ABSTRACT

ERROR CORRECTION FOR ASYNCHRONOUS COMMUNICATION AND PROBABILISTIC BURST DELETION CHANNELS

by

Chen Yi

Short-range wireless communication with low-power small-size sensors has been broadly applied in many areas such as in environmental observation, and biomedical and health care monitoring. However, such applications require a wireless sensor operating in "always-on" mode, which increases the power consumption of sensors significantly. Asynchronous communication is an emerging low-power approach for these applications because it provides a larger potential of significant power savings for recording sparse continuous-time signals, a smaller hardware footprint, and a lower circuit complexity compared to Nyquist-based synchronous signal processing.

In this dissertation, the classical Nyquist-based synchronous signal sampling is replaced by asynchronous sampling strategies, i.e., sampling via level crossing (LC) sampling and time encoding. Novel forward error correction schemes for sensor communication based on these sampling strategies are proposed, where the dominant errors consist of pulse deletions and insertions, and where encoding is required to take place in an instantaneous fashion. For LC sampling the presented scheme consists of a combination of an outer systematic convolutional code, an embedded inner marker code, and power-efficient frequency-shift keying modulation at the sensor node. Decoding is first obtained via a maximum a-posteriori (MAP) decoder for the inner marker code, which achieves synchronization for the insertion and deletion channel, followed by MAP decoding for the outer convolutional code. By iteratively decoding marker and convolutional codes along with interleaving, a significant reduction in terms of the expected end-to-end distortion between original and reconstructed signals can be obtained compared to non-iterative processing. Besides investigating the

rate trade-off between marker and convolutional codes, it is shown that residual redundancy in the asynchronously sampled source signal can be successfully exploited in combination with redundancy only from a marker code. This provides a new low complexity alternative for deletion and insertion error correction compared to using explicit redundancy. For time encoding, only the pulse timing is of relevance at the receiver, and the outer channel code is replaced by a quantizer to represent the relative position of the pulse timing. Numerical simulations show that LC sampling outperforms time encoding in the low to moderate signal-to-noise ratio regime by a large margin.

In the second part of this dissertation, a new burst deletion correction scheme tailored to low-latency applications such as high-read/write-speed non-volatile memory is proposed. An exemplary version is given by racetrack memory, where the element of information is stored in a cell, and data reading is performed by many read ports or heads. In order to read the information, multiple cells shift to its closest head in the same direction and at the same speed, which means a block of bits (i.e., a non-binary symbol) are read by multiple heads in parallel during a shift of the cells. If the cells shift more than by one single cell location, it causes consecutive (burst) non-binary symbol deletions.

In practical systems, the maximal length of consecutive non-binary deletions is limited. Existing schemes for this scenario leverage non-binary de Bruijn sequences to perfectly locate deletions. In contrast, in this work binary marker patterns in combination with a new soft-decision decoder scheme is proposed. In this scheme, deletions are soft located by assigning a posteriori probabilities for the location of every burst deletion event and are replaced by erasures. Then, the resulting errors are further corrected by an outer channel code. Such a scheme has an advantage over using non-binary de Bruijn sequences that it in general increases the communication rate.

ERROR CORRECTION FOR ASYNCHRONOUS COMMUNICATION AND PROBABILISTIC BURST DELETION CHANNELS

by
Chen Yi

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann
Department of Electrical and Computer Engineering

August 2020

APPROVAL PAGE

ERROR CORRECTION FOR ASYNCHRONOUS COMMUNICATION AND PROBABILISTIC BURST DELETION CHANNELS

Chen Yi

Dr. Jörg Kliewer, Dissertation Advisor Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Alexander Haimovich, Committee Member Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Osvaldo Simeone, Committee Member Date
Professor of Information Engineering, King's College London, United Kingdom

---

Dr. Wei Tang, Committee Member Date
Associate Professor of Electrical and Computer Engineering,
New Mexico State University, Las Cruces, NM

BIOGRAPHICAL SKETCH

Author: Chen Yi

Degree: Doctor of Philosophy

Date: August 2020

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2020
- Master of Engineering in Signal and Information Processing,
  Chongqing University of Posts and Telecommunications, Chongqing, China, 2010
- Bachelor of Science in Ecology,
  Shandong University, Jinan, Shandong, China, 2007

Major: Electrical Engineering

Presentations and Publications:

C. Yi, J. Kliewer, "Error correction for low power sensors in asynchronous communication," submitted to *Signal Processing*, May, 2020.

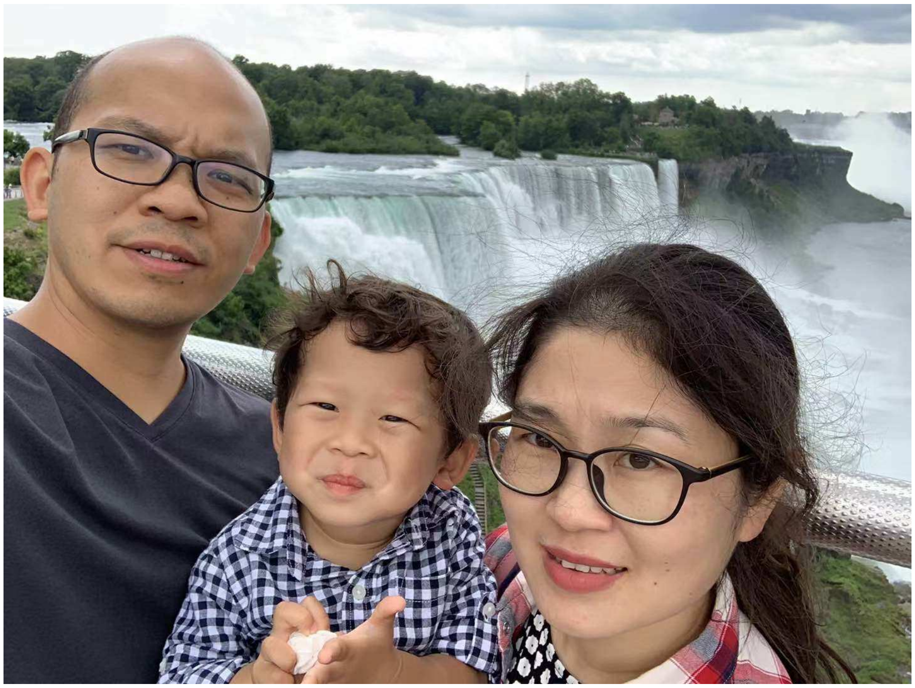
C. Yi, J. Kliewer, "Correcting deletions in probabilistic non-binary segmented burst deletion channels," *IEEE 53th Asilomar Conference on Signals, Systems and Computers*, pp 1349-1353, Pacific Grove, CA, November, 2019.

C. Yi, J. Kliewer, "On error correction for asynchronous communication based on iterative processing," *IEEE 50th Asilomar Conference on Signals, Systems and Computers*, pp 240-244, Pacific Grove, CA, November, 2016.

C. Yi, J. Kliewer, "Error correction for asynchronous communication," *IEEE 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pp 310-314, Brest, France, September, 2016.

Q. Hu, C. Yi, J. Kliewer, and W. Tang "Asynchronous communication for wireless sensors using ultra wideband impulse radio," *IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp 1-4, Fort Collins, CO, August, 2015.

Dedicated to my beloved parents, my wife & son
For their love, endless support,
encouragement & sacrifices



ACKNOWLEDGMENT

First, I would like to express my sincere gratitude to my advisor, Professor Jörg Kliewer for his continuous support of my PhD study and related research, for his countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process. His guidance helped me in all the time of research and writing of this dissertation.

I would also like to thank you Dr. Alexander Haimovich, Dr. Ali Abdi, Dr. Osvaldo Simeone, and Dr. Wei Tang for agreeing to serve on my committee. Thank you for your insightful comments and encouragement, and the hard questions to widen my research from various perspectives.

Last but not the least, I would like to thank my parents, ZhengFa Yi and FenQiu Lu, my wife, An Wang and my son, Benjamin Yi for supporting me spiritually throughout my life in general.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

Since the far-reaching work of Shannon [33], coding and information theory has been significantly improved. Fundamental limits and efficient coding solutions which approach these limits are well known for many communication channels. However, for all those vast majority of coding schemes invented, it is assumed that the receiver is perfectly synchronized with the transmitter, i.e., the symbol arrival times are known at the receiver. If the perfect synchronization is not guaranteed, random symbol deletions and/or insertions (synchronization errors) occur in the received sequence. This phenomenon poses a great challenge for error correction because the positions of the inserted/deleted symbols are unknown at the receiver, even a single uncorrected insertion/deletion can result in a catastrophic burst of errors. Thus, conventional error-correcting codes fail at these deletion/insertion channels.

In this dissertation, we investigate coding designs and their corresponding probabilistic decoding schemes for some typical applications associated with deletion and insertion channels: 1) Forward error correction for low power sensors in short-range wireless asynchronous communication via level crossing (LC) sampling and time encoding, where the dominant errors consist of pulse deletions and insertions, and where encoding is required to take place in an instantaneous fashion; 2) Error correction for probabilistic burst deletion channels in non-volatile storage.

Short-range wireless communication with low-power small-size sensors has been broadly applied in many areas during the last years (see, e.g., [10, 42, 12]), such as in environmental observation, biomedical, and health care monitoring. However, such applications require a wireless sensor operating in “always-on” mode, which increases the power consumption of sensors significantly. For instance, the battery of a surveilance wireless image sensor [8] needs to be replaced every two days, which limits the deployment of such sensors in the wild. Therefore, extremely power efficient small size integrated sensors being capable of providing reliable wireless links under a long battery life and with low circuit complexity are required.

In order to address the challenges of such a power-efficient sensing operation, it has been proposed to replace the classical Nyquist-based synchronous signal processing by an asynchronous sensing architecture. This is motivated by the observation that the Nyquist "sampling and quantization" approach is not always the optimal solution for recording sparse continuous-time signals [17]. For instance, if the input signal is inactive or if no changes are detected, sampling and transmission do not need to be carried out, which has the potential of significant power savings. Another advantage in asynchronous sensing is that the sensor hardware is not required to have a clock circuit [32], which produces lower electromagnetic interference [36] and thus allows for a very energy efficient and compact size hardware design. In contrast to the sensor node, it is often assumed that the base station is wall powered and thus is able to run complex algorithms. We make the same assumption in this dissertation.

Although there exists many works related to theory and implemention of asynchronous sampling, no results have been published so far addressing the difficulty in communicating asynchronous samples reliably over noisy channels to the best of our knowledge, apart from our work [16]. In [16], we have shown that the drawback of uncoded asynchronous communication lies in the fact that the channel noise may lead to symbol insertions and deletions at the receiver, thus destroying the data synchronization.

On the other hand, a large amount of work has addressed insertion and/or deletion error correction for synchronous communication systems [37, 15, 28, 13]. Recently, concatenated codes have been considered to maintain synchronization of bit streams (see, e.g., [24, 7, 9, 39]), where an outer forward error correcting code and an inner code for resynchronization are concatenated. Also, convolutional codes for insertion/deletion channels have been investigated by either extending the state space of the code or by modifying the path metric of the Viterbi decoder [29]. Note that synchronous insertion/deletion error correcting schemes from above cannot directly be applied to the asynchronous setting. For example, the fact that in asynchronous communication the information about the underlying waveform signal is mostly contained in the timing information of the transmitted signal pulses only allows to embed code redundancy via extending the modulation alphabet and not by adding extra pulses. For the same reason, the code must be necessarily systematic.

These constraints are addressed in this dissertation by proposing a novel deletion/insertion correction scheme for asynchronous communication based on a combination of a systematic convolutional code, an embedded marker code, and power-efficient frequency-shift keying (FSK) modulation at the sensor node tailored to the asynchronous setting. Note that employing a convolutional code allows for encoding in a streaming fashion with low latency and only requires a buffer length of a few bits at the sensor node. We show that burst deletions/insertions exist in asychronous communication, and by solely interleaving a short block of parity bits in combination with an iterative decoder a significant iterative gain can be obtained both in terms of symbol error probability and the expected end-to-end distortion between original and reconstructed signals. We also show that residual redundancy in the asynchronously sampled source signal can successfully be exploited for synchronization in combination with a marker code, thus providing an extremely low complexity alternative to using explicit redundancy from a channel code. We employ two different asynchronous sampling techniques in this dissertation, level crossing (LC) sampling [25, 31, 11, 38, 35, 16], and for comparison purposes, time encoding (TE), which was proposed in [19, 20]. For TE, we replace the systematic channel code encoding by a quantization for the relative position of each timing with respect to its previous and next position in time.

In asynchronous communication, we employ the encoder with an additional interleaver and employ iterative processing in the decoder to correct burst deletions and insertions. However, due to the imposed latency by this strategy, iterative processing is not an appropriate solution for other important applications as for example high-read/write-speed non-volatile memory, which also suffers from burst deletion errors. Take an emerging non-volatile memory technology called racetrack memory for example. This technology has attracted significant attention in recent years due to its promising ultra-high storage density and low power consumption [30, 34]. In racetrack memory [30], the element of information storage is called a cell. The data reading in this memory is operated by many read ports or heads. In order to read the information, each cell is shifted to its closest head, once a cell is shifted, all the other cells are also shifted in the same direction and at the same speed, which means a block of bits (i.e., a non-binary symbol) are read by multiple heads in parallel during a shift of the cells. If the cells shift more than by one single cell location, one or more cells will not be read by each head, which causes not only a single non-binary symbol deletion but also consecutive (burst) non-binary symbol deletions.

In practical systems, the maximal length of consecutive non-binary deletions is limited [47]. This process can be considered as that a non-binary symbol is transmitted through a specific channel, i.e., a non-binary deletion channel with an additional segmentation assumption [22, 40]. According to this assumption, a fixed number of consecutive transmitted non-binary symbols are considered as a block or a segment, and the number of the non-binary deletions within each segment is limited to a certain number. This channel is referred to as the non-binary segmented burst deletion channel in the following. For instance, in racetrack memory, when the cells shift by more than one single cell location, which correspond to a burst deletion event, the next burst event only appears after some time. This fact can also be referred to as intermittent errors occuring in high density magnetic channels [27].

Correcting non-binary deletions has been addressed, for example, in [6, 5]. In [6] codes which are used to perfectly locate deletions in the non-binary segment burst deletion channel were proposed, where each non-binary code symbol consists of both a non-binary de Bruijn symbol and a Maximum Distance Separable (MDS) code symbol. The non-binary de Bruijn symbols are responsible for locating the deletions perfectly if the number of deletions per segment is limited; by inserting erasures appropriately, the symbol sequence is resynchronized. After that, the resulting errors introduced in the resychronization stage are corrected by an error correcting code (e.g., an MDS code or a convolutional code). However, in order to perfectly locate deletions in the non-binary segment burst deletion channel, the code rate of these codes decreases as the maximal burst length increases because non-binary de Bruijn symbols of larger alphabet size are required. For example in [5], extra heads are added to reconstruct the data, which leads to a decreasing code rate. In [13, 14], so called Guess & Check codes are proposed which also employ MDS codes to correct non-binary i.i.d. deletions. The deletions in this approach cannot be perfectly located but the code rate is higher than the codes proposed in [6]. Nevertheless, this approach assumes that boundaries between the MDS code words are known at the receiver.

In this dissertation we relax the requirement of locating the deletions perfectly for the burst deletion channel and instead compute the likelihood for potential positions of the deletions. Instead of using non-binary de Bruijn sequences, we restrict ourselves to binary marker patterns. This increases the code rate independently of the maximal burst length and thus is advantageous for storage applications. The binary marker patterns are obtained by minimizing a simple union bound on the resulting error probability, with deletions replaced by erasures, via an exhaustive search. This search is feasible for both moderate segment and burst lengths. The resynchronized sequence is obtained via a soft-decision forward-backward algorithm adapted to a Markov model which describes the burst deletion model at hand, and soft values of the resynchronized sequence are further utilized by an outer channel code decoder. We observe that the reduced overhead of our proposed scheme only incurs a slightly increased deletion locating error probability under the soft-decision decoding compared to using non-binary marker patterns. Also, the proposed scheme provides better error correction performance than existing schemes for the same code rate.

