Distributed Lossy Source Coding Using BCH-DFT Codes

Mojtaba Vaezi

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

February 2014

A dissertation submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Distributed source coding, separate encoding (compression) and joint decoding of statistically dependent sources, arises in an increasing number of applications like sensor networks and multiview video coding. Many of those applications are highly interactive, requiring the development of low-delay, energy-limited communication and computing schemes. Currently, this compression is performed by using capacity-approaching binary channel codes. As a natural extension, distributed lossy source coding is realized by cascading a quantizer and Slepian-Wolf coding in the binary domain. Despite big strides in practical distributed source coding techniques, this problem is still demanding in terms of processing power, bandwidth, and delay.

In this dissertation, we develop a new framework for distributed lossy source coding, in which we use real-number codes for binning. Specifically, we use a class of Bose-Chaudhuri-Hocquenghem (BCH) codes in the real/complex field known as the discrete Fourier transform (DFT) codes. Contrary to the conventional scheme, we first compress the continuous-valued sources and then quantize them. The new scheme exploits the correlation between continuous-valued sources, rather than quantized ones, which is more accurate. Also, by using short BCH-DFT codes, it reduces the complexity and delay and offers the potential to avoid the problems of the conventional quantization and binning approach, with relatively simple encoder/decoder.

We propose both syndrome- and parity-based schemes, and we extend the parity-based scheme to distributed joint source-channel coding based on a single DFT code. Further, to adapt to uncertainty in the degree of statistical dependence between the sources, we construct rate-adaptive BCH-DFT codes. This allows the encoder to switch flexibly between encoding sample rates, if the degree of statistical dependence varies. The construction of rate-adaptive codes is based on transmission of additional syndrome samples and a simple extension of the subspace-based decoding.

Another major contribution of this dissertation is to generalize the encoding/decoding of BCH-DFT codes. We prove that the parity frequencies of a BCH-DFT code, or equivalently the zeros of codewords in the frequency domain, are not required to be adjacent; we provide the decoding algorithm as well. This offers flexibility in constructing BCH-DFT codes and further improvement in the decoding which can be exploited in channel coding as well.

Résumé

Le codage de sources distribué, c'est-á-dire l'encodage séparé et le décodage conjoint de sources statistiquement dépendantes, survien dans un grand nombre d'applications telles que les réseaux de capteurs et le codage de vidéos multi-vues. Plusieurs de ces applications sont hautement interactives, ce qui demande le développement de communications et de schémas de calcul á faibles délais et limités énergétiquement. Actuellement, cette compression est effectuée en utilisant des codes binaires pour approcher la capacité de canal. En tant qu'extension naturelle, le codage de sources distribué avec pertes est réalisé en mettant en cascade un quantificateur et un codage Slepian-Wolf dans le domaine binaire. Malgré les grands progrés effectués dans les techniques pratiques de codage de sources distribué, ce probléme reste exigeant en termes de puissance de traitement, de bande passante et de délai.

Dans cette dissertation, nous développons un nouvel axe d'étude pour le codage de sources distribué avec pertes, dans lequel nous utilisons des codes á nombre réels pour l'encapsulage. Plus spécifiquement, nous utilisons une classe de codes Bose-Chaudhuri-Hocquenghem (BCH) dans le domaine réel, plus connus sous le nom de codes de la transformée de Fourier discréte (TFD). Contrairement au schéma conventionnel, nous comprimons d'abord les sources valeurs continues et nous les quantifions ensuite. Le nouveau schéma exploite la corrélation entre les sources valeurs continues plutôt que celles quantifiées, ce qui offre plus de précision. De plus, l'utilisation de codes BCH-TFD de petite taille, réduit la complexité et le délai et permet d'éviter les problmes liés á la quantification conventionnelle et á l'approche d'encapsulage, avec un encodeur/décodeur relativement simple.

Nous proposons deux schémas: un basé sur le syndrome et l'autre sur la parité, et nous étendons le schéma basé sur la parité au codage source-canal conjoint distribué á partir d'un seul code TFD. Par la suite, pour s'adapter á l'incertitude du degré de dépendance statistique entre les sources, nous construisons des codes BCH-TFD avec taux adapté. Cela permet á l'encodeur de permuter facilement entre les taux de code, si le degré de dépendance statistique varie. La construction de codes á taux adapté est basée sur la transmission d'échantillons-syndrome supplémentaires et sur une simple extension du décodage de type sous-espace.

Une autre contribution majeure de cette dissertation est la généralisation du processus

encodage/décodage des codes BCH-TFD. Nous prouvons que les fréquences de parité d'un code BCH-TFD, équivalents aux zéros des mots de code dans le domaine fréquentiel, n'ont pas besoin d'être adjacents; nous fournissons l'algorithme de décodage également. Nous apportons ainsi de la flexibilité lors de la construction des codes BCH-TFD et d'autres améliorations au niveau du décodage, lesquelles peuvent également être exploitées au niveau du codage de canal.

Acknowledgments

It has been an enlightening experience to work with so many talented people during my time at McGill. I significantly benefited from interacting with professors, fellow students, and other collaborators during my Ph.D. study.

My foremost gratitude goes to my adviser, Prof. Fabrice Labeau, for not only giving me the opportunity to pursue my Ph.D., but also for several other reasons: ideas, guidance, patience, endless support, and introductions. I am especially thankful to him for making my academic experience at McGill truly enjoyable.

I thank Professors Ioannis Psaromiligkos and Warren Gross for being on my committee and providing valuable comments. I am also thankful to professors from whom I learned a great deal attending their courses as a part of my Ph.D. work.

I would like to gratefully acknowledge the financial support of the McGill Engineering Doctoral Award, NSERC and Hydro-Québec throughout my Ph.D. study.

