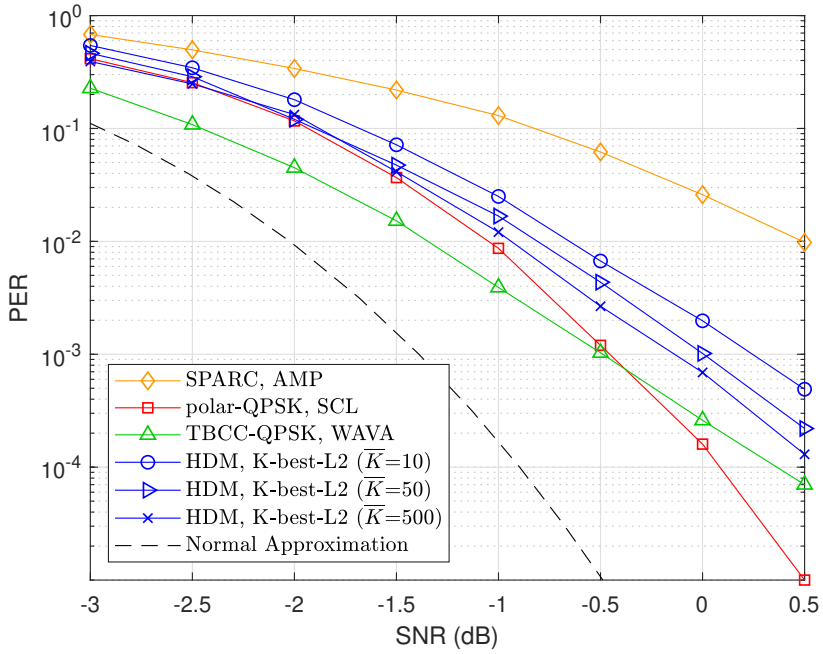


Figure 2.5: PER performance in the complex AWGN channel, Rate-1/2 packet.

Next we examine the trade-off between PAPR reduction and SNR loss caused by intentional clipping on the transmitted HDM signal. Figure 2.7 shows the resulting PAPR and SNR loss at different clipping levels c for a power normalized HDM vector. A root-raised-cosine (RRC) filter with roll-off factor 0.5 is used for pulse shaping. Different operating conditions regarding SNR (after clipping) are tested and the corresponding SNR losses are plotted. SNR loss is defined as the difference between original SNR before clipping and the resulting SDNR after clipping. Figure 2.7 shows that the PAPR can be reduced to $\leq$ 6.5 dB with only 0.5 dB S(D)NR degradation. It is observed that with RRC filter, the PAPR is further increased by roughly a constant 2.5 dB. Note that PSK signaling also undergoes the same/similar PAPR increase after pulse shaping. Hence the PAPR gap (in dB) in maintained the

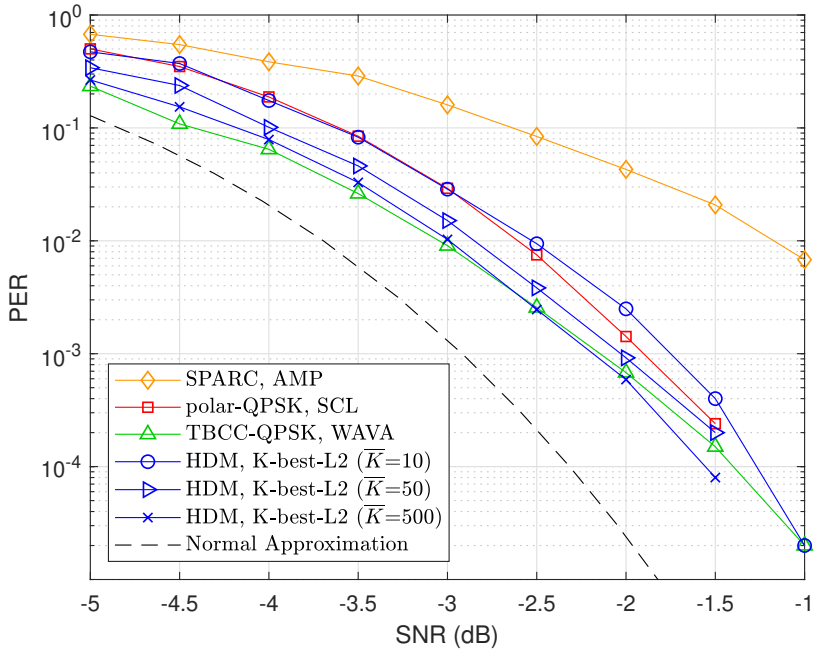


Figure 2.6: PER performance in the complex AWGN channel, Rate-1/3 packet.

same with or without RRC filtering. It confirms that aggressive intentional clipping can be applied to HDM without significant degradation in PER performance.

We also quantify the performance degradation caused by the quantization during analog-to-digital conversion (ADC). Any value whose amplitude is outside the quantization range is saturated to the highest quantization point, and thus it can be interpreted as signal clipping in the receiver. In the simulation, the largest amplitude of the quantized signal is determined by the clipping level that results in 0.1 dB SNR degradation using the analysis in the previous PAPR tradeoff simulation. Given this saturation level, the number of ADC bits determines the other quantized levels which are uniformly spaced. Figure 2.8 shows the PER with different numbers of ADC bits. We observe that 4 or 5 ADC bits are sufficient for PER $\sim 10^{-3}$ at 0 dB SNR. Note that constant-envelope BPSK / QPSK schemes require a similar number of ADC bits in order to reliably operate without excessive signal distortion in a low SNR condition where PAPR is increased and set by the noise.

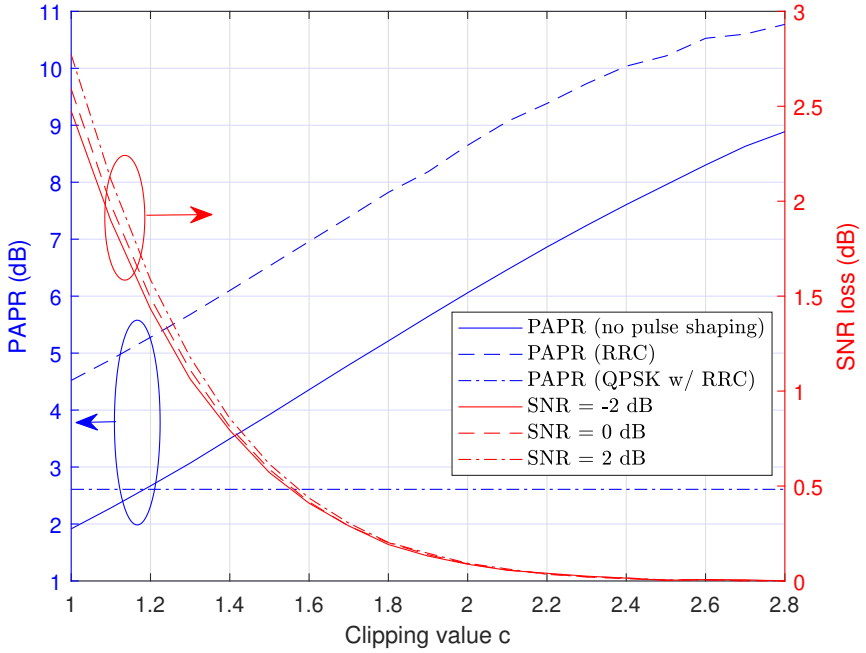


Figure 2.7: PAPR and SNR loss with intentional clipping

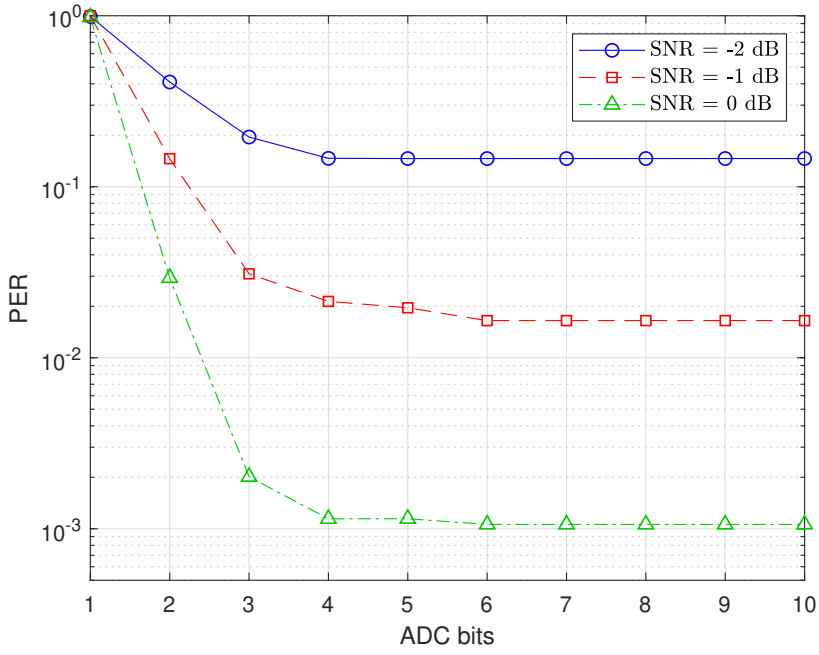


Figure 2.8: PER performance with ADC quantization

To evaluate the robustness of HDM against intensive interference in an mMTC network, we consider a narrowband mMTC system with 1kHz bandwidth that coexists

with relatively wideband systems such as WiFi and BLE. Following a star network topology with grant-free pure ALOHA, an HDM packet may collide with one or more packets from other devices in the network, leading to *intra*-network interference whose timing and power information is available (estimated) at the receiver. Packets from other non-mMTC networks may also cause interference to an HDM packet. Each *inter*-network interference packet is assumed to have a fixed length of 2 ms, which is much shorter than the length of 128 ms for 1kHz-bandwidth HDM ($D = 128$) and polar/TBCC-QPSK (128-symbol) packets. We evaluate PER for intra- and inter-network interference cases separately using different HDM decoding algorithms.

Figure 2.9 shows the PER when a desired packet collides with another interference packet with different overlapping ratios (1 indicates complete overlap and 0.5 means one half of the packet is overlapped). The interference packet is set to have 2 times stronger power than the desired packet to simulation an interference dominated scenario. The background (interference-free) SNR is set to 1 dB, which is sufficient for all schemes to achieve PER less than 10^{-3} when the interference is absent. The timing and power information of the interference is assumed to be perfectly estimated at the receiver so it can either adopt the weighted-L2 K-best algorithm for HDM, or calculate a more accurate log-likelihood ratio (LLR) for each bit for polar codes and TBCC. Note that for QPSK packets, the resulting noise plus interference is not Gaussian distributed, thus the LLR calculation is not exact even if the interference power variance is known. As shown in the figure, HDM with the weighted-L2 K-best algorithm and polar-QPSK outperform TBCC, which turns out to be significantly more vulnerable to a packet collision. This is because of its trellis-based structure, which is vulnerable to *consecutive* corrupted symbols observed during collision. Note that for all three schemes, the performance (in dotted lines) significantly degrades when interference information is unavailable.

Figure 2.10 shows the PER in strong inter-network interference scenarios. The

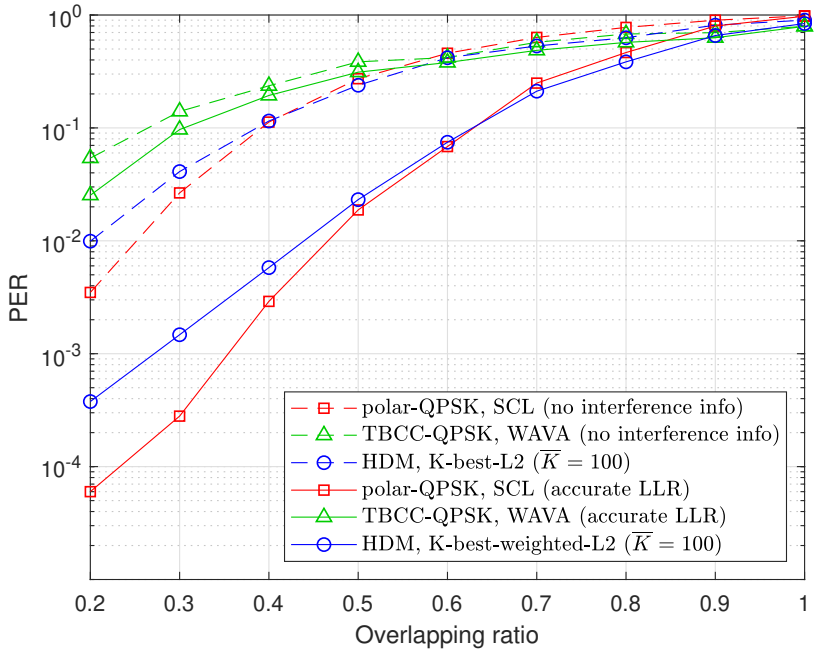


Figure 2.9: PER performance with intra-network interference

background (interference-free) SNR of the desired packet is set to 3 dB for this simulation to evaluate an interference-dominant condition. The power of interference packet follows a log-normal distribution with variance of 10 dB [30] while their mean power is set by the simulated signal-to-interference ratio (SIR), which is the ratio between average signal power of the desired packet and average interference power. Note that SIR is defined by only the part where the interference burst overlaps with the desired packet. The arrival of interference packets follows a Poisson arrival process with a mean interval of 5 ms while each interference packet is 2 ms long. Each sample in an interference packet is an i.i.d. Gaussian random variable that emulates the amplitude of OFDM signals. The desired signal has 1kHz bandwidth, thus each sample/symbol in the desired packet spans 1 ms. To increase the robustness against the outlier samples caused by strong short interference packets, HDM with the L2-norm minimization K-best algorithm sets an amplitude saturation threshold of 2 for each I and Q channel for a power-normalized HDM packet. For polar codes and TBCC, an LLR

mapping method [29] designed to works with a wide range of interference-to-noise ratio (INR) between -5 and 40 dB is used to enhance the robustness to pulse-like interference. Note that HDM does not require SI(N)R information for decoding while polar/TBCC-QPSK uses the (average) SNR information for LLR computation. Using the (average) SINR for LLR computation degrades PER for polar/TBCC-QPSK since interference is short and sporadic.

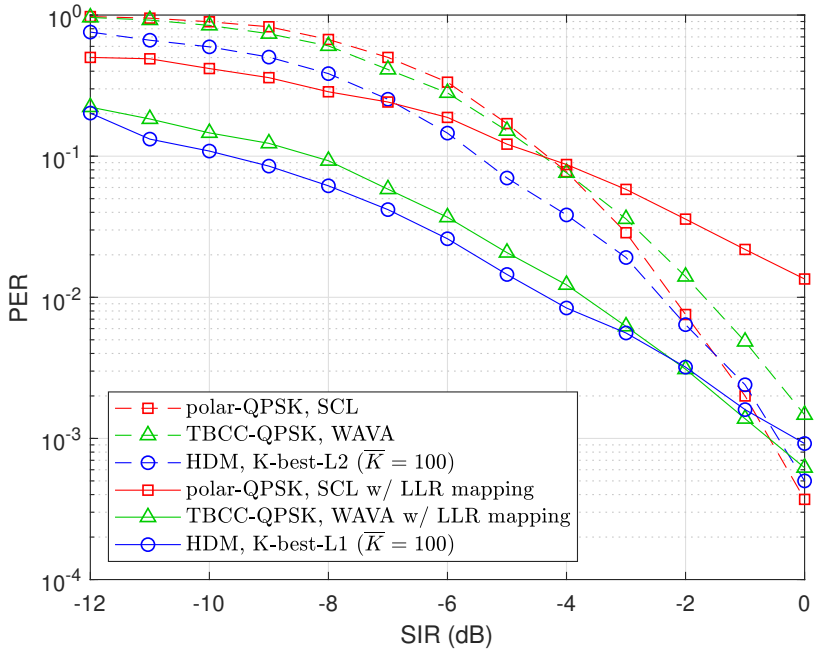


Figure 2.10: PER performance with inter-network interference

Figure 2.10 confirms that HDM with the L1-norm minimization K-best algorithm yields the best performance. It is observed that the LLR mapping method improves polar codes very little while it can be even harmful when SIR is high. This is related to the polar decoding process, where the effects of inaccurate LLR propagates through successive cancellation, often causing unrecoverable errors. On the other hand, HDM and TBCC are more robust to the sporadic outliers.

Note that the gap between L1- and L2-norm minimizations for HDM reduces as SIR increases and eventually the L2-norm minimization scheme outperforms the L1-

norm counterpart when interference does not dominate the channel noise any more.

From Figures 2.5, 2.6, 2.9 and 2.10, we have shown that HDM works reliably for all scenarios unlike polar and TBCC-based schemes which are vulnerable to some scenarios. Specifically, TBCC is relatively more vulnerable to heavy intra-network interference scenarios (Fig. 2.9) whereas polar codes suffer in inter-network interference scenarios (Fig. 2.10).

2.6.2 Real-World Experiments

To evaluate the performance in real-world scenarios, a wireless end-to-end system testing setup is constructed using a software define radio platform, USRP X310 [43]. Two USRPs are used as a transmitter and receiver pair for wireless communication in uncontrolled real-world channels which may corrupt the signal by noise and interference. The signal has 10kHz bandwidth and each packet contains a pre-defined preamble. The payload is modulated either by HDM or QPSK with polar/TBCC encoding for the length of 128 symbols (12.8 ms) to contain 64 information bits with 1/2 rate. The preamble is used for packet detection and channel estimation. We assume the channel is block fading with a constant amplitude and phase over one packet. We then use the estimated channel to equalize the received packet. We design the preamble to be sufficiently long (60 ms) so that the packet detection and channel estimation does not limit the decoding error performance. The USRP transmitter and receiver pair exhibits inevitable small carrier frequency offset which causes slow phase rotation of the received signal. Hence the transmitter also sends an unmodulated pilot tone on a different carrier frequency along with the signal to assist frequency offset tracking. To control the transmit power of the packet, signal attenuators are used in addition to the digital gain control feature provided by the USRP.

We first test the performance in the 915MHz ISM band, which is less crowded with fewer interference sources compared to the 2.4GHz ISM band. Figure 2.11

shows that HDM outperforms polar/TBCC-QPSK even with a moderate $\overline{K} = 10$ in the real-world channel, unlike the simulated AWGN case shown in Figure 2.5. This may be due to the uncontrolled interference and inaccurate SNR (consequently, inaccurate LLR) estimation, which cause more significant performance degradation to polar codes and TBCC than HDM. The presence of interference and/or other non-idealities can be observed by the offset between the expected sensitivity of -134 dBm (in ideal AWGN) and the measured sensitivity of -126.75 dBm for the PER of 10^{-3}.

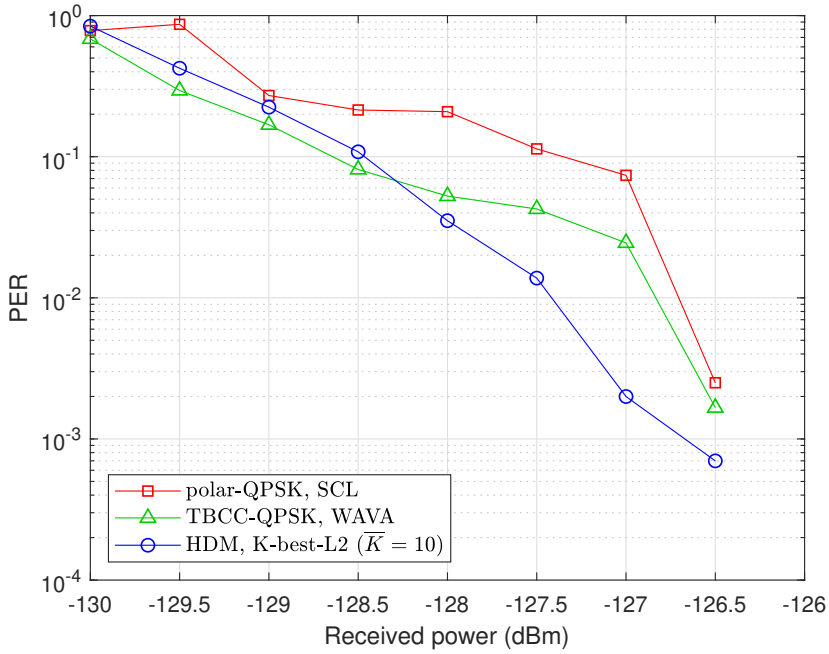


Figure 2.11: PER measurement at 915MHz

Next, we test the performance in the 2.4GHz ISM band, which contains severe uncontrolled interference including WiFi and Bluetooth. Figure 2.12 shows an example spectrogram of the signal captured in the 2.4GHz band with 20MHz sampling rate, where wideband WiFi signals ($\geq$ 20MHz) and frequency hopping Bluetooth signals ($\geq$1MHz) dominate the spectrum. These interference sources are both wideband and short compared to the desired narrowband (10kHz) signal, which justifies our assumption made in the previous sections. Although not visible in the spectrogram

because of the limited time resolution, there are also many very short ($\ll$1ms) interference signals, which may be short Bluetooth control packets or other wireless devices operating in the university campus network.

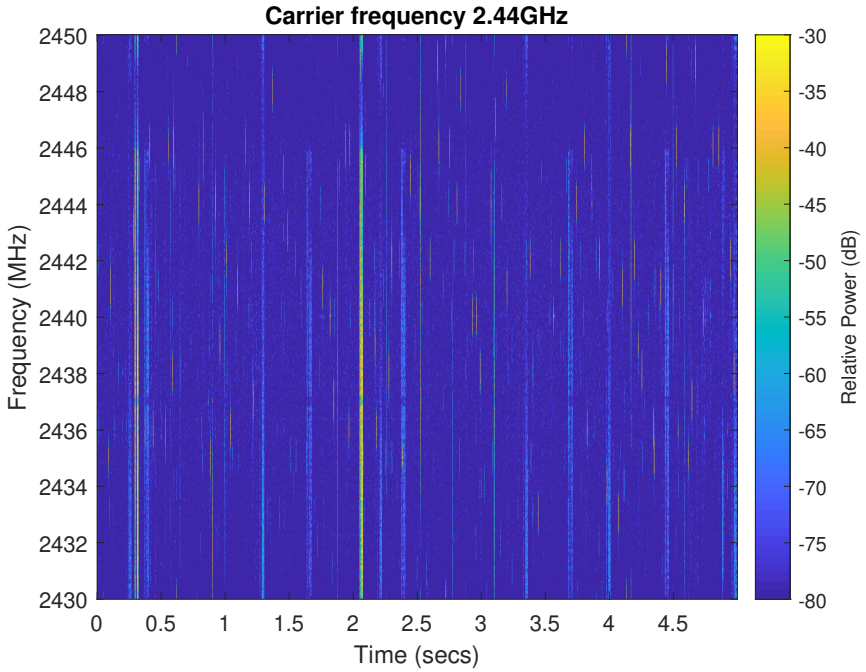


Figure 2.12: Spectrogram at 2.44GHz

Since the interference in the environment is not controlled, it is not practically feasible to completely distinguish the interference from the noise. Therefore, in our experiments we define the interference any signal that has 10 dB higher power than average noise power. By definition, INR is $>$ 10dB for our experiment. The power distribution of the measured interference signal is shown in Figure 2.13, which reveals that the instantaneous INR can be higher than 30 dB in the real-world 2.4GHz channel.

Figure 2.14 shows the distribution of the interference duration and the interval between two closest interference signals. It is observed that most of the interference is short ($\leq$1 ms). The intervals between interference are also relatively short compared to the narrowband HDM and polar/TBCC-QPSK packets. This implies that

one packet may encounter more than one interference burst with high probability. Although this observation is based on our own definition of interference with INR $>$ 10dB, it is generalizable for a busy 2.44GHz band such as on-campus networks where WiFi and Bluetooth dominate the wireless traffic.

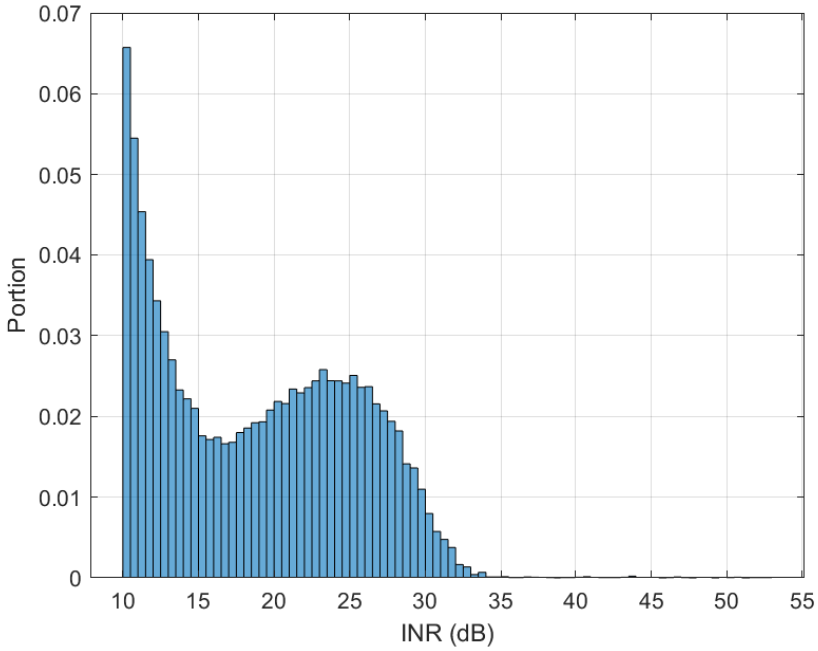


Figure 2.13: Power distribution of the interference

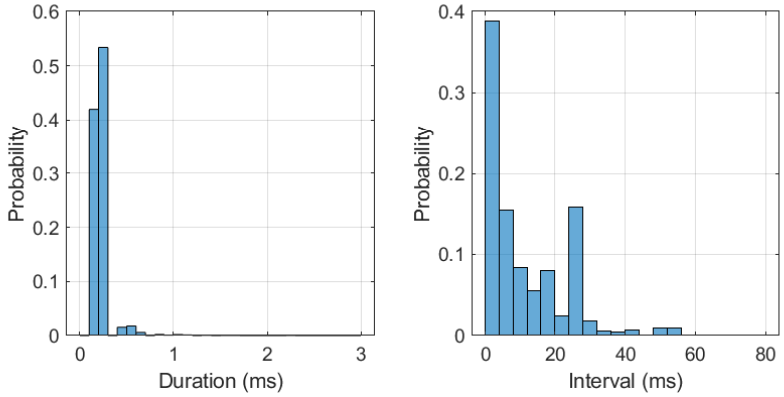


Figure 2.14: Duration and interval statistics of the interference

Figure 2.15 shows the PER measurement in the 2.4GHz ISM band. Both L2-norm and L1-norm minimization algorithms are shown for HDM with a modest $\overline{K} =$

10 setting. The LLR mapping method [29] is adopted and verified to improve the performance during the real-world experiments. Figure 2.15 shows that, for the same PER, the required received signal power is higher for the 2.4GHz band than that of the 915MHz band because of the stronger background interference. It is observed that HDM with L1-norm minimization K-best algorithm has the best performance for the target PER of $< 10^{-3}$. For HDM, no error is observed for at least 5000 packets when the received power is ≥ -113 dBm. The results do not closely follow the simulated results in Figure 2.10 because of the mismatch in interference characteristics between the uncontrolled real-world environment and our simulation setup. More burst errors are observed in this real-world experiment that cause more degradation for TBCC and HDM with the L2-norm minimization algorithm.