This dissertation is organized as follows. In Chapter 2, we present error correction scheme for low power sensors in short-range wireless asynchronous communication. In Chapter 3, based on our work in Chapter 2, we provide an error correction scheme for probabilistic burst deletion channels, occuring for example in non-volatile memory settings. Finally, we provide concluding remarks in Chapter 4.

1.1 Notation

Analog waveforms are denoted as $s(t)$ with the continuous time t in the argument. Discrete-time quantities are denoted as x_{t_k}, indexed by the time instant t_k, enumerated by a nonnegative integer $k \in [0, 1, \dots]$. Vectors are boldfaced roman, as $\mathbf{x}$, and may depend on the continuous time, as in $\mathbf{x}(t)$ or a discrete time instant, as in $\mathbf{x}_{t_k}$. Binary vectors are specifically denoted by a boldfaced italics notation, as in $\boldsymbol{c}$. Matrices are denoted with upper case boldfaced letters, as $\mathbf{X}$. The probability of a random variable X with realization x is denoted as $P(x)$ or with $\Pr(\cdot)$, if we refer to the probability of an event, similarly for random vectors. $Q(\cdot)$ indicates the Q-function. Finite sets are denoted in calliographic notation, as in $\mathcal{D}$. We also indicate the length of the support of a set of non-overlapping time intervals $\mathcal{T}$ as $|\mathcal{T}|$. A finite alphabet with q elements is denoted as $\mathbb{F}_q$, and we indicate the set of all positive integers with $\mathbb{N}$ and all reals with $\mathbb{R}$, respectively. Finally, integer sets $\{a, a+1, \dots, b\}$ are denoted with $[a : b]$.

CHAPTER 2

ERROR CORRECTION SCHEMES FOR LOW POWER ASYNCHRONOUS COMMUNICATION

Short-range wireless communication with low-power small-size sensors has been broadly applied in many areas [10, 42, 12] such as in environmental observation, biomedical, and health care monitoring. However, such applications require a wireless sensor operating in "always-on" mode, which increases the power consumption of sensors significantly. Therefore, extremely power efficient small size low circuit complexity integrated sensors being capable of providing reliable wireless links are required.

In this chapter[1], we replace the classical Nyquist-based synchronous signal sampling by asynchronous sampling strategies, i.e., sampling via level crossing (LC) sampling and time encoding (TE). The main advantage is that these asynchronous sampling strategies provide a larger potential of significant power savings for recording sparse continuous-time signals [17], a smaller hardware footprint, and a lower circuit complexity compare to Nyquist-based synchronous signal sampling.

We propose a novel deletion/insertion correction scheme for asynchronous communication based on a combination of a systematic convolutional code, an embedded marker code, and power-efficient frequency-shift keying (FSK) modulation at the sensor node tailored to an asynchronous setting such as LC sampling. For TE, we replace the systematic channel code encoding by a quantization for the relative position of each timing with respect to its previous and next position in time.

---

[1] This chapter is based on [43, 44, 46].

2.1 System Model and Preliminaries

In this section, we introduce the system model for two different asynchronous communication schemes employing LC sampling [25, 31, 11, 38, 35, 16] and TE [19, 20], respectively.

2.1.1 Asynchronous Communication Model Based on Level Crossing Sampling

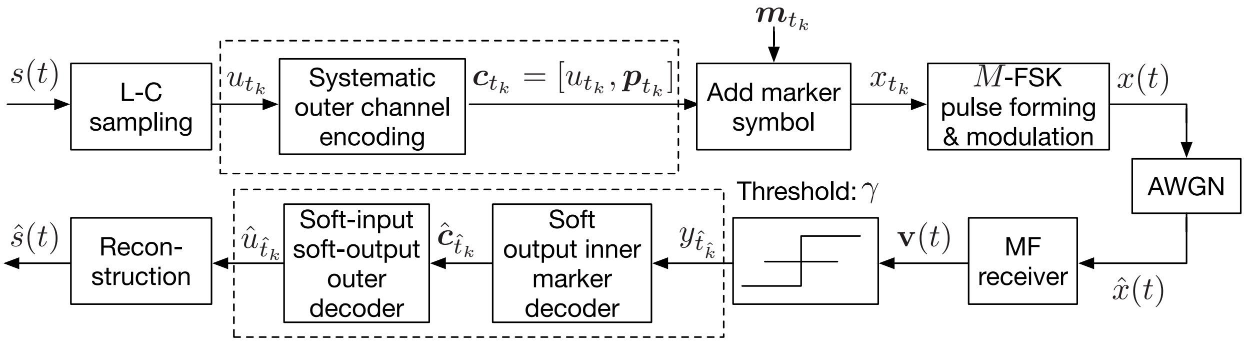


Figure 2.1 System model for LC sampling.

The system model for asynchronous communication based on LC sampling is shown in Figure 2.1. The dashed blocks in Figure 2.1 will be replaced by a modification for iterative decoding discussed in Section 2.4. In the transmitter, $s(t)$ is first sampled into the discrete-time sample/information bit u_{t_k} at time t_k, $k \in \mathbb{N}$, via LC sampling, as described in Section 2.1.3. Then, after systematic channel encoding is performed, a binary codeword $\boldsymbol{c}_{t_k} = [c_{t_k}^{(1)}, \ldots, c_{t_k}^{(n_{t_k})}] = [u_{t_k}\ \boldsymbol{p}_{t_k}]$, $c_{t_k}^{(n)} \in \mathbb{F}_2$, $n = 1, \ldots, n_{t_k}$, is generated where $\boldsymbol{p}_{t_k}$ is a binary parity sequence of length $n_{t_k} - 1$. Note that the length of the codeword n_{t_k} can in general be (periodically) time varying, i.e., it can depend on the position in the source sequence. After that, a marker sequence $\boldsymbol{m}_{t_k}$ of length j_{t_k} is added to ensure synchronization. This yields the code symbol $x_{t_k} \in \mathbb{F}_M$, $M = n_{t_k} + j_{t_k}$, by multiplexing code and marker bits for a fixed time instant t_k. Then, each x_{t_k} is subsequently modulated into a M-FSK orthogonal sinc waveform [3]. The modulated waveform $x(t)$ is obtained, which is transmitted over a continuous-input AWGN channel. Note that in order to preserve the timing information generated in the sampling process, redundancy can only be added by extending the FSK symbol alphabet, which extends the transmission bandwidth. In the system models discussed in this chapter, we employ M-FSK-modulation due to the fact that (i) FSK modulation has a simple structure, which can be implemented easily in sensor hardware; (ii) FSK is power efficient and suitable for ultra wideband radio operation (see our work in [16]).

In the receiver, the observation at the channel output $\hat{x}(t)$ is applied to a matched filter (MF) receiver. In the MF receiver, the output of the matched filters $\mathbf{v}(t) = [v_1(t), \ldots, v_M(t)]^T$, with $v_m(t)$, $m = 1, \ldots, M$, indicating the MF output for the m-th FSK frequency f_m, is compared with a pre-given threshold γ on a fine time grid by running a high frequency local clock at the receiver. Whenever this threshold is exceeded for one of the outputs $v_m(t)$, the corresponding index m is encoded in the M-ary discrete-time symbol $y_{\hat{t}_k} \in \mathbb{F}_M$ associated with an estimated time $\hat{t}_k$. The symbol $y_{\hat{t}_k}$ is then passed to the inner soft output marker decoder which ensures resynchronization (i.e., insertion and deletion error correction) to obtain the log-likelihood ratio (LLR) $L(c_{\hat{t}_k}^{(n)}) = \ln \frac{P(Y_1^R | c_{\hat{t}_k}^{(n)} = 0)}{P(Y_1^R | c_{\hat{t}_k}^{(n)} = 1)}$, where Y_1^R denotes the whole received sequence of R symbols at the output of the threshold test over all time instances. Then, $L(c_{\hat{t}_k}^{(n)})$ is passed to the outer soft input soft output (SISO) channel decoder to obtain an estimate $\hat{u}_{\hat{t}_k}$. The LC reconstruction then provides the waveform estimate $\hat{s}(t)$.

2.1.2 Asynchronous Communication Model Based on Time Encoding

The system model for TE sampling is shown in Figure 2.2. In contrast to the system model for LC sampling in Figure 2.1, the signal information is solely contained in the sign change of a rectangular signal $z(t)$, indicated by the times t_k, and not in the amplitude. Specifically, the relative position of sign change t_{k-1} of $z(t)$ between

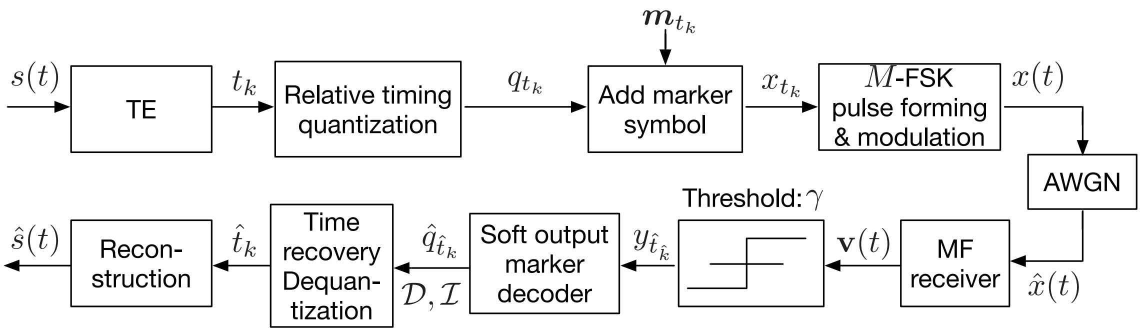


Figure 2.2 System model for TE sampling.

its previous and next sign changes, i.e., t_{k-2} and t_k, is quantized and then mapped into a W-bit non-binary symbol q_{t_k}, $W \leq \log_2 M$, to encode the signal timing, as explained in Section 2.2.1 below. As another difference to Figure 2.2, $y_{\hat{t}_{\hat{k}}}$ is first passed to the soft output marker decoder to obtain the estimate $\hat{q}_{\hat{t}_k}$, along a with a list of deleted $\mathcal{D}$ and inserted symbol positions $\mathcal{I}$ in the whole block. These positions can be determined according to the approach outlined in Section 2.2.4 below. The timing recovery block is only active, whenever the k-th transmit pulse is deleted or an additional k-th transmit pulse is inserted. In the case of a deletion, the symbol value of $\hat{q}_{\hat{t}_k}$ is dequantized to $\hat{t}_k$.

2.1.3 Asynchronous Sampling

In the following, we discuss the two above mentioned asynchronous signal acquisition schemes employed in this chapter in more detail.

Level Crossing Sampling. The LC asynchronous sampling process [17, 16, 43] at the sensor node is shown in Figure 2.3.

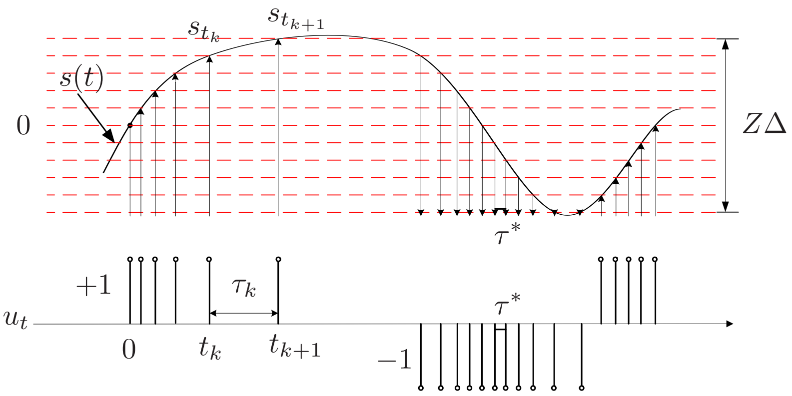


Figure 2.3 LC sampling [17, 16, 43].

In Figure 2.3, $s(t) \in \mathbb{R}$ is the real-valued amplitude of the waveform signal at time t, and $u_t \in \{+1, -1\}$ denotes the sample sequence obtained from LC sampling, containing the timing information where the samples take place. At each time, the waveform is compared with Z pre-determined uniformly distributed levels, with the decision interval defined as $\Delta = \frac{2 \max\limits_t (|s(t)|)}{Z}$. Whenever the waveform triggers a decision level in the direction of increasing amplitude a "+1" sample is recorded at that specific time; otherwise, a "-1" sample is placed. "+1" and "-1" are mapped to bits 1 and 0, respectively. In the LC sampling stage, no clock circuit is required at the sensor node compared to a traditional periodical sampling based on a "sample and quantize" operation, which significantly reduces the power consumption of the sensor node. Therefore, the timing interval τ_k for any adjacent samples t_k and t_{k+1} is a variable instead of a constant, and the smallest possible sampling interval is defined as $\min(\tau_k) \triangleq \tau^*$, which is a function of the source waveform signal, Z, and Δ. Also, if the signal is sparse, the number of samples generated by LC sampling will become significantly smaller than periodical sampling for the same duration of the signal waveform [17, 16].

Time Encoding. TE [20] represents the amplitude information $s(t)$ at the input as a time sequence $z(t)$ at its output (see Figure 2.4). The analog input signal $s(t)$ is bounded by $|s(t)| \leq \epsilon < \varphi$, and the analog output $z(t)$ assumes one of two different steady states, i.e., $+\varphi$ or $-\varphi$. Before entering the integrator, $s(t)$ is added to the feedback from the output $z(t)$, which is given by the constant $\pm\varphi$. At the beginning, the output of TE is assumed to be in the state $z(t) = -\varphi$, and the feedback is given by φ. The signal $s(t) - z(t)$ is then fed into an integrator, and the output of the integrator, i.e. $w(t)$, grows from $-\epsilon$ to ϵ. When $w(t)$ reaches the maximum value ϵ, $z(t)$ transits from $-\varphi$ to φ instantly, the output state becomes $z(t) = \varphi$ and the feedback becomes $-\varphi$, respectively.

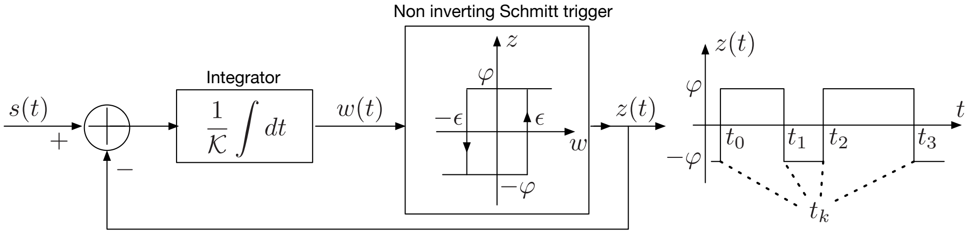


Figure 2.4 TE sampling [20].

We note that $z(t)$, which is utilized for the reconstruction of $s(t)$, contains both the amplitude value $\pm\varphi$ and the time where the amplitude of $z(t)$ transits, as shown in Figure 2.4. However, since the amplitude of $z(t)$ alternates between $-\varphi$ and φ, $z(t)$ can be represented by the times t_k only, provided that the initial amplitude value of $z(t)$ is known to the receiver. Therefore, only the real-valued timing information t_k is required in the waveform reconstruction stage.

2.1.4 Errors in Asynchronous Communication

Note that the dominant errors in the asynchronous settings in Figures 2.1 and 2.2 are deletion and insertion errors as elaborated below. To this end, let $\mathcal{T}_m^{i,+}$ be the time interval associated with the i-th out of T FSK pulses of frequency f_m in the overall transmit signal $x(t)$ and $\mathcal{T}_m^+ = \bigcup_{i=1}^{T} \mathcal{T}_m^{i,+}$ the union of these intervals. Then $\mathcal{T}^+ = \bigcup_{m=1}^{M} \mathcal{T}_m^+$ is the set of time intervals where the transmitter is active. If the source signal $s(t)$ has total duration $\mathcal{T}$, all time segments where the transmitter is not active are included in the set $\mathcal{T}^- = \mathcal{T} \backslash \mathcal{T}^+$. Likewise, $\mathcal{T}_m^-$ describes the set of non-active intervals for frequency f_m.