All my colleagues in the Telecommunications and Signal Processing group were sources of inspiration and friendship. I wish to give a special thanks to Sina for so many stimulating discussions, but also for his constant willingness to help, and Fabian for his meticulous translation of the abstract of this dissertation to French. It was a great pleasure to share, not just ideas and thoughts, but the daily ups and downs of research life with Hessam, Fabian, Sina, Mohsen, Alice, Ahmad, Mahsa, Hoda, Di, Leila, and Saeed, to mention a few. Since arriving at McGill, I have had the companionship and encouragement of friends, old and new; I should especially thank Yasin, Nima, and Ali.

Finally, my deepest gratitude is for my family for their unflagging love and unconditional support throughout my life and my studies. My parents who raised me, loved and supported me, and taught me to be the person I am today. My wife, Azam, who has always been with me and supported me through all the ups and downs of my Ph.D. studies. Without her constant support and patience this dissertation would have been impossible.

Contents

List of Figures

List of Tables

List of Acronyms

BCH	Bose-Chaudhuri-Hocquenghem
BSC	Binary Symmetric Channel
CS	Compressive Sensing
CEQNR	channel-error-to-quantization-noise ratio
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DJSCC	Distributed Joint Source Channel Coding
DMC	Discrete Memoryless Channel
DMS	Discrete Memoryless Source
DOA	Direction Of Arrival
DSC	Distributed Source Coding
DST	Discrete Sine Transform
DVC	Distributed Video Coding
EM	Expectation-Maximization
GBG	Gaussian-Bernoulli-Gaussian
GE	Gaussian-Erasure
IDFT	Inverse Discrete Fourier Transform
i.i.d.	independent and identically distributed
JSCC	Joint Source Channel Coding
LDPC	Low-Density Parity-Check
LS	Least Squares
MSE	Mean-Squared Error
ML	Maximum Liklihood
MT	Multi-Terminal

MUSIC	MUltiple SIgnal Classification
OFB	Oversampled Filter Bank
pdf	probability density function
PGZ	Peterson-Gorenstein-Zierler
RD	Rate Distortion
SI	Side Information
SWCQ	Slepian-Wolf-Coded Quantization
TCQ	Trellis Coded Quantization
VQ	Vector Quantization

Chapter 1

Introduction

We are living in a time of information explosion with ever-increasing demand for data storage and communication. It is estimated that 90% of all the data the human race has ever produced has been generated over the last two years [31]. The information age has been revolutionized by Shannon's fundamental work in 1948 [98]. In yet another change, during the past decades, communications has shifted from simple, point-to-point communication to network communication, with many senders and/or receivers. New network applications are distributed in nature (peer-to-peer) and highly interactive, requiring the development of new low-delay, energy-limited communication and computing schemes. In this thesis, we are interested in a specific kind of networks, namely, wireless sensor networks, a new, emerging field that is expected to play an important role in the future society and revolutionize environmental sensing. The applications of sensor networks are diverse; some broad range of applications areas are health, military, agriculture, and home [4], and new applications are continuously being discovered as the technology develops.

The emergence of distributed systems has intensified requirement for the efficient use of expensive and limited resources, such as bandwidth and power, so that a major concern in design and operation of wireless sensor networks is the energy-efficiency. This is because the sensors usually run on batteries and it is required that the battery last for the entire lifetime of the sensor. Reducing the amount of information to be transmitted is then sought-after to reduce the energy spent on communication and increase the battery lifetime. To this purpose, *distributed source coding* (DSC) is a key "enabling" technology. DSC refers to the compression of multiple, statistically dependent sources (e.g., neighboring sensors'

output) that *do not* communicate with each other and therefore are encoded in a distributed manner. In such a setting, the sensor nodes send their compressed outputs to a common center where joint decoding is performed. Distributed (separate) encoding is crucial to prevent unnecessary energy consumption for communication and coordination between the sensors.

A seminal work by Slepian and Wolf in 1973 [99] laid the foundation of distributed source coding, in which the authors proved the counter-intuitive result that separate encoding (with joint decoding) achieves the same compression rate as joint encoding does. Essentially, to realize a Slepian-Wolf encoder, one can partition all possible source outcomes into bins indexed by *syndromes* of some "good" linear *channel code* for the specific source correlation model [128]. In other words, the alphabets of the source are divided into cosets, and the cosets' indices are transmitted.[1] The Slepian-Wolf theorem deals with *lossless* compression of two correlated signals. A few years after this work, Burger and Tung [108] studied the lossy counterpart of this problem, i.e., *lossy* compression of correlated signals. A special case of distributed lossy source coding, namely, lossy source coding with *side information* at the decoder, which had already been studied by Wyner and Ziv [126], received much more attention in the literature.

In spite of these theoretical results, advancing the theory to less simple settings has been problematic and started only nearly a decade and a half ago. Specifically, to move toward the gains promised by information theory, Slepian-Wolf coders have been privileged by the advancement of practically implementable, capacity-approaching channel codes (e.g., turbo [12] and LDPC [42] codes). In a similar fashion, practical lossy distributed source coding is realized by converting the continuous-valued sources to the binary sources and applying Slepian-Wolf coding afterwards. Although big strides have been made in devising practical schemes for distributed source coding, this problem is still challenging, in various applications, in terms of *processing power* and *delay*. This is because, strictly speaking, capacity-achieving channel codes require unboundedly complex encoder/decoder with infinite number of signaling degrees of freedom (blocklength). Therefore, in practice, especially in real-time applications, where delay and complexity limitations are more stringent, the performance of DSC based on turbo and LDPC codes is highly affected. In general, in

[1] Instead, one may use a parity-based approach, in which the parity resulting from the application of the source to a *systematic* linear block code is used to label the cosets and thus is transmitted in lieu of the source.

practice, where delay and complexity are bounded, quantization and binning losses arise since it is proved that infinite dimension source and channel codes are needed to reach the boundary [132, 128].