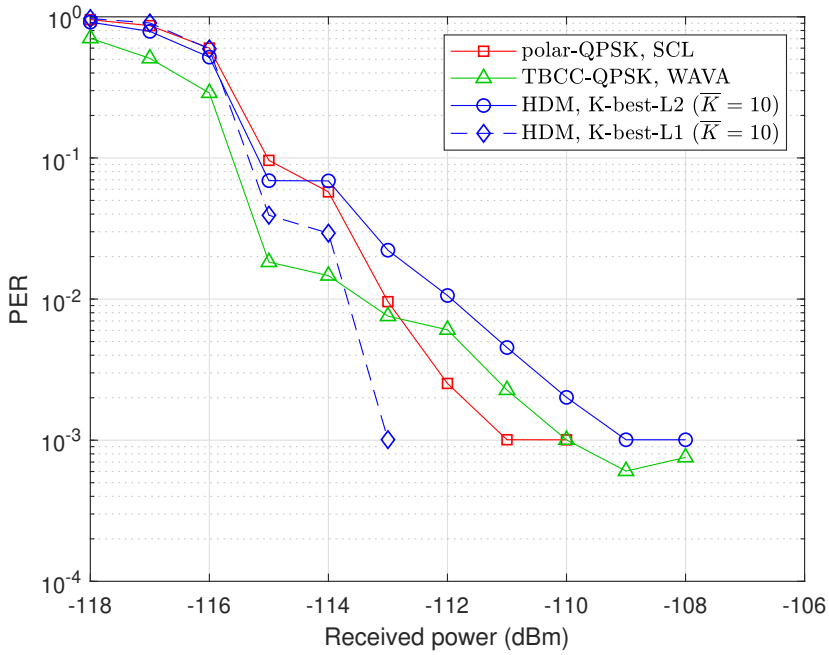


Figure 2.15: PER measurement at 2.4GHz

2.7 Related Works

The modulation process of HDM involves sparse vector mapping via index modulation. There are related prior works that also use sparse vectors with a random dictionary or index-based modulation for robust communication, thus they possess similar properties as in HDM. Here we describe prior schemes and their differences compared to our scheme.

Sparse superposition codes (SPARC) [19, 20, 44] are capacity achieving schemes in the AWGN channel. Their codewords are constructed by sparse linear combinations of entries in a dictionary, or equivalently, superposition of matrix-(sparse)vector products as in (2.1). Therefore, HDM can been considered as a special case of SPARC. For decoding of SPARC, approximate message passing algorithm [44] is widely used. SPARC is proven to achieve the channel capacity if the size of the dictionary is large enough under some parameter constraints. The main distinctions between HDM and SPARC are the size of the codeword and the design of the dictionary. SPARC typically uses a very long codeword length (e.g., 5000 bits) to achieve low error rates and the dictionary is constructed with randomly generated entries. On the other hand, HDM is designed for a relative short message length (e.g., $D = 128$) and it uses a structured modulation process that combines sparse index encoding, fast linear transform, and vector permutation to enable computationally-efficient yet powerful algorithms to achieve superior (compared to general SPARC schemes) error rate performance for short packets. Whereas SPARC typically operates with a relatively long outer code such as LDPC, the proposed HDM adopts a CRC-assisted error correction scheme which is more efficient for short packets.

Multi-dimensional modulation (MDM) [45, 46] is a modulation scheme that uses multi-dimensional lattices. By exploiting coding gain from the lattice and shaping gain, a well-designed constellation for MDM can outperform conventional QAM schemes with less energy per information bit for the same error rate without sacri-

ficing bandwidth efficiency. Although higher dimensions improve both coding and shaping gain, a practical well-designed multi-dimensional modulation scheme usually has a moderate dimension M because it becomes very difficult to design a good constellation in a high dimension space as the demodulation complexity dramatically increases with M. MDM is more beneficial in high SNR conditions when the constellation size can be relatively large and the spectral efficiency is ≥ 1 bps/Hz. Whereas, HDM is designed to operate in a low SNR or interference-dominated channel with a relatively low spectral efficiency of < 1 bps/Hz for reliable communications of short messages. Unlike MDM that requires a deliberately designed codebook/constellation for a specific, relatively small dimension space of size M, HDM uses a fast linear transform and random permutation defined with a much larger dimension $D \gg M$.

Sparse vector coding (SVC) [47] is a non-orthogonal encoding scheme based on the theory of compressive sensing. The encoding process is similar to HDM as it selects columns from a dictionary according to a sparse vector. Dictionaries in SVC are typically constructed by randomly sampling from Gaussian or Bernoulli distribution without any elaborate structure, which is widely assumed for compressive sensing. A predefined table is used for mapping information bits to a sparse vector, whose non-zero elements are not restricted to be placed in different layers/sections as in HDM or SPARC. Multipath matching pursuit for sparse recovery[48] is a popular algorithm to decode SVC. In contrast, HDM uses an elaborate layer structure with a common linear transformation across all layers for efficient encoding and decoding. Sparse recovery algorithms generally do not work well for HDM because of its unique structure and elaborate constraints, which lead to dedicated (and efficient) decoding algorithms.

Orthogonal frequency division multiplexing with index modulation (OFDM-IM) [49] uses indices of active subcarriers and modulated symbols on these subcarriers to embed information message bits via OFDM. Since index selection is followed by

IFFT for OFDM, the modulation process resembles HDM. The goal of OFDM-IM is to improve robustness to inter-carrier interference caused by high mobility in OFDM systems. However, OFDM-IM only involves strictly orthogonal subcarriers and does not combine multiple non-orthogonal vectors. Thus its demodulation process consists of finding the most probable active subcarriers and demodulating their symbols without considering any interference caused by the superposition of non-orthogonal vectors. On the other hand, HDM is a non-orthogonal modulation scheme that applies element-wise permutations to fast linear transform (which does not have to be FFT) results before combining non-orthogonal vectors. And it employs a dedicated decoding algorithm to mitigate interference among superimposed vectors within the same packet.

Integer-HDM [50] is a modulation scheme inspired by our original HDM [24] and thus it has a modulation structure similar to this work. Instead of using fast linear transformation on complex-valued vectors, Integer-HDM constrains the superimposed vectors to take values only from the binary set $\{1, -1\}$. This enables an even simpler decoding algorithm without compromising the error rate compared to the original HDM in [24].

Despite that all these prior schemes share some similarities in the modulation structure, their decoding algorithms significantly differ because of differences in their design principle and target operating scenarios. In this work, we propose advanced decoding methods for HDM to further improve the performance in AWGN and also in interference-limiting scenarios which are not explicitly considered in aforementioned prior schemes.

2.8 Summary

In this Chapter, we present hyper-dimensional modulation (HDM) specifically designed for interference-heavy mMTC networks. HDM is a special case of sparse su-

perposition codes as a non-orthogonal modulation scheme that superimposes multiple independent vectors for concurrent transmission. The proposed decoding scheme uses a CRC-aided K-best algorithm with L2-norm minimization to achieve robust performance in the AWGN channel. Furthermore, the algorithm is extended to weighted L2- and L1-norms to combat intra- and inter-network interference, respectively. Both simulations and real-world experiments are provided to show that the proposed schemes greatly improves SPARC for short packets and HDM can outperform conventional orthogonal transmission schemes that use strong channel coding such as polar codes and TBCC. The proposed HDM is particularly advantageous in interference-heavy scenarios as a promising solution for practical mMTC networks.

CHAPTER III

OSLA: Instantaneous Feedback-based Opportunistic Symbol Length Adaptation for Reliable Communication

3.1 Introduction

It is well known that feedback cannot increase the capacity of a memoryless additive white Gaussian noise (AWGN) channel or memoryless discrete channel [51]. It, however, can significantly increase the error exponent to improve the error rate* [54]. This benefit is important for short blocklength codes as capacity-achieving codes typically rely on long blocklengths. Therefore, feedback-based transmission is of high interest in the regime of short blocklength communications for emerging applications such as real-time control of autonomous vehicles to enable more reliable communications with an enhanced error rate exponent.

Since the study on feedback codes in Shannon's famous work [54], several communication schemes have been proposed that use feedback to pursue high reliability without relying on complicated code designs. The classic Schalkwijk-Kailath (SK) scheme can achieve a super exponential decaying rate of error probability w.r.t the

*This is not always true for symmetric discrete memoryless channels where fixed-length codes can have the same error exponent without feedback [52, 53].

blocklength (infinite error exponent) in AWGN channel with a simple yet elegant strategy when the feedback is noiseless [55]. However, the SK scheme is extremely sensitive to noise in the feedback channel, to the extend that even some small arithmetic imprecision prohibits it from attaining the claimed performance, thus it has been regarded as a practically infeasible scheme. Several works have proposed to solve the problem with modified algorithms [56–59]. Although the assumptions in [56–59] for the feedback channel quality and computation precision are significantly relaxed compared to the original SK scheme, they are still unattainable in practical communication systems especially when the blocklength is relatively long. On the other end of the spectrum, Deepcode [60] was shown to have superior performance even with noisy feedback by exploiting the modeling power of deep neural networks. Deepcode has been proven to be very effective for codeword-level feedback-based communications. However, it comes with certain drawbacks such as limited scalability to longer blocklengths, and inflexible network models that need to be specifically trained for each code rate and length configuration.

Variable-blocklength codes have been well studied as a feedback-based reliable communication scheme. Burnashev [61], and Yamamoto and Itoh [62] proposed variable-blocklength codes that achieve the channel capacity with an optimal error exponent. Variable-blocklength codes can provide performance advantages over fixed-blocklength codes [52, 53], but they often come with the difficulty of maintaining state-synchronization between two communicating ends. To address that issue, a feedback coding-based synchronization scheme for noisy feedback channels was proposed in [63]. However, out-of-sync recovery in [63] still requires long delays, which is not consistent with applications requiring short blocklengths. Therefore, designing simple and robust feedback schemes for variable-blocklength codes is a challenge of great interest.

Surprisingly, very limited feedback information such as informing the encoder to

terminate the transmission for variable blocklength can attain considerably faster convergence to capacity as analyzed in [3]. A similar concept of using feedback to terminate the transmission can also be found in an early work by Viterbi [64] using sequential decision feedback. Unlike a variable-blocklength scheme that still has constant symbol length, the scheme proposed in [64] transmits the signal for a bit or a symbol (consisting of multiple bits) with variable duration and it is shown to provide up to 6dB SNR gain compared to a fixed symbol length (uncoded) transmission. However, the work did not gain much attention probably because of the impractical instantaneous feedback assumption in the era of analog communications as well as the fact that transmission was uncoded.

Thanks to modern digital integrated circuit (IC) technology for fast and low latency feedback decision computation, it is now practically possible to utilize instantaneous feedback (i.e., with much shorter delay than the symbol length) to improve communication reliability. Inspired by [64], we introduce Opportunistic Symbol Length Adaptation (OSLA), a feedback-based scheme that opportunistically adjusts the symbol length based on the noise realization observed at the receiver with sub-symbol granularity. OSLA is non-trivial generalization of [64] and it operates in discrete time for coded communication with feedback. Unlike Viterbi's work, where an M-ary signal is used to transmit a message of $\log_2 M$ bits to improve reliability, our system uses a multi-dimensional BPSK to transmit multiple coded bits without the constraint of using the same symbol length for $\log_2 M$ coded bits that are concurrently transmitted in an M-ary symbol. We propose a deliberate feedback scheme that prevents catastrophic feedback errors in noisy feedback channels, while using a tail-biting convolutional code (TBCC) or a turbo code in the forward channel.

OSLA is a *practical* instantaneous feedback-based scheme that can be applied to a communication system utilizing a convolutional code. Besides its superior reliability, OSLA possesses additional advantages of constant envelope signaling and

low complexity transmitters unlike deep learning-based schemes such as Deepcode [60], which may be important for Internet-of-Things applications where low-cost and low-complexity transmitters are favored. Furthermore, OSLA utilizes binary decision feedback, which does not suffer from any arithmetic imprecision, and is robust to feedback noise. We therefore consider our proposed system as a strong candidate for ultra-reliable low-latency communications (URLLC) with short blocklengths in next generation wireless communication standards.

The contributions of this work are summarized as follows:

1. We propose OSLA, a feedback-based variable-symbol-length transmission scheme, to reliably transmit convolutional (including TBCC and turbo) coded messages.

2. We generalize [64] and propose a new feedback decision scheme that determines the length of each coded bit based on the real-time sub-symbol-granularity realization of state (and branch) metrics of the decoding algorithm executed at the receiver.

3. It is shown that OSLA combined with TBCC outperforms fixed-length state-of-the-art short codes as well as the recently proposed deep learning-based feedback scheme Deepcode [60].

4. Scalability of OSLA combined with turbo codes is shown with various blocklengths and signal-to-noise ratios (SNRs) to provide consistent gains over a conventional fixed-blocklength scheme that does not utilize feedback.

5. We propose and evaluate a novel feedback scheme for OSLA that uses a pulse-based feedback signal with a hidden Markov model to synchronize the transmitter and receiver in noisy feedback channels for robust communications.

6. We extend OSLA to low latency applications with hard deadline by formulating policy optimization problems with constraint. The problems are solved with

Markov decision process and reinforcement learning tools.

The remaining parts of this chapter are organized as follows. Section 3.2 presents the proposed OSLA system for uncoded BPSK transmission. We then generalize the system to incorporate trellis-based coding including TBCC and turbo codes in section 3.3. Our proposed feedback scheme for OSLA is discussed in section 3.4. Evaluation and comparison to other schemes are provided in Section 3.6. Section 3.7 discusses the limitations and future directions of this work. Finally, Section 3.8 summarizes the work.

3.2 OSLA for Uncoded BPSK

3.2.1 OSLA-BPSK System Model

In this section, we introduce the discrete-time system model of sequential decision feedback for uncoded BPSK transmission. Although a similar continuous-time counterpart has been introduced in [64], we describe our discrete-time model for completeness of the work. We use the term *OSLA-BPSK* for the uncoded system which will be generalized to a trellis coded scheme in a subsequent section.

The OSLA-BPSK system involves a source and destination with two channels connecting them: the forward and feedback channel which are discrete in time as illustrated in Figure 3.1. In this section, we assume noiseless feedback and thus the source receives perfect feedback from the destination.

The source modulates the kth bit b_k with a (variable length) BPSK symbol that consists of a series of chips $x_{k,i}, i = 1, \cdots, N_k$, where a chip refers to the smallest unit of transmission and is generally much shorter than a symbol in timescale. Here i is the chip index and N_k denotes the number of chips for b_k. As will be explained below, N_k is a random quantity that depends on the channel noise realization. Considering

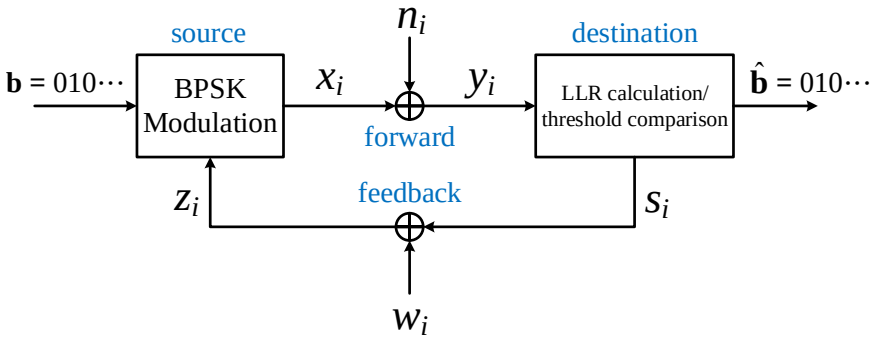


Figure 3.1: OSLA-BPSK system model

an AWGN model for the forward channel, the received chip is modeled as

$$y_{k,i} = x_{k,i} + n_{k,i} \tag{3.1}$$

where $x_{k,i} = (-1)^{b_k}\sqrt{P}$ with P denoting the signal power and $n_{k,i} \sim \mathcal{N}(0, \sigma_n^2)$ is zero-mean Gaussian noise with variance σ_n^2. The log likelihood ratio (LLR) of each chip can be written as

$$\Delta L_{k,i} = \log \frac{Pr(y_{k,i}|b_k = 0)}{Pr(y_{k,i}|b_k = 1)} = \frac{2\sqrt{P}y_{k,i}}{\sigma_n^2}. \tag{3.2}$$

The LLR of b_k upon receiving all chips $y_{k,i}$ is the summation of the LLR of each chip (due to the independence of the noise)

$$\text{LLR}(b_k) = \sum_{i=1}^{N_k} \Delta L_{k,i}. \tag{3.3}$$

The LLR in (3.3) can be calculated recursively in time as

$$L_i(b_k) = L_{i-1}(b_k) + \Delta L_{k,i}, \qquad i = 1, \ldots, N_k \tag{3.4}$$

with initial condition $L_0(b_k) = 0$.

In OSLA-BPSK, the destination calculates the cumulative LLR, $L_i(b_k)$, of each

bit, and informs the source (via the feedback channel) when to stop sending additional chips related to bit b_k and start transmitting the next bit b_{k+1}. The decision of moving (or advancing) to the next bit is made when the destination has enough confidence on the current bit, i.e., $|L_i(b_k)| \geq L$ is satisfied for a predetermined LLR threshold L. Since we assume perfect feedback for now, the source perfectly receives the bit-advancing decision so it is always synchronized with the destination with one chip delay as shown in Figure 3.2. The design of a practical feedback scheme is discussed in the later section.

Under this model, the destination can guarantee that the LLR of each received bit is at least L and thus ensure a target bit error rate (BER) performance. Because the number of chips, N_k, for each bit is a random variable that depends on the noise realization, the length (and thus energy) of a symbol for each bit automatically adapts to the noise realization to ensure the target reliability.

The noise variance σ_n^2 is governed by the chip sampling rate f_s of the system as $\sigma_n^2 = \frac{N_0}{2} \cdot f_s$ where N_0 is the noise power spectral density. Defining $\Delta t = 1/f_s$ as the time duration of one chip, the average length of one symbol is obtained by

$$\overline{T}_{\text{sym}} = \mathbb{E}\{N_k\}\Delta t = \overline{N}\Delta t. \tag{3.5}$$

The average energy per symbol E_s in OSLA is obtained by $E_s = P \cdot \overline{T}_{\text{sym}}$. And the LLR of each sample is expressed by

$$\Delta L_{k,i} = \frac{2\sqrt{P}(\pm\sqrt{P} + n_{k,i})}{\frac{N_0}{2} f_s} = \pm\frac{4P\Delta t}{N_0} + \frac{4\sqrt{P}\Delta t \cdot n_{k,i}}{N_0} \tag{3.6a}$$

$$\sim \mathcal{N}(\pm\frac{4P\Delta t}{N_0}, \frac{8P\Delta t}{N_0}). \tag{3.6b}$$

Therefore, the bit error performance of OSLA-BPSK is fully determined by $\frac{P\Delta t}{N_0}$ and L.

An example of the OSLA-BPSK transmission is illustrated in Figure 3.2. Notice the Δt delay between the timing of bit-advancing decision from the destination and the acknowledgement at the source. It can be regarded as an *one chip delay*[†] *feedback system* whose delay is much shorter than the average *symbol* length as $\Delta t \ll \overline{T}_{\text{sym}}$ holds in general.

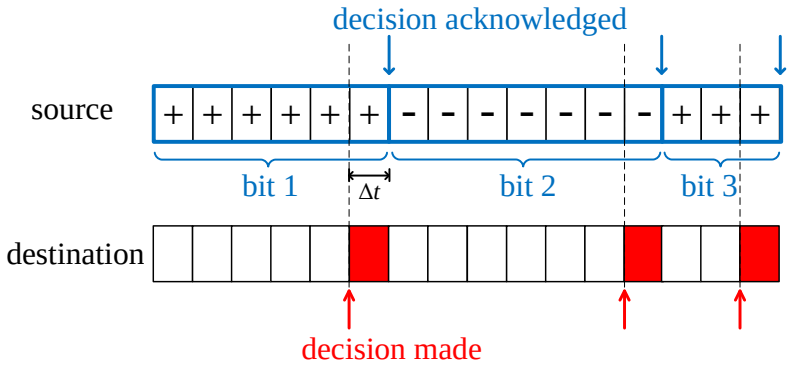


Figure 3.2: Forward and feedback signal in OSLA-BPSK on a timeline

3.2.2 Performance Analysis

The performance analysis of OSLA-BPSK is similar to the approach in [64]. A continuous time model in [64] allows analyzing the average stopping time through differential equations. We summarize the analysis result with our parameter definitions here.

As Δt approaches 0, the destination can make the LLR of each bit to be exactly $+L$ or $-L$. Without loss of generality, assume $+L$ is observed and thus $\hat{b}_k = 0$ is estimated. Under this circumstance the probability of $b_k = 1$ equals to the error probability P_e, and the relation of P_e and L can be written as

$$\log \frac{1 - P_e}{P_e} = L \quad \text{or} \quad P_e = \frac{1}{e^L + 1}. \tag{3.7}$$

[†]In this work, we consider this one chip delay 'instantaneous'.

To analyze the average symbol time $\overline{T}_{\text{sym}}$, first note that the symbol transmission ends when the cumulative LLR reaches the threshold L. So the symbol time can be written as

$$T_{\text{sym}} = \inf\{t \geq 0 : L_t \notin (-L, L)\} \tag{3.8}$$

where L_t is now continuous in time and denotes the cumulative LLR at time t. Next, define the average symbol time given the initial cumulative LLR $L_0 = l$ as

$$\overline{T}(l) = \mathbb{E}\{T_{\text{sym}} | L_0 = l\}. \tag{3.9}$$

Following the derivation in [64], the result can be obtained through solving a differential equation given by

$$\overline{T}'(l) + \overline{T}''(l) + \frac{N_0}{4P} = 0. \tag{3.10}$$

With the boundary condition $\overline{T}(L) = \overline{T}(-L) = 0$, the solution of (3.10) is

$$\overline{T}(l) = -\frac{N_0}{4P}(L + l) + \frac{N_0 L}{2P} \frac{e^L - e^{-l}}{e^L - e^{-L}} \tag{3.11}$$

and therefore, the average symbol length with the initial $L_0 = 0$ is

$$\overline{T}_{\text{sym}} = \overline{T}(0) = \frac{N_0 L}{4P} \tanh(L/2). \tag{3.12}$$

Combining (3.7) and (3.12), and using the average energy per bit $E_b = P \cdot \overline{T}_{\text{sym}}$, we have

$$4\frac{E_b}{N_0} = (1 - 2P_e) \log \frac{1 - P_e}{P_e} \xrightarrow{P_e \ll 1} \quad P_e \approx e^{-4\frac{E_b}{N_0}}. \tag{3.13}$$

Compared to fixed-length BPSK, which has error rate $P_e = Q(\sqrt{2\frac{E_b}{N_0}}) \approx e^{-\frac{E_b}{N_0}}$ (at high SNR), OSLA-BPSK has 6 dB (factor of 4) gain w.r.t. E_b/N_0 as found in [64].

3.2.3 OSLA-BPSK Symbol Length and Signal Spectrum

For the transmission of $K+1$ symbols the transmitted signal can be expressed as

$$X(t) = \sum_{k=0}^{K} B_k \; p\Big(\frac{t - S_k}{T_k}\Big), \tag{3.14}$$

where B_k's are i.i.d. random variables (r.v.) with equal probability of being $+1$ or -1, and

$$p(t) = \begin{cases} 1 & , 0 < t < 1 \\ 0 & , \text{otherwise.} \end{cases} \tag{3.15}$$

With this expression, the pulse corresponding to the k-th symbol has duration equal to T_k and the starting time of the pulse is S_k, where S_k is defined as

$$S_k = \begin{cases} 0 & , k = 0 \\ \sum_{i=0}^{k-1} T_i & , k \geq 1. \end{cases} \tag{3.16}$$

To make the discussion more concrete, we now consider the statistics of the i.i.d random variables T_k for the system described in [64, 65]. The LLR value calculated at the destination can be considered as a Wiener process with nonzero drift [66]. The symbol length T is a stopping time, defined as the first time the LLR reaches the boundary $+L$ or $-L$. The pdf of T in this case is given by [66]

$$f_T(t) = \frac{Le^{-\gamma t}(e^{-\frac{L}{2}} + e^{\frac{L}{2}})}{\sqrt{16\pi\gamma t^3}} \sum_{k=-\infty}^{\infty} (1+4k) e^{-\frac{L^2(1+4k)^2}{16\gamma t}}, \tag{3.17}$$

or in an approximation form when $L \gg 1$

$$f_T(t) \approx \frac{L}{\sqrt{16\pi\gamma t^3}} e^{-\frac{(L-4\gamma t)^2}{16\gamma t}}, \tag{3.18}$$

where $\gamma = \frac{P}{N_0}$, P is the power of the signal and N_0 is the noise spectral density. The average energy per bit is given by $E_b = P\mathbb{E}\{T\}$.

For a discrete model, the stopping time can be apporoximated by accounting for exceeding the exact boundary (due to the coarse time resolution) with a modified boundary value L' [67]. A good model is to use $L' = L + 0.586\sigma_s$ to replace L in (3.18) [67]. A probability mass function (PMF) to replace PDF is obtained by using the chip duration Δt for integration.

The autocorrelation function of the signal in (3.14) is defined as

$$R(t,\tau) = \mathbb{E}\{X(t+\tau)X(t)\}. \tag{3.19}$$

In the following, we only consider the case $\tau > 0$. The case of $\tau \leq 0$ can be derived similarly. It is easy to see that

$$\begin{aligned} R(t,\tau) &= \sum_{k=0}^{K} \mathbb{E}\Big\{p\Big(\frac{t+\tau-S_k}{T_k}\Big)p\Big(\frac{t-S_k}{T_k}\Big)\Big\} \\ &= \sum_{k=0}^{K} A_k = A_0 + \sum_{k=1}^{K} A_k \end{aligned} \tag{3.20}$$

where A_k denotes each of the expectation terms inside the summation. In the following, anticipating the fact that we will consider $R(t,\tau)$ for $t \to \infty$ we only consider these expressions for $t > 0$.

First consider the term A_0:

$$\begin{aligned} A_0 &= \mathbb{E}\{p\Big(\frac{t+\tau}{T_0}\Big)p\Big(\frac{t}{T_0}\Big)\} \\ &= \mathbb{E}\{\mathbb{1}(0 < t < T_0 - \tau)\} \\ &= 1 - F_T(t+\tau), \end{aligned} \tag{3.21}$$

where $\mathbb{1}(\cdot)$ is the indicator function such that it equals to 1 if the condition in the parentheses is satisfied, and 0 otherwise, and $F_T(\cdot)$ is the cdf of T.

Next, consider A_k for $k \geq 1$. Note that due to (3.16), T_k and S_k are independent random variables. As a result we have

$$\begin{aligned} A_k &= \mathbb{E}_{T_k}\mathbb{E}_{S_k}\{p\Big(\frac{t+\tau-S_k}{T_k}\Big)p\Big(\frac{t-S_k}{T_k}\Big)\} \\ &= \mathbb{E}_{T_k}\Big\{\int_{-\infty}^{\infty} f_{S_k}(s)p\Big(\frac{t+\tau-s}{T_k}\Big)p\Big(\frac{t-s}{T_k}\Big)ds\Big\} \\ &= \mathbb{E}_{T_k}\Big\{\int_{t+\tau-T_k}^{t} f_{S_k}(s)\mathbb{1}(\tau < T_k)ds\Big\} \\ &= \mathbb{E}_T\Big\{\mathbb{1}(\tau < T)\big[F_{S_k}(t) - F_{S_k}(t+\tau-T)\big]\Big\}. \end{aligned} \tag{3.22}$$

Combining these two expressions, we have

$$\begin{aligned} R(t,\tau) &= 1 - F_T(t+\tau) \\ &\quad + \mathbb{E}_T\Big\{\mathbb{1}(T > \tau)\sum_{k=1}^{K}\big[F_{S_k}(t) - F_{S_k}(t+\tau-T)\big]. \end{aligned} \tag{3.23}$$

The expression in (3.23) requires the evaluation of $F_{S_k}(t)$. This can be done as

follows.

$$
\begin{aligned}
F_{S_k}(t) &= \mathbb{P}(S_k \leq t) \\
&= \int_0^\infty \mathbb{P}(S_k \leq t | T_k = x) f_T(x) dx \\
&= \int_0^\infty \mathbb{P}(S_{k-1} \leq t - x | T_k = x) f_T(x) dx \\
&= \int_0^\infty \mathbb{P}(S_{k-1} \leq t - x) f_T(x) dx \\
&= \int_0^\infty F_{S_{k-1}}(t - x) f_T(x) dx \\
&= F_{S_{k-1}}(t) * f_T(t) \\
&= F_{S_{k-2}}(t) * f_T(t) * f_T(t) \\
&= \cdots \\
&= F_{S_1}(t) * f_T(t) * \cdots * f_T(t), \qquad (3.24)
\end{aligned}
$$

where $*$ denotes convolution. Observe that

$$
F_{S_1}(t) = F_T(t) = \int_0^t f_T(x) dx = u(t) * f_T(t) \qquad (3.25)
$$

where $u(t)$ is the step function. Therefore, $F_{S_k}(t)$ can be obtained by convolving $u(t)$ with $f_T(t)$ k times.