Insertion errors can now only happen in time segments contained in $\mathcal{T}^-$, as there may exist a channel noise realization inducing a signal $v_m(t)$ at the MF output for which $v_m(\bar{t}) > \gamma$, $m \in [1 : M]$, and at least one $\bar{t} \in \mathcal{T}^-$, i.e., even though the transmitter was not active. Likewise, deletion errors can only happen for time segments in $\mathcal{T}^+$, as even if the transmitter is active, there may exist a channel noise realization for which $v_m(t) < \gamma$ for all $t \in \mathcal{T}_m^{i,+}$, $i \in [1 : T]$, $m \in [1 : M]$. Note that we assume for the rest of the chapter that deletion and insertion error events are stationary and occur uniformly distributed for all $t \in \mathcal{T}^+$ and $t \in \mathcal{T}^-$, respectively.

For the simple case of uncoded transmission and $M = 2$ we can compute insertion and deletion error probabilities analytically as a function of the threshold γ. To elaborate, consider the amplitude values $x_m(\cdot) \in \mathbb{R}$, $m = 1, 2$, of every point in the sampled FSK transmit pulse as $\mathbf{x}_m = [x_m(t_{-\mu}), \ldots, x_m(t_{-\mu+i}), \ldots, x_m(t_0), \ldots, x_m(t_{\mu-i}), \ldots, x_m(t_\mu)]$. Here μ is an integer, which depends on the time grid employed at the receiver. Further, the FSK pulse is assumed to be even-symmetric around $x_m(t_0)$, which is true for any practical FSK pulse [3]. If we further assume that the time-domain FSK waveform is limited to τ^* seconds, the maximum value at the output of the MF is given as

$$V_m(t_0) = \frac{\tau^*}{2\mu+1} \sum_{i=0}^{2\mu} x_m(t_{-\mu+i}) \cdot x_m(t_{\mu-i}) + N = \frac{\tau^*}{2\mu+1} \sum_{i=0}^{2\mu} x_m^2(t_{\mu+i}) + N,$$

where $N \sim \mathcal{N}(0, \sigma_m^2)$. Now, as outlined above, a deletion error is caused if $v_m(t) < \gamma$ for any $t \in \mathcal{T}_m^+$. Likewise, an insertion error is obtained if $v_m(t) > \gamma$ for at least one t with $t \in \mathcal{T}_m^-$. For convenience we define a normalized threshold $\tilde{\gamma} \in [0, 1]$. Then, the deletion probability is given as

$$
\begin{aligned}
P_d &= \sum_{m=1}^{2} \Pr\left(v_m(t) < v_m(t_0)\tilde{\gamma} \,\middle|\, t \in \mathcal{T}_m^+\right) \Pr\left(t \in \mathcal{T}_m^+\right) \\
&= \sum_{m=1}^{2} Q\left(\frac{v_m(t_0)(1-\tilde{\gamma})}{\sigma_m}\right) \frac{|\mathcal{T}^+|}{|\mathcal{T}|}. \qquad (2.1)
\end{aligned}
$$

Likewise, for the insertion probability we obtain

$$
\begin{aligned}
P_i &= \sum_{m=1}^{2} \Pr\left(v_m(t) > v_m(t_0)\tilde{\gamma} \,\middle|\, t \in \mathcal{T}_m^-\right) \Pr\left(t \in \mathcal{T}_m^-\right) \\
&= \sum_{m=1}^{2} Q\left(\frac{v_m(t_0)\tilde{\gamma}}{\sigma_m}\right) \left(1 - \frac{|\mathcal{T}^+|}{|\mathcal{T}|}\right), \qquad (2.2)
\end{aligned}
$$

where $\frac{|\mathcal{T}^+|}{|\mathcal{T}|}$ is the duty cycle [2] of $x(t)$. For an example of a sparse medical signal and LC sampling with $Z = 63$ Figure 2.5 shows both deletion and insertion probability versus the AWGN channel signal-to-noise ratio (SNR) E_s/N_0 and different selected thresholds $\tilde{\gamma}$, computed from (2.1) and (2.2), respectively. Here, the noise variance after MF is given as $\sigma_m^2 = N_0/(2E_s)$ where E_s is the energy to transmit each bit and N_0 the one-sided power spectral density of the noise. We observe that, as the

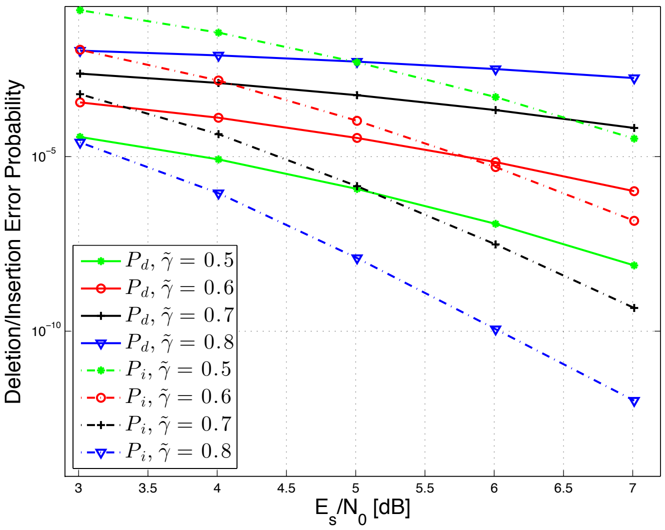


Figure 2.5 Deletion (P_d) and insertion (P_i) error probability versus AWGN channel SNR for different normalized thresholds $\tilde{\gamma}$ ($M = 2$ and LC sampling with $Z = 63$.)

threshold increases, the deletion probability increases but the insertion probability decreases, so the intersection points of these curves move left, and vice versa. Finally, we remark that even if it cannot be directly compared, this analysis nevertheless also provides guidance on suitable threshold values for the coded case with $M > 2$.

2.2 Error Correction for Asynchronous Communication

2.2.1 Code Design

For the *synchronous* communication case, in [9] the authors introduce a code construction for correcting insertion, deletion, and substitution errors, where all the redundant bits are added by introducing additional transmit symbols. Note that such a solution is not feasible in an asynchronous setting. Therefore, in this section we propose novel deletion and insertion correcting constructions with the feature that redundancy is added while maintaining the symbol timing.

Construction for the LC Sampling Scheme. In this section, we propose a novel code construction to add redundant bits for LC sampling. Figure 2.6 shows an example how asynchronously sampled information bits u_{t_k}, encoded parity bits $p_{t_k}^{(j)}$, and periodic marker bits $m_{t_k}^{(j)}$ can be arranged in a two-dimensional array to form M-ary modulation symbols, where the superscript indicates the corresponding layer $j \in [1 : L-1]$. Here, we have defined $L = \log_2 M$ for brevity. As can be seen from Figure 2.6, the mapping to FSK symbols indicated by columns in this array can be characterized in different horizontal layers, namely information, parity, mixture, and marker layers. Apart from the information layer, all other layers can in principle be absent or present in multiple quantities, providing a significant flexibility to the proposed design and a structured way to generate higher order M-FSK symbols. In this scheme, channel encoding is organized in blocks of K information bits each. During the first block of K symbols, the parity bits are set to zero because parity bits have not been generated yet. In Figure 2.6, the parity bits are denoted by red circles,

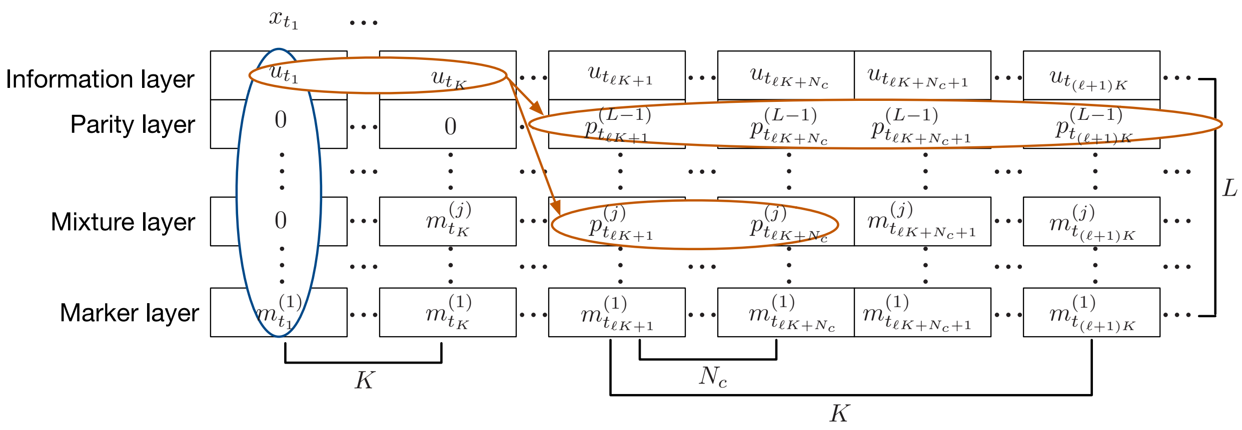


Figure 2.6 Allocation of information and redundant bits for M-FSK pulse forming with LC sampling.

where $p^{(j_{t_k})}_{t_k}$ and $p^{(L-1)}_{t_k}$ represent parity bits in a parity symbol $\boldsymbol{p}_{t_k} = [p^{(j_{t_k})}_{t_k}, \ldots, p^{(L-1)}_{t_k}]$. We here introduce a novel chaining construction for the parity bits in order to deal with the causality constraints and the demand for instantaneous transmission: Once a sensor sample is generated at time t_k, the parity bits in the corresponding block are generated causally from the information bits in the previous length-K block. In Figure 2.6, the quantity N_c represents the number of parity bits in each block in the mixture layer. The marker bits in Figure 2.6 ($m^{(1)}_{t_k}$ and $m^{(j_{t_k})}_{t_k}$ represent marker bits in a marker symbol $\boldsymbol{m}_{t_k} = [m^{(1)}_{t_k}, \ldots, m^{(j_{t_k})}_{t_k}]$ shown in Figure 2.1) are assumed to be known at the decoder and are used to resynchronize the received symbol sequence from deletion and insertion errors. Note that the parity and information bits in the same symbol x_{t_k} are only moderately dependent as they are associated with different codewords, which disperses the bit errors in each codeword and improves the error correcting performance. Since K is directly related to the memory requirements and the chip area of the sensor [16], it is required to be not too large. To this end, we employ systematic convolutional codes whose code rates are given as R_c. Also, let R_m denote the code rate associated with the marker symbols. Then, it is easy to see that the fundamental trade off between the two code rates is given as

$$R_c R_m = \frac{1}{L}. \tag{2.3}$$

Construction for the TE scheme. Different from LC sampling, where both the value and the timing of the samples are required in the waveform reconstruction stage, only the timing of the samples is relevant in TE. A similar two-dimensional mapping as in Figure 2.6 adapted to the TE case is shown in Figure 2.7, where at least one marker layer is employed for deletion/insertion error protection (see the bottom layer in Figure 2.7). Let $q_k = 1+\sum_{i=L-W+1}^{L} q_{t_k}^{(i)} 2^{i-(L-W+1)}$, $q_{t_k}^{(i)} \in \{0,1\}$, $i \in [L-W+1 : L]$, represent the symbol value of the W-bit symbols q_{t_k}, which provides the quantized relative position in time of the sample $x_{t_{k-1}}$ between its previous sample $x_{t_{k-2}}$ and the subsequent sample x_{t_k}. The details related to the calculation of the relative position of each pulse timing and its quantization will be discussed in Section 2.2.4 below.

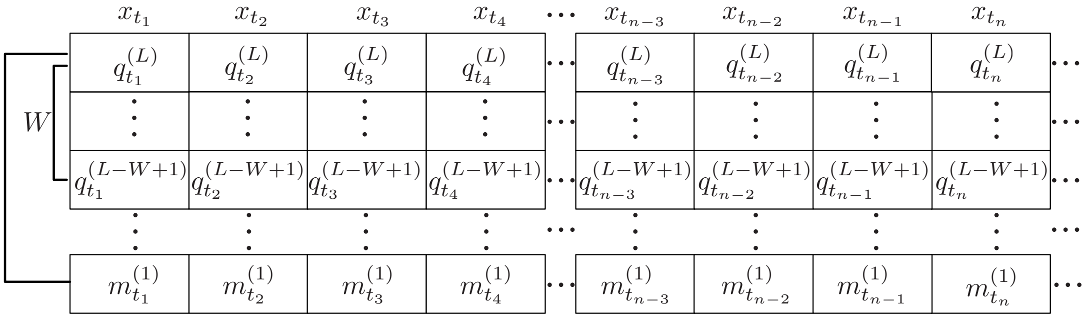


Figure 2.7 Allocation of quantized relative position and marker bits for M-FSK pulse forming with TE.

2.2.2 Generating Soft Outputs from Marker Redundancy

In this subsection, we introduce a generalized symbol-based forward-backward recursion (FBA) used to locate the deletions/insertions by computing the corresponding likelihoods. For notational brevity, in this section the time values t_k and its estimate $\hat{t}_k$ are denoted by their enumeration k. Hence, the transmitted symbol sequence of length T before modulation is defined as $X_1^T \triangleq (x_1, x_2, \ldots, x_T)$, and the received symbol sequence of length R after detection is given by $Y_1^R \triangleq (y_1, y_2, \ldots, y_R)$. Both R and T are assumed to be known at the receiver. As in the classical BCJR algorithm [1], we define $\alpha_{k,n} = P(D_{k,n}, Y_1^n)$ and $\beta_{k,n} = P(Y_{n+1}^R | D_{k,n})$, where $D_{k,n}$

denotes the event that k symbols are transmitted and n symbols are received. The change of events $D_{k-1,n-1} \Rightarrow D_{k,n}$ and $D_{k-1,n} \Rightarrow D_{k,n}$ indicates symbol transmission and deletion, respectively. A symbol insertion is represented by a change of events as $D_{k,n-1} \Rightarrow D_{k,n}$. Let P_i, P_d, and P_s denote the insertion, deletion and substitution probabilities, respectively, and $P_t = 1 - P_i - P_d$. Further, let $P(\boldsymbol{x}_k)$ denote the *a priori* probability of the transmitted symbols, where $\boldsymbol{x}_k$ is the binary vector notation associated with the M-ary symbol x_k, likewise for the received vector $\boldsymbol{y}_k$. Then, the symbol-based forward recursion [1] is given as

$$\alpha_{k,n} = \frac{P_i}{M^2}\alpha_{k,n-1} + P_d\alpha_{k-1,n} + P_t\alpha_{k-1,n-1}\sum_{\mathbf{x}_k} P(\boldsymbol{x}_k)\delta_{\boldsymbol{x}_k,\boldsymbol{y}_n}, \tag{2.4}$$

where

$$\delta_{\boldsymbol{x}_k,\boldsymbol{y}_n} = \begin{cases} 1 - P_s & \text{if marker bits in } \boldsymbol{y}_n \text{ are equivalent to the marker bits in } \boldsymbol{x}_k, \\ P_s & \text{if marker bits in } \boldsymbol{y}_n \text{ are not equivalent to the marker bits in } \boldsymbol{x}_k, \\ 1 & \text{if } \boldsymbol{x}_k \text{ only contains information and/or parity bits.} \end{cases} \tag{2.5}$$

In each time slot k, for LC sampling $\boldsymbol{x}_k$ contains j_k marker bits $\boldsymbol{m}_k = [m_k^{(1)}, ..., m_k^{(j_k)}]$, and $\bar{j}_k = \log_2(M) - j_k$ uniformly distributed binary non-marker bits $\boldsymbol{c}_k = [p_k^{(j_k+1)}, ..., p_k^{(L-1)}, u_k]$, $j_k \in [0 : L-2]$, respectively. For TE, we have $\boldsymbol{c}_k = [q_k^{(j+1)}, \ldots, q_k^{(L)}]$ with $\bar{j} = W$, $j = L - W$, independent of k. We then obtain

$$P(\boldsymbol{x}_k = [\boldsymbol{c}_k, \boldsymbol{m}_k]) = P(\boldsymbol{c}_k) = \prod_{i=1}^{\bar{j}_k} P(c_k^{(i)}) = \frac{1}{2^{\bar{j}_k}}, \tag{2.6}$$

as the marker bits are assumed to be known at the receiver. For example, consider 4-FSK: If $\boldsymbol{x}_k$ contains a single marker bit, then $\bar{j} = 1$ and $j = 1$ and we have $P(\boldsymbol{x}_k = [0,0]) = P(\boldsymbol{x}_k = [1,0]) = 0.5$ or $P(\boldsymbol{x}_k = [0,1]) = P(\boldsymbol{x}_k = [1,1]) = 0.5$. On the other hand, if $\boldsymbol{x}_k$ contains a parity bit instead of the marker bit, then $\bar{j} = 2$ and $j = 0$, which yields $P(\boldsymbol{x}_k) = P(c_k^{(1)}) \cdot P(c_k^{(2)}) = 0.25$.