In addition to the above losses, there is also another loss due to inaccuracy of the correlation model between the sources. This issue arises both in lossless and lossy DSC;[2] however, it is more pronounced in the latter. The reason is that the correlation between the sources is usually modeled after quantization, i.e., in the discrete domain; most often it is assumed to be a binary symmetric channel with a known crossover probability. Clearly, though, to capture the correlation between continuous-valued sources as accurately as possible, it is required to model correlation in the continuous domain. In other words, since quantization is a nonlinear operation, the correlation between the quantized signals is not known accurately even if it is known in the continuous domain, indicating that a model in the binary domain may not be accurate for continuous-valued sources, so it incurs some extra loss.

1.1 Contributions

In this dissertation, we develop a new framework for distributed lossy source coding, in which we introduce the use of real-number codes for binning. Transmission is, however, through a digital communication channel, i.e., binning is followed by quantization.[3] That is, contrary to the conventional scheme, we propose to first compress the continuous-valued sources and then quantize them. In the new framework, the compression is in the real field, aiming at representing the source with fewer samples. We still use channel codes for this purpose; the codes are analog, however. The new scheme exploits the correlation between continuous-valued sources, rather than quantized ones, to perform compression. It offers the potential to avoid the problems of the conventional quantization and binning approach, with relatively simple encoder/decoder and short codes. The following summarizes the major contributions of this dissertation:

- We establish a new framework for lossy DSC in general, and the Wyner-Ziv coding

[2] Throughout this dissertation the terms "lossy distributed source coding" is used interchangeably with "distributed lossy source coding." A similar convention is valid for the lossless case.

[3] Such a scheme benefits from the advantages of digital communication systems, such as the advanced error correction techniques and prevalence of digital processors; it can also be used in analog communication systems by removing the quantization block.

as a special case, where binning is performed in the real field by using real-number codes; this helps model the correlation noise more accurately and before quantization. Specifically, we use short Bose-Chaudhuri-Hocquenghem (BCH) codes in the discrete Fourier transform (DFT) domain which can reduce the complexity and delay to a large extent. We introduce both syndrome- and parity-based schemes and emphasize their advantages and disadvantages in different contexts.

- To adapt to uncertainty in the degree of statistical dependence between the sources, we construct rate-adaptive BCH-DFT codes. This allows the encoder to switch flexibly between encoding sample rates, without switching the code. In a rate-adaptive DSC, the encoder need not know the degree of statistical dependence in advance, if there is feedback from decoder to encoder.

- We extend the parity-based DSC to the case where the transmission channel is noisy; i.e., we use a single DFT code both to compress the signal and protect it against channel variations. This gives rise to distributed joint source-channel coding (DJSCC) based on DFT codes, and is a key for low-delay coding since it maps source blocks to short channel blocks.

- In order to develop the parity-based DSC and DJSCC, we introduce the notion of *systematic* DFT frames (or codes). For an (n, k) DFT code, there are $\binom{n}{k}$ systematic codes each of which may result in a different reconstruction error. We first demonstrate that the performance of these frames differs depending on the relative position of the systematic and parity samples in the codewords. We then prove that evenly spaced systematic (or parity) samples result in the minimum mean-squared reconstruction error, whereas the worst performance is obtained when the parity samples are consecutive. We also prove that a *tight*, systematic DFT frame can be realized if and only if the frame is performing integer oversampling.

- An extension of subspace-based error localization of BCH-DFT codes is developed in order to characterize rate-adaptation for this class of codes. This extended subspace-based approach, simply extends and improves the existing subspace-based algorithms by enlarging the dimension of the quantization noise subspace, or equivalently, increasing the number of polynomials obtainable for error localization. To this purpose, extra syndrome samples are required. The same idea can be applied to other real

codes which are based on orthogonal transform matrices, such as the *discrete cosine transform* (DCT) and *discrete sine transform* (DST) codes.

- We prove that for an (n, k) DFT code there are $\phi(n)$ syndrome matrices for decoding, where $\phi(n)$ is the number of positive integers less than n that are relatively prime to n. This diversity is exploited to improve the error localization by combining the error localizing polynomials corresponding to the matrices. Apart from the diversity in decoding, the generalized subspace method brings another novelty at the encoder side; that is, the parity frequencies of a BCH-DFT code (equivalently, the zeros of codewords in the frequency domain) are not required to be adjacent. This provides substantial flexibility in constructing BCH-DFT codes.

The above contributions have been reported in the following publications.

Articles published in refereed journals and conferences

1. M. Vaezi and F. Labeau, "Distributed Source-Channel Coding Based on Real-Field BCH Codes" *IEEE Trans. Signal Process.*, vol. 62, pp. 1171-1184, Mar. 2014.

2. M. Vaezi and F. Labeau, "Generalized and Extended Subspace Algorithms for Error Correction with Quantized DFT Codes" *IEEE Trans. Commun.*, accepted for publication, Dec. 2013.

3. M. Vaezi and F. Labeau, "Systematic DFT Frames: Principle, Eigenvalues Structure, and Applications" *IEEE Trans. Signal Process.* vol. 61, pp. 3774-3885, August 2013.

4. M. Vaezi, A. Combernoux, and F. Labeau, "Low-Delay Joint Source-Channel Coding with Side Information at the Decoder," *in Proc. DSP/SPE 2013*, Napa, California, 11-14 Aug. 2013.