It is easier to express these convolutions in frequency domain, where convolution can be replaced by multiplication. Let $\Phi_k(\omega), \Phi(\omega)$ and $U(\omega)$ be the Fourier transform

of $F_{S_k}(t)$, $f_T(t)$ and $u(t)$, respectively[‡]. Then

$$\Phi_k(\omega) = U(\omega)\Phi^k(\omega). \tag{3.26}$$

Similarly, to obtain $F_{S_k}(t+\tau-T)$, we can use its Fourier transform $\Phi_k(\omega)e^{-j\omega(T-\tau)}$.

Therefore, the expectation term in (3.23) becomes (for $\omega > 0$)

$$\begin{aligned}
&\mathbb{E}\Big\{ \mathbb{1}(T>\tau) \sum_{k=1}^{K} \mathcal{F}^{-1}\big\{U(\omega)\Phi^k(\omega) - U(\omega)\Phi^k(\omega)e^{-j\omega(T-\tau)}\big\}\Big\} \\
&= \mathbb{E}\Big\{ \mathbb{1}(T>\tau) \sum_{k=1}^{K} \mathcal{F}^{-1}\big\{U(\omega)\Phi^k(\omega)(1-e^{-j\omega(T-\tau)})\big\}\Big\} \\
&= \mathbb{E}\Big\{ \mathbb{1}(T>\tau) \mathcal{F}^{-1}\big\{U(\omega)\big[\sum_{k=1}^{K} \Phi^k(\omega)\big](1-e^{-j\omega(T-\tau)})\big\}\Big\} \\
&= \mathbb{E}\Big\{ \mathbb{1}(T>\tau) \mathcal{F}^{-1}\big\{U(\omega)\frac{\Phi(\omega)(1-\Phi^K(\omega))}{1-\Phi(\omega)}(1-e^{-j\omega(T-\tau)})\big\}\Big\} \\
&= \mathcal{F}^{-1}\Big\{ \frac{U(\omega)\Phi(\omega)(1-\Phi^K(\omega))}{1-\Phi(\omega)} \mathbb{E}_T\big\{ \mathbb{1}(T>\tau)\big[1-e^{-j\omega(T-\tau)}\big]\big\}\Big\} \\
&= \mathcal{F}^{-1}\Big\{ \frac{U(\omega)\Phi(\omega)(1-\Phi^K(\omega))}{1-\Phi(\omega)} \int_{\tau}^{\infty} f_T(s)(1-e^{j\omega(\tau-s)})ds\Big\}.
\end{aligned} \tag{3.27}$$

Let

$$Q(\omega;\tau) = \int_{\tau}^{\infty} f_T(s)(1-e^{j\omega(\tau-s)})ds \tag{3.28}$$

and

$$G(\omega;\tau) = \frac{\Phi(\omega)(1-\Phi^K(\omega))}{1-\Phi(\omega)} Q(\omega;\tau), \quad \omega > 0. \tag{3.29}$$

[‡]Although the Fourier Transform of $u(t)$ does not exist, we use it formally in our expressions. As can be seen from the final result in (3.30), the expressions depend on Fourier transforms that are well defined.

Then, for $t, \tau > 0$ we have

$$\begin{aligned} R(t,\tau) &= 1 - F_T(t+\tau) + \mathcal{F}^{-1}\Big\{U(\omega)G(\omega;\tau)\Big\} \\ &= 1 - F_T(t+\tau) + u(t) * g(t;\tau) \\ &= 1 - F_T(t+\tau) + \int_{-\infty}^{t} g(s;\tau)ds \end{aligned} \tag{3.30}$$

where $g(t;\tau)$ is the inverse Fourier transform of $G(\omega;\tau)$ (w.r.t the first argument).

Note that the autocorrelation function $R(t,\tau)$ depends on both t and τ as expected. Unlike the standard case for constant T where the autocorrelation becomes cyclostationary and a time randomization argument is invoked to average out the dependence on t, here we follow a different approach. First we consider the limit of an infinite sequence of symbols, i.e., we consider $K \to \infty$ for fixed t, τ. Taking this limit results in exactly the same equations with the exception of (3.29) that simplifies[§] to $G(\omega;\tau) = \frac{\Phi(\omega)}{1-\Phi(\omega)}Q(\omega;\tau)$.

Furthermore, it is observed (as shown in Figure 3.10) that for T is not a constant, the autocorrelation function $R(t,\tau)$ converges for large t. To get rid of t, we can choose $t \to \infty$ when $K = \infty$, and define $R(\tau) = \lim_{t\to\infty} R(t,\tau)$. This is a reasonable choice if we don't know where $t = 0$ is and we just sample the signal at a random time.

[§]Note that $|\Phi(\omega)| = |\mathbb{E}\{e^{-j\omega T}\}| < \mathbb{E}\{|e^{-j\omega T}|\} = 1$ for $\omega > 0$ thanks to Jensen's inequality and the assumption that T is not a constant and has a continuous pdf. Therefore we have $\Phi^K(\omega) \to 0$ and thus the limit result.

From (3.30), we have

$$\begin{aligned}
\lim_{t\to\infty} R(t,\tau) &= \int_{-\infty}^{\infty} g(s;\tau)ds \\
&= \lim_{\omega\to 0} G(\omega,\tau) \\
&= \lim_{\omega\to 0} \frac{\Phi(\omega)Q(\omega;\tau)}{1-\Phi(\omega)} \\
&= \lim_{\omega\to 0} \frac{Q(\omega;\tau)}{1-\Phi(\omega)} \\
&= -\frac{Q'(\omega;\tau)}{\Phi'(\omega)}\Big|_{\omega=0}, \qquad (3.31)
\end{aligned}$$

where the last equality is due to L'Hospital's rule. For the numerator we have

$$\begin{aligned}
Q'(\omega;\tau)\Big|_{\omega=0} &= \frac{d}{d\omega}\int_{\tau}^{\infty} f_T(s)(1-e^{j\omega(\tau-s)})ds\Big|_{\omega=0} \\
&= \int_{\tau}^{\infty} f_T(s)\frac{d}{d\omega}(1-e^{j\omega(\tau-s)})ds\Big|_{\omega=0} \\
&= \int_{\tau}^{\infty} f_T(s)(-j)(\tau-s)e^{j\omega(\tau-s)}ds\Big|_{\omega=0} \\
&= -j\int_{\tau}^{\infty}(\tau-s)f_T(s)ds. \qquad (3.32)
\end{aligned}$$

For the denominator we have

$$\begin{aligned}
\Phi'(\omega)\Big|_{\omega=0} &= \frac{d}{d\omega}\int_{-\infty}^{\infty} f_T(s)e^{-j\omega s}ds\Big|_{\omega=0} \\
&= \int_{-\infty}^{\infty} f_T(s)\frac{d}{d\omega}e^{-j\omega s}ds\Big|_{\omega=0} \\
&= \int_{-\infty}^{\infty} f_T(s)(-js)e^{-j\omega s}ds\Big|_{\omega=0} \\
&= -j\int_{-\infty}^{\infty} sf_T(s)ds \\
&= -j\mathbb{E}\{T\}. \qquad (3.33)
\end{aligned}$$

Therefore, we have for $\tau > 0$

$$\begin{aligned}
R(\tau) &= -\frac{-j\int_{\tau}^{\infty}(\tau - s)f_T(s)ds}{-j\mathbb{E}\{T\}} \\
&= \frac{1}{\mathbb{E}\{T\}}\int_{\tau}^{\infty}(s-\tau)f_T(s)ds. \qquad (3.34)
\end{aligned}$$

The signal spectrum $S(f)$ can be obtained by taking the Fourier transform of $R(\tau)$. However, without a close-form expression of $R(\tau)$, it is hard to obtain an analytical solution of $S(f)$. Nevertheless, given a known pdf f_T, the spectrum can be assessed numerically.

3.3 OSLA for Convolutional Codes

3.3.1 OSLA with Viterbi Algorithm

We now further enhance the performance of OSLA-BPSK by applying channel coding. Note that directly concatenating an outer channel encoder and decoder in an OSLA-BPSK scheme can only provide the uncoded performance gain because the

importance of each later bit is not equal given the information of early bits that are already received. On the other hand, if we want to fully exploit feedback, we need to design the entire coding scheme to take advantage of feedback. This will result in a unwanted complexity increase. In order to strike a balance between performance and complexity, we generalize the OSLA-BPSK to include convolutional codes (CC) with a modified Viterbi algorithm (VA) named OSLA-VA, resulting in a new coding scheme that we call OSLA-CC.

Figure 3.3 shows the OSLA-CC system with three main blocks: encoding, modulation, and OSLA-VA. At the encoding stage, information bits $\mathbf{b}$ are first encoded by a (tail-biting) CC into the coded bits $\mathbf{c}$, which is done without utilizing any feedback. On the other hand, the modulation of the coded bits $\mathbf{c}$ takes advantage of feedback and it is intertwined with OSLA-VA to improve reliability. Finally, OSLA-VA serves as the demodulator as well as the decoder. During the transmission, OSLA-VA evaluates path metrics in the trellis as the decoder and provides feedback so that the length of each coded bit is adjusted.

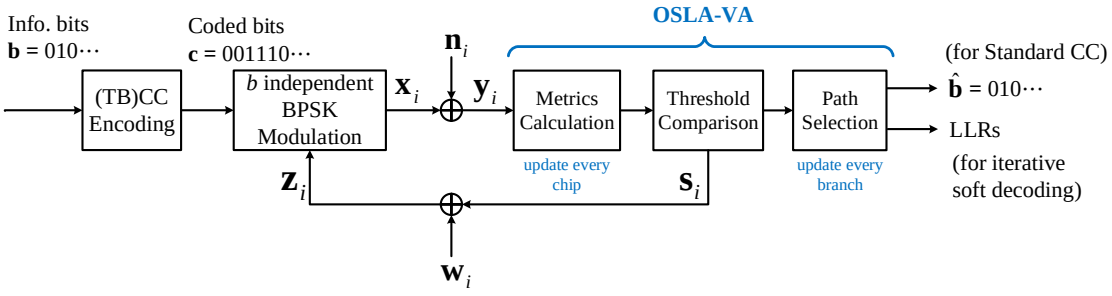


Figure 3.3: OSLA for convolutional codes (OSLA-CC)

Consider a rate a/b convolutional code, where every a information bits are encoded into b coded bits. The b coded bits that correspond to the kth branch in the trellis are denoted as $\mathbf{c}_k$ or $\{c_{km}, m = 1, \cdots, b\}$ with index m denoting the mth bit in that branch. To transmit the b coded bits concurrently, OSLA uses b independent BPSK channels (each using orthogonal resources other than time such as orthogonal

frequencies). The channel model is the same as (3.1) except for that subscript k is changed to km to denote mth coded bit in kth branch. Note that the length (or number of chips N_{km}) of each coded bit is not fixed but variable.

If N_{km} were deterministic or known, the decoding algorithm would be the same as a conventional VA [68]. The main difference of OSLA-VA is that the decoding process itself is responsible for determining the length of each coded bit N_{km} and ensuring that the decoder has sufficient confidence on the most probable codeword.

In VA, the confidence of a codeword can be represented by the state metric of the corresponding trellis path following classic definitions [69]. And the state metric $M_k(s)$ for the state s after the kth branch is defined as the log likelihood (LL), following the update rule with path selection given by

$$M_k(s) = \max_{(s' \to s) \in \mathcal{T}} \{M_{k-1}(s') + \mu_k(\mathbf{c}^{s' \to s})\}. \tag{3.35}$$

In (3.35), $\mathcal{T}$ is the set of all valid state transitions, $\mathbf{c}^{s' \to s}$ denotes the coded bits associated with state transition $s' \to s$, and $\mu_k(\mathbf{c}^{s' \to s})$ is the kth branch metric given by

$$\mu_k(\mathbf{c}^{s' \to s}) = \sum_{m=1}^{b} \sum_{i=1}^{N_{km}} \Delta\mathrm{LL}_{km,i}(c_m^{s' \to s}) \tag{3.36}$$

with $\Delta\mathrm{LL}_{km,i}(c_m) = \log Pr(y_{km,i}|c_m)$ denoting the chip LL of the coded bit c_m calculated by the received chip $y_{km,i}$.

To determine N_{km} for each coded bit *during* the decoding process, the decoder is designed to observe additional received chips (increasing N_{km}) for the same coded bit until a predetermined criterion is met, which indicates the reliability of the most probable codeword so far. It is important to note that although the length of a coded bit N_{km} only affects the reliability of the corresponding coded bit, the criterion should be a function of all coded bit candidates $\mathbf{c}^{(j)}, j = 1, \cdots, 2^b$ because coded bits are not

independent.

To quantify the confidence of coded bits and ensure the reliability of the most probable codeword at the same time, the confidence of $\mathbf{c}^{(j)}$ is defined as the largest state metric for a candidate $\mathbf{c}^{(j)}$, written as

$$W_k(\mathbf{c}^{(j)}) = \max_{s,s':\mathbf{c}^{s'\to s}=\mathbf{c}^{(j)}} \{M_{k-1}(s') + \mu_k(\mathbf{c}^{s'\to s})\}. \tag{3.37}$$

Figure 3.4 shows the relation of the metrics $M_k(s)$ and $W_k(\mathbf{c}^{(j)})$ in an example trellis diagram.

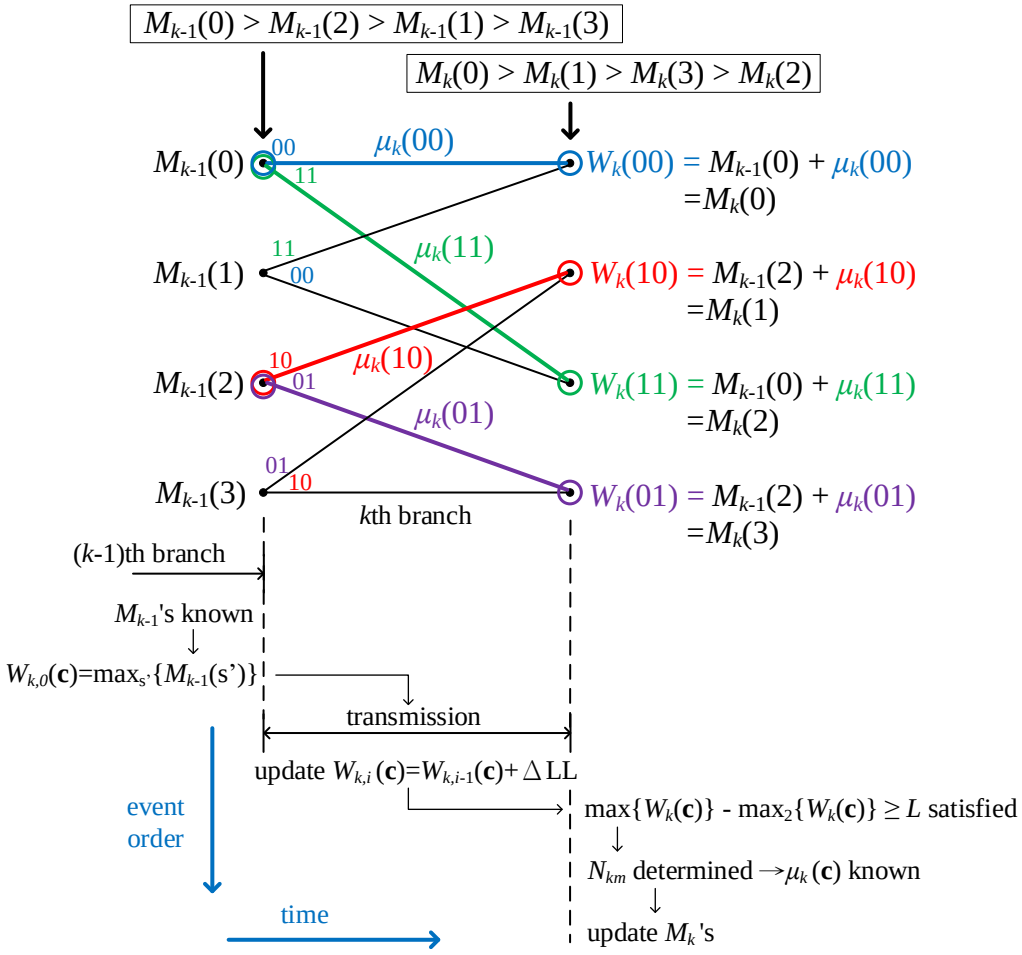


Figure 3.4: Metrics in the trellis. Different colors are used for different coded bits candidate $\mathbf{c}^{(j)}$.

Notice that (3.37) involves the length of each coded bits N_{km} via $\mu_k(\mathbf{c}^{s'\to s})$ that is increasing as the decoder receives additional chips for each coded bit. The decoder has enough confidence on the largest $W_k(\mathbf{c}^{(j)})$ (corresponding to the most probable coded bits) when the metric is sufficiently larger than the second largest. Thus, one possible (but naive) bit-advancing criterion to stop increasing N_{km} for the kth state transition is defined as

$$\max_j W_k(\mathbf{c}^{(j)}) - \max_j{}_2 W_k(\mathbf{c}^{(j)}) \geq L \tag{3.38}$$

where $\max_2(\cdot)$ denotes the second largest value.

The naive criterion (3.38) makes the bit-advancing decision based on the set of coded bits $\mathbf{c}^{(j)}$. Thus it forces the same length N_{km} for all bit indices m for the kth branch. When this method is adopted, $W_k(\mathbf{c}^{(j)})$ can be sequentially updated with time index i as in

$$W_{k,i}(\mathbf{c}^{(j)}) = W_{k,i-1}(\mathbf{c}^{(j)}) + \sum_{m=1}^{b} \Delta\mathrm{LL}_{km,i}(c_m^{(j)}) \tag{3.39}$$

with an initial condition

$$W_{k,0}(\mathbf{c}^{(j)}) = \max_{s':\mathbf{c}^{(j)}\in\mathcal{C}_{s'}} \{M_{k-1}(s')\} \tag{3.40}$$

where $\mathcal{C}_{s'}$ denotes the set of all possible coded bits with associated state transition starting from s'. The decoder evaluates (3.38) using $W_{k,i}(\mathbf{c}^{(j)})$ instead of $W_k(\mathbf{c}^{(j)})$ as time index i increases until the criterion is satisfied. At that point, N_{km} is set to the current time index i for all m.

Criterion (3.38) is highly inefficient because the coded bits for $\max_j W_k(\mathbf{c}^{(j)})$ and those for $\max_j{}_2 W_k(\mathbf{c}^{(j)})$ are often identical except for one. Determining the length N_{km} for all m's based on a single non-identical coded bit can result in unnecessarily

longer lengths for other common coded bits without improving the reliability because the excessive length (i.e., energy) for those equally contributes to the max and $\max_2$ terms.

To resolve this issue, we propose an *asynchronous multi-channel bit-advancing scheme* where each coded bit sequence with a particular index m is transmitted using a dedicated orthogonal BPSK channel (e.g., orthogonal frequency carrier) to allow separate asynchronous bit-advancing decisions for each channel. For that, we set another individual and asynchronous bit-advancing criterion for each bit/channel in addition to the criterion (3.38). The individual criterion has a form very similar to (3.38) except that the inspected metrics only account for one bit. Following the same strategy, the metric (3.39) is separated for each coded bit in $\mathbf{c}^{(j)} = \{c_1^{(j)}, ..., c_b^{(j)}\}$ such as

$$W_{km,i}(c) = W_{km,i-1}(c) + \Delta \mathrm{LL}_{km,i}(c) \tag{3.41}$$

where the coded bit c is either 0 or 1, with an initial condition
$W_{km,0}(c) = \max_{j:c_m^{(j)}=c} W_{k,0}(\mathbf{c}^{(j)})$.

The decoder with this asynchronous bit-advancing scheme evaluates the first criterion (3.38) using (3.39) as well as the individual criterion version using $W_{km,i}(c)$ for each channel. It determines the length N_{km} for a particular coded bit when *either* of the two criteria is satisfied and advances the channel to the next coded bit without waiting for the decision on the other channels/bits. When the mth bit is advanced with length N_{km}, the calculation of (3.39) stops at $i = N_{km}$ and (3.36) only sums up to $i = N_{km}$ for that particular bit.

When the length N_{km} is determined for all coded bits, the state metrics (3.35) based on branch metrics (3.36) are calculated for the next branch decoding. Some BPSK channels may advance to the next branch index $k+1$ before the completion of

the state metric updates for the other coded bits for the kth branch. Those channels can start *pre-calculating* the second term $\Delta \mathrm{LL}_{k+1,m,i}(c_m^{(j)})$ in (3.41) for the $(k+1)$-th branch.

The algorithm pseudo-code of OSLA-VA is described in Algorithm 1.

3.3.2 OSLA with Iterative Decoding of TBCC and Turbo Codes

State-of-the-art codes often involve iterative decoding that uses LLR as the decoder (soft) input. To extend OSLA to such coding schemes, we propose to concatenate an iterative decoder to trellis coded OSLA. Specifically, we apply OSLA to TBCC (or turbo) codes, which turns out to be particularly effective for relatively short (or long) code lengths.

The OSLA-VA scheme discussed in Section II.A is designed to identify the most probable trellis path with the largest metric as the decoded bits after the last path selection at the end of the trellis. As the length N_{km} of each coded bit is determined during the OSLA-VA operation, the LLR for each coded bit c_{km} can be obtained as a byproduct using the following equation:

$$\mathrm{LLR}(c_{km}) = \sum_{i=1}^{N_{km}} \frac{2\sqrt{P} y_{km,i}}{\sigma_n^2}. \tag{3.42}$$

These LLRs for all coded bits can be fed into a conventional soft-input decoder for TBCC or turbo codes to decode the original information bits. The encoder and decoder are exactly the same as in a fixed length scheme, whereas the OSLA transmission scheme can be viewed as a part of the modulation/demodulation process.

TBCC is one of the state-of-the-art short codes [70]. It is widely used for short message communications including the LTE control channel [71]. Unlike a standard convolutional code, the trellis of a TBCC starts with the state determined by the tail bits as the name indicates. Since the starting state is not pre-determined, decoding

Algorithm 1: OSLA-VA

input : Trellis $\mathcal{T}$

`// Initialization`

$M_0(s) = 0$ for every state s in $\mathcal{T}$

for *each branch* k **do**

  `// Reset metrics`

  for *each coded bits sequence* j **do**

    $W_k(\mathbf{c}^{(j)}) = \max_{s,s':\mathbf{c}^{s'\to s}=\mathbf{c}^{(j)}}\{M_{k-1}(s') + \mu_k(\mathbf{c}^{s'\to s})\}$

  for *each dimension* m **do**

    $W_{km}(c) = \max_{j:c_m^{(j)}=c} W_k(\mathbf{c}^{(j)})$

  `// Keep receiving chips`

  $advanceFlag = [0, 0, \cdots, 0]$

  while *not all advanceFlag* **do**

    for *each dimension* m **do**

      if *advanceFlag[m]* **then**

        store received chip in a buffer for future use

      else

        calculate $\Delta\mathrm{LL}_{km}(0)$ and $\Delta\mathrm{LL}_{km}(1)$

    `// Check branch`

    for *each coded bits sequence* j **do**

      $\Delta\mathrm{LL}_k(\mathbf{c}^{(j)}) = \sum_{m=1}^{b} \Delta\mathrm{LL}_{km}(c_m^{(j)})$

      $\mu_k(\mathbf{c}^{(j)}) \leftarrow \mu_k(\mathbf{c}^{(j)}) + \Delta\mathrm{LL}_k(\mathbf{c}^{(j)})$

      $W_k(\mathbf{c}^{(j)}) \leftarrow W_k(\mathbf{c}^{(j)}) + \Delta\mathrm{LL}_k(\mathbf{c}^{(j)})$

    if $\max W_k(\mathbf{c}^{(j)}) - \max_2 W_k(\mathbf{c}^{(j)}) \geq L$ **then**

      $advanceFlag = [1, 1, \cdots, 1]$

    `// Check each dimension`

    for *each dimension* m **do**

      $W_{km}(0/1) \leftarrow W_{km}(0/1) + \Delta\mathrm{LL}_{km}(0/1)$

      if $|W_{km}(0) - W_{km}(1)| \geq L$ **then**

        $advanceFlag[m] = 1$

  `// Update branch`

  for *each state* s **do**

    $M_k(s) = \max_{(s'\to s)\in\mathcal{T}}\{M_{k-1}(s') + \mu_k(\mathbf{c}^{s'\to s})\}$

needs to start with equal metrics for all states treating them as a potential valid state. In our proposed scheme, LLRs (3.42) obtained from OSLA-VA are fed into a wrap-around Viterbi algorithm (WAVA) decoder [40] for additional iterative TBCC decoding. In WAVA, a standard VA is performed (using LLRs from OSLA-VA as the soft-input) for each iteration, which is repeated until the stopping criterion is met for the metrics of all tail-biting paths. WAVA gaurantees that the output tail-biting path is the optimal solution if the stopping criterion is met before the maximum iteration number is reached. In this scheme, OSLA-VA can be regarded as the first iteration of WAVA (or simply part of the demodulation). The proposed combination of OSLA and TBCC is termed OSLA-TBCC.

Turbo codes, on the other hand, are well known for their excellent performance for long blocklengths. A widely used turbo code adopted in the LTE standard [71] is a rate 1/3 code that uses two recursive systematic convolutional (RSC) codes with an interleaver. It consists of four parts: a payload sub-block, two parity sub-blocks, and 12 tail bits. As the payload sub-block is identical to the (uncoded) information bits, it can be regarded as a systematic code. The second and third parity sub-blocks are the output bits of the two RSC codes whose constraint length is 4. As shown in Figure 3.5, the second RSC encoder takes the interleaved information bits as the input. The 12 tail bits are used to terminate the RSC trellis paths with zero states.

Application of OSLA to turbo codes is straight-forward. Since the payload sub-block is identical to the uncoded information bits, OSLA-BPSK can be directly applied. The second and third parity sub-blocks are the outputs of RSC codes. Therefore OSLA-VA is used for the transmission of these. Finally, the tail bits are transmitted with OSLA-BPSK. These four parts can be transmitted either sequentially or simultaneously with orthogonal resources (e.g., orthogonal sub-carriers) as independent messages. After completing OSLA transmissions of all four parts, the LLRs are concatenated together and sent to a conventional turbo decoder for iterative decod-

ing. Figure 3.5 shows the structure of combining the proposed OSLA scheme with turbo codes (tail bits are omitted for simplicity). This scheme is termed OSLA-turbo.