The recursion for $\beta_{k,n}$ is defined similarly as in (2.4). Finally, by combining $\alpha_{k,n}$ and $\beta_{k,n}$, we obtain the soft-output of the FBA as

$$\begin{aligned} P(Y_1^R|\boldsymbol{x}_k) &= \sum_n \frac{P(D_{k,n}, Y_1^R)}{P(\boldsymbol{x}_k)} && (2.7) \\ &= \frac{1}{P(\boldsymbol{x}_k)} \sum_n \{ \overbrace{P(D_{k-1,n}, D_{k,n}, Y_1^R)}^{\text{deletion term}} && (2.8) \\ &+ \overbrace{P(D_{k-1,n-1}, D_{k,n}, Y_1^R)}^{\text{transmission term}} + \overbrace{P(D_{k,n-1}, D_{k,n}, Y_1^R)}^{\text{insertion term}} \}, && (2.9) \end{aligned}$$

which can be readily used for estimating the position of deletions/insertions as elaborated below. Also, it can be used to estimate $\boldsymbol{x}_k$ or as soft-input $L(c_{\hat{t}_k}^{(n)})$ for the outer SISO channel decoder in the LC sampling case as shown in Figure 2.1.

2.2.3 Exploiting Residual Source Redundancy

In this section, we introduce a simple data-driven model for the bits obtained by asynchronously sampling the waveform source signal. This allows us to exploit residual source redundancy in the outer decoder of the LC scheme in Figure 2.1 instead of explicit redundancy from an outer channel code. The proposed model makes use of the simple observation that for an asynchronously sampled waveform signal with a moderate choice of parameter Z the obtained data sequence contains alternating contiguous runs of the form “00…” and “11…”. We consider the most conservative case by addressing only the two-bit “runs” “00” and “11”, which leads to the four-state Markov chain depicted in Figure 2.8. Based on the transition probabilities α and β we can compute the steady state probabilities μ_i for state $S_i \in \mathcal{I}$, $\mathcal{I} = \{S_0, S_1, S_2, S_3\}$, as $\mu_0 = \frac{\bar{\alpha}\bar{\beta}}{2\bar{\alpha}\bar{\beta}+\bar{\alpha}+\bar{\beta}} = \mu_2$, $\mu_1 = \frac{\bar{\beta}}{2\bar{\alpha}\bar{\beta}+\bar{\alpha}+\bar{\beta}}$, and $\mu_3 = \frac{\bar{\alpha}}{2\bar{\alpha}\bar{\beta}+\bar{\alpha}+\bar{\beta}}$ with $\bar{\alpha} \triangleq 1-\alpha$ and $\bar{\beta} \triangleq 1-\beta$, respectively.

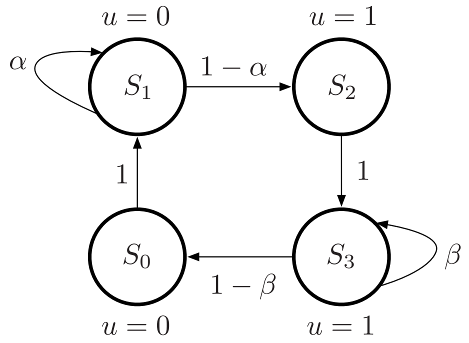


Figure 2.8 Four-state Markov chain with states $S_0, \ldots, S_3$ for the data bits obtained by LC sampling. The parameter u indicates the output bit for each state.

We now use a variant of the BCJR algorithm [1] to compute the *a posteriori* probabilities $P(\Psi_k = p|Y_1^R)$ for the state Ψ_k at time t_k, where $p, q, \Psi_k \in \mathcal{I}$, as

$$\begin{aligned} P(\Psi_k = p|Y_1^R) &= \sum_{q \in \mathcal{I}_p} P(\Psi_k = p, \Psi_{k+1} = q|Y_1^R) \\ &= \sum_{q \in \mathcal{I}_p} \alpha_k(p)\gamma_k(p,q)\beta_{k+1}(q), \end{aligned} \tag{2.10}$$

where $\mathcal{I}_p$ represents the set of all states at time t_{k+1} which are connected to state $\Psi_k = p$. We further have

$$\alpha_{k+1}(q) = \sum_{p \in \mathcal{I}} \alpha_k(p)\gamma_k(p,q), \tag{2.11}$$

with a similar definition for the $\beta_k(p)$ term in (2.10). Finally, the $\gamma_k(p,q)$-term is given as

$$\gamma_k(p,q) = P(\Psi_{k+1} = q|\Psi_k = p) \cdot \underbrace{P(Y_1^R|\Psi_k = p)}_{P(Y_1^R|\boldsymbol{x}_k)} \cdot \underbrace{P(Y_1^R|\Psi_{k+1} = q)}_{P(Y_1^R|\boldsymbol{x}_{k+1})}, \tag{2.12}$$

where $P(\Psi_{k+1} = q|\Psi_k = p)$ is the state transition probability of the Markov chain in Figure 2.8 for $p, q \in \mathcal{I}$. Further, the probabilities $P(Y_1^R|\Psi_k = p)$ and $P(Y_1^R|\Psi_{k+1} = q)$ are determined by the output of the soft output inner marker decoder (see equation (2.10)) and constitute the *a priori* input for the source BCJR algorithm.

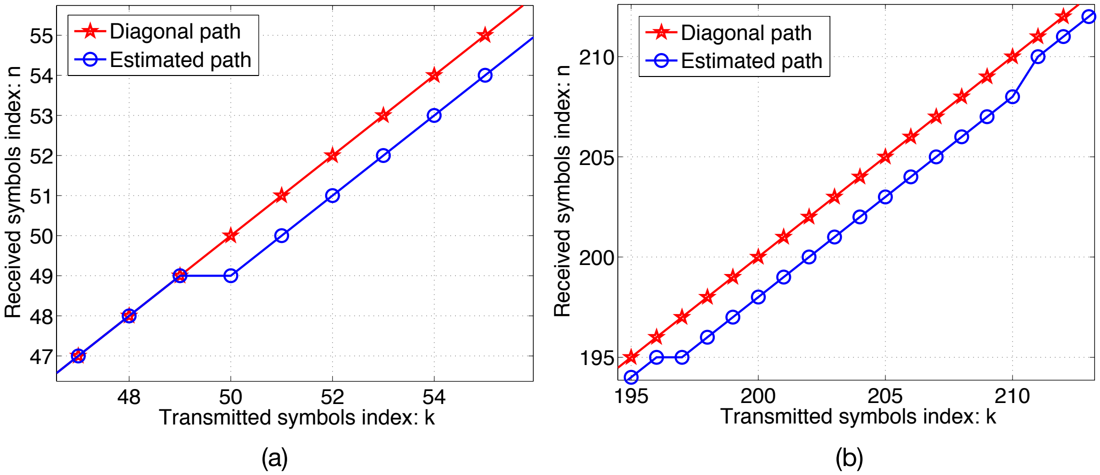


Figure 2.9 Most likely path to locate deletions and insertions. In (a), the estimated path moves downwards at transmitted position 50, indicating a deletion at this position; In (b), the estimated path moves downwards and upwards at transmitted positions 197 and 211, indicating a deletion and an insertion at these two positions, respectively.

2.2.4 Localization of Deleted and Inserted Symbols

In the following we elaborate how the soft output at the marker decoder can be employed to localize the position of deleted and/or inserted FSK pulses by visualizing the evolution of transmitted and received symbols on a two-dimensional grid of dimension $T \times R$. Examples are shown in Figure 2.9. A diagonal path with $n(k) = k$ means that no insertions or deletions have occurred during transmission. In contrast, for synchronization errors, the path deviates from the diagonal line, where insertion errors cause the path to move upwards, and deletion errors downwards, respectively. For example, in Figure 2.9(a), there is a deletion at transmitted position 50, and Figure 2.9(b) shows both a deletion and an insertion at transmitted positions 197 and 211, respectively.

By leveraging the results from the symbol-based FBA in equation (2.7), this path can be estimated by obtaining the most likely grid point $n(k)$ at position k belonging to this path as

$$n(k) = \arg\max_{n} \big\{ P(D_{k,n}, Y_1^R) \big\}. \tag{2.13}$$

Note that $P(D_{k,n}, Y_1^R)$ is given by the numerator on the right-hand side of equation (2.7). For insertion errors, we can remove the extra pulses directly where they are located. However, the exact position in time of a deleted pulse between its neighbor pulses is unknown and cannot be recovered at the receiver by the above mentioned approach. In the following, workarounds to fix the timing of deleted symbols for both LC sampling and TE are introduced.

Deletion Timing Localization for LC Sampling. In LC sampling, even though the employed marker bits may correctly estimate the position of the deleted pulse, the exact timing of the deleted pulse is unknown. As a remedy, we place a deleted pulse at the midpoint between its previous and next sample. Note that this is the optimal solution if the pulse position is uniformly distributed between the neighboring samples.

Deletion Timing Localization for TE. Let the relative position of timing of the sample $x_{t_{k-1}}$ between its previous sample $x_{t_{k-2}}$ and next sample x_{t_k} be given as

$$\nu_k = \frac{t_{k-1} - t_{k-2}}{t_k - t_{k-2}}, \tag{2.14}$$

where $t_{k-2} < t_{k-1} < t_k$, and $0 < \nu_k < 1$. The value ν_k is then quantized to q_k (see Section 2.2.1) via Lloyds-Max quantization [23] according to the distribution of ν_k, where each q_k corresponds to one of 2^W-level optimum quantization values. If a symbol $x_{t_{k-1}}$ is deleted but $x_{t_{k-2}}$ and x_{t_k} are intact, the quantization value $\hat{q}_k$ and its corresponding dequantized value $\hat{\nu}_k$ are obtained in the decoder. Then, an estimated timing of $x_{t_{k-1}}$, i.e., $\hat{t}_{k-1}$, can be recovered according to (2.14) as $\hat{t}_{k-1} = \hat{\nu}_k(t_k - t_{k-2}) + t_{k-2}$. It is easy to see that we always have $t_{k-2} < \hat{t}_{k-1} < t_k$.

2.3 Simulation Results

In this section, a 100 ms excerpt of a recorded mouse heart beat signal is employed as the test signal to evaluate the proposed error-correction strategy. This signal represents a typical sparse waveform signal. In order to carry out the simulations below on a computer, the analog signal is first sampled with 20 kHz and converted to a discrete-time signal. Then it is upsampled to a sampling frequency of 20 MHz via Whittaker–Shannon interpolation [26] to model a "quasi-analog" source signal in order to run numerical simulations.

We have considered two different setups, which are shown in Table 2.3 and only differ on the number of quantization levels Z and the total number of samples obtained from asynchronous sampling N_s. For the numerical simulations below we fix a normalized threshold of $\tilde{\gamma} = 0.6$ after MF filtering for all FSK frequencies, as this, according to the considerations in Section II-E, provides a good compromise with respect to error obtained from both insertion and deletion events.

Table 2.1 Overview of the Simulation Parameters.

	LC sampling scheme		TE scheme	
	Setup 1	Setup 2	Setup 1	Setup 2
Z	63	23	–	–
N_s	996	340	1002	336
Duty cycle of $x(t)$: $\frac{\lvert\mathcal{T}^+\rvert}{\lvert\mathcal{T}\rvert}$	0.1389	0.3893	0.1397	0.3847
Source signal bandwidth:	7.5 kHz	7.5 kHz	7.5 kHz	7.5 kHz
Nyquist sampling rate:	15 kHz	15 kHz	15kHz	15 kHz
Effective average sampling rate after asynchronous sampling	9.96 kHz	3.4 kHz	10.02 kHz	3.36 kHz

The source signal has a length of 100 ms.

2.3.1 Performance for LC Sampling

For LC sampling, we employ an outer systematic convolutional code and an inner marker code as shown in Figure 2.6. For the mixture layer filled with both parity and marker bits (see Figure 2.6), the marker bits appear at times $t_{\ell K+N_c+1}, \ldots, t_{(\ell+1)K}$, $\ell \in \mathbb{N}$, periodically, and the minimum length of the periodical marker bits which is able to effectively locate single non-consecutive deletions/insertions is two, i.e., $K - N_c = 2$. We first consider the case of 4-FSK, where the binary periodic pair of marker bits in the mixture layer is fixed as $[m_{t_{\ell K+N_c+1}}, m_{t_{(\ell+1)K}}] = [0, 1]$, $\ell \in \mathbb{N}$. The corresponding allocation scheme is shown in Figure 2.10.

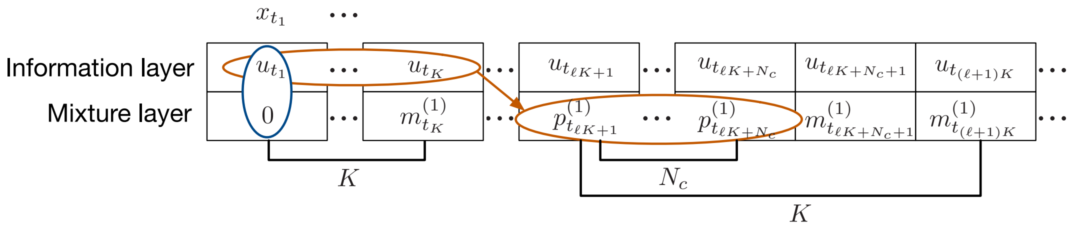


Figure 2.10 LC sampling: allocation of information and redundant bits for 4-FSK pulse forming.

For 8-FSK pulse forming, an additional marker layer is added to provide a more accurate estimated position of deletions/insertions, as shown in Figure 2.11.

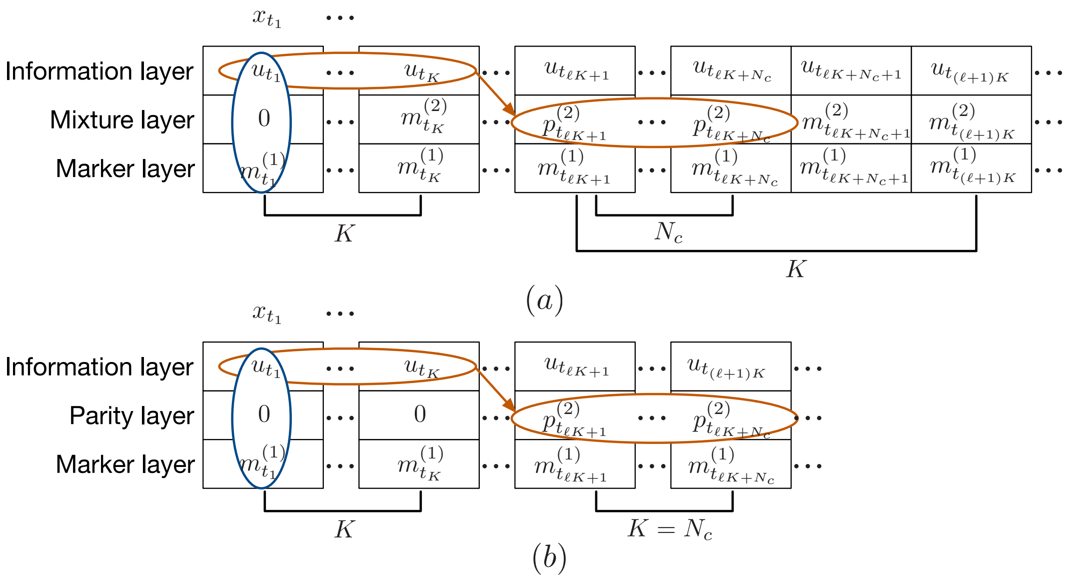


Figure 2.11 LC sampling: allocation of information and redundant bits for 8-FSK pulse forming. (a) Case $K > N_c$, (b) case $K = N_c$.