5. M. Vaezi and F. Labeau, "Extended Subspace Error Localization for Rate-Adaptive Distributed Source Coding," *in Proc. ISIT2013*, Istanbul, Turkey, pp. 2174-2178, 5-13 Jul. 2013.

6. M. Vaezi and F. Labeau, "Improved Modeling of the Correlation Between Continuous-Valued Sources in LDPC-Based DSC," *in Proc. Asilomar2012*, Pacific Grove, California, 4-7 Nov., 2012.

7. M. Vaezi and F. Labeau, "Distributed Lossy Source Coding Using Real-Number Codes," *in Proc. VTC2012-Fall*, Québec City, Canada, 3-6 Sep. 2012.

8. M. Vaezi and F. Labeau, "Least Squares Solution for Error Correction on the Real Field Using Quantized DFT Codes," *in Proc. EUSIPCO-2012*, Bucharest, Romania, 27-31 Aug., 2012.

9. M. Vaezi and F. Labeau, "Systematic DFT Frames: Principle and Eigenvalues Structure," *in Proc. ISIT2012*, Boston, MA, USA, pp. 2436-2440, 1-6 Jul. 2012.

1.2 Organization

This dissertation is organized as follows. Chapter 2 provides an introduction to the problem of distributed source coding both in lossless and lossy cases. We start with theory and state some well-known, relevant theorems in distributed source coding. We then review practical code construction techniques and some applications. This is followed by a set of open issues and challenges in practical distributed source coding. In Chapter 3, we briefly explain the process of encoding and decoding information in BCH-DFT codes. We then establish a new framework for lossy DSC, in which BCH-DFT codes are used for binning. In this chapter, our main focus is on the syndrome-based Wyner-Ziv coding and its extension to rate-adaptive case. We showcase different aspect of performance for several codes. Numerical results show that, even with short codes, the end-to-end distortion in the new scheme is better than quantization error level. In Chapter 4, we introduce parity-based DSC and extend it to the case where the transmission channel is noisy. Such a setting gives rise to distributed joint source-channel coding based on DFT codes, in which one DFT code is employed both to compress the signal and protect it against channel variations, which is appealing for low-delay communication. Also, in this chapter we introduce and study the properties of systematic DFT codes. We find the best and worst systematic codes from the minimum mean-squared reconstruction error sense. Chapter 5 is devoted to improving the decoding of BCH-DFT codes, where we extend and generalize the subspace-based decoding of BCH-DFT codes based on extra syndrome samples. Flexibility in encoding and code rate as well as diversity in decoding are offered by virtue of the developed algorithms in this chapter. Simulation results, for several codes, demonstrate the capability of the extended and generalized subspace-based algorithms to perform much better than the

existing subspace-based error localization in the presence of quantization noise. Chapter 6 concludes the dissertation and proposes avenues for future research.

Chapter 2

Distributed Source Coding Background

This chapter presents an overview of distributed source coding and sets the stage for subsequent chapters. Section 2.1 and Section 2.2 introduce the problem and outline theoretical foundations, respectively. Section 2.3 outlines practical techniques which asymptotically achieve the theoretical limits. Section 2.4 briefly outlines the state-of-the-art selected applications of distributed source coding. In Section 2.6, we address recent research and open issues in distributed source coding techniques which motivate the contributions of this dissertation.

2.1 Introduction

Consider a communication system with two separate, correlated signals X and Y, as shown in Figure 2.1. X and Y come from two sources that cannot communicate with each other. This setting is also known as *distributed* source coding (DSC) because the signals are not encoded jointly; encoding is done independently or in a distributed manner. The receiver, however, can perform *joint decoding* since it can see both encoded signals. An example of such a system is a sensor network composed of spatially separated sensor nodes, sending correlated observations to a common fusion center. The question is to find the minimum required encoding rate such that both signals can still be recovered perfectly, i.e., without any loss.

In their fundamental study [99], Slepian and Wolf laid the foundation of this prob-

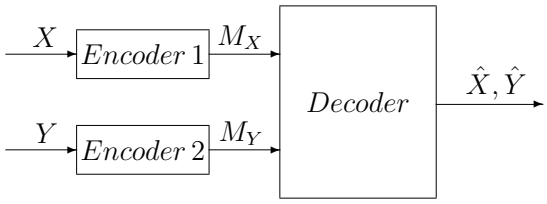


Figure 2.1 Distributed lossless source coding (Slepian-Wolf coding) with separate encoding and joint decoding.

lem, namely separate lossless compression of two correlated sources. They proved the counter-intuitive result that separate encoding (with joint decoding) achieves the same rate compression as joint encoding does. Lossless source coding with *side information* at the decoder is a special case of Slepian-Wolf coding where one signal, known as side information, is available at the decoder. Wyner and Ziv [126] extended this special case to a more general one, namely *lossy* source coding with side information at the decoder. Intriguingly, when the source and side information are jointly Gaussian and the distortion measure is the MSE, Wyner-Ziv coding does not suffer a rate loss compared to the case where side information is also available at the encoder. That is, for this case of lossy source coding, separate encoding is as efficient as joint encoding. However, in general, Wyner-Ziv coding incurs some loss in rate when compared to lossy source coding with side information available at both the encoder and decoder.

2.2 Background Theory

Perfect representation of a continuous-valued random variable requires an infinite number of bits, thus any description of such a variable with finite number of bits is imperfect and incurs some distortion. A basic problem in *rate-distortion theory* is to find the minimum expected distortion for a particular rate, given a source distribution and a distortion measure [28]. The rate-distortion function $R(D)$, which can also be applied to discrete random variables, is defined by the infimum of rates R such that (R, D) is achievable [37], for a given distortion value D. In other words, the rate-distortion function is the achievable lower bound on bit-rate for a distortion D. Here we take a look at rate-distortion functions of some well-known source coding problems.