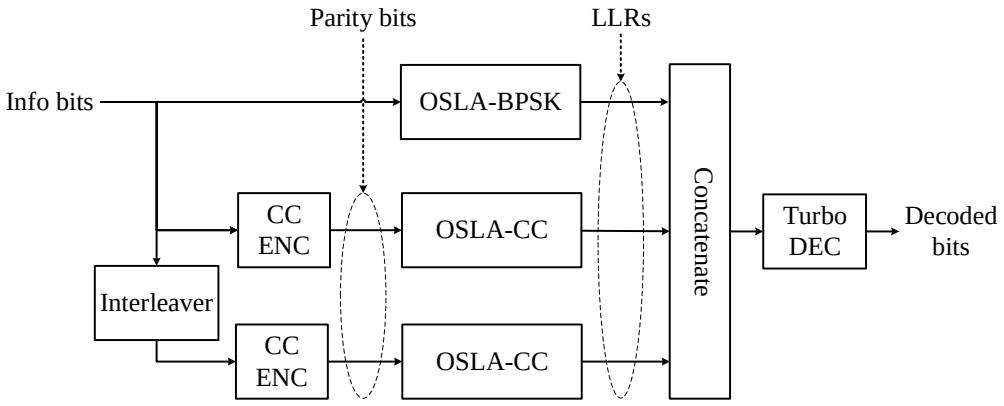


Figure 3.5: OSLA with turbo codes

3.3.3 Complexity of OSLA

Instantaneous feedback implies that the latency to compute and send the feedback decision to the source/transmitter is negligible compared to the average symbol length. The proposed OSLA assumes a chip-based feedback scheme where the decision must be made for each chip in real-time. Complexity of OSLA feedback decision, therefore, is a main concern for applying OSLA to practical systems. Note that the complexity of OSLA-BPSK is significantly lower than that of OSLA-VA. The additional iterative decoding step such as WAVA involved in OSLA-TBCC or OSLA-turbo is irrelevant to real-time feedback decision computation as it can be performed *after* receiving all coded bits via OSLA-VA or OSLA-BPSK. Therefore, in this section, we focus on analyzing the complexity of OSLA-VA to assess the feasibility of OSLA in real-time systems.

Computation of OSLA-VA can be categorized into two parts: *chip update* and *branch update*. After every chip update, the destination/receiver updates the metric and decides whether to advance to the next bit/symbol or not. When every coded bit

of the k-th branch has been advanced, branch metrics are updated by (3.35), followed by add-compare-select (ACS) operations for path selection.

Chip update involves computing the metrics W_k in (3.39) and W_{km} in (3.41), and also checking if any advancing criterion is satisfied. For every chip, we first obtain $\Delta\text{LL}_{km,i}(c)$ for all m and $c \in \{0, 1\}$. For BPSK signaling $x = \pm\sqrt{P}$ in the AWGN channel, $\Delta\text{LL}_{km,i}(c) = \frac{y_{km,i}x}{\sigma_n^2} + c_0$ for a constant c_0 holds. Calculating its scaled version (by a factor of a given constant σ_n^2) can be further simplified without explicit computations by ignoring the constant term and postponing the sign operation in $y_{km,i}x$ to a later add operation. These values are then used for updating (3.39) and (3.41), which involve 1 and b additions, respectively. A total of $2b + b2^b$ additions are needed to evaluate all possible (scaled versions of) W_k and W_{km}. The remaining steps involve finding the two largest metrics and comparing their difference to a fixed (and scaled) threshold value as in (3.38) and the criteria for individual coded bits. These steps further require at most $2^{b+1} + b + 1$ comparisons and $b + 1$ additions. Therefore, the total number of operations required in *chip update* is $(b + 2)2^b + 4b + 2$.

Branch update is performed after all branch metrics are available, i.e., N_{km} has been determined for all m's of the k-th branch. Given N_{km}, branch update is identical to that of the conventional Viterbi algorithm, where an ACS is executed for every state. The branch metric $\mu_k(\mathbf{c})$ in (3.35) can be obtained by the difference between $W_{k,0}(\mathbf{c})$ and $W_{k,N}(\mathbf{c})$, which requires b additions. An ACS requires 2 additions and 1 compare-select (a combined single operation), therefore there are a total of $2^{K-1} \cdot 3 + b$ operations for *branch update*, where K denotes the constraint length of the code. The big-O complexity representation for OSLA-VA is $O(k(\overline{N}b2^b + 2^{K-1}))$, where the first term is for *chip update* and the second term is from the ACS branch update in a Viterbi decoder.

Since *chip update* happens at a much higher rate than *branch update*, the chip update complexity is more critical for real-time OSLA-VA transmission. Notice that

the number of operations involved in *chip update* does not scale with the number of states of the trellis but it only relates to the number of output bits per branch b, which is usually small. It is possible to store the pre-calculated $\Delta\mathrm{LL}_{km,i}$ in a memory before they are needed to update the metrics once the initial conditions of (3.40) are ready upon the completion of previous branch update. Although the number of operations involved in *branch update* exponentially grows with K, fully parallel ACS computing to update all branch metrics simultaneously with low latency using 2^{K-1} parallel hardware instances in modern digital ICs is certainly feasible [72, 73] for a practical K (e.g., $K = 12$).

3.3.4 Feasibility of Feedback within One-Chip Delay

For a reasonable example configuration (that is used to evaluate OSLA-TBCC's performance in the later section) with $b = 2$, $K = 11$, and $\overline{N} = 20$, the average number of operations per chip is 128.5 according to the analysis in the previous subsection. We argue that this real-time computation complexity is practically feasible in modern digital ICs where a large number of parallel computation units are instantiated (e.g., Xilinx UltraScale XCVU440 FPGA [74] has 2,880 DSP units). An example URLLC application with 1 ms latency and 32-byte packet (after encoding) using the aforementioned configuration has the chip length of $\Delta t = 0.39\mu$s given $\overline{N} = 20$. For a digital IC that can run at $\geq$1 GHz, this chip duration corresponds to $\geq$390 cycles, which is sufficient to perform (on average) 128.5 operations which are mostly computed in parallel using dedicated computing hardware instances.

Besides the computation at the destination, a practical system also needs to consider propagation delay and processing time at the source. An example of a practical feedback mechanism with propagation delay and processing time is illustrated in Figure 3.6. Assume the distance between the source and the destination is 30m, this introduces an one-way propagation delay of 100ns. Also assume a worst-case seri-

alized implementation with a 1GHz processor for the processing at the destination thus 129ns processing time, and assume the processing time at the source (estimating the binary decision) takes 10 cycles, or 10ns. Note that the additional delay is the combination of two-way propagation delay ($2T_{\text{prop}}$) and the processing time at both sides ($T_{\text{proc,d}}$ and $T_{\text{proc,s}}$), which is 339 ns and is less than one chip. By using a shorter feedback signal (T_{fb}, details introduced in the later section), the delay due to feedback is kept as one chip. With this example, we argue that a practical OLSA system with $\leq$ 1-chip feedback latency (Fig. 2) is feasible as assumed throughout this work.

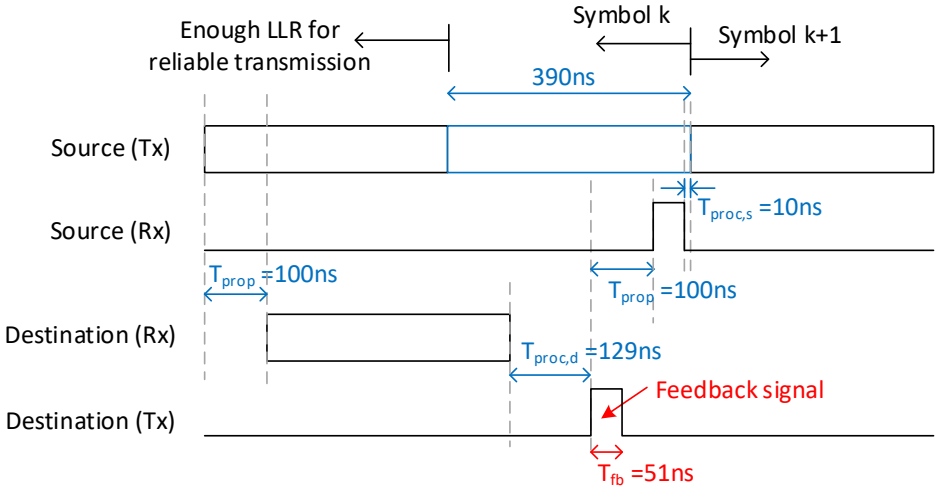


Figure 3.6: A practical OSLA system example timeline to attain $\leq$1-chip feedback delay including propagation and processing delays.

3.4 Feedback Signaling in OSLA

3.4.1 Pulse Feedback Signal

OSLA uses a feedback channel to inform the source when to advance to the next symbol/bit. Synchronizing the source and destination on the chip and symbol indices is the main goal of the feedback signaling. The only information conveyed in the feedback channel is the *timing* of the symbol-advancing decision.

We propose a pulse position/timing based feedback scheme to transmit a pulse when the destination makes a symbol-advancing decision. The feedback channel remains idle (i.e., no transmission) when the destination expects more chips/samples for the same symbol. Thus the feedback signal s_i for chip index i is given by

$$s_i = \begin{cases} \sqrt{P_{\text{fb}}} & \text{if symbol-advancing decision made at } i-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.43}$$

where P_{fb} is the power of the feedback pulse. Note that each forward channel using an orthogonal resource requires a dedicated feedback channel as shown in Figure 3.1 and 3.3.

Under this design, the probability of feedback detection error is

$$P_d = Q\Big(\sqrt{\frac{P_{\text{fb}}}{4\sigma_w^2}}\Big) \tag{3.44}$$

where σ_w^2 is the noise variance in the feedback channel of each chip/sample. Note that a single detection error can destroy the synchronization between the source and destination, causing a catastrophic failure of the transmission. With this pulse based signaling, the feedback error rate will be $1 - \mathbb{E}\{(1-P_d)^{N_{\text{total}}}\}$, where N_{total} is the random number of total chips on the feedback channel for the entire codeword transmission.

Note that this feedback scheme allocates the transmit power only for the time slot where symbol advancing occurs. The downside of this scheme is the relatively wide bandwidth usage which is inversely proportional to the chip duration Δt, not the average symbol duration.

3.4.2 Enhanced Synchronization with HMM

The source uses the received feedback signal to estimate the symbol-advancing decision made by the destination at every chip. However, the aforementioned naive scheme does not consider the fact that each chip has a different probability to advance to the next symbol. For example, the first chip of a symbol is very unlikely to be the last chip advancing to the next symbol. This chip-dependent probability can be taken into consideration to improve the reliability of the feedback estimation.

OSLA transmission can be viewed as a state transition process where each state corresponds to a unique symbol. The system either stays in the same state (symbol) or advances to the next state, whereas the transition probability changes with the chip index (or the elapsed time in a state).

We propose a 2-D state structure for state estimation, where one dimension is the symbol index and the other dimension is the number of chips spent on a symbol (i.e., chip index of a symbol) as illustrated in Figure 3.7. A state denoted as $S_{i,j}$ indicates that the system is transmitting the j-th chip of the i-th symbol. For the next chip, the state $S_{i,j}$ can transition to one of only two possible next states $S_{i+1,1}$ or $S_{i,j+1}$ depending on symbol advancing or not, respectively. We assume the system is a hidden Markov model (HMM), and the transition probability only depends on the current state, not on the past history. This assumption holds for uncoded OSLA-BPSK, but it is not necessarily true for trellis coded OSLA. However, we make this simplifying assumption as it greatly reduces the complexity of the state estimation process at the source, without affecting significantly the practical performance of the proposed scheme.

The source uses a classic forward algorithm [75] for HMM to estimate the likelihood of each state given the received feedback signal z_i and the knowledge of symbol advancing probability $q_{i,j} = Pr(S(t+1) = S_{i+1,1} | S(t) = S_{i,j})$, where t is the transmission time index in chip units. For each chip, the forward algorithm uses current

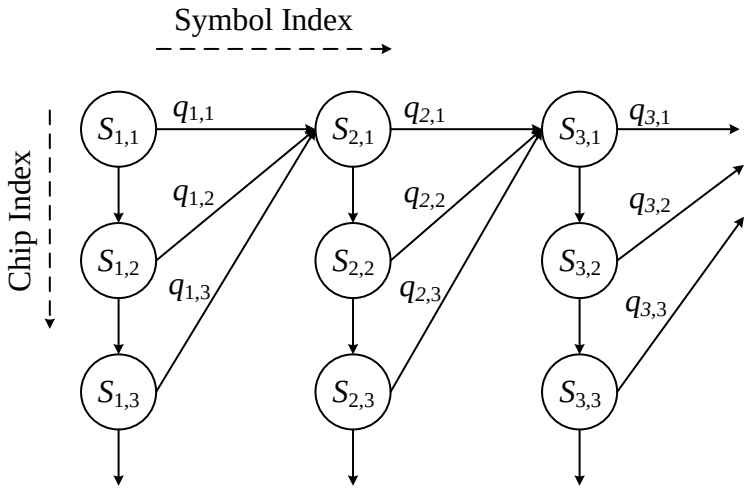


Figure 3.7: 2-D state transition model for enhanced synchronization with HMM

beliefs of all states at time t denoted as $\alpha_t(S)$ as well as the received signal z_{t+1} to update the belief of each state based on the following three steps:

1. Calculate the sum of probabilities of transitions into state $S_{i,j}$ by

$$F(S_{i,j}) = \begin{cases} \sum_{k=1}^{\infty} q_{i-1,k} \cdot \alpha_t(S_{i-1,k}) & j = 1 \\ (1 - q_{i,j-1}) \cdot \alpha_t(S_{i,j-1}) & j \neq 1. \end{cases} \tag{3.45}$$

2. Calculate the emission probability of state $S_{i,j}$ by

$$Pr(z_{t+1}|S_{i,j}) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp(-\frac{(z_{t+1}-\sqrt{P_{\text{fb}}})^2}{2\sigma_w^2}) & j = 1 \\ \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp(-\frac{z_{t+1}^2}{2\sigma_w^2}) & j \neq 1. \end{cases} \tag{3.46}$$

3. Update the belief for each state by

$$\alpha_{t+1}(S_{i,j}) = F(S_{i,j}) \cdot Pr(z_{t+1}|S_{i,j}). \tag{3.47}$$

The initial beliefs are set as $\alpha_1(S_{1,1}) = 1$ and $\alpha_1(S_{i,j}) = 0$ for all other states. For

each chip, the source updates the belief of each state, and selects the maximum one as the most likely state to transmit the corresponding chip/symbol indicated by that state. Note that synchronization between the source and destination is maintained without performance degradation as long as the symbol index i is correct even if the estimated state $S_{i,j}$ is not (i.e., chip index j is incorrect). Therefore, this scheme is robust to temporary chip index mismatches between the source and destination.

The number of states with non-zero beliefs in the HMM forward algorithm grows with the time index t to $1+\frac{t(t-1)}{2}$. This can potentially cause computation complexity issues for a relatively long packet. However, it is observed that the number of states with non-negligible probability (e.g., > 0.0001) stays very small for a feedback channel that has sufficiently high SNR. Thus, we can prune the vast majority of states to limit the number of states to evaluate.

The forward algorithm assumes that the transition probabilities for each state pair, $q_{i,j}$, are known. While it is possible to obtain them analytically for uncoded OSLA-BPSK, the analysis is difficult for trellis coded OSLA. Thus, in this work, we empirically obtain these probabilities via Monte Carlo simulations. Note that these probabilities depend only on the forward channel SNR but not the feedback channel SNR.

3.4.3 Trade-off between asynchronous and synchronous advancing schemes

In Section III-A, we introduced an asynchronous symbol/bit-advancing scheme where b coded bits that determine one state transition in the Viterbi trellis are transmitted using b dedicated orthogonal channels so that each can be advanced to the next symbol asynchronously. Such an asynchronous scheme outperforms the synchronous version (where b-bits advance at the same time to be synchronized with Viterbi trellis state transition) when the perfect feedback reliability is assumed. For a realistic feedback channel with a finite SNR, however, there is a potential advantage to use a

synchronous symbol-advancing scheme for more reliable feedback.

Since all the channels advance symbols at the same time in the synchronous scheme, only one advancing schedule (and thus one HMM) needs to be maintained. Therefore, feedback signaling power can be concentrated into a single feedback channel (as opposed to splitting the power into b separate feedback channels for asynchronous feedback for each), improving the SNR and reliability of the feedback signal. Moreover, the bandwidth of the feedback signal is also reduced thanks to less number of feedback channels. The synchronous advancing scheme saves b times power and bandwidth in the feedback channel for the same feedback robustness.

This implies that there exists a trade-off between the asynchronous and synchronous symbol-advancing schemes. The former has better forward channel reliability, whereas the latter enables more reliable feedback given the same feedback SNR. It is worth pointing out that the limiting factor on error probability can be clearly identified as either forward or feedback SNR. The error rate is limited by the largest of forward error rate (with noiseless feedback) and synchronization error rate (due to noisy feedback). Trade-off between synchronous and asynchronous schemes is easy to resolve. As evaluated in Section V, one can select a better scheme depending on the forward and feedback channel SNR.

3.5 OSLA with a hard deadline latency constraint

In the previous sections, we consider variable symbol length without any time constraint, and hence the block (or packet) length is also variable. Due to the law of large numbers, for a relative long data packet, the transmission time is expected to concentrate around the mean value. However, for short data packets, this will not be the case. Moreover, in some time-sensitive applications, the timing constraint is designed such that a packet is consider invalid if it is delivered after a deadline, while the extra time between the finish of the transmission and the deadline is not

rewarded. Typically, variable length codes are not suitable for this case. However, we demonstrate that OSLA can still provide gain while satisfying the constraints by dynamically allocate resources (e.g., time budget) in order to complete transmission within a given deadline.

3.5.1 Markov Decision Process for OSLA-BPSK

We consider the problem of transmitting K uncoded bits over a given time interval T_{tot} and with average power P. At the beginning of bit k's transmission, the source knows the remaining time T_k and sets a log-likelihood threshold $L_k = g_k(T_{1:k}, L_{1:k-1})$ that will determine the expected error probability for this bit. We seek optimal policies $g = (g_1, g_2, \cdots, g_{K-1})$ that will minimize relevant metrics, J^g. We consider the average bit error probability as the metric to be minimized.

$$J^g = \mathbb{E}\{\sum_{k=1}^{K-1} P_e(T_k, L_k) + P_e(T_K)\}, \tag{3.48}$$

where the quantity $P_e(T_k, L_k)$ is the error probability of the k-th bit when we set the likelihood threshold to L_k and there is T_k time remaining, and is given by

$$P_e(T_k, L_k) = \mathbb{E}\{P_e|T_k, L_k\} = \frac{1}{1+e^{L_k}} F_T(T_k; L_k) + Q(\sqrt{\frac{2PT_k}{N_0}})(1 - F_T(T_k; L_k)) \tag{3.49}$$

for $T_k, L_k \geq 0$, where F_T is the cdf of the random bit duration T (parameterized by threshold L_k, and depending on P/N_0). The intuition behind the above expression is the following. If the random transmission time, T, of bit k is less than the remaining time, T_k, then the error probability is guaranteed by the preselected log-likelihood level L_k according to the previous analysis in section 3.2. Else, if T is greater than T_n, then the transmission will stop at T_k and the final log-likelihood for bit k is a Gaussian r.v. with distribution $N(\frac{4PT_k}{N_0}, \frac{8PT_k}{N_0})$ and the error probability is given by

the appropriate Q function. Similarly, for the last bit K there is no decision to be made and the remaining time is used for this transmission, resulting in $P_e(T_K) = Q(\sqrt{\frac{2PT_K}{N_0}})$.

One can easily identify this problem as an $(K-1)$-horizon Markov decision process (MDP) with state T_k, action L_k, instantaneous costs $P_e(T_k, L_k)$ and terminal cost at time K (last bit) equal to $P_e(T_K)$. This implies that optimal strategies as Markovian, i.e., of the form $L_k = g_k(T_k)$ can be found through dynamic programming (DP) as follows:

$$V_K(T_K) = Q(\sqrt{\frac{2PT_K}{N_0}}), T_K \geq 0 \tag{3.50}$$

$$V_k(T_k) = \min_{L_k} P_e(T_k, L_k) + \mathbb{E}\{V_{k+1}(\max(T_k - T, 0))|T_k, L_k\}, T_k \geq 0, k = K-1, \cdots, 1 \tag{3.51}$$

The policies (i.e., functions $g_1, \cdots, g_K$) can be obtained offline, while during the transmission the threshold L_k for the next bit is calculated on the fly given the knowledge of remaining time T_k.

The idea can be extended to minimize the block error rate (BLER) of an uncoded packet with modified state (T_k, M_{k-1}), where M_{k-1} is a binary value (0 or 1) representing the error of the subblock that contains all the bits before k-th bit. Following the similar strategy, we have

$$V_K(T_K, M_{K-1}) = \begin{cases} Q(\sqrt{\frac{2PT_K}{N_0}}) & , M_{K-1} = 0, T_K \geq 0 \\ 1 & , M_{K-1} = 1 \end{cases}$$

$$V_k(T_k, M_{k-1}) = \begin{cases} \min\limits_{L_k} \mathbb{E}\Big\{V_{k+1}\big(\max(T_k - T, 0), M_k\big)|T_k, M_{k-1}, L_k\Big\} & , M_{k-1} = 0, T_k \geq 0 \\ 1 & , M_{k-1} = 1 \end{cases} \tag{3.52}$$

where the expectation is taken over both T_k and M_k. The state transition probability can be obatined by

$$\Pr(T_{k+1}, M_k | T_k, M_{k-1}, L_k) \tag{3.53}$$

$$= \Pr(T_{k+1} | T_k, M_{k-1}, L_k) \cdot \Pr(M_k | T_k, M_{k-1}, L_k, T_{k+1}) \tag{3.54}$$

$$= \Pr(T_{k+1} | T_k, L_k) \cdot \Pr(M_k | T_k, M_{k-1}, L_k, T_{k+1}) \tag{3.55}$$

where the first term can be easily obtained given the pdf f_T (which depends on L). The second term can be obtained by

$$\Pr(M_k = 1 | T_k, M_{k-1} = 1, L_k, T_{k+1}) = 1 \tag{3.56}$$

$$\Pr(M_k = 1 | T_k, M_{k-1} = 0, L_k, T_{k+1}) = \begin{cases} \frac{1}{e^{L_k}+1}, T_{k+1} > 0 \\ Q(\sqrt{\frac{2PT_k}{N_0}}), T_{k+1} = 0 \end{cases} \tag{3.57}$$

and $\Pr(M_k = 0 | T_k, M_{k-1}, L_k, T_{k+1}) = 1 - \Pr(M_k = 1 | T_k, M_{k-1}, L_k, T_{k+1})$ since M_k is binary.

The policies $g_k(T_k, M_{k-1})$ again can be obtained offline. During the transmission, the source always uses the policy $L_k = g_k(T_k, M_{k-1} = 0)$ given T_k and assumes there is no error before. This is because if $M_{k-1} = 1$, the block will always result in error while any policy makes no difference. Therefore, we adopt the best effort method, hoping we can at least optimize for the best case scenario.

3.5.2 Reinforcement Learning Policy

Although the optimal policy for uncoded OSLA-BPSK with a hard deadline latency constraint can be attained following standard MDP, it is hard to obtain the optimal policy for trellis coded OSLA following the same strategy. The dimension of the state in the MDP can be huge due to large constraint length of the code.

Moreover, the state transition probability given the action taken cannot is almost impossible to track due to the enormous state space. Therefore, to find a good (but may not be optimal in terms of BLER) policy for trellis coded OSLA, we propose to use reinforcement learning (RL) as an alternative solution.

A RL problem is formulated by defining the state, action, and reward (or cost) function. We aim at finding the optimal policy that maximizes (or minimizes) the average sum of reward (or cost) given an uncontrolled environment. The policy takes an action (we only consider deterministic action in this work) given the state, and the environment will update the state based on the current state and action, and return an immediate reward (or cost).

In the uncoded OSLA-BPSK case for minimizing average BER, it is easy to define the state as T_k, the action as L_k, and the cost as the BER, which is calculated by $1/(e^L+1)$ given the amplitude of the resulting LLR after the bit transmission, denoted as L. Since we already have the optimal policy by MDP, we can use that to verify the idea of using RL to solve the problem, and later extend it to the trellis coded OSLA.

We propose to use deep Q-learning with normalized advantage function (NAF) [76]. A Q function[¶] $Q(s, a)$ in RL is defined as the expected rewards for an action a taken in a given state s. In this problem, the state is the remaining time T_{rem}, and the action is the threshold L for the next bit to transmit. However, it is convenient to consider the number of remaining bit b_{rem} as part of the state so that only one policy is needed. We model the Q function as

$$Q(s, L) = \sigma(-\frac{1}{2}P(\mathbf{s})(L - \mu(\mathbf{s}))^2 + V(\mathbf{s})), \tag{3.58}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, and $\mathbf{s} = (T_{\text{rem}}, b_{\text{rem}})$ is the state. $P(\mathbf{s}), \mu(\mathbf{s})$ and $V(\mathbf{s})$ are neural networks that may contain multiple layers. Instead of directly using a quadratic function as suggested in [76], we add another layer of

[¶]This Q function is not the Gaussian Q-function.

sigmoid function to make it more related to probability, which is our optimization goal. The quadratic function indicates that there exists an optimal L at a sweet spot because we know that too small or too large L will either harm the current bit or later bits, both resulting in bad performance.

We follow classic Q-learning strategy to minimize the loss function given the immediate cost C and the next state $\mathbf{s}'$. The immediate cost is obtained after a symbol transmission by $C = 1/(e^{\tilde{L}} + 1)$, where $\tilde{L}$ is the amplitude of LLR when the symbol is terminated ($\tilde{L}$ may be greater or less than L due to time expiration or overshooting in discrete time OSLA). We use double Q-learning [77] where we fix a target network Q^{target} and iteratively update the policy network Q^{policy} to reduce variance. The target network Q^{target} is updated once a while by $Q^{\text{target}} = Q^{\text{policy}}$. During the training process, we run the simulation by choosing L according to the current policy Q^{policy} given $\mathbf{s}$, and the next state $\mathbf{s}'$ and cost C from the enviroment (i.e., a symbol transmission by OSLA). We then collect the data pairs $(\mathbf{s}, L, \mathbf{s}', C)$, which are used to optimize the parameters in the network with respect to a loss function. The loss function is defined as

$$\text{Loss} = \frac{1}{\sqrt{Q^{\text{target}}(\mathbf{s}, L)} + \epsilon} BCE(Q^{\text{policy}}(\mathbf{s}, L), C + \sigma(V(\mathbf{s}')) \tag{3.59}$$

where BCE is the binary cross entropy function, and ϵ is a small number to prevent numerical issue if $Q^{\text{target}}(\mathbf{s}, L) = 0$.

To extend RL solution to trellis coded case, we need to define proper state $\mathbf{s}$ and cost C by taking the trellis structure into consideration. The onjective function is obviously the BLER, which should be defined as the expected value of the sum of the costs. One easy way is to design the costs by having the immediate cost as zero and the terminal cost as the block error. However, this results in tranining inefficency because zero immediate cost makes it hard to improve the policy in early steps.

Instead, we propose to formulate a RL problem with a non-zero immediate cost. Our proposal is inspired by the observation that given the received symbol at each branch, the reliability of the surviving path can be calculated iteratively by reliability output Viterbi algorithm (ROVA) [78].

The probability of the event that the survivor path at branch k terminating at trellis state s with incoming state s' is correct is given by

$$\Pr(P_k^s) = \frac{1}{\Delta} e^{\mu_k(\mathbf{c}^{s'\to s})} \Pr(P_{k-1}^{s'}) \tag{3.60}$$

and the probability of the event that one of the *nonsurviving* path terminating at node s (i.e., being discarded at branch k) is correct is given by

$$\Pr(\overline{P}_k^s) = \frac{1}{\Delta}\Big[e^{\mu_k(\mathbf{c}^{s'\to s})} \Pr(\overline{P}_{k-1}^{s'}) + \sum_{q\neq s'} e^{\mu_k(\mathbf{c}^{q\to s})} [\Pr(P_{k-1}^q) + \Pr(\overline{P}_{k-1}^q)]\Big] \tag{3.61}$$

where

$$\Delta = \sum_{(s'\to s)\in\mathcal{T}} e^{\mu_k(\mathbf{c}^{s'\to s})} \times [\Pr(P_{k-1}^{s'}) + \Pr(\overline{P}_{k-1}^{s'})]. \tag{3.62}$$

Recall that $\mu_k(\mathbf{c}^{s'\to s})$ is the branch metric defined in (3.36), which is the log likelihood of state transition $(s' \to s)$ given the received symbol.