Table 2.2 Code Rates and Puncturing Schemes for the Employed Optimum Systematic Convolutional Codes [21] and the Corresponding Marker Code Rates for 4-FSK and 8-FSK Modulation.

R_0	Generators $\mathbf{G}(D)$	Puncturing matrix	R_c	K	N_c	R_m $(Q=4)$	R_m $(Q=8)$
$\frac{2}{3}$	$\begin{bmatrix} 1 & 0 & \frac{D^2+D+1}{D^3+D^2+D+1} \\ 0 & 1 & \frac{D^3+D+1}{D^3+D^2+D+1} \end{bmatrix}$	—	$\frac{2}{3}$	4	2	$\frac{3}{4}$	$\frac{3}{6}$
$\frac{1}{2}$	$\begin{bmatrix} 1 & D^3+D+1 \end{bmatrix}$	—	$\frac{1}{2}$	2	2	1	$\frac{2}{3}$
$\frac{1}{2}$	$\begin{bmatrix} 1 & D^3+D+1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$	$\frac{5}{9}$	10	8	$\frac{9}{10}$	$\frac{3}{5}$

Table 2.2 shows the employed outer optimum convolutional codes [21] for different values of N_c along with the corresponding mother code rates R_0, punctured

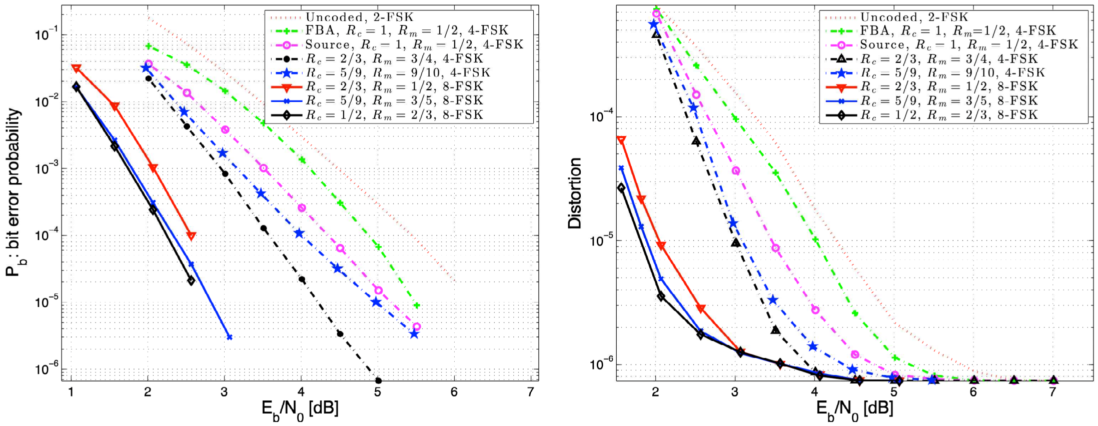
rates R_c, and marker code rates R_m. In order to preserve the timing information of the pulses, only parity bits are punctured. Also, by enumerating all possible puncturing patterns, the employed puncturing patterns are selected such that the combined deletion, insertion, and substitution error probabilities are minimized. For the sake of simplicity, only the performances for 2-FSK, 4-FSK, and 8-FSK modulation are compared. For 8-FSK pulse forming, information and redundant bits for punctured convolutional codes with rate of $R_c = 2/3$, $R_c = 5/9$ are allocated as shown in Figure 2.11(a), and for $R_c = 1/2$ are allocated as shown in Figure 2.11(b), respectively. For the numerical results shown below, we employ Whittaker–Shannon interpolation [26] in the waveform reconstruction stage at the receiver.

The residual signal distortion after reconstruction is given as the mean square error (MSE)

$$D = \frac{1}{|\mathcal{T}|} \int_0^{|\mathcal{T}|} |s(t) - \hat{s}(t)|^2 dt. \tag{2.15}$$

Note that this distortion depends on Z, the number of pre-determined uniformly distributed amplitude levels for the LC sampling (see Figure 2.3), and also on how accurately the sample times t_k are recovered.

For Setup 1 and LC sampling, we compare the total residual bit error probability $P_b = \Pr(u_{t_k} \neq \hat{u}_{t_k})$ versus the AWGN channel SNR with $E_b = E_s R_c R_m$, over all insertion, deletion, and substitution errors for different amounts of marker redundancy N_c, given the outer concatenated codes shown in Table 2.2. The results are displayed in Figure 2.12(a). The total code rates $R_c R_m$ are listed according to equation (2.3).



(a) (b)

Figure 2.12 Numerical results for Setup 1 and LC sampling, the codes displayed in Table 2.2, and a MF threshold of $\tilde{\gamma} = 0.6$. The scheme with $R_c = 1$ uses residual source redundancy and explicit redundancy solely from a marker code. For all 4-FSK schemes, the total rate is given as $R = \frac{1}{2}$, for all 8-FSK schemes as $\frac{1}{3}$ according to (2.3). (a) Total bit error probability versus channel SNR, (b) MSE distortion versus channel SNR.

Note that these comparisons ignore errors due to incorrect localization of the position of deleted and inserted pulses, so they can be interpreted as a genie-aided approach where the exact pulse times are perfectly known. Howerver, they still provide a reasonable assessment for the performance of the schemes considered in this chapter.

Figure 2.12(a) also shows the performance for a scheme based on the ideas in Section 2.2.3, in which all explicit redundancy is constituted by a marker code (i.e., $R_m = 1/2$, $R_c = 1$) and where source redundancy instead of explicit redundancy is used to fix the value of the information bits after deletion/insertion inner decoding. Using the LC sampling source model in Figure 2.8 with $Z = 63$ levels we obtain for the employed test signal the values $\alpha = 0.9533$ and $\beta = 0.9537$. The entropy rate of this source can be computed as $H(\mathcal{X}) = -\sum_{i \in \mathcal{I}} \mu_i \sum_{j \in \mathcal{I}} p_{ij} \log_2 p_{ij} = 0.1798$ bits with the following transition probabilities p_{ij} between the states S_i and S_j: $p_{01} = p_{23} = 1$, $p_{11} = \alpha$, $p_{12} = \bar{\alpha}$, $p_{33} = \beta$, $p_{30} = \bar{\beta}$, all other p_{ij} are zero.

We observe from Figure 2.12(a) that increasing marker redundancy also leads to a smaller total error probability if a convolutional code is used to clean up the residual errors after insertion/deletion decoding. However, the scheme based on source redundancy suffers from a higher error probability even if it uses $R_c = 1$. This is due to the fact that despite the entropy rate induced by the source redundancy is low, this does not provide any advantage in minimum distance compared to the uncoded case (i.e., we have $d_{\min} = 1$). This is also supported by the observation that for $R_c < 1$ additionally exploiting source redundancy after channel decoding does not provide a significant further gain in performance. We can also see from Figure 2.12(a) that solely employing source redundancy provides a significant gain compared to just using the inner marker code and a hard decision after the output of the inner FBA decoder. Nevertheless, for 8-FSK modulation, a smaller R_c provides a better overall error correcting performance. This is due to the fact that the additional marker layer of the pulse forming scheme in Figure 2.11 already provides sufficient protection for the information bits from synchronization errors. Therefore, the overall error correcting performance gain for 8-FSK is dominated by convolutional codes with lower code rates.

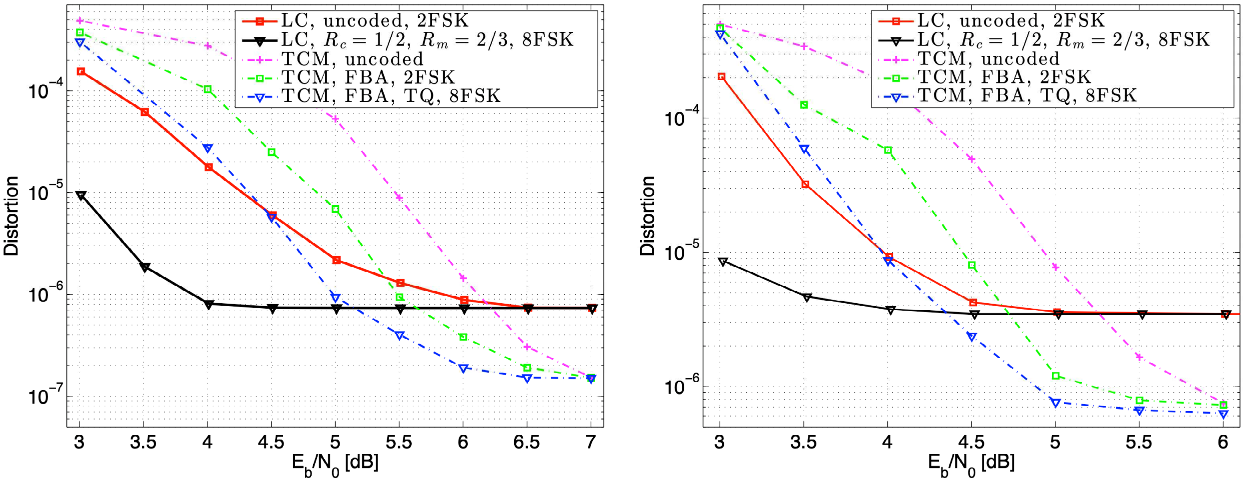
The resulting end-to-end MSE distortion (2.15) is shown in Figure 2.12(b), which now also includes errors due to inaccurate localization and repositioning of deleted and inserted transmit pulses. We observe that all the schemes using explicit redundancy have similar performance, whereas the scheme using implicit source redundancy suffers from a performance penalty of around 0.5 dB in SNR for small to moderate channel SNRs. Also, solely employing a marker code without any additional explicit or implicit redundancy incurs an additional penalty of 0.5 dB for moderate SNRs.

For Setup 2, similar observations as for Setup 1 with respect to the error correcting performance can be made (see also Figure 2.13 below).

2.3.2 Distortion Comparison between TE and LC Sampling

We now compare the MSE distortion performance of TE with that of LC sampling. For the sake of brevity, only the performances of 2-FSK and 8-FSK modulation are discussed. For case of 2-FSK in TE, each symbol x_{t_k} (see Figure 2.7) becomes a single marker bit with the pattern $[m_{t_{2k-1}}, m_{t_{2k}}] = [0, 1]$, $k \in \mathbb{N}$. For 8-FSK, $W = 2$ quantization bits are added with each marker bit to form a symbol $x_{t_k} \in \mathbb{F}_8$ which is able to provide a more accurate position estimate for symbol deletions.

We roughly fix the same average sampling rate for both LC sampling and TE and compare the distortion for these two schemes under different SNR regimes. The resulting MSE distortion versus SNR results are shown in Figure 2.13 for both Setup 1 and 2.



(*a*) (*b*)

Figure 2.13 MSE distortion comparison between TE and LC sampling schemes versus channel SNR ($\tilde{\gamma} = 0.6$). (a) Setup 1, (b) Setup 2.

For the TE scheme, the uncoded scheme provides the worst distortion performance. In the 2-FSK modulation scheme marker bits/codes are employed to locate the deletions/insertions, placing the pulses in the estimated deleted positions and removing the located insertions, which reduces the distortion of the reconstructed signal. For 8-FSK, the added redundancy for the pulse timing provides a more accurate position estimate for each deletion. In the low SNR regime, LC sampling introduces a smaller distortion than TE because LC sampling encodes the sample information in *both* pulse value and timing. As a consequence, LC sampling is more robust to noise than TE. For instance, if the noise corrupts the pulse timing in TE, its associated sample value is corrupted, while this is not necessarily the case in LC sampling. When the SNR increases, the distortion decreases as expected, and eventually becomes independent of SNR. The resulting MSE distortion is due to the sub-Nyquist average sampling rate in our work [20] (see Table 2.3) and depends on Z for LC sampling and φ, ϵ for TE, respectively.

2.4 Deletion/Insertion Error Correction for LC Sampling Based on Iterative Processing

In this section, we first show how burst deletions/insertions take place in asynchronous communication. Then we propose to extend our error correction results outlined above by introducing iterative processing at the decoder to improve the error correction performance. We show that by solely interleaving a short block of parity bits in combination with an iterative decoder a significant iterative gain can be obtained both in terms of symbol error probability and the expected end-to-end distortion between original and reconstructed signals.

Note that any time position $t \in \mathcal{T}_m^-$ (see Section 2.1.4) may trigger the threshold when random noise exists. Thus, it is possible that there will be more than one insertion between the adjacent pulses, thus creating burst insertion errors. Likewise, it is possible that no time position $t \in \mathcal{T}_m^+$ at adjacent pulses triggers the threshold which causes burst deletion errors.

For example, for the continuous-time recorded mouse heart beat signal employed in Section 2.3, Figure 2.14 shows the length distribution of consecutive insertion errors at an SNR of 2dB, where approximately 35% of the insertion errors are burst errors.

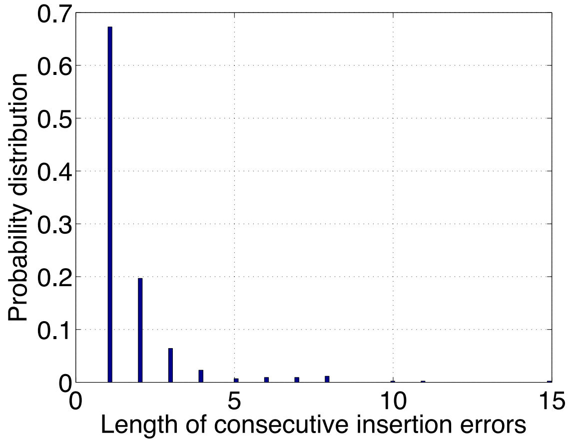


Figure 2.14 Length distribution of consecutive insertion errors.

In order to combat burst insertion/deletion errors, a random interleaver π is now employed shown as in Figure 2.15.

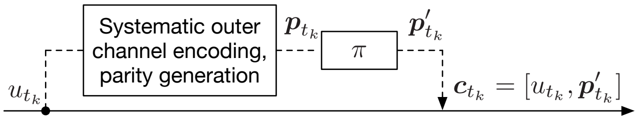


Figure 2.15 Encoding with interleaver, which replaces the dashed block in the encoder of Figure 2.1.

which replace the dashed block in encoder of Figure 2.1. In the encoder, the parity sequence within the layer shown as red circle in Figure 2.6 are permuted by π whose interleaving pattern is known to the receiver.

$$\pi(\boldsymbol{p}_{k+1}, \boldsymbol{p}_{k+2}, ..., \boldsymbol{p}_{k+L_\pi}) = (\boldsymbol{p}'_{k+1}, \boldsymbol{p}'_{k+2}, ..., \boldsymbol{p}'_{k+L_\pi}) \tag{2.16}$$

The length of interleaver denoted as L_π is kept moderate due to the limited size of the sensor chip storage and given transmission delay constraints. Since the order of the information bits and therefore the timing information must be preserved, and since the marker bits appear periodically and are known to the receiver, only the parity bits are interleaved.

The iterative decoder setup employed in this work is shown in Figure 2.16, which replaces the dashed block in the decoder of Figure 2.1.

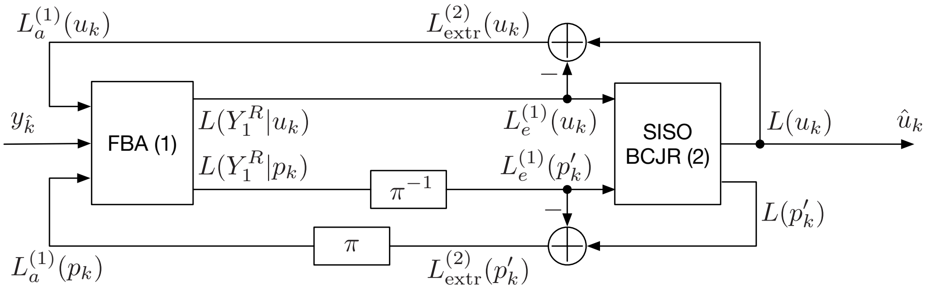


Figure 2.16 Iterative decoding, which replaces the dashed block in the decoder of Figure 2.1.