2.2.1 Point-to-Point Source Coding

Theorem 2.1. *Shannon's Lossy Source Coding Theorem [28]*
The rate-distortion function for a discrete memoryless source (DMS) X with a distribution $p(x)$ and a distortion measure $d(x, \hat{x})$ is equal to the associated information rate-distortion function

$$R(D) = \min_{p(\hat{x}|x):\mathbb{E}(d(X,\hat{X}))\leq D} I(X;\hat{X}), \tag{2.1}$$

for $D \geq D_{min} \triangleq \min_{\hat{x}} \mathbb{E}(d(X,\hat{x}))$, where $I(X;\hat{X})$ is the mutual information of X and $\hat{X}$.

It means that a rate R is achievable with distortion D for $R > R(D)$, but it is not achievable for $R < R(D)$. It is easy to check that lossless source coding theorem is a special case of the lossy source coding theorem. To do so, in (2.1), let $D = 0$ which implies $\hat{x} = x$; and thus $R(0) = I(X, X) = H(X)$, which is the optimal rate for lossless source coding.

2.2.2 Source Coding with Side Information

Several interesting extensions of Theorem 4.1 have been studied in the literature when causal or non-causal *side information* (SI) is available at the encoder or decoder [37]. Here we only consider "non-causal" side information, that is when the entire side information sequence is available. When side information Y is available *both* at the encoder and decoder, it can be easily shown that the rate-distortion function becomes

$$R_{SI-ED}(D) = \min_{p(\hat{x}|x,y):\mathbb{E}(d(X,\hat{X}))\leq D} I(X;\hat{X}|Y). \tag{2.2}$$

This is known as *conditional* source coding problem, usually represented by the conditional rate-distortion function $R_{X|Y}(D)$. It potentially decreases the required encoding rate to achieve the same distortion [32], when compared with the case where side information is not available at the encoder or decoder.

A more fascinating variation however, is the problem of lossy source coding with side information available only at the decoder. Wyner and Ziv [126] studied this problem, and obtained its rate-distortion function.

Theorem 2.2. *Wyner-Ziv's Theorem [126]*
If (X, Y) are two DMS, the rate-distortion function of X with distortion measure $d(x, \hat{x})$

when side information Y is available only at the decoder is equal to

$$R_{SI-D}(D) = \min_{p(u|x),\, \hat{x}(u,y):\, \mathbb{E}(d(X,\hat{X}))\leq D} I(X;U|Y) \tag{2.3}$$

where U is an auxiliary random variable and $U \to X \to Y$ forms a Markov chain.

This rate-distortion function is mostly represented by $R_{WZ}(D)$ or $R_{X|Y}^{WZ}(D)$ in the literature. It can be verified [37] that the difference between $R_{SI-D}(D)$ and $R_{SI-ED}(D)$ is in the sense that the minimum is taken over $p(u|x)$ and $p(u|x,y)$, respectively. Thus, $R_{SI-D}(D) \geq R_{SI-ED}(D)$ is immediate. This indicates that a rate loss $R_{SI-D}(D) - R_{SI-ED}(D) \geq 0$ is incurred when the encoder does not know side information, which seems natural. However, Wyner and Ziv proved the intriguing result that $R_{SI-D}(D) = R_{SI-ED}(D)$ for Gaussian memoryless sources and mean-squared error distortion. [1] In particular, without loss of generality, for $X \sim \mathcal{N}(0, \sigma_X^2)$ and side information $Y = X + U$ with $U \sim \mathcal{N}(0, \sigma_U^2)$ independent of X, we have

$$R_{WZ}(D) = R_{X|Y}(D) = \begin{cases} \frac{1}{2}\log_2\left(\frac{\sigma_{X|Y}^2}{D}\right) & \text{if} \quad 0 \leq D \leq \sigma_{X|Y}^2 \\ 0 & \text{if} \quad D > \sigma_{X|Y}^2 \end{cases} \tag{2.4}$$

where $\sigma_{X|Y}^2 = \frac{\sigma_X^2 \sigma_U^2}{\sigma_X^2 + \sigma_U^2}$. It should be emphasized that, although in the quadratic case there is no loss if the encoder does not have side information, the underlying coding scheme is very different from the case that side information is available to both parties. In fact, the same performance is achieved at the cost of coding complexity. We should also point out that when side information is not perfect (e.g., suffers from rate loss) this result does not hold, and the exact solution is unknown to date. In general, for continuous memoryless sources under the MSE distortion constraint, Zamir [131] proves that $R_{\mathrm{WZ}}(D) - R_{X|Y}(D) \leq \frac{1}{2}$ bits. Yet, such a bound is not known for sources with memory.

2.2.3 Distributed Lossless Source Coding

Slepian and Wolf, in a seminal work [99], laid the foundation of distributed source coding (DSC), where statistically dependent signals are encoded in a distributed manner but de-

[1] The dual of this result in channel coding, the well-known "dirty paper coding," was later proved by Costa [26].

coded jointly. They proved, unexpectedly, that separate encoding can be as effective as joint encoding for this system. Let us first take a look at theoretical limits of the case where *joint encoding* can be performed; that is, the two sources are able to communicate. From Shannon's source coding theorem [28], for probability of decoding error to approach zero, the minimum sum rate is simply the joint entropy $H(X,Y)$. Surprisingly, though, the same combined rate is sufficient even if the signals are encoded *separately*, as described in the following theorem.