The probability of the correct trellis path being discarded at k-th branch, denoted as P_d^k, can then be obtained by summing all states

$$P_k^{\text{discard}} = \sum_s \Pr(\overline{P}_k^s). \tag{3.63}$$

The immediate cost after k-th branch transmission is defined as $C_k = P_k^{\text{discard}} - P_{k-1}^{\text{discard}}$, with initial condition $P_0^{\text{discard}} = 0$. Note that the cost can be negative, unlike the definition in uncoded case. To accommodate the negative cost, we replace

the sigmoid function in (3.59) by tanh. The terminal cost after the whole block is transmitted is defined as

$$C_T = \sum_{s \neq s_*} \Pr(P_K^s), \qquad s_* = \underset{s}{\operatorname{argmax}}\, M_K(s). \tag{3.64}$$

These definitions are made by noting that the sum of the immediate costs and the terminal cost equals to the BLER if the trellis path with the largest metric $M_K(s)$ at the end is selected as the decoded output.

The state definition in trellis RL problem may be defined as the latest reliability outputs for all trellis states, in addition to the remaining time T_{rem} and the number of remaining branches b_{rem}.

3.6 Evaluation

We evaluate the performance of OSLA with Monte Carlo simulations. In all simulations, the average symbol length of OSLA is controlled by setting a proper threshold L for symbol advancing criteria evaluation to match the symbol length of a fixed-length scheme. For a fair error rate comparison, all schemes are evaluated with the same E_b/N_0 and spectral efficiency. Rectangular pulse-shaping is assumed for both schemes. Chip duration Δt in OSLA is set to be $1/(10b)$ of the average symbol length for a rate $1/b$ coding ($b = 1$ for the uncoded cas

3.6.1 OSLA-BPSK

Figure 3.8 shows the BER performance of uncoded OSLA-BPSK for various $\Delta t/\overline{T}_{\text{sym}}$ settings. The system approaches the continuous model when $\Delta t/\overline{T}_{\text{sym}}$ is smaller, and the simulation result is well aligned with the analysis (3.13). Performance degradation occurs when $\Delta t/\overline{T}_{\text{sym}}$ is larger with longer feedback delay of Δt that results in wasted transmit energy or symbol length. The performance gap be-

tween OSLA-BPSK and fixed-length BPSK (both uncoded) is approximately 6 dB at high SNR as expected from the analysis (3.13).

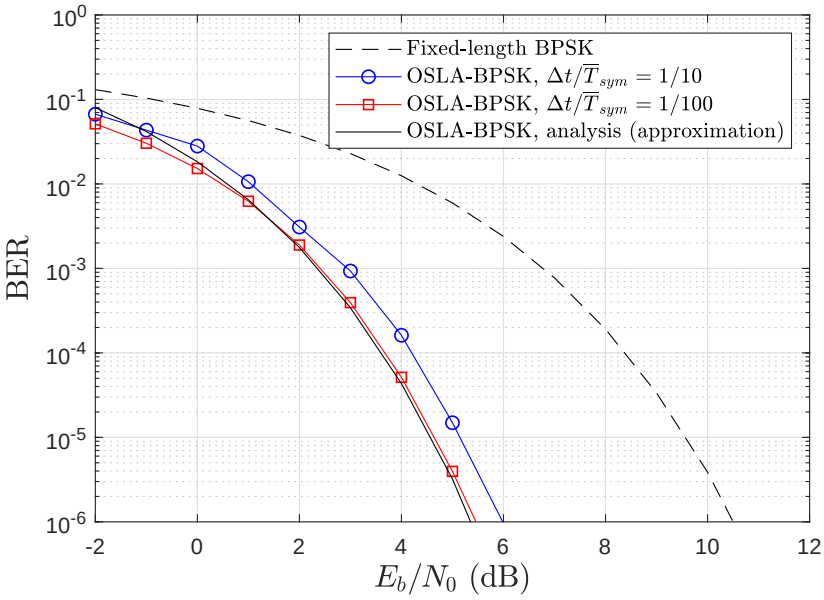


Figure 3.8: OSLA-BPSK BER performance and analysis

Figure 3.9 plots the distribution of $N = T_{\text{sym}}/\Delta t$, the number of chips per symbol, of OSLA-BPSK for different SNR scenarios. The expected number of chips per symbol $\overline{N} = \mathbb{E}\{N\}$ is set (by controlling L) to 10 in the simulation. As the figure shows, the analysis (3.18) matches the simulation results very well. Notice the distribution of N is dependent on the SNR for a given $\overline{N}$ (= 10 in Fig. 3.9). For a higher SNR, the variance of N is smaller, and its distribution can be approximated by a Gaussian distribution.

Next, to justify our claim of fair comparison with other fixed-symbol-length schemes, we numerically evaluate the spectrum of uncoded OSLA signal. We first evaluate $R(t, \tau)$ using the analytical results in (3.30), and compare to the simulation results. Note that the pdf in (3.18) depends on $\gamma = P/N_0$ and L. Since the analysis aims at providing a spectrum comparison with the fixed-symbol-length scheme, here we always choose a L such that $\mathbb{E}\{T\} = 1$ for all γ. In this case, $E_b/N_0 = P \cdot \mathbb{E}\{T\}/N_0 = \gamma$.

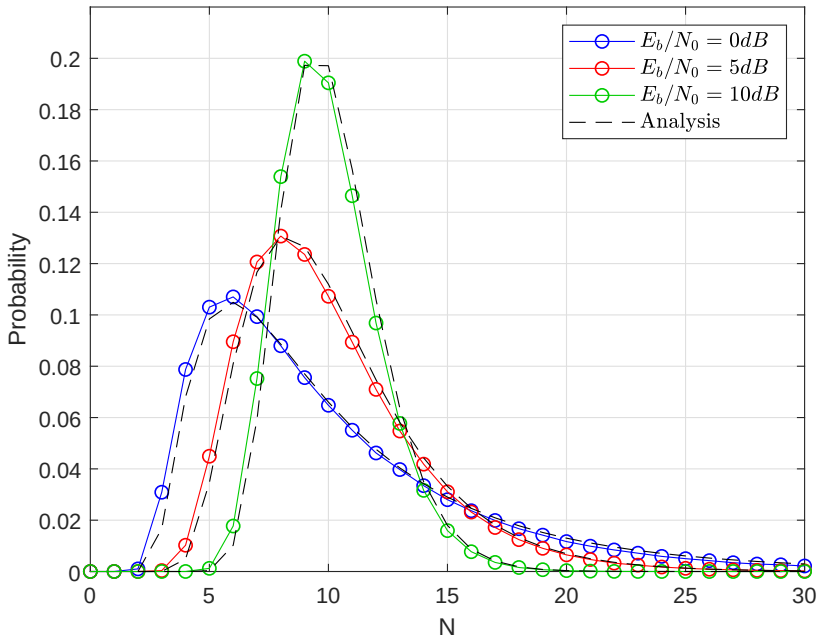


Figure 3.9: Distribution of $N = T_{\text{sym}}/\Delta t$ in OSLA-BPSK

In the following, we show numerical results of the analysis and the simulation by generating T_k with pdf f_T in (3.17). We use $K = 100$ to simulate a relatively long sequence.

In Figure 3.10, it is observed that for a fixed τ, the value of $R(t, \tau)$ depends on t, but converges to a fixed value as $t \to \infty$.

Figure 3.11 shows the analysis from (3.34) and simulation results of the autocorrelation function at different $\gamma = E_b/N_0$. It is observed that for high SNR, the shape of $R(\tau)$ approachs a fixed-length scheme. This is expected since at large γ, the pdf in (3.17) is more concentrated around $\mathbb{E}\{T\}$, and the r.v. T_k approaches a constant $T = 1$.

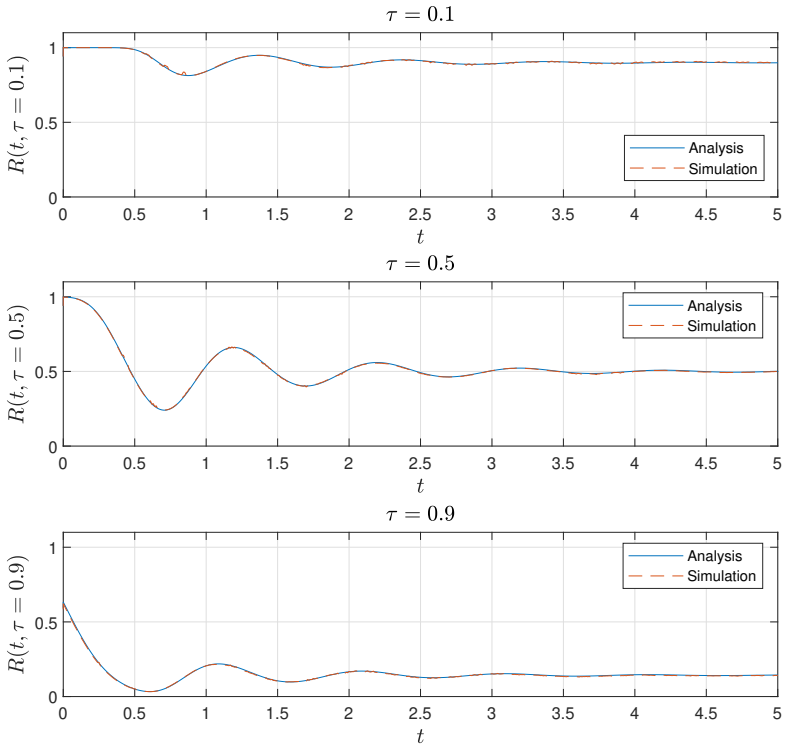


Figure 3.10: $R(t, \tau)$ analysis and simulation.

Finally, Figure 3.12 shows the numerically evaluated PSD $S(f)$, which is the Fourier transform of $R(\tau)$. From the figure, it is observed that the shape of spectrum changes with $\gamma = E_b/N_0$. 3GPP defines the occupied bandwidth as the bandwidth containing 99% of the total integrated power of the transmitted spectrum [79]. Under this definition, the occupied bandwidth of OSLA with three different settings (E_b/N_0 =5dB, 10dB, 15dB), and fixed-symbol-length BPSK is 9.65, 9.55, 9.56, and 9.42 (for $\mathbb{E}\{T\} = 1$). Therefore, when a proper L is chosen that $\mathbb{E}\{T\}$ is kept the same as a fixed-symbol-length scheme, there is no obvious spectral efficiency change. From this observation, we claim that the error rate evaluation in [65] is a fair comparison between the variable/fixed-symbol-length schemes with same E_b/N_0 and $\mathbb{E}\{T\}$.

Compared to a fixed-length scheme, it is observed in Fig. 3.13 that OSLA occupies similar cumulative power profile and bandwidth. Note that in practical systems, non-

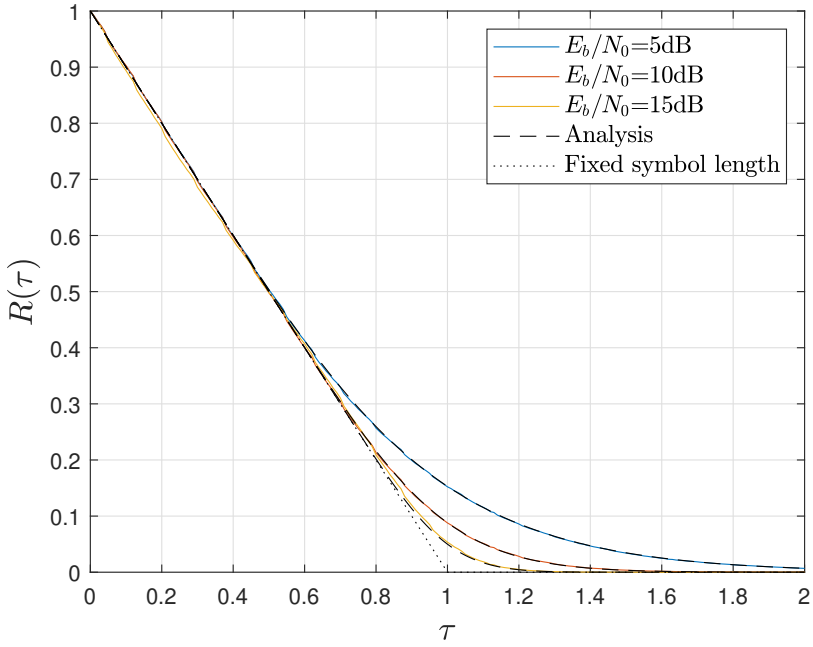


Figure 3.11: $R(\tau)$ of OSLA signal at various E_b/N_0.

rectangular (e.g., root-raised-cosine) pulse shaping can be applied to a fixed-length scheme for reducing the occupied bandwidth. Similarly, head/tail ramping up/down can be applied to OSLA, but it is non-trivial and is left as a future work.

3.6.2 Trellis Coded OSLA

Figure 3.14 shows the block error rate (BLER) performance evaluation of a rate-1/2 (128,64) OSLA-TBCC compared to state-of-the-art non-feedback short codes. BLER performance a rate-1/3 (150,50) OSLA-TBCC compared to other feedback-based schemes is shown in Figure 3.15. All schemes that have the identical coding rate also have the same spectral efficiency as shown in Figure 3.13. The constraint length of TBCC is set to 11 and Δt in OSLA is set to $1/(10b)$ of the average symbol length for the coding rate of $1/b$. Smaller Δt is avoided for faster simulation although it would enhance the error performance. Noiseless feedback channel is assumed in these plots.

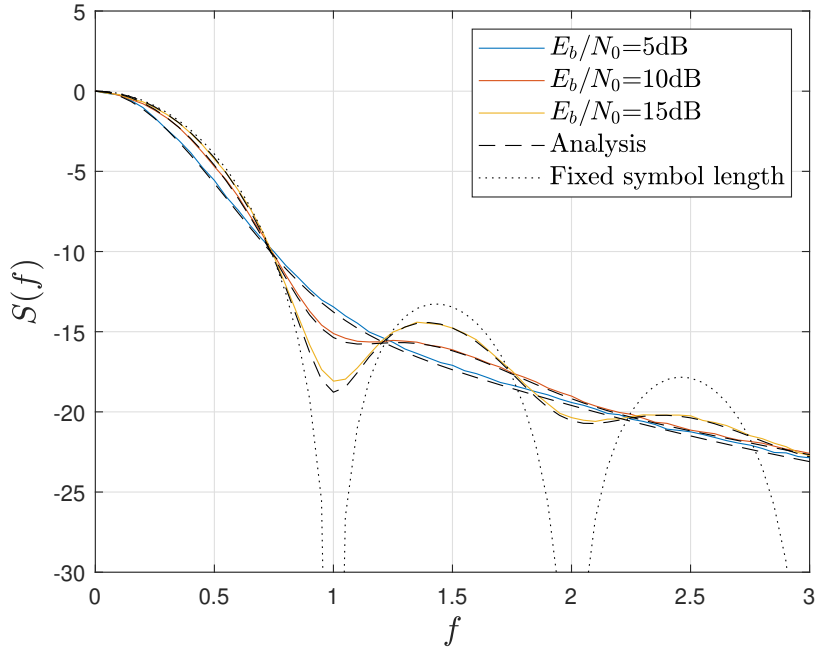


Figure 3.12: $S(f)$ of OSLA signal at various E_b/N_0

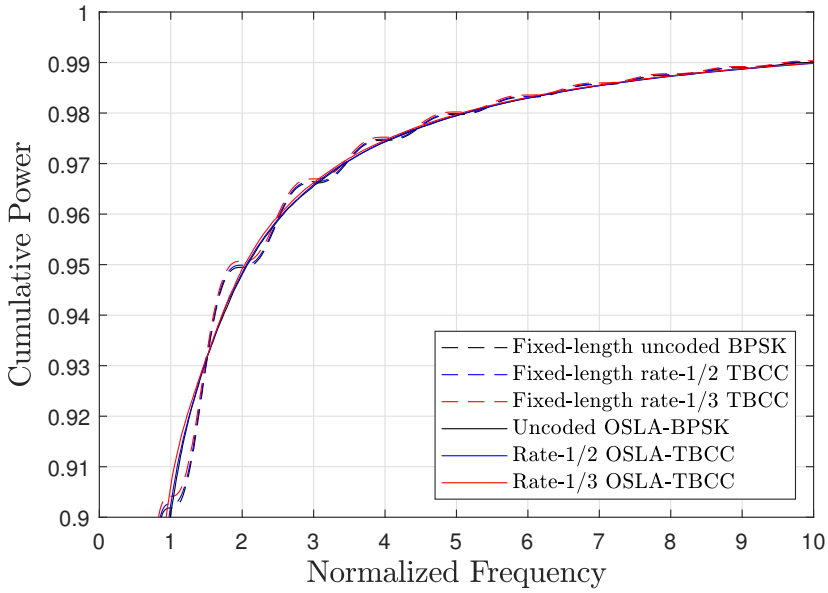


Figure 3.13: Occupied bandwidth

It is observed that (128,64)-OSLA-TBCC significantly outperforms state-of-the-art non-feedback polar, TBCC and BCH codes [9] by about 1.5dB. The normal approximation [8] of an (128,64) non-feedback code in binary input AWGN channel is

also shown. OSLA-TBCC can surpass the normal approximation curve thanks to the utilization of feedback.

Figure 3.15 shows that (150,50)-OSLA-TBCC can outperform Deepcode [60], a state-of-the-art deep learning-based feedback scheme that has the same spectral efficiency, especially in the high SNR region. The BLER of Schalkwijk-Kailath (SK) scheme [80] is also shown in the same figure. Although SK scheme is better than OSLA-TBCC and closer to Shannon limit in a noiseless feedback channel, it is practically infeasible because of its noiseless feedback assumption and extreme sensitivity to the numerical precision. On the contrary, OSLA-TBCC and Deepcode are more practical schemes as both can tolerate noisy feedback and do not suffer from the arithmetic imprecision issue.

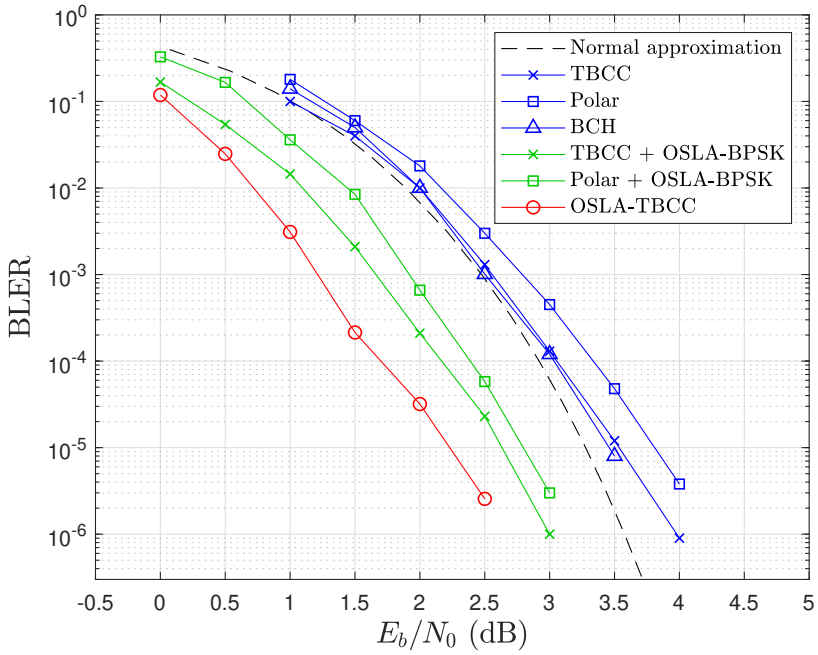


Figure 3.14: OSLA-TBCC BLER performance comparison with non-feedback schemes

Figure 3.16 shows the BER performance of OSLA-turbo with different codeword lengths under noiseless feedback. The turbo code settings in the simulation follow the LTE standard [71], and the number of decoder iteration cycles is set to 5. OSLA-turbo

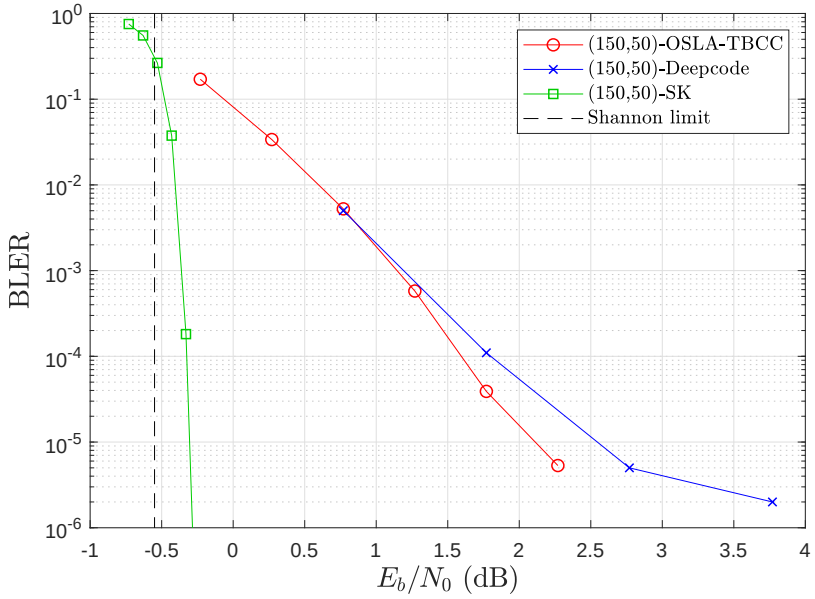


Figure 3.15: OSLA-TBCC BLER performance comparison with feedback-based schemes

outperforms fixed-length non-feedback turbo codes with 0.5∼0.7 dB. The gap between OSLA and fixed-length turbo codes slightly decreases with longer codeword lengths, showing that the OSLA feedback scheme is more advantageous for shorter codeword lengths. This behavior is expected as turbo coding is asymptotically capacity achieving. OSLA-turbo outperforms Deepcode for BER $< 10^{-4}$ for 500 information bits (other longer codeword settings are not available in [60]). One main drawback of Deepcode is its limited scalability. Unlike OSLA, its BER slope remains almost the same regardless of the codeword length. Moreover, it needs a different neural network model for each particular codeword length and rate setting. To achieve satisfactory performance for long codewords, the authors of [60] propose to use an outer turbo code with an inner Deepcode. However, it inevitably lowers the coding rate (1/9 in [60]). On the contrary, OSLA can easily scale the codeword length without changing any code structure.

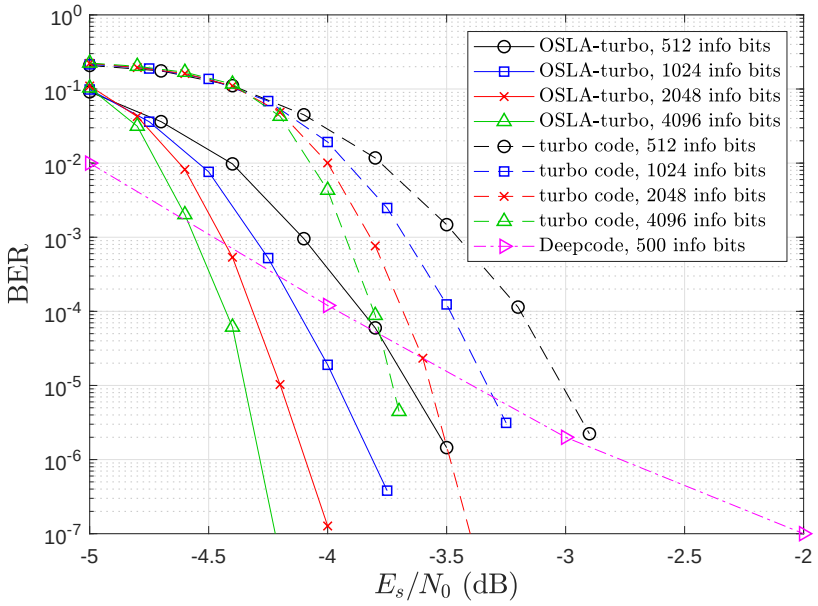


Figure 3.16: OSLA-turbo BER performance comparison

3.6.3 OSLA with Noisy Feedback

Figure 3.17 shows the BER performance of OSLA-BPSK with noisy feedback in different $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$ settings, where $E_b^{(\mathrm{fb})}$ is the feedback energy per forward-channel information bit and $N_0^{(\mathrm{fb})}$ is the noise power spectral density of the feedback channel. The forward channel E_b/N_0 is set to 3dB. With the proposed HMM-based synchronization, the required $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$ for feedback is relaxed by about 1dB compared to a naive scheme without an HMM for the same (forward channel) BER performance. The figure also shows that both $\overline{N}$ (expected number of chips per symbol) and packet length affect the required $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$. For the naive feedback scheme without an HMM, the total number of chips $\overline{N}_{\mathrm{total}} = \overline{N} \cdot$(number of information bits) of a packet governs the feedback robustness, whereas in the HMM-based feedback scheme, the number of information bits plays a more important role than $\overline{N}$. Note that BER loss is negligible when the feedback channel SNR is sufficiently high ($E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})} \geq 17$ dB) as the synchronization error probability is substantially lower than the forward channel BER. It is also observed that the number of states with non-negligible prob-

abilities (< 0.0001) in the HMM is always less than 3 at 17 dB SNR. In the high feedback SNR regime, the attainable forward channel BER is lower for a larger $\overline{N}$ (shorter Δt) as shown in Fig. 3.8.

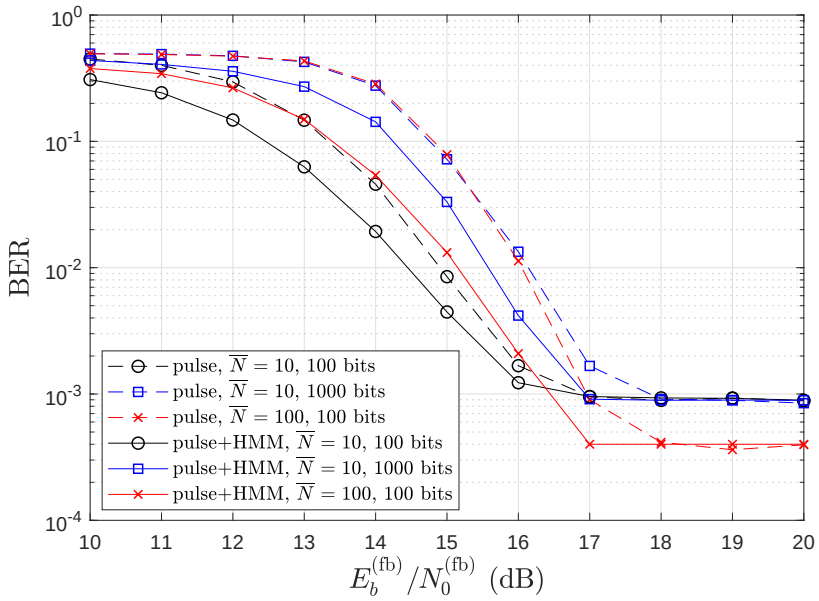


Figure 3.17: OSLA-BPSK with noisy feedback

The BLER performance with error correction coding in noisy feedback channels is shown in Figure 3.18 with respect to $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$ for various forward channel E_b/N_0 settings (1, 1.77 and 3dB). Note that $E_b^{(\mathrm{fb})} = (b/a) \cdot P_{\mathrm{fb}} \Delta t$ holds for the coding rate of a/b. The same $(150, 50)$ setting as in the previous simulation is used for comparison. First, observe that the synchronous advancing scheme for OSLA-TBCC has about 4.7 dB feedback-SNR gain over the asynchronous advancing scheme thanks to the power saving from using only one feedback channel instead of three ($= b$). However, the synchronous scheme has a worse/higher BLER floor compared to the asynchronous advancing scheme for sufficiently high $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$ conditions, exhibiting the trade-off between the forward communication and feedback reliability. OSLA-TBCC, regardless of synchronous and asynchronous advancing schemes, shows more reliable feedback in terms of $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$ compared to Deepcode and Modulo-SK scheme [56],

which is a variant of the SK scheme. Although Modulo-SK can achieve lower BLER for sufficiently high $E_b^{(\mathrm{fb})}/N_0^{(\mathrm{fb})}$, it is not scalable to significantly longer codewords because the required numerical precision for the forward channel grows with the length of the codeword (the length of >50 bits is impractical and difficult to simulate). It also requires a specific setting for each combination of forward and feedback SNRs. On the other hand, OSLA-TBCC does not suffer from the same forward channel numerical precision issue for longer codewords (as shown in Fig. 3.16) and it does not rely on the knowledge of feedback SNR.