In iterative decoding for serially concatenated codes [4], the extrinsic information of both information bits and parity bits are computed and exchanged between inner codes and outer codes. Here, we exchange the extrinsic information of both information and parity bits between inner marker codes and outer convolutional codes. The output L-values of the information bits from the FBA are fed directly to the input of the soft-input soft-output (SISO) BCJR decoder for the outer convolutional code. The output L-values of the parity bits from the FBA are deinterleaved before being processed by the BCJR. The extrinsic information of information bits and parity bits from the BCJR decoder are obtained as

$$L_{\text{extr}}^{(2)}(u_k) = L(u_k) - L_e^{(1)}(u_k), \tag{2.17}$$

$$L_{\text{extr}}^{(2)}(p'_k) = L(p'_k) - L_e^{(1)}(p'_k), \tag{2.18}$$

where

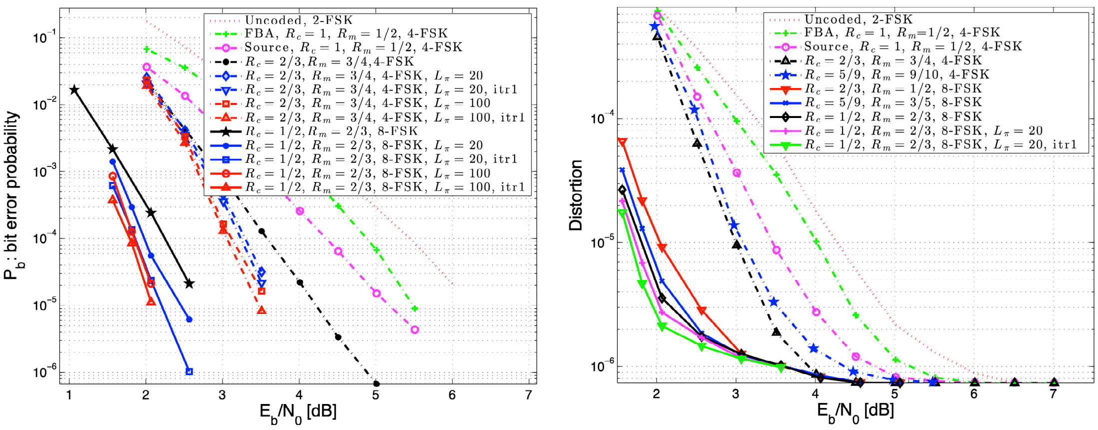
$$L_e^{(1)}(u_k = 0/1) = \log \frac{P(Y_1^R|u_k = 0/1)}{P(Y_1^R|u_k = 1/0)}, \tag{2.19}$$

$$L_e^{(1)}(p'_k = 0/1) = \log \frac{P(Y_1^R | p'_k = 0/1)}{P(Y_1^R | p'_k = 1/0)}, \tag{2.20}$$

which feeds back as a priori information to the input of the FBA to update $P(\boldsymbol{x}_k)$ in (2.4).

2.4.1 Performance for LC Sampling under Iterative Decoding

For Setup 1 and LC sampling, we also compare the total residual bit error probability and end-to-end MSE distortion under the iterative decoding for different outer convolutional codes and interleaver sizes of $L_\pi = 20$ and $L_\pi = 100$, resp., for both 4-FSK and 8-FSK modulation, as shown in Figure 2.17.



(a) (b)

Figure 2.17 Iterative decoding results for Setup 1 and LC sampling, the codes displayed in Table 2.2, and a MF threshold of $\tilde{\gamma} = 0.6$. "itr1" denotes the iterative decoding with one iteration. For all 4-FSK schemes, the total rate is given as $R = \frac{1}{2}$, for all 8-FSK schemes as $\frac{1}{3}$ according to (2.3). (a) Total bit error probability versus channel SNR, (b) MSE distortion versus channel SNR.

Convolutional codes with the best overall error correcting performance under 4-FSK and 8-FSK in Figure 2.17(a) are investigated, respectively. The overall performance for 8-FSK improves on 4-FSK because a larger amount of redundancy is employed. We can also see that a moderate increase of the interleaver size from $L_\pi = 20$ to $L_\pi = 100$ improves the error correcting performance as burst insertions are dispersed to a larger extent. Due to the small interleaver size, only one decoding iteration is employed; we have observed diminishing gains in further iterations.

Figure 2.17(b) shows the expected end-to-end MSE distortion versus the SNR, where similar observations as in Figure 2.17(a) can be made. In contrast to Figure 2.12(a) and Figure 2.17(a), the distortion includes errors due to incorrect localization of the position of inserted and deleted pulses. Again, we can see that employing iterative decoding along with interleaving of parity bits also reduces the distortion.

CHAPTER 3

PROBABILISTIC BURST DELETION ERROR CORRECTION

In this chapter[2], we will investigate a probabilistic burst deletion correction scheme for low-latency non-volatile memory. We consider racetrack memory as an example, which offers high read/write speed but is prone to deletion errors.

Note that in the asynchronous communication setting in the previous chapter, we employ interleaving at the encoder and iterative processing at the decoder to correct burst deletions/insertions. However, due to the imposed latency of this settings, iterative procssing is not advisable in low-latency settings such as in high-read/write-speed non-volatile memory.

Racetrack memory (see Figure 3.1) consists of vertical columns of a magnetic material, connected and continuous like a racetrack, standing over a silicon wafer. Magnetic cells (a single blue or red unit in Figure 3.1 is called a cell) are used to store binary information in tall columns of magnetic material arranged perpendicularly on the surface of a silicon wafer.

---

[2] This chapter is based on [45].

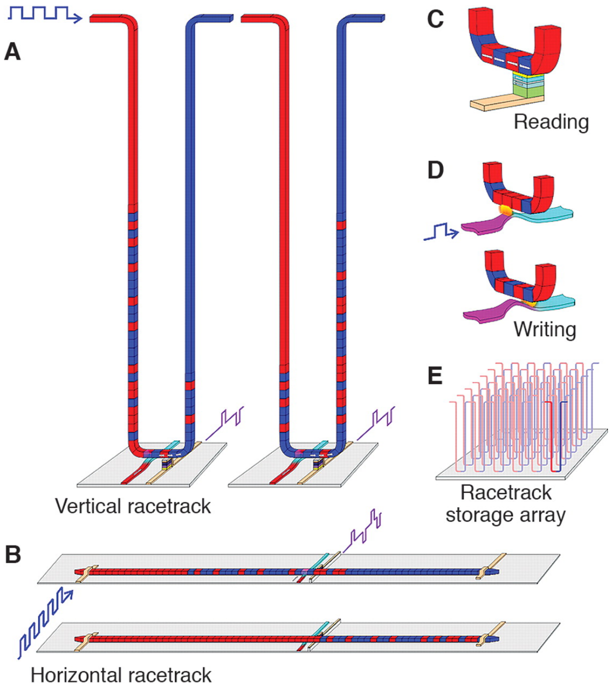


Figure 3.1 Racetrack memory [30]

The working principle of racetrack memory is given as follows: When a spin polarized current is passed through this racetrack, all cells move in the same direction along the racetrack. Data reading and writing elements can be fixed at suitable positions along the racetrack, to intersect each data bits during the motion of cells.

In order to read the information, multiple cells shift to its closest head in the same direction and at the same speed, which means a block of bits (i.e., a non-binary symbol) are read by multiple heads in parallel during a shift of the cells. If the cells shift more than by one single cell location, it causes consecutive (burst) non-binary symbol deletions.

3.1 System Model

The system model for a burst deletion locating scheme is shown in Figure 3.2.

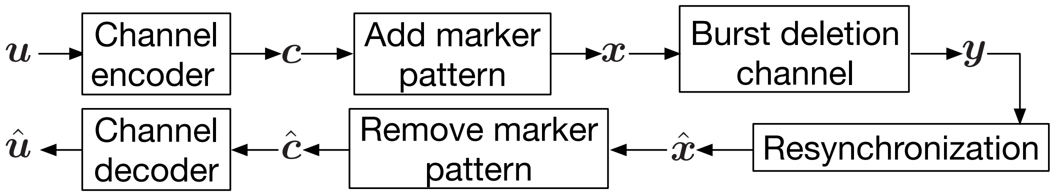


Figure 3.2 System model.

At the encoder, each information sequence $\boldsymbol{u} = [u_1, \cdots, u_K]$ with information symbols $u_\ell \in \mathbb{F}_{q_u}$, $\ell \in \{1, \cdots, K\}$ is encoded by an (N, K) block code of rate $R = K/N$ into a code sequence $\boldsymbol{c} = [c_1, \cdots, c_N]$ with code symbols $c_i \in \mathbb{F}_{q_c}$, $i \in \{1, \cdots, N\}$. Then each $\boldsymbol{c}$ is encoded by a sequence of q_m-ary marker symbols $\boldsymbol{\pi}_b = [\pi_1, \cdots, \pi_N]$, $\pi_i \in \mathbb{F}_{q_m}$. This results in a total symbol sequence $\boldsymbol{x} = [x_1, \cdots, x_N]$ via symbol alphabet size extension as shown in Figure 3.3, with symbols $x_i \in \mathbb{F}_{q_x}$, $q_x = q_c \cdot q_m$. This block of N symbols $\boldsymbol{x}$ is repeated λ times, $\lambda \in \mathbb{N}^+$, leading to a total block length of $T = \lambda N$ symbols. The final marker pattern $\boldsymbol{\pi}$ is now constructed by appending not necessarily identical patterns $\boldsymbol{\pi}_b$, i.e., $\boldsymbol{\pi} = [\boldsymbol{\pi}_{b_1}, \cdots, \boldsymbol{\pi}_{b_\lambda}]$.

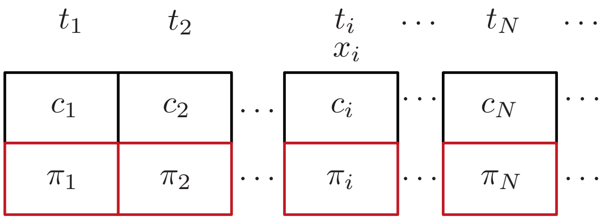


Figure 3.3 Encoding procedure.

In racetrack memory[30], for example, each x_i can be considered as a block of bits read by $\log_2(q_x)$ heads in parallel at time slot t_i. The symbol π_i is chosen as a marker or a de Bruijn symbol [6] and is assumed to be known at the receiver to ensure synchronization. Then, x is transmitted over a probabilistic non-binary burst deletion channel, where the mis-synchronized symbol y is observed at the output of the channel (see Figure 3.2).

We describe the action of the probabilistic maxburst-B non-binary deletion channel via the Markov chain depicted in Figure 3.4.

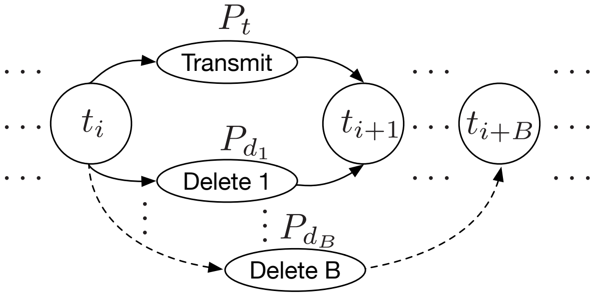


Figure 3.4 Markov chain associated with the probabilistic maxburst-B non-binary deletion channel.

At each channel use, one state transition occurs, where each state indicates one specific time instant at time i. With probability P_{d_b}, where $b \in \{1, ..., B\}$, the transmitted symbols between state t_i and t_{i+b} are deleted, and with probability $P_t = 1 - \sum_{b=1}^{B} P_{d_b}$, the current symbol is transmitted.

The estimated code sequence $\hat{\boldsymbol{x}}$ is generated by a soft-decision decoder (see below), and $\hat{\boldsymbol{c}}$ is obtained after removing $\boldsymbol{\pi}$ from each $\hat{\boldsymbol{x}}$. After decoding from $\hat{\boldsymbol{c}}$, $\hat{\boldsymbol{u}}$ is obtained.

3.2 Optimized Binary Marker Patterns for the Probabilistic Non-Binary Segmented Burst Deletion Channels

Note that in some practical magnetic recording systems such as in racetrack memory[47], the maximal length of the non-binary burst deletions observed from at the output of the probabilistic non-binary burst deletion channel is limited to a certain number with high probability. In this dissertation, we focus on this practical scenario and fix the maximum number of non-binary deletions within a sliding segment as B.

Definition 1. *Let a burst symbol deletion event of length b happen with probability P_{d_b}. In a probabilistic L-segment maxburst-B non-binary deletion channel at most a single burst symbol deletion event of length b exists in a block of L consecutive non-binary symbols, where $b \in \{1, ..., B\}$, $B \geq 2$, and $L \geq B+1$.*

In this section, optimized binary marker patterns for the probabilistic L-segment maxburst-B non-binary deletion channel are introduced. At first, let us briefly introduce de Bruijn sequences which can locate deletions perfectly in this channel.

Definition 2. *[6] A q-ary sequence $\boldsymbol{\pi} = (\pi_1, ..., \pi_\ell)$ is called a δ-bounded de Bruijn sequence of strength h if all δ consecutive length-h sub-vectors, i.e., $[\pi_i, \pi_{i+1}, ...\pi_{i+h-1}]$, $[\pi_{i+1}, \pi_{i+2}, \cdots, \pi_{i+h}]$, $\cdots$, $[\pi_{i+\delta-1}, \pi_{i+\delta}, \cdots, \pi_{i+h+\delta-2}]$ are distinct.*

Example 3.2.1. Let $\ell = 2q$ and $\delta \leq q$, then $\boldsymbol{u} = (0, 1, ..., q-1, 0, 1, ..., q-1)$, is a q-ary δ-bounded de Bruijn sequence of length ℓ, strength $h = 1$.

Example 3.2.2. Let $\boldsymbol{u} = (00110011...)$ and $\boldsymbol{v} = (011011...)$, then $\boldsymbol{u}$ and $\boldsymbol{v}$ are 2-ary $\delta = 4$ and $\delta = 3$ bounded de Bruijn sequence of strength $h = 2$, respectively.

Let us denote q_m-ary δ-bounded de Bruijn sequences of strength h as BDB(q_m, δ, h). It is shown in [6] that $\boldsymbol{\pi} \in$ BDB$(B, B, 1)$ (i.e., $B = q_m = \delta$), where $B \geq 4$, is able to perfectly locate non-binary deletions as long as the total number of deletions within a segment of length B does not exceed $B - 1$. For instance, for $B = 4$ the sequence BDB(4,4,1) can perfectly locate all deletions up to a burst length of 3. Consider now $q_m = 2$ for any $B \geq 2$ and $L \geq B + 1$. In order to perfectly locate single deletions, the bits of different value within the sequence $\boldsymbol{\pi}$ must appear alternatively, i.e., $\boldsymbol{\pi} = (0101...)$ or $\boldsymbol{\pi} = (1010...)$. Such a pattern is unable to locate any burst-$2n$ deletions, where $n \in \mathbb{N}$ and $2n \leq B$. Therefore, the binary $\boldsymbol{\pi}$ which is able to perfectly locate *all types* of burst deletions in the probabilistic L-segment maxburst-B deletion channel does not exist.

As outlined previously, we focus on $q_m = 2$ and propose a guided exhaustive search algorithm to find an optimized binary marker pattern for the probabilistic L-segment maxburst-B deletion channel.

To this end, define the following parameters:

- $\boldsymbol{\Pi}$: whole set of binary sequence of length T, $\boldsymbol{\pi} \in \boldsymbol{\Pi}$.
- run: consecutive bits with the same value.

- $R : \boldsymbol{\Pi} \mapsto \mathbb{N}$: $R(\boldsymbol{\pi})$ is the length of the longest run in $\boldsymbol{\pi}$.
- P_{d_b}: probability of a burst-b deletion event.
- $f(b, L, \boldsymbol{\pi})$: for a *binary* marker pattern $\boldsymbol{\pi}$, the maximal number of errors generated in the resynchronization stage (where deletions are replaced by erasures) associated with any single burst-b deletion event.