Theorem 2.3. *Slepian-Wolf's Theorem [99]*
The optimal rate region for distributed coding of two DMS sources $(X,Y) \sim p(x,y)$ *is the set of rate pairs* (R_X, R_Y) *that*

$$\begin{aligned} R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X) \\ R_X + R_Y &\geq H(X,Y) \end{aligned} \tag{2.5}$$

Figure 2.2 illustrates the Slepian-Wolf rate region compared to the conventional entropy coding. As it can be seen, Slepian-Wolf coding reduces the rate required for lossless transmission of correlated source; in other words, it expands the achievable rate region. Examples of significant reduction in total transmission rate are presented in [28], [37]. With conventional separate entropy encoding and separate decoding, one can only achieve $R_X \geq H(X)$ and $R_Y \geq H(Y)$; thus, $R_X + R_Y = H(X) + H(Y)$ which is greater than $H(X,Y)$ for correlated X and Y. Slepian and Wolf showed that the corner point A (represented by the first and last inequalities in (2.5)) is achievable even when sender 1 does not know Y. Thus, the sum rate $R_X + R_Y = H(X,Y)$ is achievable even though the sources are *separately* encoded. Therefore, the limit for lossless DSC can be smaller than that of separate coding. The compression, however, is no more error free. Nevertheless, the probability of error can be vanishingly small for long sequences.

Similar to Shannon's channel coding theorem, the proof of Slepian-Wolf coding is based on *random binning*. Random binning is a key, fundamental concept in information theory. In the context of Slepian-Wolf coding, the essential idea of random bins is to choose a large random index for each source sequence. Then with high probability, different source

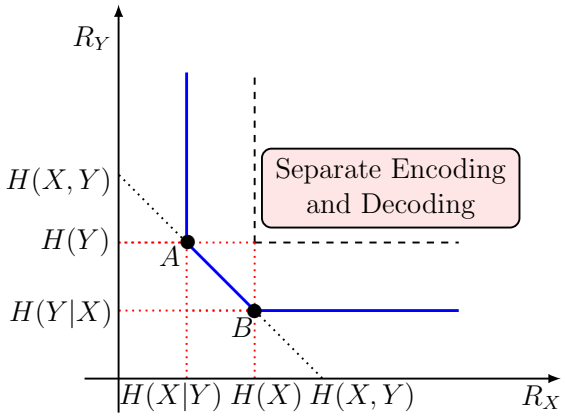


Figure 2.2 Achievable rate regions for the Slepian-Wolf coding (solid lines) and separate encoding with separate decoding (dashed lines).

sequences will have different indices, providing that the set of typical source sequences is small enough. Hence, we can recover the source sequence from the index [28].

2.2.4 Distributed Lossy Source Coding

As an extension of Slepian and Wolf problem, it is natural to study distributed lossy source coding in which reconstruction is no longer perfect. Similar to lossless DSC, two sources X and Y are separately encoded and the descriptions are sent over noiseless communication links to a common decoder. But unlike that, the compression is lossy and the decoder wishes to reconstruct the two sources with distortions D_X and D_Y, respectively. This is the most general setup for two sources; it was first studied by Berger and Tung under the name of "multiterminal source coding" [108]. Again, the question is to find the minimum required description pairs (R_X, R_Y) that achieve distortion pair (D_X, D_Y), i.e., the rate-distortion region $\mathcal{R}(D_X, D_Y)$. This problem is more involved and the solution is not known, except for the quadratic Gaussian case.

The rate-distortion region of this problem is not known in general. However, Berger and Tung [108] introduced an *inner bound* which is tight for several special cases. It includes both Wyner-Ziv and Slepian-Wolf regions as special cases. Particularly, this region reduces to Slepian-Wolf region when D_X and D_Y are Hamming distortion measure. For $R_Y \geq H(Y)$, i.e., when there is no rate limit for Y, this region gives the Wyner-Ziv rate-

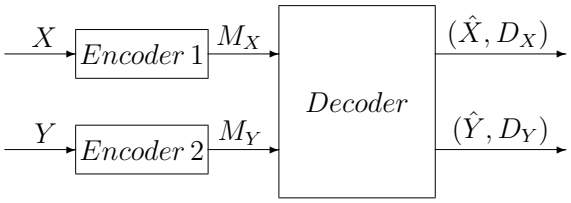


Figure 2.3 Distributed lossy source coding (multiterminal lossy source coding).

distortion function. Moreover, the Berger-Tung inner bound is optimal for the quadratic Gaussian case. However, a recently-found counterexample shows that the Berger-Tung inner bound is not tight in general [37].

Berger and Tung also introduced an *outer bound* which is tight for some special cases, including the Slepian-Wolf and Wyner-Ziv problems. There are also other cases for which this outer bound is proven to be tight [37], however it is not tight in general. For instance, it is not tight for the quadratic Gaussian case.

Having established a strictly better outer bound, Wagner *et al.* [120] recently proved the optimality of Berger-Tung's inner bound for the quadratic Gaussian case and gave the rate-distortion region. They also showed that the optimal architecture for this problem comprises separate quantization followed by Slepian-Wolf coding for each source. In addition, they proved that for a given covariance, Gaussian source has the smallest rate region. In other words, with this optimal architecture and a given covariance, higher rates are required to send Gaussian sources than any other source [120]. This indicates that Gaussian source is the "worst" source in this sense.