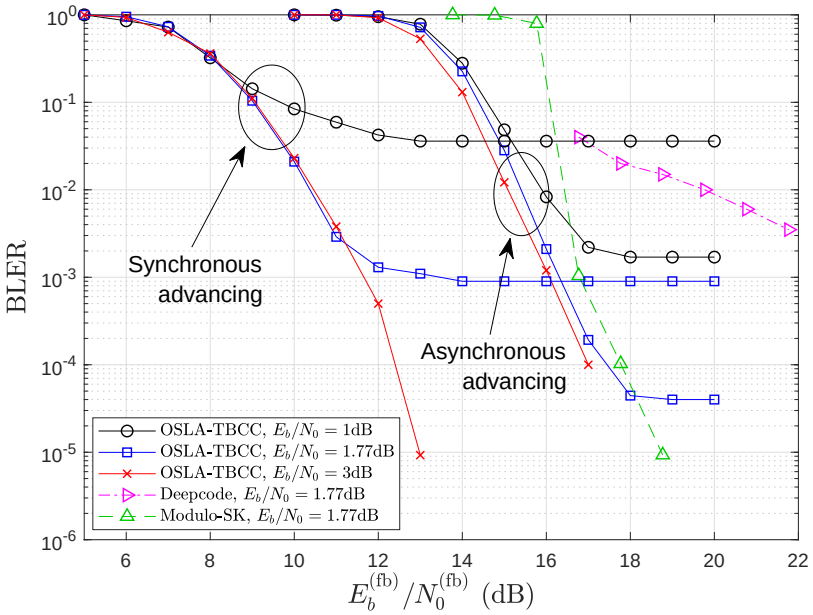


Figure 3.18: OSLA-TBCC with noisy feedback

3.6.4 OSLA with hard latency constraint

In this section we evaluate the performance of OSLA under the constraint of hard deadline latency. Note that the MDP solution as analyzed in Section 3.5 is based on the assumption of continuous time model. When the scheme is evaluated in a discrete time model, there exist a small performance degradation as shown before.

Figure 3.19 shows the BER performance for all bits in a block, where the number

of bits K varies. The scheme adopts threshold decision policy derived from MDP, which is optimal in terms of BER. When E_b/N_0 is fixed regardless of number of bits (i.e., total energy is proportional to K), it is observed that the performance improve with K. At a packet of 20 bits, the BER performance has less than 1 dB gap compared to the unconstrained OSLA-BPSK. This indicates that even when the block length is fixed, by exploiting the gain from feedback with variable symbol length, OSLA still provide close to full gain in uncoded case.

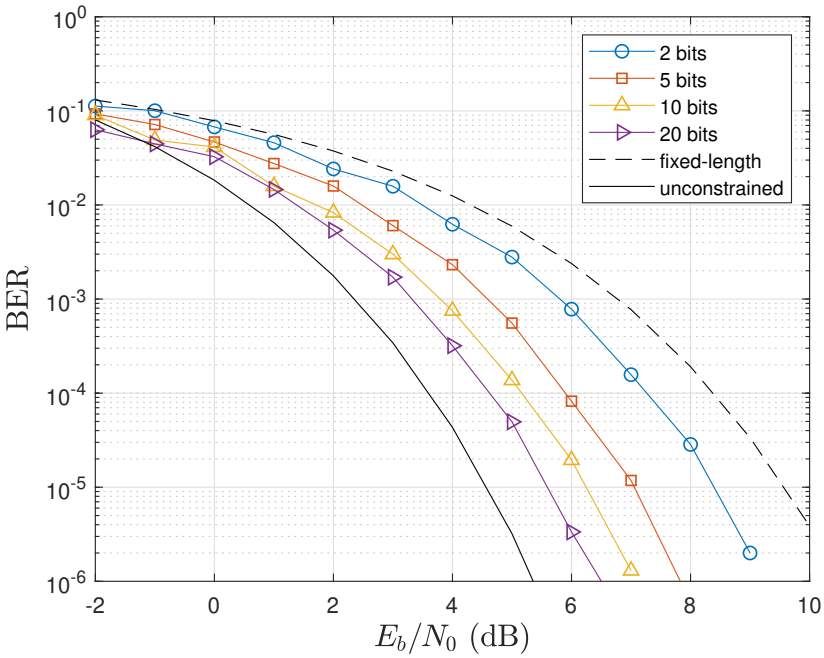


Figure 3.19: BER of limited time budget OSLA. Optimal policy by MDP

Figure 3.20 shows the normalized average length of each bit in a 20-bit uncoded packet for OSLA-BPSK with hard latency constraint using optimal policy by MDP at $E_b/N_0 = 5$ dB. It is observed that the policy is more converse at the beginning (by choosing smaller L_k), then the average symbol length gradually increases. Furthermore, the optimal policy saves plenty of time for the last bit. This is because the last bit error rate is determined by a Q function, whose tail (large BER) can dominate the performance if the remaining time is not large enough.

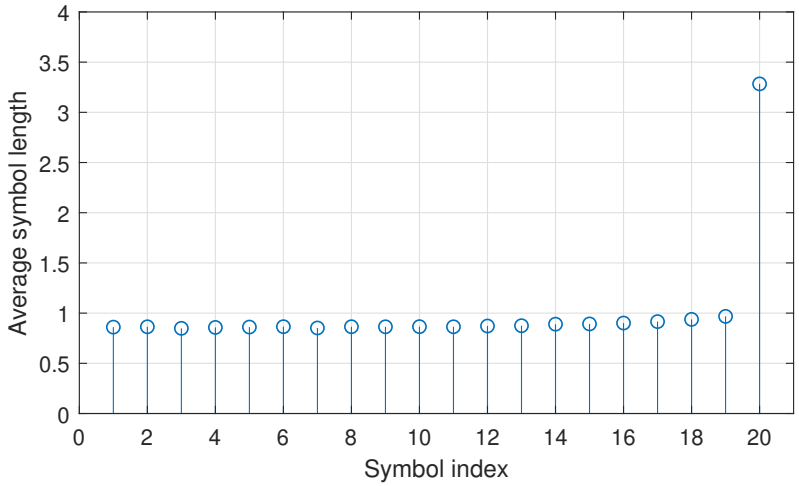


Figure 3.20: Average symbol length in a block

To evaluate the effectiveness of the proposed RL network structure, we use ADAM optimizer with learning rate 0.001 to train the network with 10^6 episodes. Figure 3.21 shows the performance of the system adopting the policy obtained by RL. The performance of the RL policy is fairly close to the optimal policy analyzed by MDP. This implies that RL with DNN is a viable tool to find a close-to-optimal policy in the uncoded case.

3.7 Limitation and Future Directions

TABLE 3.1 summarizes the comparison of different feedback-based schemes in various aspects. Note that OSLA possesses distinctive advantages in blocklength scalability and code rate flexibility. Considering other relative strengths such as relaxed computation precision requirements and robustness of the feedback, our proposed OSLA has a great potential as a practical communication scheme. However, it has certain limitations listed as follows inviting future works.

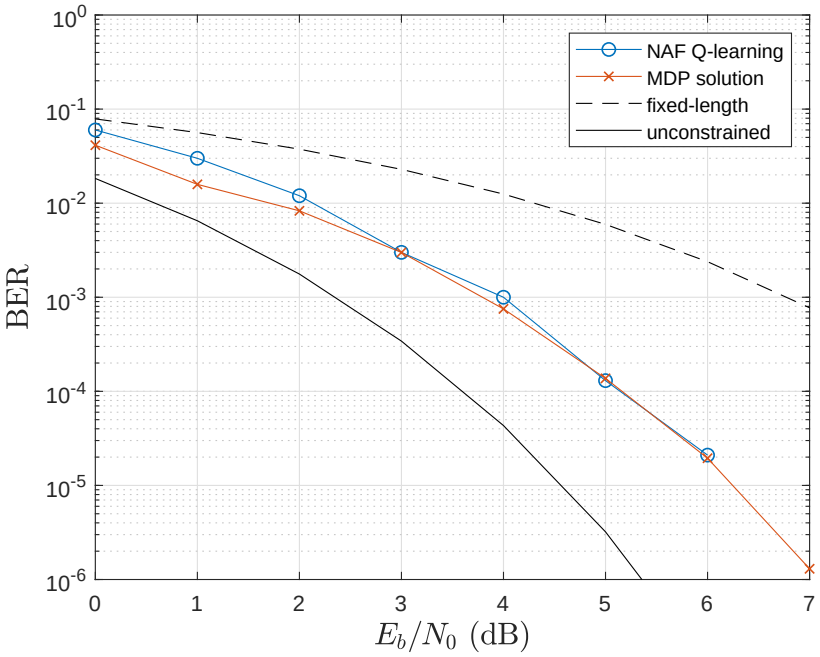


Figure 3.21: BER of uncoded OSLA with NAF Q-learning policy (10 bits)

Scheme	Reliability	Feedback Robustness	Blocklength Scalability	Complexity	Arithmetic Imprecision Tolerance	Flexibility*
Non-Feedback	Medium	N/A	Excellent	Good	Good	Excellent
(Modulo-)SK	Excellent	Good**	Poor	Excellent	Poor	Excellent
Burnashev	Excellent	Poor	Poor	Good	Medium	Good
DeepCode	Good	Good	Medium	Poor	Good	Poor
OSLA	**Good**	**Excellent**	**Excellent**	**Medium**	**Good**	**Excellent**

* Easiness to adjust code rate or SNR configuration.
** When Modulo-SK is used.

Table 3.1: Comparison of schemes on different aspects.

3.7.1 OSLA in Other Channel Models

In this work we only evaluate the performance in memoryless AWGN channels. The concept of OSLA can easily be applied to memoryless discrete channels or erasure channels with trivial modification on metric calculation in (3.2) or (3.36). Extension to fading channels is also straight-forward. Furthermore, OSLA has benefits in fading

channels thanks to its automatic adaptability to noise realization, unlike deep learning based schemes where SNR configuration is a crucial factor when training the code.

On the other hand, this work is based on the assumption of memoryless channels, where the metric can be calculated iteratively with simple addition. Application to channels with memory is non-trivial. However, similar to trellis codes case where the proposed algorithm it used to estimate the maximum likelihood sequence, we may use a similar technique to decode the packet in a channel with memory.

3.7.2 Delayed Feedback

Although this work is based on the assumption of instantaneous feedback where delay is only a single chip, the system does not immediately fail when the delay is longer than one chip. Delay with multiple chips only affects performance by wasting energy on the additional chips in a similar way as larger chip duration. However, one limitation is that the delay still needs to be substantially smaller than the average symbol length in order to exploit the gain of symbol length adaptation.

one possible way to deal with the stringent constraint on delay is to use higher order modulation (e.g., QPSK or QAM) to convey more bits in each symbol. For a fixed time budget to transmit a fix amount of bits, this requires less symbols and hence each symbol duration is larger, which relax the delay requirement. Moreover, a symbol can automatically use longer time with low SNR so that a certain reliability can still be achieved, which relaxes the SNR requirement on the modulation type. However, using higher order modulation is not as efficient as BPSK for low SNR because of the unnecessary energy waste on other bits when the demodulator is struggling on neighbor symbols who only differ in one bit. This is similar to the fact that synchronous advancing scheme is worse than asynchronous scheme in the trellis coded OSLA.

3.7.3 OSLA with Other Coding Schemes

Extension of OSLA to other coding schemes besides trellis codes is of interest but not trivial. For example, decoding algorithms for linear block codes typically require the knowledge of the whole block, which makes it difficult to combine with OSLA like Viterbi algorithm.

Many recently proposed deep learning based feedback codes utilize recurrent neural network (RNN) structure for encoding and decoding. Combination of such codes with OSLA can be a very interest research topic, where the symbol length as well as the coded symbol values are determined based on the knowledge of received symbols at the destination.

3.8 Summary

In this chapter, we propose OSLA, an instantaneous feedback-based transmission scheme that automatically adapts the symbol length based on the noise realization at the receiver via instantaneous feedback to guarantee the target reliability for communication. OSLA can be combined with trellis codes such as turbo and TBCC to boost the performance, providing lower BLER than state-of-the-art short codes including a deep learning-based feedback scheme. Moreover, OSLA can easily scale to longer codeword lengths with consistent gain over fixed-length feedback-less schemes. Using pulse-based feedback signaling and HMM-based state synchronization, OSLA operates reliably in noisy feedback channels. When subjecting to a hard deadline latency constraint for the packet, dynamic threshold decision policies for OSLA, obtained by either MDP or RL, can still provide substantial gain close to the unconstrained one.

CHAPTER IV

Packet Synchronization for Millimeter-Scale Crystal-Less Low Power Wireless Sensor Nodes

4.1 Introduction

With the fast growing business of ubiquitous Internet-of-Things (IoT), new types of applications , such as industrial automation, implantable bio-medical devices and unobtrusive surveillance systems, are being explored. These applications often rely on low-power miniaturized sensor nodes to provide seamless integration into existing technologies. Both industry and academic research have sought to satisfy the form-factor limitation for sensor nodes, trying to scale it down to centimeter, or more aggressively to millimeter scales [81, 82]. Communication is one of the challenging bottleneck, where solutions for these IoT applications need to cover a long (>20 meters) distance while maintaining the ultra-small (mm-scale) form-factor including the antenna.

While there are many existing international standards for IoT connectivity such as Bluetooth Low-Energy (BLE)[83] and ZigBee [84], their relatively high carrier frequency, complexity, and power consumption limit their non-line-of-sight (NLOS) communication distance and/or applicability to mm-scale sensor nodes. Z-wave [85] targets the longer distance indoor environment by using a lower carrier frequency

(sub-GHz). But all of the existing systems impose a stringent specification for the frequency accuracy, timing stability, and quality of the continuous waveform that are unattainable in a mm-scale energy-autonomous wireless node.

We present an energy-autonomous radio system fully integrated within a 3×3×3 mm^3 form-factor operating in the 915MHz ISM band for indoor NLOS communications [86]. The system is constructed from several layers (chips) that are stacked, connected through wire bonding [87] and placed on one side of a miniaturized printed antenna. Our system is crystal-less and solely powered by the energy harvested from a mm-scale PV cell. The complete system has a $3 \times 3 \times 3$ mm^3 form factor including the processor, radio transceiver, baseband controller, antenna, thin-film battery, and the PV cell as shown in Figure 4.1.

In this chapter, we focus on the physical layer communication solution, especially the gateway receiving algorithm for this radio system. The proposed asymmetric system design exploits abundant computation resource at the gateway to assist the transmission with low power sensor nodes, which transmit frequency and phase unstable signal. The carrier frequency and sampling frequency offsets are estimated and compensated with the proposed efficient 2D-FFT algorithm, and a custom design noncoherent sparse M-PPM signal is adopted to deal with unreliable phase and limited energy drawn from the circuit. With the proposed communication protocol that highly relies on the assistance of the gateway, the computation and power consumption at the sensor node can be cut down while still maintaining satisfactory communication distance.

The rest of this chapter is organized as follows. Section 4.2 presents the overview of the designed communication system. Section 4.3 proposes gateway synchronization algorithm for CFO and SFO estimation. Some practical designs for robustness enhancement are discussed in section 4.4. Section 4.5 demonstrate the system with performance quantification. An application using the proposed system for Monarch

butterfly tracking is introduced in section 4.6. Finally, section 4.7 summarizes this chapter.

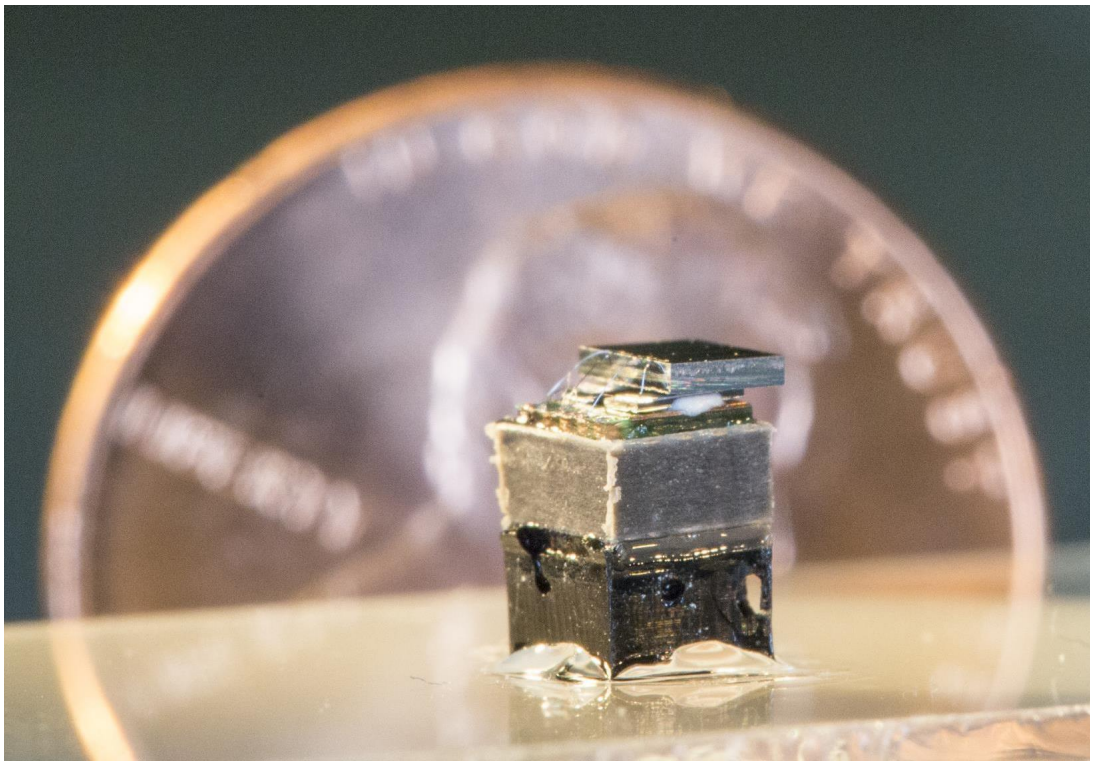


Figure 4.1: The proposed fully integrated system with the processor, radio, PV cell, battery, and printed antenna

4.2 Communication System Design

4.2.1 Sparse M-PPM Modulation Scheme

A major issue of the mm-scale battery is its high internal resistance, which prevents drawing large peak current (mA range) to transmit RF signals. We tackle this issue by powering the transceiver with a trickle charged energy buffer/reservoir capacitor. Instead of pulling current directly from the battery, the transmitter pulls high instantaneous current from the capacitor to generate RF pulses. The battery is continuously recharged through the mm-scale PV cell with variable harvested en-

ergy depending on the ambient light condition, ranging from a few nW in a dimmed indoor room to tens of μW under outdoor sunlight. After each pulse transmission, the battery trickle charges the capacitor to its nominal voltage. Since the recharging time is much longer than the pulse duration, the transmitted pulses are inevitably sparse in time domain.

To accomedate the circuit needs, we propsoed to pulse position modulatio (PPM) to convey information. PPM is a modulation scheme that the information is embedded in the position of the transmitted pulse. It can be regarded as an orthogonal signaling scheme [69], where the symbols when presented in a vector form are mutually orthogonal to each other. Although optimal demodulation for orthogonal signaling scheme (and hence PPM) requires the knowledge of phase of the signal, noncoherent detection can be utilized to relax the phase requirement, therefore is more friendly to phase unstable signal. A gerneral M-PPM is able to carry $\log_2 M$ bits with a pulse transmission, where the position (i.e., timing) is chosen from M time slots according to a bit-to-position mapping. The symbol error probability is given by

$$P_e = \sum_{n=1}^{M-1} \frac{(-1)^{n+1}}{n+1} \binom{M-1}{n} e^{\frac{-n}{n+1}\frac{E_s}{N_0}}. \tag{4.1}$$

For this work, we mostly consider 2-PPM, in which case (4.1) simplifies to

$$P_e = \frac{1}{2} e^{-\frac{E_b}{2N_0}}. \tag{4.2}$$

To further enhance the quality of the communication, coded packet may be used in the transmission. A soft decoder that takes soft input such as log-likelihood ratio (LLR) is required to exploit the full gain of a error correction code. Therefore, one may also obtain the LLR of each bit in 2-PPM scheme by calculating the likelihood of 0 and 1 being transmitted given the received singal amplitude in a vector form $\mathbf{y} = [y_1, y_2]^T$. Note that for noncoherent detection, the likelihood is obtained by the

product of a Rician and a Rayleigh distribution. Therefore, the LLR is given by

$$\text{LLR} = \log \frac{I_0(2y_1\sqrt{E_s}/N_0)}{I_0(2y_2\sqrt{E_s}/N_0)} \tag{4.3}$$

where $I_0(\cdot)$ is the modified Bessel function of the kirst kind with order zero, and is give by

$$I_0(x) = \sum_{k=0}^{\infty} \frac{(\frac{1}{4}x^2)^k}{(k!)^2} \tag{4.4}$$

The modulation and recharging schemes are illustrated in Fig. 4.2

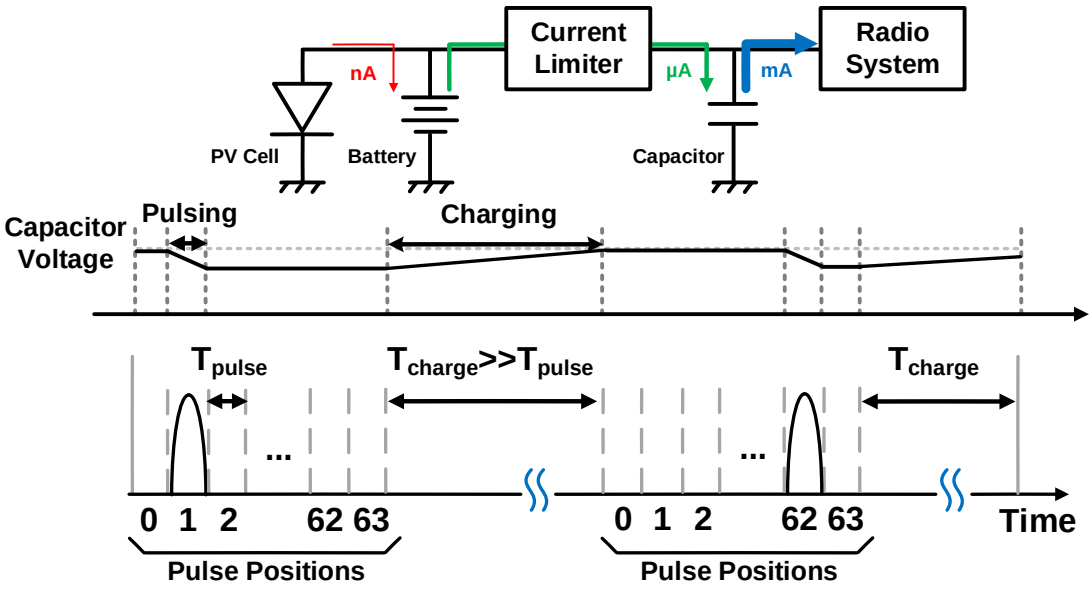


Figure 4.2: M-PPM Modulation and Recharging Scheme

4.2.2 Communication Protocol

The proposed communication system uses an asymmetric link between the gateway and distributed sensor nodes as the gateway has much relaxed constraints on power, complexity, and form-factor dimension. The proposed system is based on a star network topology, where every sensor node is individually linked to a nearby gateway. The real-time gateway realized on the USRP [88] platform has excellent receiver sensitivity, high transmitter power, and abundant FPGA resources for digital signal processing. In the proposed sensor initiating protocol, the sensor node is mostly in sleep in order to save energy, while the gateway receiver is always listening to the channel to find connection messages initiated by nearby sensor nodes. Extending the concept originally proposed in [89], we demonstrate *real-time bi-directional communication* between mm-scale sensor nodes and an USRP based gateway.

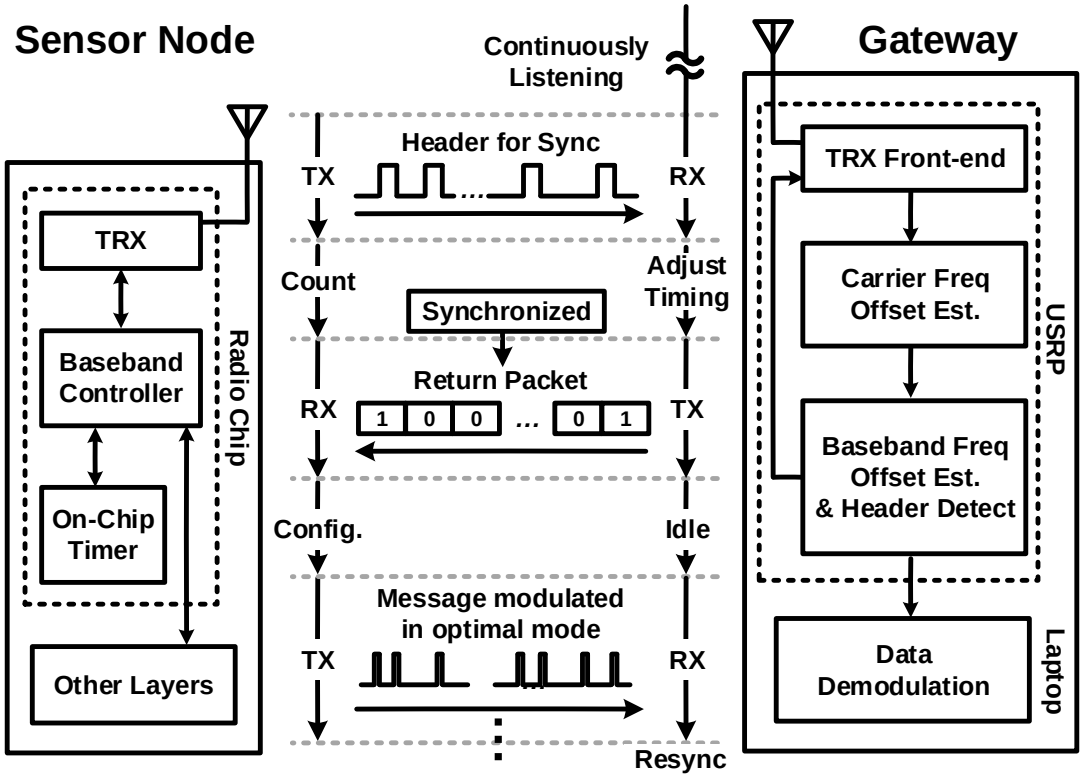


Figure 4.3: Adaptive sensor-initiation synchronization communication protocol

Synchronization between the mm-scale sensor node and the gateway is entirely performed on the gateway. It is gateway's responsibility to track and adapt to the baseband and carrier frequency offset of each sensor node. The gateway analyzes the channel quality and send configuration commands to the sensor node with optimal modulation settings, enabling dynamic *link adaptation*. Fig. 4.3 shows the timing diagram of the protocol.

The sensor node first initiates the communication by sending a header. The gateway is continuously listening to the channel until it finds a valid header. After the header is received, the gateway starts analyzing and calculating the carrier and baseband sampling frequency offset. The sensor node that initiated the communication enters the receive mode after a pre-defined waiting time measured by a ultra-low power timer in the sensor node. During this turn-around time, the gateway estimates and adjusts its baseband timing and carrier frequency to compensate the offset estimated from the sensor node packet. Hence the return packet is synchronized to each sensor node's local timer without explicit synchronization or header detection process performed on the sensor node. The demodulation process on the sensor node is greatly simplified as it starts demodulating symbols at a pre-defined time slot measured by the local low-power relaxation oscillator based timer [90].