$f(b, L, \boldsymbol{\pi})$ is obtained by the following procedure: it is assumed that a single burst-b deletion takes place in the i-th position of the marker pattern $\boldsymbol{\pi}$, i.e., b consecutive bits $(\pi_i, ..., \pi_{i+b-1})$ are deleted, where $i \in \{1, 2, ..., T-b\}$. In the resynchronization stage, the likely positions of these consecutively deleted bits are estimated by the feature of the specific $\boldsymbol{\pi}$ and replaced by erasures. The number of resulting errors generated in the resynchronization stage is counted, and the largest one is denoted as $f(b, L, \boldsymbol{\pi}, i)$. Finally, we obtain $f(b, L, \boldsymbol{\pi}) \triangleq \max_i (f(b, L, \boldsymbol{\pi}, i))$ via an exhaustive search.

Then, for any binary marker pattern, we obtain the following simple union bound on the error probability.

Proposition 1. *In a probabilistic L-segmented maxburst-B non-binary deletion channel, any given binary marker pattern* $\boldsymbol{\pi}$ *after the resynchronization stage has an error probability upper bounded by* $R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} \frac{f(b,L,\boldsymbol{\pi})}{b} P_{d_b}$.

Proof. The number of segments corrupted by burst-b deletion events in the whole non-binary sequence is given as $\frac{TP_{d_b}}{b}$. Then, the total maximal number of errors generated by the burst-b deletion events, where $b \in \{1, \cdots B\}$, in the resynchronization stage amounts to $\sum_{b=1}^{B} \frac{TP_{d_b}}{b} f(b, L, \boldsymbol{\pi})$ by the union bound. Note that for $b = 1$, we have $f(1, L, \boldsymbol{\pi}) = R(\boldsymbol{\pi})$, thus $\sum_{b=1}^{B} \frac{TP_{d_b}}{b} f(b, L, \boldsymbol{\pi}) = TP_{d_1} R(\boldsymbol{\pi}) + \sum_{b=2}^{B} \frac{TP_{d_b}}{b} f(b, L, \boldsymbol{\pi})$, leading to the claimed upper bound of $R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} f(b, L, \boldsymbol{\pi}) \frac{P_{d_b}}{b}$.

□

The optimized binary marker pattern is given as $\boldsymbol{\pi}^* \triangleq \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \left(R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} f(b, L, \boldsymbol{\pi}) \frac{P_{d_b}}{b} \right)$. The algorithm to find such a pattern is given in the following

Algorithm 1 Find the optimized binary marker pattern $\boldsymbol{\pi}^*$ in the probabilistic L-segment maxburst-B deletion channel

```
 Input: $\boldsymbol{\Pi}$
 Output: $\boldsymbol{\pi}^*$
while $1 \le j < T$ do
  search $\boldsymbol{\pi}$ from whole set of binary sequences of length $T$, i.e. $\boldsymbol{\Pi}$, such that $R(\boldsymbol{\pi}) = j$
  $UB \Leftarrow R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} T\frac{P_{d_b}}{b}$
  while $2 \le b \le B$ do
    while $i \le T - b$ do
      $f(b, L, \boldsymbol{\pi}) \Leftarrow f(b, L, \boldsymbol{\pi}, 1)$
      if $f(b, L, \boldsymbol{\pi}) < f(b, L, \boldsymbol{\pi}, i)$ then
        $f(b, L, \boldsymbol{\pi}) \Leftarrow f(b, L, \boldsymbol{\pi}, i)$
      end if
      $i$++
    end while
    $b$++
  end while
  if $R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} f(b, L, \boldsymbol{\pi})\frac{P_{d_b}}{b} < UB$ then
    $UB \Leftarrow R(\boldsymbol{\pi})P_{d_1} + \sum_{b=2}^{B} f(b, L, \boldsymbol{\pi})\frac{P_{d_b}}{b}$
    $\boldsymbol{\pi}^* \Leftarrow \boldsymbol{\pi}$
  end if
  $j$++
end while
```

Example 3.2.3. Let $B = 2$, $L = 3$, $P_{d_1} > 0$ and $P_{d_2} > 0$. We determine $\boldsymbol{\pi}$ from the complete set of binary sequence of length T, i.e. $\boldsymbol{\Pi}$, such that $R(\boldsymbol{\pi}) = 1$. This means that only the runs of length 1 exist in $\boldsymbol{\pi}$, thus the bits of different value appear alternatively in $\boldsymbol{\pi}$. The maximal number of errors generated in the resynchronization stage, when $\boldsymbol{\pi}$ is corrupted by a single burst-2 deletion event, is $f(2, 3, \boldsymbol{\pi}) = T$. We have $\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}, R(\boldsymbol{\pi})=1} \left(R(\boldsymbol{\pi})P_{d_1} + f(2, 3, \boldsymbol{\pi})\frac{P_{d_2}}{2} \right) = P_{d_1} + \frac{T}{2}P_{d_2}$. Now, let us search for a pattern $\boldsymbol{\pi}$ from the set $\boldsymbol{\Pi}$ such that $R(\boldsymbol{\pi}) = 2$. If $\boldsymbol{\pi}$ only contains runs of length 2, $\boldsymbol{\pi}$ is either $\boldsymbol{\pi} = (00110011...)$ or $\boldsymbol{\pi} = (11001100...)$. Assume that a burst-2 deletion event happens in the positions where the values of these deleted bits in $\boldsymbol{\pi}$ are not the same, e.g., the corrupted $\boldsymbol{\pi}$ is $\hat{\boldsymbol{\pi}} = (001\text{xx}011...)$ or $\hat{\boldsymbol{\pi}} = (110\text{xx}100...)$, where x indicates a deleted symbol. Then it is possible that the decoder wrongly decodes the pattern $\hat{\boldsymbol{\pi}}$ into $\tilde{\boldsymbol{\pi}} = (00\text{x}10\text{x}11...)$ or $\tilde{\boldsymbol{\pi}} = (11\text{x}01\text{x}00...)$, where 4 errors are generated. So we have $f(2,3,\boldsymbol{\pi}) = 4$. If $\boldsymbol{\pi}$ contains runs of both length 1 and length 2, let $\boldsymbol{\pi} = (01100110110011...)$, yielding $f(2,3,\boldsymbol{\pi}) = 4$; let $\boldsymbol{\pi} = (011011...)$ we get $f(2,3,\boldsymbol{\pi}) = 3$, etc.. We search for an overall $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and calculate their corresponding $f(2,3,\boldsymbol{\pi})$ exhaustively, and find $f(2,3,\boldsymbol{\pi}) \geq 3$. Thus, we have $\min\limits_{\boldsymbol{\pi}\in\boldsymbol{\Pi}, R(\boldsymbol{\pi})=2} \left(R(\boldsymbol{\pi})P_{d_1} + f(2,3,\boldsymbol{\pi})\frac{P_{d_2}}{2} \right) = 2P_{d_1} + \frac{3}{2}P_{d_2}$.

By continuing searching for $\boldsymbol{\pi}$ from $\boldsymbol{\Pi}$ such that $R(\boldsymbol{\pi}) \geq 3$, we have $f(2,3,\boldsymbol{\pi}) \geq 3$, thus $\min\limits_{\boldsymbol{\pi}\in\boldsymbol{\Pi}, R(\boldsymbol{\pi})\geq 3} \left(R(\boldsymbol{\pi})P_{d_1} + f(2,3,\boldsymbol{\pi})\frac{P_{d_2}}{2} \right) > \min\limits_{\boldsymbol{\pi}\in\boldsymbol{\Pi}, R(\boldsymbol{\pi})=2} \left(R(\boldsymbol{\pi})P_{d_1} + f(2,3,\boldsymbol{\pi})\frac{P_{d_2}}{2} \right) = 2P_{d_1} + \frac{3}{2}P_{d_2}$.

In summary, the optimized binary marker patterns $\boldsymbol{\pi}$ for a probabilistic 3-segment maxburst-2 non-binary deletion channel are those ones with an error probability upper bounded by $\min\left(P_{d_1} + \frac{T}{2}P_{d_2}, 2P_{d_1} + \frac{3}{2}P_{d_2}\right)$. If $2P_{d_1} + \frac{3}{2}P_{d_2} > P_{d_1} + \frac{T}{2}P_{d_2}$, i.e., $\frac{P_{d_1}}{P_{d_2}} > \frac{T-3}{2}$, then for $T \gg 3$ single deletions happen with much larger probability than burst-2 deletions. Thus, the optimized binary marker patterns are the ones with an error probability upper bounded by $P_{d_1} + \frac{T}{2}P_{d_2}$, i.e., $\boldsymbol{\pi} = (0101...)$ or $\boldsymbol{\pi} = (1010...)$. $\square$

3.3 Soft-Decision Decoding

In this section, we employ a soft-decision resynchronization scheme to locate the burst deletions, i.e., a forward-backward recursion algorithm (FBA) corresponding to the Markov chain depicted in Figure 3.4.

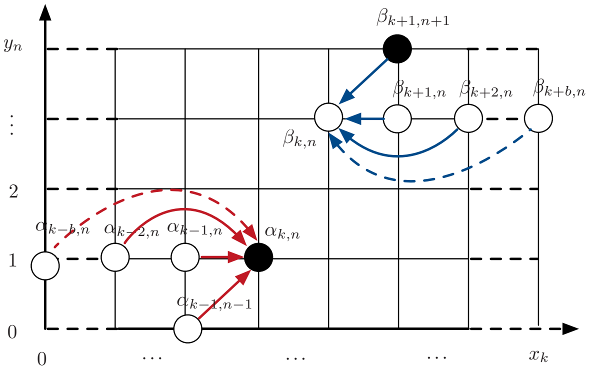


Figure 3.5 Corresponding probabilities for the FBA associated with the Markov model in Figure 3.4.

Figure 3.5 shows the corresponding probabilities for the underlying FBA associated with the model in Figure 3.4. In Figure 3.5, x_k and y_n denote the transmitted and the received symbols, respectively, where $k \in \{1, 2, ..., T\}$ and $n \in \{1, 2, ..., R_e\}$ are the index of T transmitted and R_e received symbols. Define the code sequence and the mis-synchronized sequence after the channel as $X_1^T \triangleq (x_1, x_2, ..., x_T)$ and $Y_1^{R_e} \triangleq (y_1, y_2, ..., y_{R_e})$, respectively. The equations α and β are the forward and backward probabilities in FBA given as

$$\alpha_{k,n} = P(Y_1^n, D_{k,n}), \tag{3.1}$$

$$\beta_{k,n} = P(Y_{n+1}^{R_e} | D_{k,n}), \tag{3.2}$$

where $D_{k,n}$ denotes the event k symbols are transmitted and n symbols are received. The α and β terms can be calculated by means of the following forward and backward recursions,

$$\alpha_{k,n} = \sum_{b=1}^{B} P_{d_b} \alpha_{k-b,n} + P_t \alpha_{k-1,n-1} \delta_{\boldsymbol{x}_k, \boldsymbol{y}_n}, \tag{3.3}$$

$$\beta_{k,n} = \sum_{b=1}^{B} P_{d_b} \beta_{k+b,n} + P_t \beta_{k+1,n+1} \delta_{\boldsymbol{x}_k, \boldsymbol{y}_n}, \tag{3.4}$$

where $\boldsymbol{x}_k$ is the binary vector notation associated with the q_x-ary symbol x_k, likewise for the received vector $\boldsymbol{y}_k$, $P_t = 1 - \sum_{b=1}^{B} P_{d_b}$, and

$$\delta_{\boldsymbol{x}_k,\boldsymbol{y}_n} = \begin{cases} 1 & \text{if marker in } \boldsymbol{y}_n = \text{marker in } \boldsymbol{x}_k; \\ 0 & \text{otherwise} \end{cases}. \tag{3.5}$$

We can compute the target conditional probability as

$$\begin{aligned} P(Y_1^{R_e}|x_k) = \frac{1}{P(x_k)}\Big\{ &\sum_{b=1}^{B}\sum_{n=1}^{R_e} P_{d_b}\alpha_{k-b,n}\beta_{k,n} \\ &+ P_t \sum_{n=1}^{R_e} \alpha_{k-1,n-1}\beta_{k,n}\delta_{\boldsymbol{x}_k,\boldsymbol{y}_n}\Big\}. \end{aligned} \tag{3.6}$$

After that, the posterior probability $P(x_k|Y_1^{R_e})$ is calculated, and $\hat{x}_k$ is obtained via a maximum a posteriori estimation.

3.4 Numerical Results

Some experimental results show that the typical maximal length of the burst deletions is two in practical systems[41, 27, 40, 47]. Therefore, we set $B = 2$ and $L \geq B + 1$. Here we select an arbitrary $L = 12$, and compare the deletion location performance (see Figure 3.6) for different decoding schemes and choices of marker patterns in the probabilistic 12-segment maxburst-2 non-binary deletion channel. The performance of the non-binary burst deletion locating schemes is evaluated by the error probability $P_s \triangleq Pr(c_i \neq \hat{c}_i)$ (see Figure 3.3) associated with replacing deletions by erasures in the resynchronization stage. The probability of the single and burst-2 deletion events, i.e., P_{d_1} and P_{d_2} are shown on the horizontal axis. For the sake of simplicity, we set $P_{d_1} = P_{d_2}$ in Figure 3.6. The optimized binary marker pattern $\boldsymbol{\pi} \in \text{BDB}(2, 4, 2)$ and the simplest de Bruijn sequence of field size 4, i.e., $\boldsymbol{\pi} \in \text{BDB}(4, 4, 1)$, are employed for comparison. First, we set $q_c = 8$, $q_m = 2$ (see Figure 3.3), so the rate of the marker code is $R_m \triangleq \frac{\log q_c}{\log(q_m q_c)} = \frac{3}{4}$. It is obvious that for the same optimized binary marker pattern the proposed soft-decision decoding scheme (red curve) provides a gain

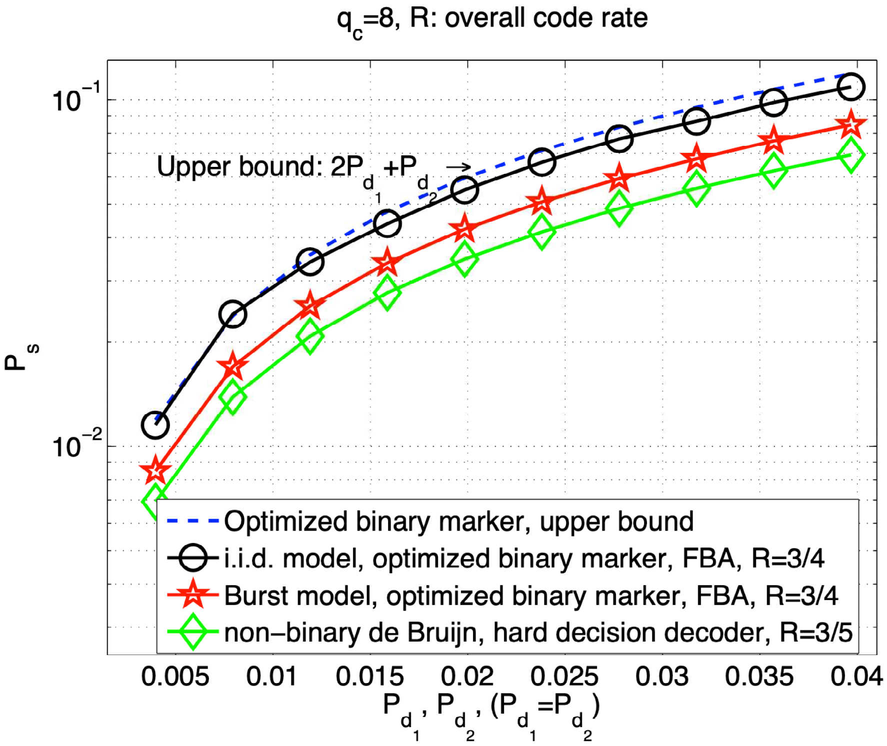


Figure 3.6 Error probability after the resynchronization stage versus deletion error probabilities P_{d_1}, P_{d_2} with $P_{d_1} = P_{d_2}$ of different coding schemes for the burst deletion channel with $B = 2$, $L = 12$.

over the soft-decision decoding scheme in [44] (black curve) which addresses all the deletions as i.i.d. This is due to the fact that the proposed decoder correctly utilizes a priori information from the probabilistic 12-segment maxburst-2 non-binary deletion channel. The upper bound for the optimized binary marker pattern is obtained from algorithm 1 as $2P_{d_1} + P_{d_2}$ (blue curve). We then set $q_c = 8$, $q_m = 4$, and employ a 4-ary de Bruijn sequence as the marker pattern, so the code rate becomes $R_m = \frac{3}{5}$. Although this sequence can perfectly locate *all types* of burst deletions in the probabilistic 12-segment maxburst-2 non-binary deletion channel at hand, the located deletions are still replaced by erasures in the resynchronization stage. We observe that the proposed scheme with binary marker sequences in general incurs only a minor loss in performance compared to the approach based on non-binary de Bruijn sequences [6] (green curve) while providing a higher rate.