2.3 Practical Code Construction

The Slepian-Wolf theorem, which dates back to the early 1970's, provides a seminal, conceptual basis for distributed source coding. However, it is nonconstructive since it uses *random binning* in the proof. Serious practical code constructions for this theorem was motivated, only during the past decade and half, by emerging applications in distributed sensor networks and distributed video coding. According to the Slepian-Wolf coding the-

orem, the achievable rate region for distributed sources X and Y is given by (2.5). This region is illustrated in Figure 2.2. Most practical Slepian-Wolf codings first try to approach a corner point (e.g., A) with $R_X + R_Y = H(X,Y) = H(X|Y) + H(Y)$. This is called *asymmetric* Slepian-Wolf coding. Although, to prove the achievability of this point, Slepian and Wolf uses binning technique, the proof contains the main intuition for practical construction as well. To highlight this, from the fact that $p_{X,Y}(x,y) = p_X(x)p_{Y|X}(y|x)$, one can think of the Y-sequences that are generated by applying the X-sequences as inputs to a discrete memoryless channel with probability of $p_{Y|X}(y|x)$. In other words, one can view X and Y as input and output of a noisy communication channel with probability transition matrix of $p_{Y|X}(y|x)$. Thus, designing a capacity-achieving channel code for this channel ensures reliable transmission of a subset of X, effectively the index of that subset. More precisely, a capacity-achieving channel code for this channel with $C = I(X;Y)$, can distinguish $2^{I(X;Y)} = 2^{H(X)-H(X|Y)}$ different codes. Since there are on the average $2^{H(X)}$ input symbols, we can divide them into $2^{H(X|Y)}$ disjoint sets where each set on average contains $2^{H(X)}/2^{H(X|Y)} = 2^{I(X;Y)}$ elements. Hence, by designing $2^{H(X|Y)}$ disjoint capacity-achieving codes, any symbol produced by X can be associated to one capacity-achieving code.

In a similar fashion, the other corner point (B) can be achieved. Afterwards, any point between A and B will be achievable by time sharing, i.e., using each code for a fraction of time. For instance, to achieve the point in the middle of A, B each code must be used 50% of time. One can also design codes that directly approach any point on the line segment AB; this approach is referred as *symmetric* Slepian-Wolf coding [55]. In what follows, we discuss the asymmetric Slepian-Wolf coding design techniques.

2.3.1 Slepian-Wolf Coding

Although it took more that 30 years to design practical codes to approach this theoretical limit, a basic idea for practical Slepian-Wolf code construction was proposed in 1974 by Wyner [125], a year after Slepian and Wolf's work was published. The idea is to partition all possible source outcomes into bins indexed by syndromes of some "good" linear *channel code* for the specific correlation model [128]. In other words, the alphabets of X can be divided into cosets, and the cosets' indices are transmitted. Figure 2.4 illustrates the idea of distributed lossless source coding, where cosets are indexed based on syndrome. Alternatively, one may use a parity-based approach, in which a systematic linear block

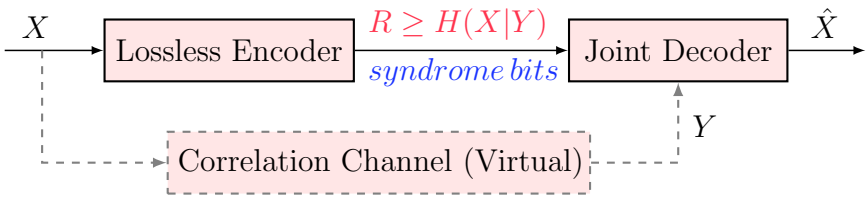


Figure 2.4 Practical *lossless* distributed source coding in asymmetric case, where Y has the role of side information at the decoder. X and Y are binary sequences; the *virtual* correlation channel is usually modeled as a binary symmetric channel. Alternatively, we may transmit parity instead of syndrome.

code is applied and the resulting parity is transmitted.

The Slepian-Wolf source coding problem is actually a channel coding problem and binary channel codes have been the main driver of most DSC techniques. Using convolutional codes for coset construction, both scalar and trellis, Pradhan and Ramchandran [80, 81] first introduced a constructive coding technique for binary and Gaussian sources. Later, more sophisticated channel codes, including turbo and low-density parity-check (LDPC) codes [93, 42, 61], were adapted to this problem by several groups independently. Now, it is known that capacity-approaching turbo [9, 2, 75, 45], and LDPC codes [69, 46, 95] result in a compression rate that approaches toward the Slepian-Wolf bound [50, 128].

In summary, as shown in Figure 2.4, in order to independently compress two correlated sources, one source (Y) is compressed to its entropy while the other one (X) is channel encoded using a capacity-approaching code, in which only the *redundant* bits (i.e., parity or syndrome bits) are transmitted, resulting in compression. At the decoder, the first code is decoded independently as it was encoded simply by entropy coding. Decoding of the second source, however, is more demanding. Due to the correlation between the sources, the first source can be thought of as a "noisy" version of the second one. Then, the syndrome bits, which are used to describe the second code, are concatenated to its noisy interpretation. This is as if we have channel coded the second source and sent the code and redundancies through the channel. Thus, we can run channel decoding and recover this channel codeword.

A *virtual* channel with input X and output Y, as illustrated in Figure 2.4, represents the correlation between the two sources. In fact, existing channel codes can be used in DSC provided that the correlation can be modelled by a simple communication channel.

Then, if a channel code is good for this correlation channel, Slepian-Wolf compression will work to a perfect degree. This virtual channel can be modeled in different ways. Many approaches model it as a binary symmetric channel (BSC) with a known crossover probability p. Others consider p as an unknown and estimate it during the iterative decoding algorithm [45] or use a Markov model to measure the correlation between the sources. In yet other approaches, the correlation parameter p changes over time. This gives rise to the use of *adaptive* algorithms to predict and track the change of p, in order to enhance decoding. The estimation of statistical correlation between the source and side information is particularly important when dealing with non-stationary sources with unknown statistics, like distributed video coding (DVC). The BSC has attracted more interest mainly because in *channel coding*, a large body of research has been carried out on this channel, and a number of capacity-approaching codes are available for that.