4.3 Gateway Synchronization

4.3.1 Timing Offset, CFO and SFO

The modulation scheme and comunication protocol introduced in section 4.2 are based on the assumption that the synchronization has been established between the sensor node and the gateway. However, for frequency and phase unstable signal, packet synchronization is a challenge, and may even limit the reliability of the communication system if not taken care of well.

Obviously, the M-PPM modulation scheme relies on timing accuracy since the information is embbeded in the time information. When there is time offset, the boundaries between time slots recognized by the gateway are not aligned with the actual signal, and hence the matched filter output is not exact. To make thing worse, when this happens, we not only lose SNR for the signal time slot, but also leak power into other time slots, increasing the error probability with double effects. Therefore, timing synchronization is extremely critical in this system.

On the other hand, frequency synchronization also plays a important role. To filter out the out-of-band noise, accurate carrier frequency estimation is required. Moreover, sampling clock precision is also critical since even little clock offset will result in timing offset after accumulation. Therefore, both carrier frequency offset (CFO) and sampling frequency offset (SFO) need to be well compensated.

The main challenge in the gateway design is to identify the frequency and timing offsets with the sensor node in real-time. The TX-RX turn-around time of the sensor node imposes a strict latency constraint on this real-time synchronization process. The sensor node does not have a crystal oscillator. Instead, its baseband sampling clock is generated by a RC relaxation oscillator [90]. Its carrier frequency is determined by the inductance value of the 3D magnetic dipole antenna and the matching on-/off-chip capacitors without a PLL. Thus, it is inevitable that the sensor node has significant baseband SFO (up to 0.5%) as well as CFO (up to 2%) affected by PVT variations. Calculating accurate SFO / CFO and compensating these offsets for the return packet in real-time is performed on the FPGA of the USRP platform.

In this section, we propose an efficient synchronization algorithm deploited on the gateway, including packet detecion, timing and frequency synchronization, and drift tracking.

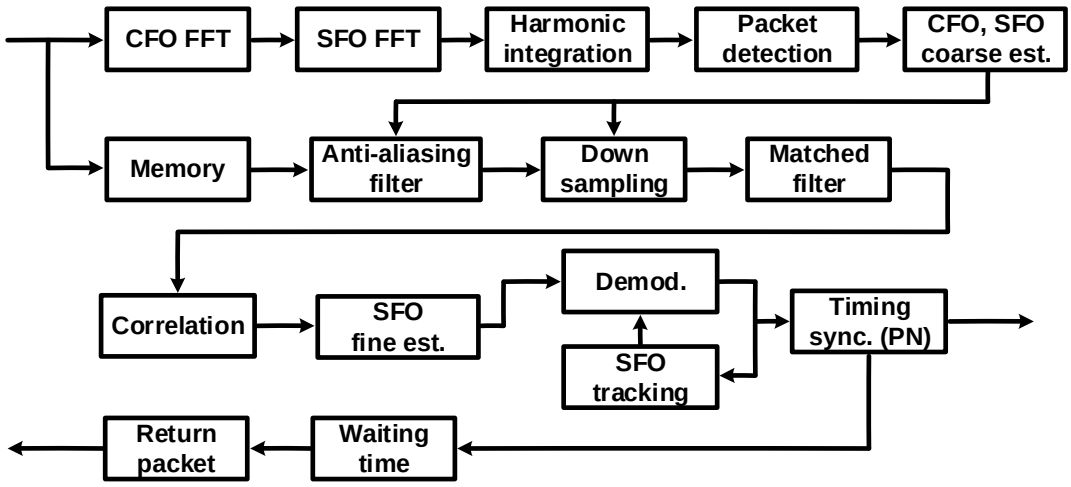


Figure 4.4: DSP datapath implemented on the gateway FPGA

4.3.2 2D-FFT Frequency Offset Estimation and Packet Detection

Fig. 4.4 shows the detailed signal processing datapath implemented on the gateway's FPGA. The header from the sensor node always starts with an RF pulse train with a constant pulse interval, as illustrated in figure 4.5. Thus, we propose a 2D-FFT based process that identifies the SFO and CFO at the same time. The incoming signal is first divided into multiple time domain signal frames, whose length is equivalent to the half of the pulse width. A 1D-FFT is performed on each signal frame and signal power is computed for each frequency offset bin, which correspond to a specific CFO hypotheses. A second FFT is performed on the sample power of frequency domain samples (output of the first FFT) that belong to that same bin (one specific CFO). This process is repeated for all frequency bins. Each bin of the second FFT output now corresponds to a specific SFO fundamental frequency.

In mathematical form, consider a window with MN recevied samples $x[i], i = 0, 1, \cdots, N-1, N, \cdots, MN-1$. We devide the window into M subwindows, where each subwindow has N samples. After applying the first FFT for each subwindow,

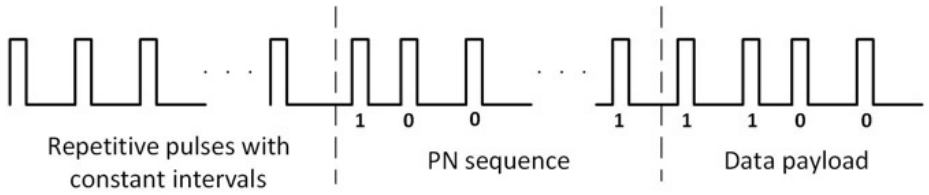


Figure 4.5: Pulse train of the transmiited signal

we will then form a matrix with size $N \times M$ as

$$X_{km}^{\text{CFO}} = \sum_{n=0}^{N-1} x[mN+n] e^{-j\frac{2\pi kn}{N}}, \tag{4.5}$$

for $k = 0, \cdots, N-1$ and $M = 0, \cdots M-1$. The second FFT is applied to each row of the matrix, resulting in

$$X_{kl}^{\text{SFO}} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[mN+n] e^{-j2\pi(\frac{kn}{N}+\frac{lm}{M})}, \tag{4.6}$$

for $k = 0, \cdots, N-1$ and $l = 0, \cdots M-1$. In the matrix X^{SFO}, each row corresponds to a CFO hypothesis, while each column correspones to a sampling frequency (not offset), both depending on the sampling rate and choice of M, N.

To accurately estimate the actual pulse repetition frequency of the header, we add the power of all harmonic frequency bins corresponding to a specific fundamental frequency. This is done by first setting a sampling frequency hypothesis, then identify the column indices that correpond to its fundamental and harmonic frequencies. The final output of the 2D-FFT block is the ratio between the power in harmonic bins and nonharmonic bins for each CFO and SFO hypothesis. The values in a row (i.e., for a CFO hypothesis) can be expressed as

$$P_{\text{ratio}}[k,i] = \frac{\sum_{l\in\mathcal{H}_i} P_{kl}}{\sum_l P_{kl} - \sum_{l\in\mathcal{H}_i} P_{kl} - P_{k0}} \tag{4.7}$$

where P_{kl} is obtained by $P_{kl} = |X_{kl}^{\text{SFO}}|^2$, which represents the power. The set $\mathcal{H}_i$ is

consist of all the column index that correspond to fundamental and harmonic frequencies of the i-th SFO hypothesis. Note that we subtract P_{k0} to prevent unwantted effect from DC. If the value $P_{\text{ratio}}[k, i]$ exceeds a predetermined threshold, the gateway announces the arrival of the signal and continues the rest of the demodulation datapath.

Fig. 4.6 shows an example of the 2D-FFT harmonic integration output from the header processing, where the y-axis corresponds to the CFO bin and x-axis is the SFO fundamental frequency hypothesis. By finding the maximum power from the 2D-FFT result, the gateway identifies the SFO as well as the CFO at the same time. The CFO FFT resolution is inversely proportional to the header pulse width, which is 1–250 kHz in our system. Fig. 4.6 is the result for 6.5MHz CFO and 5kHz SFO from the 915MHz and 250kHz ideal carrier and sampling frequencies.

4.3.3 Timing Detection and SFO tracking

The initial SFO estimation is followed by a fine SFO estimation, since the initial resolution might not be enough. Multiple hypothesis are tested by calculating the correlation between the received header signal and the hypothesis signal. The highest value is corresponding to the correct SFO and also the signal timing. The estimated timing is then used to determined where the legal sample starts after the signal is downsampled.

Although the time boundary at the gateway should now align well with the transmitted signal, we still need to identify the data payload start time. The data payload start time is detected by finding the PN sequence in the header, as shown in figure 4.5. Since the transmission is noncoherent, we correlate the ampltitude of the received signal (after downsampling) with the expected PN sequence pattern (i.e., the sparse M-PPM output signal of the PN sequence). The timing is determiined by finding the peak value of the correlation.

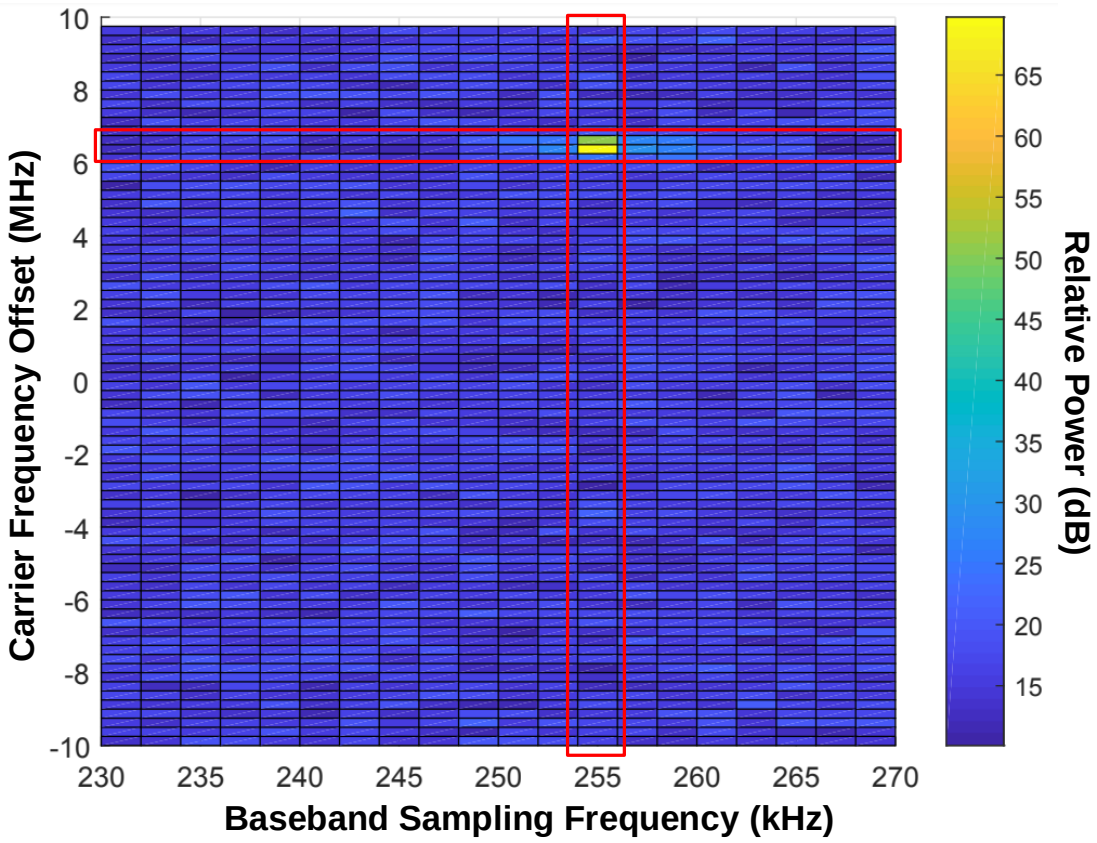


Figure 4.6: 2-D FFT map for sensor node CFO and baseband clock frequency evaluation

After the SFO estimation process for header detection, the gateway keeps tracking SFO during data payload demodulation to eliminate residual SFO and to mitigate the time-drifting offset. At the end of the payload demodulation, the final SFO is applied to calculate the return packet transmission time and its symbol/sampling rate.

4.4 Robustness Enhancement

Besides the unreliable frequency and timing, there are other issues that needed to deal with. To make the system more robust, we further add three practical designs, discussed as follows.

4.4.1 Guard Interval

Due to the jitter of the reference clock in the node, the transmitted signal might have time ambiguity. This will cause additional difficulty on PPM demodulation even if the timing synchronization has been established. To make the system more robust to jitter, a guard interval is put between every position that in the M-PPM. Although doing this will not degrade SNR, the data rate will be halved for a conventional M-PPM since now $2M$ slots have to been used. Nevertheless, the data rate does not decrease a lot in the proposed system because the existence of the recharging cycle, as shown in figure 4.2. Since the recharging cycle is already much longer than the PPM cycle, doubling the PPM portion does not make large difference. For 2-PPM, this data rate reduction is negligible. However, these guard intervals greatly increase the robustness out the system, observed by our empirical results.

4.4.2 Frequency Hopping

In the real world environment, especially in the ISM band, interference is inevitable. Strong interference signal within the signal band is hard to filter out, thus will cause signal-to-interference-plus-noise ratio (SINR) degradation and failure in demodulation. To circuvent this issue, we adopt a commonly used frequency hopping scheme. Multiple duplicated packets in different frequency bands are transmitted for the same information to decrease the probability of overlapping with some interference, at the price of spending more power and time. As long as not all frequency bands are jammed, the gateway will be able to detect a clean packet and demodulate it with sufficient SINR. In the system, the duplicated packets are not overlapped in time because the gateway is not capable to process multiple packets at different bands simultaneously.

4.4.3 CFO Bin Masking

Out-of-band interference can also affect the performance of the system. The 2D-FFT method described in section 4.3 has the ability to detect header signal even when interference presents because it is designed to search for a certain pattern. However, if the interefernce is modulated, it may behave like the singal of interest and result in false detection. A false detection is harmful to the system because it will initiate the demodulation process, which keeps the gateway busy for a certain amount of time (equals to the packet length), and hence the real packet will be overlooked during this time. To alleviate the effect, assuming the modulated interference stays at the same frequency, we can block the CFO bins overlapped with the interference to prevent the freqeunt false triggers. We propose to use an algorithm to mask the noisy channels (i.e., CFO bins in the synchronization algorithms) by monitoring the channel qualitiy of CFO bins. When the quality value goes below a predefined threshold, the gateway will mask the corresponding row in the matrix used for detection (defined in (4.7)) and ignore the values in that row. The channel quality is defined as

$$\mathrm{CQ} = 1 - N_{\mathrm{FA}}/N_{\mathrm{win}} \tag{4.8}$$

where N_{FA} is the number of false alarm in a fix amount of time, and N_{win} is the number of synchronization instances in the same time window. The false alarm is defined as the event that a packet detection is triggered (i.e., both (4.7) and correlation from PN seqeuence have high values) but no valid packet is found, by checking the CRC in the data payload. The mask will be lifted after a certain amount of time to allow detection again.

4.5 Evaluation

4.5.1 Signal Visualization

In Figure 4.7, wireless traffic was captured on an additional USRP device monitoring the channel. Uplink communication is initiated by sparse pulses from the sensor node and the return packet from the gateway is immediately followed by the end of the uplink message.

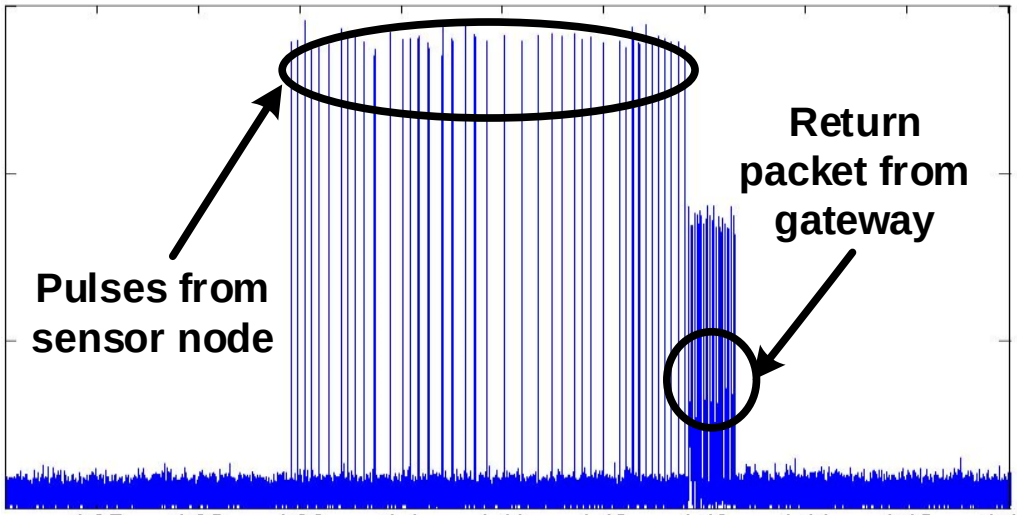


Figure 4.7: Sensor node-gateway communication channel monitoring

In this experiment, we validate that the gateway can correctly and reliably demodulate uplink messages from the sensor node in realistic (uncontrolled) channels as shown in Figure 4.8. Three duplicated packets in different channels from the frequency hopping scheme can be observed in Figure 4.8. There is also stong interference signal residing, where those bins are masked can will not trigger the packet detection and demodulation process thanks to the implementation of real-time CFO bin masking algorithm.

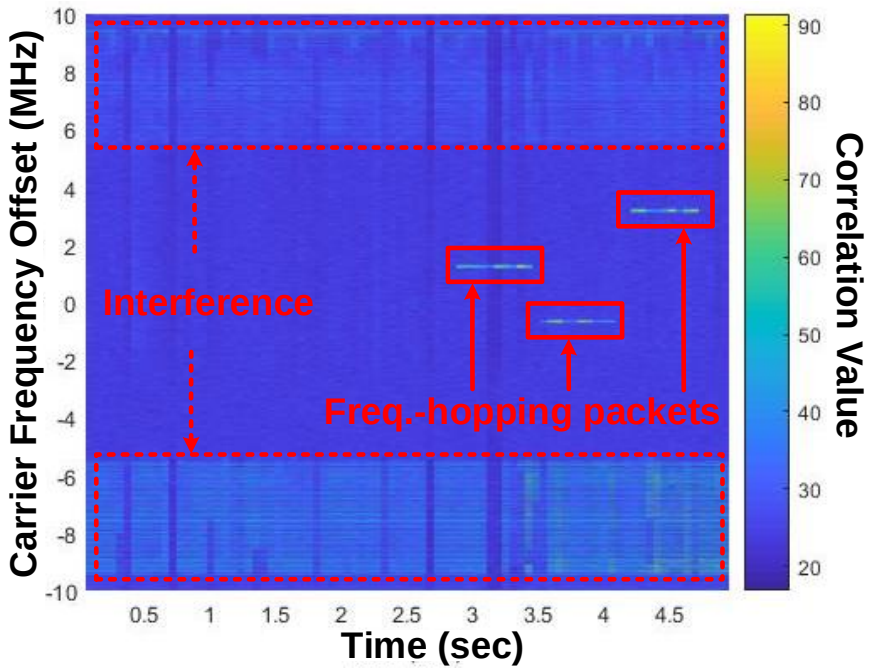
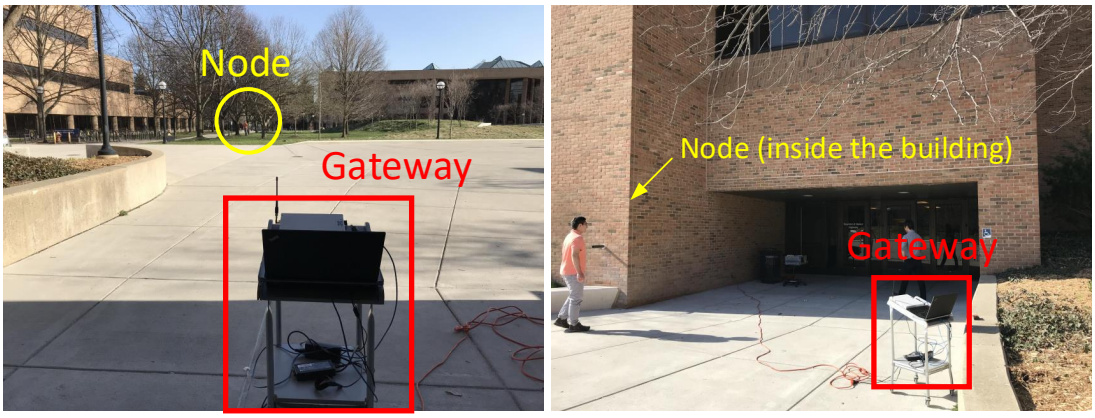


Figure 4.8: A 20 MHz bandwidth spectrogram snapshot of the communication channel

4.5.2 Distance Measurement

The system went through a demostration in the north campus of the University of Michigan. The gateway is build on a URSP X310 pairing with a laptop. Figure 4.9 (a) shows the LOS measurement environment. Both the gateway and the sensor node are located outdoor with distance more than 50 meters. Figure 4.9 (b) shows the NLOS measurement environment, where the gateway is placed outdoor while the sensor node is put inside a classroom in the building with a think wall. Under these environment settings, the system can achieve reliable uplink transmission with $< 10^{-1}$ packet error rate (including packet miss detection). The measurement results are shown in the TABLE 4.1.



(a) (b)

Figure 4.9: Measurement environment. (a) LOS measurement environment (b) NLOS measurement environment

Case	Node location	Gateway location	Distance
LOS	Outdoor	Outdoor	70 m
NLOS	Indoor	Outdoor	12 m
NLOS	Outdoor	Indoor	35 m

Table 4.1: Reliable transmission distance in different test cases.

4.6 Application: Monarch Butterfly Migration Tracking

4.6.1 Motivation, Challenges and Solution

Animal migrators are critical ecosystem indicators because their long-distance travels, often on continental scales, integrate information over broad and diverse geographical locales and seasonal time scales. Tracking technologies have allowed us unprecedented access to these paths, offering insights not only into how migration works, but also into how environments are changing and how species interactions are impacted by changing movements and distributions. However, currently only the larger animal migrators can be tracked for significant portions of their migratory flight. Long-term tracking devices require large amounts of energy and power for information processing and storage and large transceivers and antennas for data

transmission, all of which increase the size and weight of the system. This makes them unsuitable for insect migrators, which make up a large percentage of the total abundance of migrators (e.g., 2.1 billion birds between Europe and Africa versus 3.4 trillion insects over the southern United Kingdom alone). Insect migration detection has been limited to *en masse* detection by vertical-looking entomological radar or tracking for short periods of time or over short distances. Thus, the ability to track small individual migrators over their entire migratory path will offer a tremendous advance in our understanding of migration biology, the impacts of changing climate on small migrators, and effects of migrants on local and global ecosystems.

One of the most enthralling animal tracking stories has been that of the iconic eastern North American Monarch butterfly (*Danaus plexippus*). Each fall, millions of monarchs across the US and Canada migrate up to 4,000 km to overwinter in the same cluster of mountain tops in central Mexico. In spring, these migrants mate and remigrate northwards to repopulate their northern breeding territory over 2–4 partially overlapping generations. Because each migrant monarch completes only part of this round trip, and does not return to the overwintering site, this navigational task cannot be learned from the prior generation or involve commonly employed systems such as path integration, landmark-guidance, or (magnetic field) imprinting.

The number of monarchs completing the journey has steadily declined in the past decades, coincident with the decreased availability of their milkweed host plant. The US, Mexico, and Canada have invested tremendous resources into monarch conservation efforts, including enacting specific policy initiatives, public outreach programs, and habitat protection and restoration projects. The US invested over $11 million between 2015—2017 alone. Developing a tracking technology for monarch can be key in these efforts, for instance through detailed understanding of habitat use during migratory flight and dependence on weather conditions. Furthermore, it can significantly benefit animal research, and agricultural and environmental science.

A monarch tracker must assure daily localization of the butterfly as it progresses on its journey while not interfering with its flight. As such, any deployed sensor must perform this task while having a weight in the tens of milligram (mg) and measuring a few millimeters (mm) in size. The conventional method for determining location is to use GPS. However, the received signal from the satellites is very weak (-155 dBm) and hence requires a power-hungry, very low noise amplifier. To power such a system requires, at minimum, a coin cell sized battery which by itself already weighs ~ 200 mg. Furthermore, the GPS carrier frequency of 1.58 GHz requires a relatively large, centimeter (cm) scale antenna. As a result, the smallest commercial GPS system has a total weight of 1.1 g and size of 5 cm. An alternate to GPS is the Motus system which uses a radio beacon with tens of km transmit distance attached to each specimen combined with geographically distributed receive towers. However, while receive towers are relatively dense in Ontario and along the eastern seaboard, there are very few along the primary monarch migration region in the Midwest. Also, they require a long antenna (multiple cm) and have a weight > 230 mg, which was found to significantly impede monarch flight. Finally, daylight trackers were proposed to compute location based on sunrise/set times. However, their data readout requires physical access to the sensor which is impractical in the case of monarch migration. Furthermore, their size/weight (320 mg and $12 \times 5 \times 4$ mm) remain well beyond that required for the monarch and, with only daylight-based sensing, location accuracy is limited, especially during the equinox.

Monarchs do not migrate in flocks as many bird migrators do. Rather, they primarily migrate as individuals, although sometimes monarchs are seen in large groups as they cross geographical barriers (e.g. crossing large bodies of water). One of the longer-term objectives of this work is to achieve understanding of how variable the individual paths are, as this will inform the type of navigational system that they employ on their trip (e.g. vector-based vs. true map-based). Tracking the

mass has several limitations in determining how monarchs utilize different habitats and resources along their migratory flight. Also, for the same reason, a distributed approach with cameras or other sensors is not effective although there are 35,122 Monarch Waystations, as of July 13, 2021, across US, Canada, Mexico, and the Caribbean. Recently, a distributed approach with a radio transmitter on monarchs utilized the Motus Wildlife Tracking System, an automated radio-telemetry network of over 100 towers. While an important advance, this small number of towers severely limits the coverage of signal detection.

We propose a new wireless sensing platform, mSAIL (Figure 4.10) [91], specifically designed for the monarch migration study based on previously developed custom-designed ICs. mSAIL is an energy-harvesting, 62 mg device with a $8 \times 8 \times 2.6$ mm form-factor (including antenna), that 1) simultaneously measures light intensity and temperature using non-uniform temporal sampling; 2) compresses the recorded data in 16 kB memory; and 3) wirelessly communicates data up to 150 m distance using a crystal-less radio at the overwintering site in a realistic non-line of sight (NLOS) scenario to custom designed gateways. An integrated, chip-size battery, continuously recharged using a custom-designed light-harvesting IC with eight photovoltaic (PV) cells, provides energy autonomy.

mSAIL addresses the technical challenges in Millimeter/milligram form factor, energy autonomy, limited data storage and wireless communication. Specifically, mSAIL adopts the synchronization algirithms for low power sensor nodes proposed in this work, following the same RF radio system and wireless protocol designs. Since monarchs overwinter in dense clusters, often high up in trees, long distance (> 100 m) wireless communication is essential. However, mSAIL imposes extreme constraints on antenna size (8×8 mm) resulting in low antenna efficiency of -8 dBi. In addition, the mm-scale battery can sustain a maximum current draw of only 60 μA due to its high internal resistance. We address this challenge by using a custom sparse pulse position

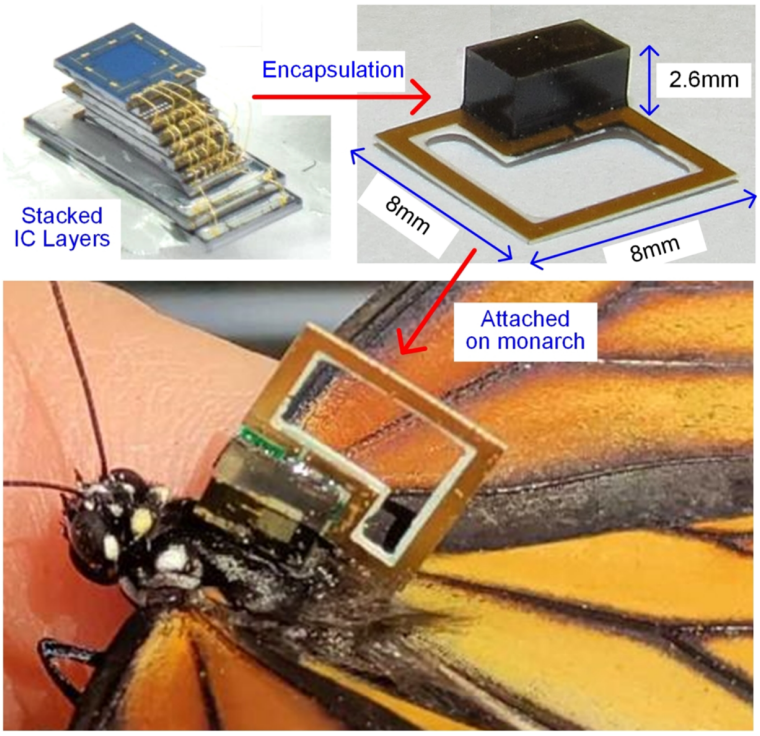


Figure 4.10: mSAIL Monarch butterfly tracking system.

modulation transceiver IC which accumulates charge on a capacitor between pulses, enabling 3.3 dBm transmit power. However, the transceiver IC exhibits high carrier frequency uncertainty and offset because it operates without a RF-reference crystal (typically $>$ MHz), nor a PLL, to reduce the system size, weight, and power. This makes narrowband wireless communications more challenging. We address this using a new 2D-FFT based carrier-and sampling-frequency offset joint estimation algorithm on a USRP X310 [88] compatible, custom gateway for real-time RF communication.