Now we concatenate the proposed marker code with an outer MDS code or convolutional code which corrects errors generated in the resynchronization stage. We employ soft decoders for both codes [18], [1]. For an (n, k) convolutional code, the encoding parameters stated in section 3.1 amount to $N = T(\lambda = 1)$, $K = T$, $q_u = 2^k$, $q_c = 2^n$. The bit error probability is defined as $P_b \triangleq Pr(u_{\ell,w} \neq \hat{u}_{\ell,w})$, where $u_{\ell,w} \in \mathbb{F}_2$ is the w-th bit in the binary representation of the information symbol $u_\ell \in \mathbb{F}_{q_u}$. We set $q_c = 8$, and choose both an MDS(6,4) code with the rate $R_c = \frac{4}{6} = \frac{2}{3}$ and minimum distance 3, and a binary convolutional code of rate $R_c = \frac{2}{3}$ and free distance $d_{free} = 4$ (denoted as CC(3,2)) with the following generator matrix [21],

$$\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & \frac{D^2+D+1}{D^3+D^2+D+1} \\ 0 & 1 & \frac{D^3+D+1}{D^3+D^2+D+1} \end{bmatrix}, \tag{3.7}$$

respectively.

We set $B = 2$, $L = 3$, and let $P_{d_1} = P_{d_2}$, where the optimized binary marker pattern becomes $\boldsymbol{\pi} \in \text{BDB}(2, 3, 2)$. For the optimized binary marker code with $q_m = 2$ we obtain an overall code rate of $R \triangleq R_m R_c = \frac{1}{2}$. As we observe from Figure 3.7, the proposed soft-decision decoding scheme for the outer convolutional code CC(3,2) (red curve) provides a moderate gain over the soft-decision decoding scheme in [44] for the CC(3,2) code (black curve). We also consider the same 4-ary non-binary de Bruijn sequence from Figure 3.6 as the inner marker code. Note that in this case the number of errors exceeds the error correcting capabilities of both the MDS(6,4) and the CC(3,2) codes and leads to a non-zero bit error probability in Figure 3.7. For the selected channel at hand the CC(3,2) outperforms the MDS code due to its larger distance. Note that the concatenated constructions using the BDB(4,4,1) code outperform the constructions using a binary marker pattern, in particular for the CC(3,2) outer code, at the expense of a smaller overall code rate of $R = \frac{2}{5}$ compared to $R = \frac{1}{2}$ for the binary case. Note that in practice, the proposed coding scheme

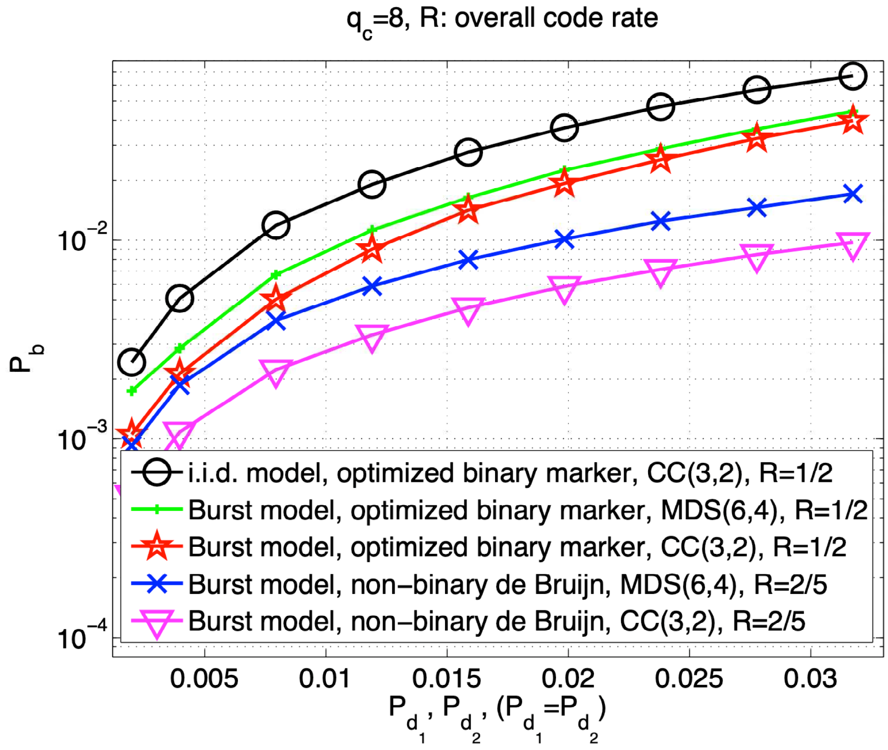


Figure 3.7 Bit error probability versus deletion error probabilities P_{d_1}, P_{d_2} with $P_{d_1} = P_{d_2}$ for different coding schemes for the burst deletion channel with $B = 2$, $L = 3$.

needs to be concatenated with an outer error correcting codes to obtain the very low error probabilities required for storage applications.

CHAPTER 4

CONCLUSION

In this dissertation, we have investigated code designs and their corresponding probabilistic decoding schemes for both low power sensors in short-range wireless asynchronous communication and probabilistic non-binary segmented burst deletion channels in racetrack memory.

In Chapter 2, two marker-based error-correction schemes, namely LC sampling and TE, have been proposed as asynchronous sampling solutions by adding redundant bits via a modulation alphabet extension. The proposed scheme for LC sampling comprises a combination of a systematic outer convolutional code and an embedded inner marker code at the sensor node. Simulation results have shown that using 8-FSK modulation symbols provides a performance gain of over 1 dB in SNR compared to 4-FSK. If the number of marker bits is not large enough to protect every information bit (e.g., not all transmitted FSK symbols contain marker bits for 4-FSK), increasing the number of marker bits provides a better error correcting performance and a smaller end-to-end distortion. In contrast, if a sufficient number of marker bits is already employed (e.g., every transmitted FSK symbol contains a marker bit as in the proposed 8-FSK scheme), the overall performance is dominated by convolutional codes with lower code rates. Also, iterative decoding with a small number of iterations in combination with interleaving the parity bits further reduces both the number of residual substitution errors and the end-to-end distortion. In TE, each FSK symbol contains a marker bit for locating deletion/insertion errors, combined with quantization indices for the relative position of the pulse timing. In the low SNR regime, LC sampling combined with concatenated marker and channel codes outperforms TE due to its larger robustness to noise than TE. In the high SNR regime however, TE performs better than LC sampling because the recovery from time-encoded samples is able to offset the residual distortion introduced by the varying sampling rate in LC sampling.

In Chapter 3, we have proposed a scheme to approximately locate the deletions in the probabilistic L-segment maxburst-B non-binary deletion channel, where optimized binary marker patterns of small field size are employed. We optimize the binary marker pattern by an exhaustive search based on an union bound type upper bound for the error probability. Further, a soft-decision decoding scheme which correctly utilizes a priori information from this channel is proposed. Simulations show that the reduced overhead of our proposed scheme for small deletion error probabilities only incurs a slightly increased probability of error in locating the deletions, compared to using non-binary marker patterns.

REFERENCES

[1] L. R. Bahl and J. Cocke and F. Jelinek and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, pages 284–287, 1974.

[2] S. F. Barrett and D. J. Pack. Microcontrollers fundamentals for engineers and scientists. *Synthesis Lectures on Digital Circuits and Systems*, 1(1):1–124, 2005.

[3] J. R. Barry, E. A. Lee, and D. G. Messerschmitt. *Digital Communication*. Springer Science & Business Media, 2012.

[4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding. *IEEE Transactions on Information Theory*, 44(3):909–926, 1998.

[5] Y. M. Chee, H. M. Kiah, A. Vardy, and E. Yaakobi et al. Coding for racetrack memories. *IEEE Transactions on Information Theory*, 64(11):7094–7112, 2018.

[6] Y. M. Chee, H. M. Kiah, E. Vardy, K. Van Vu, and E. Yaakobi. Codes correcting limited-shift errors in racetrack memories. In *IEEE International Symposium on Information Theory (ISIT)*, pages 96–100, Vail, CO, November, 2018.

[7] J. Chen, M. Mitzenmacher, C. Ng, and N. Varnica. Concatenated codes for deletion channels. In *IEEE International Symposium on Information Theory*, pages 218–218, Yokohama, Japan, June 2003.

[8] S. Chen, W. Tang, X. Zhang, and E. Culurciello. A 64x64 pixels UWB wireless temporal-difference digital image sensor. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(12):2232–2240, 2011.

[9] F. Wang and D. Fertonani and T. M. Duman. Symbol-level synchronization and LDPC code design for insertion/deletion channels. *IEEE Transactions on Communications*, 59(5):1287–1297, 2011.

[10] J. Foerster, E. Green, S. Somayazulu, and D. Leeper. Ultra-wideband technology for short- or medium-range wireless communications. *Intel Technology Journal*, 5(2):1–11, May, 2001.

[11] K. Guan and A. C. Singer. A level-crossing sampling scheme for non-bandlimited signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing.*, volume 3, pages III–III, Toulouse, France, May, 2006.

[12] V. C. Gungor and G. P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Trans. on Industrial Electronics*, 56(10):4258 –4265, October, 2009.

[13] S. K. Hanna and S. E. Rouayheb. Guess and check codes for deletions and synchronization. In *IEEE International Symposium on Information Theory*, pages 2693–2697, Aachen, Germany, June, 2017.

[14] S. K. Hanna and S. E. Rouayheb. Guess and check codes for deletions, insertions, and synchronization. *IEEE Transactions on Information Theory*, 65(1):3–15, 2018.

[15] A. S. J. Helberg and H. C. Ferreira. On multiple insertion/deletion correcting codes. *IEEE Transactions on Information Theory*, 48(1):305–308, 2002.

[16] Q. Hu, C. Yi, J. Kliewer, and W. Tang. Asynchronous communication for wireless sensors using ultra wideband impulse radio. In *IEEE 58th International Midwest Symposium on Circuits and Systems*, pages 1–4, Fort Collins, CO, August 2015.

[17] J. A. Tropp and J. N. Laska and M. F. Duarte and J. K. Romberg and R. G. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Transactions on Information Theory*, 56(1):520–544, 2010.

[18] R. Koetter and A. Vardy. Algebraic soft-decision decoding of reed-solomon codes. *IEEE Transactions on Information Theory*, 49(11):2809–2825, 2003.

[19] A. A. Lazar and L. T. Tóth. Time encoding and perfect recovery of bandlimited signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages VI–709, HongKong, China, April, 2003.

[20] A. A. Lazar and L. T. Tóth. Perfect recovery and sensitivity analysis of time encoded bandlimited signals. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(10):2060–2073, 2004.

[21] S. Lin and D. J. Costello. *Error Control Coding.* Pearson Education, 2004.

[22] Z. M. Liu and M. Mitzenmacher. Codes for deletion and insertion channels with segmented errors. *IEEE Transactions on Information Theory*, 56(1):224–232, 2010.

[23] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[24] M. C. Davey and D. J. MacKay. Reliable communication over channels with insertions, deletions, and substitutions. *IEEE Transactions on Information Theory*, 47(2):687–698, 2001.

[25] J. Mark and T. Todd. A nonuniform sampling approach to data compression. *IEEE Transactions on Communications*, 29(1):24–32, 1981.

[26] R. Marks. *Introduction to Shannon Sampling and Interpolation Theory.* Springer Science & Business Media, 2012.

[27] A. Mazumdar and A. Barg. Channels with intermittent errors. In *IEEE International Symposium on Information Theory*, pages 1753–1757, St. Petersburg, Russia, August, 2011.

[28] L. McAven and R. Safavi-Naini. Classification of the deletion correcting capabilities of reed–solomon codes of dimension 2 over prime fields. *IEEE Transactions on Information Theory*, 53(6):2280–2294, 2007.

[29] H. Mercier and V. K. Bhargava. Convolutional codes for channels with deletion errors. In *IEEE 11th Canadian Workshop on Information Theory*, pages 136–139, Ottawa, Canada, May, 2009.

[30] S. S. Parkin, M. Hayashi, and L. Thomas. Magnetic domain-wall racetrack memory. *Science*, 320(5873):190–194, 2008.

[31] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan. A level-crossing sampling scheme for A/D conversion. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(4):335–339, 1996.

[32] B. Schell and Y. Tsividis. A continuous-time ADC/DSP/DAC system with no clock and with activity-dependent power dissipation. *IEEE Journal of Solid-State Circuits*, 43(11):2472–2481, November, 2008.

[33] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[34] Z. Y. Sun, W. Q. Wu, and H. Li. Cross-layer racetrack memory design for ultra high density and low power consumption. In *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, Austin, TX, June, 2013.

[35] W. Tang, A. Osman, D. Kim, and B. Goldstein. Continuous time level crossing sampling ADC for bio-potential recording systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(6):1407–1418, 2013.

[36] Y. Tsividis. Event-driven data acquisition and digital signal processing—a tutorial. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(8):577–581, 2010.

[37] R. R. Varshamov and G. M. Tenenholtz. Correction code for single asymmetric error. *Automation and Remote Control*, 26:286–290, 1965.

[38] C. Vezyrtzis and Y. Tsividis. Processing of signals using level-crossing sampling. In *IEEE International Symposium on Circuits and Systems.*, pages 2293–2296, Taipei, Taiwan, May, 2009.

[39] F. Wang, D. Aktas, and T. M. Duman. On capacity and coding for segmented deletion channels. In *49th Annual Allerton Conference on Communication, Control, and Computing*, pages 1408–1413, Monticello, IL, September 2011.

[40] F. Wang, T. M. Duman, and D. Aktas. Capacity bounds and concatenated codes over segmented deletion channels. *IEEE Transactions on Communications*, 61(3):852–864, 2013.

[41] R. L. White, R. Newt, and R. F. W. Pease. Patterned media: a viable route to 50 Gbit/in/sup 2/and up for magnetic recording? *IEEE Transactions on Magnetics*, 33(1):990–995, 1997.

[42] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2):102 –124, May, 2008.

[43] C. Yi and J. Kliewer. Error correction for asynchronous communication. In *IEEE 9th International Symposium on Turbo Codes and Iterative Information Processing*, pages 310–314, Brest, France, September, 2016.

[44] C. Yi and J. Kliewer. On error correction for asynchronous communication based on iterative processing. In *IEEE 50th Asilomar Conference on Signals, Systems and Computers*, pages 240–244, Pacific Grove, CA, November, 2016.

[45] C. Yi and J. Kliewer. Correcting deletions in probabilistic non-binary segmented burst deletion channels. In *IEEE 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1349–1353, Pacific Grove, CA, November, 2019.

[46] C. Yi and J. Kliewer. Error correction for low power sensors in asynchronous communication. *submitted to Signal Processing*, 2020.

[47] C. Zhang, G. Sun, X. Zhang, W. Zhang, W. Zhao, T. Wang, Y. Liang, Y. Liu, Y. Wang, and J. Shu. Hi-fi playback: Tolerating position errors in shift operations of racetrack memory. In *ACM SIGARCH Computer Architecture News*, volume 43, pages 694–706, 2015.