4.6.2 Monarch Migration Tracking Application Scenario

mSAIL records light intensity and temperature with accurate, 32 kHz crystal-based timestamps along the monarch migration path. Standard light-based locationing determines the sunrise and sunset time using a light intensity threshold and then determines the geolocation using a Sun-Earth system model and Lambertian Law.

The day length and center time depend on the geolocation and date and has been used in long-term larger animal tracking studies. However, it has the fundamental limitation of large latitude ambiguity around the equinox (September 22 and March 20) when the day length is the same regardless of latitude. A second challenge is the significant light intensity variation due to weather and terrain that an ideal sunlight intensity model is unable to capture.

mSAIL adopts a novel data-driven algorithm for monarch migration tracking that leverages the principle of correlating multiple sensors. It achieves superior accuracy by applying deep neural network (DNN) models and multimodal fusion to effectively combine multiple sensor readings, including light intensity and temperature. The objective of the DNN approach is to identify cross-correlation between the multimodal readings and the sunlight intensity pattern as well as temperature information on a particular date.

Although the details of the trajectory are not known, the final destination of the monarch migration is known. Overwintering monarchs will distribute over a limited number of sites within central Mexico with 21-78% of the total population reliably congregating at a single site (El Rosario sanctuary, at the Monarch Butterfly Biosphere Reserve) each year. mSAIL nodes will be programmed with a predefined rendezvous time to start wireless data offloading to multiple gateways deployed at that overwintering site. This scenario allows retrieval from both fallen (dead) and live monarchs as long as they are within the communication range. Since an estimated $\sim$90% of monarchs survive at the overwintering site, the data recovery rate is expected to be significantly improved compared with current paper tagging methods that can only access dead butterflies. After a gateway retrieves the data log from an mSAIL, the entire butterfly trajectory can be constructed using the DNN localization algorithm proposed in [92]. The DNN is trained and evaluated by the data collected through a data measurement campaign with 306 volunteers from 2018—2020 across

the US, Canada, and Mexico. They recorded light intensity and temperature using commercial cm-scale sensors [93] as an emulation of mSAIL during the monarch migration season. The localization algorithm [92] shows a geocoordinate accuracy of $< 0.6^\circ$ and $< 1.7^\circ$ in longitude and latitude respectively (1° is $\sim$ 85.2 km in longitude and $\sim$ 111.2 km in latitude in the midwestern US), which is sufficient for monarch studies.

4.6.3 Wireless Communication Evaluation

The mSAIL radio transmission distance was tested at two locations. First, in an unobstructed outdoor environment, mSAIL was placed on a pole such that it was an average 20 m higher than the gateway in order to emulate conditions at the overwintering site when a monarch will be in a tall conifer. Results of the test are shown in Figure 4.11 (a). A 75 cm Yagi antenna with 11 dBi gain was attached to the gateway. Transmission was line-of-sight up to 494 m. A similar test was performed in a heavily wooded area at the overwintering site in Mexico, shown in Figure 4.11 (b) using a 135 cm Yagi with 15 dBi gain which showed excellent results at 150 m with a low ($< 5\%$) packet loss. In both cases, the mSAIL was powered by attaching larger batteries using 10 mm leads to allow for longer operation duration under continuous radio transmission for ease of testing. However, in separate testing this was shown not to impact radio distance noticeably.

Our approach leverages the fact that monarchs cluster at impressive densities at the Mexican overwintering sites. The median density estimate is 20 million butterflies (range 6-60 million) per hectare (10,000 m^2). Therefore, even with hundreds of meters of communication range, we expect to cover millions of butterflies.

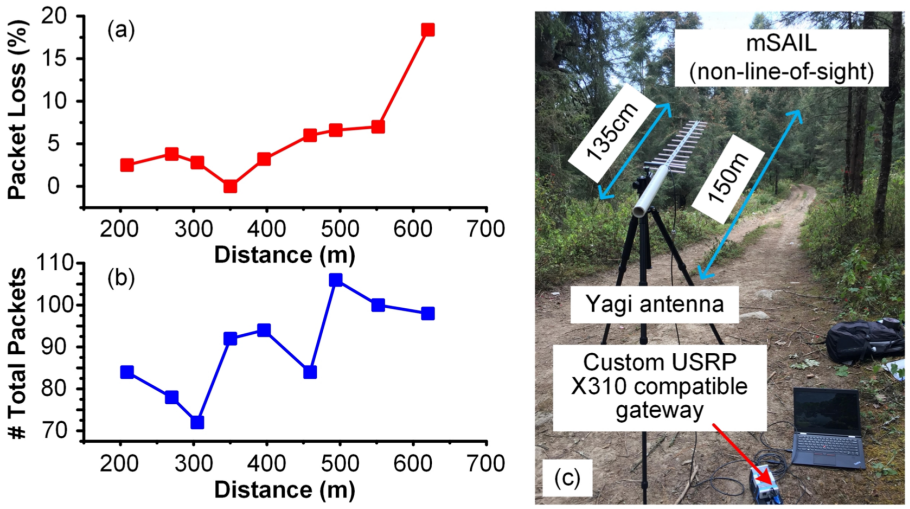


Figure 4.11: RF communication distance test. (a) Result from the open-area test; (b) Test in wooded area.

4.7 Summary

In this chapter, we present a IoT communication system with a fully integrated $3 \times 3 \times 3$ mm^3 ultra-low power sensor node. A low-power asymmetric communication protocol was proposed and implemented on an FPGA based software-defined radio platform for real-time demonstration. To accomendate the frequency and phase unstable signal from the sensor, we propose efficient receiving algorithm at the powerful gateway, including packet detection, CFO/SFO compensation and demodulation. The complete radio system demonstrates $>$ 50 meter and $>$ 20 meter communication distance in LOS and NLOS environment, respectively. The proposed system and synchronization algorithms are applied to a monarch butterfly migration tracking caller mSAIL, where millimeter scale trackers with $>$ 100 m signal receiving distance are utilized to resolve the technical problems of tracking insects.

CHAPTER V

Conclusion and Outlook

5.1 Summary of Contributions

This dissertation focuses on three main challenges faced in IoT: short packet reliability, low latency constraint and low power consumption, aiming at expanding the potential applications in mMTC and URLLC, two essential classes in 5G and beyond. Unlike human-oriented applications that concentrate on data rate with large data, IoT applications in mMTC and URLLC demand distinct features with a much smaller message size. mMTC requires support for massive devices in the same network with satisfactory reliability and decent traffic handling. URLLC requests ultra-reliability and stringent low latency constraint at the same time. Both pose new challenges to the design of communication systems for emerging IoT applications. The goal of this dissertation is to study the potential of different communication systems with novel designs in systems and algorithms.

Chapter II proposes hyper-dimensional modulation (HDM), a novel non-orthogonal modulation scheme, for short packet communication in interference-heavy mMTC scenarios. This work provides a viable solution that addresses severe interference coming from either intra-network due to packet collision from unsynchronized devices, or inter-network due to the coexistence with other technologies. The major contributions of this work include: 1) HDM, a special case of SPARC with a designed

dictionary and dedicated decoding algorithm to extend the usage into short length regime; 2) Extensions of the proposed decoding algorithm to handle both intra- and inter-network interference; 3) Decoding complexity and practical design discussion for the HDM; and 4) Extensive performance evaluation with simulation results and real-world measurements.

Chapter III proposes opportunistic symbol length adaptation (OSLA), a variable symbol length code scheme based on instantaneous feedback, for URLLC applications. This work serves as a strong candidate for ultra-reliable delay-sensitive applications with advantages in computation lightweight encoder and scalability in block length. The major contributions of this work include: 1) OSLA, a new type of variable-length code that adapts symbol length to noise realization with the help of instantaneous feedback, inspired by an early work by Viterbi; 2) Autocorrelation and spectrum analysis for uncoded OSLA signal; 3) Trellis coded OSLA with proposed modified Viterbi algorithm that outperforms state-of-the-art non-feedback short codes and a deep learning-based feedback code; 4) HMM-based feedback scheme for OSLA that shows outstanding robustness in noisy feedback channel; and 5) Extension of OSLA with hard deadline latency constraint.

Chapter IV presents a low power miniaturized radio system with a custom-designed communication scheme, focusing on synchronization for frequency and phase unstable devices. This work demonstrates the idea of using powerful gateway resources to compensate for the non-ideality from a low power crystal-less transmitter node. The major contributions of this work include: 1) Efficient joint CFO and SFO estimation algorithm for frequency and phase unstable signal; 2) Robustness enhancement design for the radio system in ISM band; 3) Field trial experiments of the proposed system with custom-design sensor nodes; and 4) Application to monarch butterfly migration tracking.

5.2 Future Directions

The works presented in this dissertation show potential of novel system designs for IoT applications. Although each work already demonstrates success in its target focus, many aspects of the proposed solution can still be further explored or extended to fulfill a more complete system.

First, the works in this dissertation mostly focus on point-to-point communication, where multi-user performance is omitted. In Chapter II, we discuss the performance of HDM under packet collision. The performance from a network perspective can be analyzed, given the packet arrival and power statistics. Reliability enhancement with packet-level successive interference cancellation can be easily applied once a packet is decoded, however, the optimization of the decoding order is non-trivial and worth exploring. In Chapter IV, only single-user performance is evaluated, and extension to multi-user scenarios is highly desirable. The coexistence of multiple sensor nodes sharing the same ISM band will cause potential packet collision in both time and frequency, so the system, including sensor nodes and the gateway receiver, must be co-designed to improve reception rate by optimizing traffic planning.

Second, performance in fading channels can be characterized. In Chapter III, we formulate the problem for OSLA in AWGN channel, assuming the channel is perfectly equalized. However, OSLA can potentially provide advantages in a fading channel thanks to its automatic adaptation to noise realization given a reliability target. Most non-adaptable systems must be designed for the worst-case scenario to satisfy the application requirements, and possibly waste the resource when the channel condition is good. On the contrary, OSLA is designed to terminate the packet earlier if the target reliability is already achieved in a good channel, thus preventing the waste of time or energy. It is worth quantifying the gain of OSLA in a fading channel, in terms of energy saving or latency reduction.

Third, exploiting the modeling power from deep neural networks (DNN) can po-

tentially improve the proposed schemes. The codes in HDM in Chapter II and OSLA in Chapter III are both hand-designed with linear operation. By unleashing the power of nonlinear operation from DNN, the code structure may be improved and thus the performance. Instead of a block box method, adding DNN on top of HDM or OSLA structure is of interest and may provide potential benefits in scalability and complexity. For example, replacing the encoding and metric calculation in the K-best decoding algorithm for HDM with DNN may be a viable solution to balance the performance and tedious training process for larger size codes. Using recurrent neural networks to replace the convolution codes in OSLA is also an interesting direction to explore, since OSLA is suitable for sequential decoding.

Last but not least, prototyping for the proposed communication systems is desired. In Chapter IV, we demonstrate a completed radio system, in which non-ideality from the hardware must be dealt with. We also verify the effectiveness of HDM in Chapter II with real-world measurements, but the experimental success is based on the setting of a relatively long preamble, bypassing the packet detection and channel equalization. Integration of HDM with a newly designed header structure can be an interesting research direction. On the other hand, Chapter III discusses OSLA in theoretical aspects, and the evaluation is only done by simulation. Since the system is based on instantaneous feedback, real-time processing is critical to the realization of OSLA. Hardware implementation of OSLA must be done to substantiate the idea and to prove the practicality of the system.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] L. S. Vailshery, "Internet of Things (IoT) revenue worldwide from 2019 to 2030 (in billion U.S. dollars), by vertical," 2022. [Online]. Available: https://www.statista.com/statistics/1183471/iot-revenue-worldwide-by-vertical/#statisticContainer

[2] M. Series, "IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond," *Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation M.2083-0*, Sep 2016.

[3] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Feedback in the non-asymptotic regime," *IEEE Trans. on Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, 2011.

[4] C. Shannon, "A Mathematical Theory of Communication," *Bell Sys. Tech. J.*, no. 3, pp. 379–423, 1948.

[5] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[6] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[8] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," vol. 56, no. 5, pp. 2307–2359, 2010.

[9] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code design for short blocks: A survey," *CoRR*, vol. abs/1610.00873, 2016.

[10] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wireless Communications*, vol. 24, pp. 120–128, 2017.

[11] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive Machine-Type Communications in 5G: Physical and MAC-Layer Solutions," vol. 54, no. 9, pp. 59–65, 2016.

[12] G. Durisi, T. Koch, and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.

[13] L. Gaudio, T. Ninacs, T. Jerkovits, and G. Liva, "On the Performance of Short Tail-Biting Convolutional Codes for Ultra-Reliable Communications," in *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*, 2017, pp. 1–6.

[14] M. Baldi, F. Chiaraluce, N. Maturo, G. Liva, and E. Paolini, "A Hybrid Decoding Scheme for Short Non-Binary LDPC Codes," vol. 18, no. 12, pp. 2093–2096, 2014.

[15] H. Nikopour and H. Baligh, "Sparse Code Multiple Access," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 332–336.

[16] S. Chen, B. Ren, Q. Gao, S. Kang, S. Sun, and K. Niu, "Pattern Division Multiple Access—A Novel Nonorthogonal Multiple Access for Fifth-Generation Radio Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3185–3196, 2017.

[17] L. Ping, L. Liu, K. Wu, and W. Leung, "Interleave Division Multiple-Access," *IEEE Transactions on Wireless Communications*, vol. 5, no. 4, pp. 938–947, 2006.

[18] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "A Novel Analytical Framework for Massive Grant-Free NOMA," vol. 67, no. 3, pp. 2436–2449, 2019.

[19] A. Joseph and A. R. Barron, "Fast Sparse Superposition Codes Have Near Exponential Error Probability for $R < \mathcal{C}$," vol. 60, no. 2, pp. 919–942, 2014.

[20] K. Hsieh and R. Venkataramanan, "Modulated sparse superposition codes for the complex awgn channel," *IEEE Transactions on Information Theory*, vol. 67, no. 7, pp. 4385–4404, 2021.

[21] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed Sensing for Wireless Communications: Useful Tips and Tricks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1527–1550, 2017.

[22] X. Xiong, K. Zheng, R. Xu, W. Xiang, and P. Chatzimisios, "Low Power Wide Area Machine-to-Machine Networks: Key Techniques and Prototype," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 64–71, Sep. 2015.

[23] P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors," *Cognitive Computation*, vol. 1, no. 2, pp. 139–159, 2009.

[24] H. Kim, "HDM: Hyper-Dimensional Modulation for Robust Low-Power Communications," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[25] Zhan Guo and P. Nilsson, "Algorithm and Implementation of the K-best Sphere Decoding for MIMO Detection," vol. 24, no. 3, pp. 491–503, 2006.

[26] C. Goursaud and Y. Mo, "Random Unslotted Time-Frequency ALOHA: Theory and Application to IoT UNB Networks," in *2016 23rd International Conference on Telecommunications (ICT)*, 2016, pp. 1–5.

[27] S. Popli, R. K. Jha, and S. Jain, "A Survey on Energy Efficient Narrowband Internet of Things (NBIoT): Architecture, Application and Challenges," *IEEE Access*, vol. 7, pp. 16 739–16 776, 2019.

[28] C.-W. Hsu and H.-S. Kim, "Collision-Tolerant Narrowband Communication Using Non-Orthogonal Modulation and Multiple Access," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[29] M. Mehrnoush and S. Roy, "Coexistence of WLAN Network With Radar: Detection and Interference Mitigation," vol. 3, no. 4, pp. 655–667, 2017.

[30] C.-W. Hsu and H.-S. Kim, "Non-Orthogonal Modulation for Short Packets in Massive Machine Type Communications," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[31] C. Martınez, "Partial quicksort," *Proc. 6th ACMSIAM Workshop on Algorithm Engineering and Experiments and 1st ACM-SIAM Workshop on Analytic Algorithmics and Combinatorics*, pp. 224–228, 2004.

[32] 3GPP R1-1610141, "Short block-length design," *3rd Generation Partnership Project*, 2016.

[33] M. Ajtai, J. Komlós, W. Steiger, and E. Szemerédi, "Optimal Parallel Selection Has Complexity O(Log Log N)," *Journal of Computer and System Sciences*, vol. 38, no. 1, pp. 125–133, 1989.

[34] B. Li, H. Shen, and D. Tse, "Parallel decoders of polar codes," *CoRR*, vol. abs/1309.1026, 2013. [Online]. Available: http://arxiv.org/abs/1309.1026

[35] H. S. Kim and B. Daneshrad, "Power Optimized PA Clipping for MIMO-OFDM Systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 9, pp. 2823–2828, Sep. 2011.

[36] H. Ochiai and H. Imai, "Performance of the Deliberate Clipping with Adaptive Symbol Selection for Strictly Band-Limited OFDM Systems," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 11, pp. 2270–2277, 2000.

[37] T. Koike-Akino, K. J. Kim, M. Pajovic, and P. V. Orlik, "Universal multi-stage precoding with monomial phase rotation for full-diversity m2m transmission," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–7.

[38] 3GPP TS 38.212, "NR; Multiplexing and Channel Coding (Release 15)," *3rd Generation Partnership Project*, 2018.

[39] I. Tal and A. Vardy, "List Decoding of Polar Codes," vol. 61, no. 5, pp. 2213–2226, 2015.

[40] R. Y. Shao, Shu Lin, and M. P. C. Fossorier, "Two Decoding Algorithms for Tailbiting codes," *IEEE Trans. on Commun.*, vol. 51, no. 10, pp. 1658–1665, 2003.

[41] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner, "Efficient Error-Correcting Codes in the Short Blocklength Regime," *Physical Communication*, vol. 34, pp. 66 – 79, 2019.

[42] W. Yang, Y. Wang, J. Soriaga, T. Ji, and K. Mukkavilli, "Coding performance modeling for short-packet communications," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 820–826.

[43] Ettus Research X310, https://www.ettus.com/all-products/x310-kit/.

[44] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-Achieving Sparse Superposition Codes via Approximate Message Passing Decoding," *IEEE Transactions on Information Theory*, vol. 63, no. 3, pp. 1476–1500, 2017.

[45] G. Forney and L.-F. Wei, "Multidimensional constellations. i. introduction, figures of merit, and generalized cross constellations," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 877–892, 1989.

[46] T. Koike-Akino and V. Tarokh, "Sphere packing optimization and exit chart analysis for multi-dimensional qam signaling," in *2009 IEEE International Conference on Communications*, 2009, pp. 1–5.

[47] H. Ji, S. Park, and B. Shim, "Sparse Vector Coding for Ultra Reliable and Low Latency Communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6693–6706, 2018.

[48] S. Kwon, J. Wang, and B. Shim, "Multipath Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2986–3001, 2014.

[49] E. Basar, U. Aygolu, E. Panayırcı, and H. V. Poor, "Orthogonal Frequency Division Multiplexing With Index Modulation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5536–5549, 2013.

[50] M. Hersche, S. Lippuner, M. Korb, L. Benini, and A. Rahimi, "Near-Channel Classifier: Symbiotic Communication and Classification in High-Dimensional Space," *Brain Informatics*, vol. 8, no. 1, p. 16, 2021.

[51] T. M. Cover and J. A. Thomas, *Elements of Information Theory, Second Edition*. John Wiley & Sons, Inc., 2006.

[52] R. L. Dobrushin, "An asymptotic bound for the probability error of information transmission through a channel without memory using the feedback," *Probl. Peredachi Inf.*, vol. 8, pp. 161–160, 1962.

[53] E. A. Haroutunian, "Lower bound for error probability in channels with feedback," *Probl. Peredachi Inf.*, vol. 13, pp. 36–44, 1977.

[54] C. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. on Inf. Theory*, vol. 2, no. 3, pp. 8–19, 1956.

[55] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback–i: No bandwidth constraint," *IEEE Trans. on Inf. Theory*, vol. 12, no. 2, pp. 172–182, 1966.

[56] A. Ben-Yishai and O. Shayevitz, "Interactive schemes for the awgn channel with noisy feedback," *IEEE Trans. on Inf. Theory*, vol. 63, no. 4, pp. 2409–2427, 2017.

[57] Y. Urman and D. Burshtein, "Feedback channel communication with low precision arithmetic," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2067–2072.

[58] N. C. Martins and T. Weissman, "Coding for additive white noise channels with feedback corrupted by quantization or bounded noise," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 4274–4282, 2008.

[59] Z. Chance and D. J. Love, "Concatenated coding for the awgn channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6633–6649, 2011.

[60] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *IEEE J. on Sel. Areas in Inf. Theory*, vol. 1, no. 1, pp. 194–206, 2020.

[61] M. V. Burnashev, "Data transmission over a discrete channel with feedback. Random transmission time," *Probl. Peredachi Inf.*, vol. 12, no. 4, pp. 10–30, Oct.-Dec. 1976.

[62] H. Yamamoto and K. Itoh, "Asymptotic performance of a modified schalkwijk-barron scheme for channels with noiseless feedback (corresp.)," *IEEE Trans. on Inf. Theory*, vol. 25, no. 6, pp. 729–733, 1979.

[63] S. Draper and A. Sahai, "Variable-length coding with noisy feedback," *Eur. Trans. Telecommun.*, vol. 19, pp. 355–370, Jun. 2008.

[64] A. J. Viterbi, "The effect of sequential decision feedback on communication over the gaussian channel," *Information and Control*, vol. 8, no. 1, pp. 80–92, 1965.

[65] C.-W. Hsu, A. Anastasopoulos, and H.-S. Kim, "Instantaneous feedback-based opportunistic symbol length adaptation for reliable communication," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[66] A. N. Borodin and P. Salminen, *Handbook of Brownian Motion - Facts and Formulae*. Birkhauser, 1996.

[67] P. L. Smith, "A note on the distribution of response times for a random walk with gaussian increments," *Journal of Mathematical Psychology*, vol. 34, no. 4, pp. 445–459, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022249690900233

[68] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[69] J. G. Proakis and M. Salehi, *Digital Communications, 5th Edition*. McGraw-Hill Education, 2007.

[70] H. Ma and J. Wolf, "On tail biting convolutional codes," *IEEE Trans. on Commun.*, vol. 34, no. 2, pp. 104–111, 1986.

[71] 3GPP TS 36.212, "Evolved universal terrestrial radio access (e-utra); multiplexing and channel coding."

[72] G. Fettweis and H. Meyr, "High-rate viterbi processor: a systolic array solution," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 8, pp. 1520–1534, 1990.

[73] ——, "High-speed parallel viterbi decoding: algorithm and vlsi-architecture," *IEEE Communications Magazine*, vol. 29, no. 5, pp. 46–55, 1991.

[74] Xilinx UltraScale XCVU440 FPGA, https://www.xilinx.com/products/boards-and-kits/1-66ql3z.html.

[75] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[76] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 2829–2838.

[77] H. Hasselt, "Double q-learning," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010.

[78] A. Raghavan and C. Baum, "A reliability output viterbi algorithm with applications to hybrid arq," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1214–1216, 1998.

[79] 3GPP TS 34.121, "Technical Specification Group Terminals; Terminal conformance specification; Radio transmission and reception (FDD) (Release 6," *3rd Generation Partnership Project*, 2005.

[80] A. Ben-Yishai and O. Shayevitz, "SK and Modulo-SK Matlab Code." [Online]. Available: https://github.com/assafbster/Modulo-SK

[81] G. Chen, M. Fojtik, D. Kim, D. Fick, J. Park, M. Seok, M. T. Chen, Z. Foo, D. Sylvester, and D. Blaauw, "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb 2010, pp. 288–289.

[82] M. Tabesh, N. Dolatsha, A. Arbabian, and A. M. Niknejad, "A power-harvesting pad-less millimeter-sized radio," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 962–977, April 2015.

[83] "Bluetooth Core Version 5.0 specification," Online.

[84] "Zigbee products," Online.

[85] "Z-Wave," Online.

[86] L. X. Chuo, Y. Shi, Z. Luo, N. Chiotellis, Z. Foo, G. Kim, Y. Kim, A. Grbic, D. Wentzloff, H. S. Kim, and D. Blaauw, "7.4 a 915mhz asymmetric radio using q-enhanced amplifier for a fully integrated 3x3x3mm^3 wireless sensor node with 20m non-line-of-sight communication," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 132–133.

[87] Y. Lee, S. Bang, I. Lee, Y. Kim, G. Kim, M. H. Ghaed, P. Pannuto, P. Dutta, D. Sylvester, and D. Blaauw, "A modular 1 mm^3 die-stacked sensing platform with low power i^{2}c inter-die communication and multi-modal energy harvesting," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 229–243, Jan 2013.

[88] Universal Software Radio Peripheral (USRP), Ettus Research LLC, Online.

[89] Y. Shi, M. Choi, Z. Li, Z. Luo, G. Kim, Z. Foo, H. S. Kim, D. D. Wentzloff, and D. Blaauw, "A 10 mm^3 inductive coupling radio for syringe-implantable smart sensor nodes," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 11, pp. 2570–2583, Nov 2016.

[90] M. Choi, T. Jang, S. Bang, Y. Shi, D. Blaauw, and D. Sylvester, "A 110 nw resistive frequency locked on-chip oscillator with 34.3 ppm/^{o}c temperature stability for system-on-chip designs," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2106–2118, Sept 2016.

[91] I. Lee, R. Hsiao, G. Carichner, C.-W. Hsu, M. Yang, S. Shoouri, K. Ernst, T. Carichner, Y. Li, J. Lim, C. R. Julick, E. Moon, Y. Sun, J. Phillips, K. L. Montooth, D. A. Green, H.-S. Kim, and D. Blaauw, *MSAIL: Milligram-Scale Multi-Modal Sensor Platform for Monarch Butterfly Migration Tracking.* New

York, NY, USA: Association for Computing Machinery, 2021, p. 517–530. [Online]. Available: https://doi.org/10.1145/3447993.3483263

[92] M. Yang, R. Hsiao, G. Carichner, K. Ernst, J. Lim, D. A. Green, I. Lee, D. Blaauw, and H.-S. Kim, "Migrating monarch butterfly localization using multi-modal sensor fusion neural networks," in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1792–1796.

[93] ONSET., "Hobo pedant mx temperature/light data logger." Online.