The linear channel code design and its rate depend on the correlation model. Essentially, the objective of the correlation channel is to help find a good linear channel code to compress X. To be specific, it indicates how many syndrome bits are required so that the decoder can decode X with small amount of error. The more accurate the modeled correlation channel between X and Y, the better (higher) the realizable compression for X, provided that an appropriate channel code, i.e., a code that approaches the capacity of this virtual channel, is used to compress X. Then, the rate required to describe X moves toward the Slepian-Wolf bound, $H(X|Y)$.

In general, finding a good code for Slepian-Wolf coding problem is not very easy. Even for apparently simple BSC correlation model, with Slepian-Wolf limit $H(X|Y) = H(p)$, finding a solution for the Slepian-Wolf coding problem is not trivial. However, since a number of capacity-approaching codes are available for this channel, this model is widely considered in this context. With this model, lower $H(p)$ corresponds to higher correlation and vice versa. Simulation results show that the lower the correlation between the two sources is, the more powerful the channel code needed to achieve the same probability of error [128]. To estimate the virtual channel, one approach is to transmit as few as possible syndrome bits initially. Upon receiving the syndrome bits, the decoder estimates the correlation parameter and decides whether the received syndrome bits are enough for a successful decoding. Otherwise, it informs the encoder to send more syndrome bits, until the correlation parameter is estimated accurately enough; at this stage, the decoder begins to decode [39]. This is in contrast to the *rate-adaptive* scheme with feedback [119] in which

the decoder attempts decoding with a short syndrome first and requests an augmented syndrome only if decoding fails. The expectation-maximization (EM) algorithm (using the log-likelihood ratio) is proposed in [39]. However, the estimation accuracy depends on the initialization of the EM algorithm, particularly for sources with low correlation. In [104] an algorithm based on maximum likelihood (ML) estimation is proposed which is performed prior to decoding, so it can be used to initialize the expectation maximization algorithm. It should be mentioned that despite giant strides in this field, universal Slepian-Wolf coding is still an open problem.

Apart from being an important problem individually, when cascaded with a quantizer, Slepian-Wolf coding provides a practical approach to lossy DSC problems, such as the Wyner-Ziv problem. This is discussed in the following.

2.3.2 Wyner-Ziv Coding

In many emerging applications like sensor networks, we are dealing with continuous sources rather than discrete sources. This gives rise to the problem of lossy distributed source coding or *rate-distortion* for distributed correlated sources. Particularly, for the asymmetric case, we may consider code construction for the corner point $(R_X, R_Y) = (H(X|Y), H(Y))$ only. Since $R_Y = H(Y)$, we can encode and decode Y independently; thus, we assume Y is available at the decoder. It then becomes the well-known Wyner-Ziv problem, namely, source coding with side information at the decoder. In general, Wyner-Ziv coding incurs some rate loss compared to conditional source coding, in which the side information is available to both encoder and decoder. However, as explained in Section 2.2, there is no rate loss in the quadratic Gaussian case, where (X, Y) are jointly Gaussian and the distortion measure is the MSE. This "no rate loss" condition, which was introduced by Wyner and Ziv, later extend by Pradhan *et al.* to the more general case of $X = Y + Z$, where X and Y can have arbitrary distributions and Z is Gaussian and independent from X, Y [128].

In general, a Wyner-Ziv encoder can be modeled by cascading a quantizer and a Slepian-Wolf encoder, as depicted in Figure 2.5. Since the Slepian-Wolf coding is based on channel coding, Wyner-Ziv becomes a source-channel coding and two different losses are introduced: Source coding loss (or quantization loss) and channel coding loss (or binning loss). Thus, to move toward the Wyner-Ziv bound, we need to concurrently minimize quantization

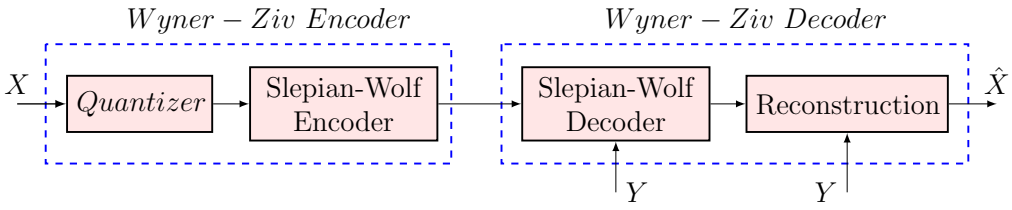


Figure 2.5 Practical *lossy* source coding with side information (Y) at the decoder (the Wyner-Ziv coding).

loss and maximize rate compression in the Slepian-Wolf encoder. Zamir and Shamai [132] proved that infinite dimension source and channel codes are required in general to reach the Wyner-Ziv boundary. Hence, practically, the Wyner-Ziv limit can be reached only *asymptotically*.

Inspired by the information-theoretic work by Zamir and Shamai, the design of quantizers for this problem first started by Pradhan and Ramchandran [80], in 1999. Later, they completed this work in [81], where they constructed computationally efficient scalar and trellis codes attempting to approach the Wyner-Ziv limit. They combined two source codes, i.e., scalar quantization and trellis coded quantization (TCQ), with two channel codes, scalar and trellis-based coset construction, and compared the performance of these schemes. Another practical approach is to combine scalar quantization with powerful Slepian-Wolf codes like LDPC and turbo codes. In this approach, there is no binning in quantization and all binning is left to the Slepian-Wolf code. This allows to use a powerful channel code and virtually limits the performance degradation only to the source coding (quantization) part. Further improvement is achieved when the source code is TCQ and the channel code is a turbo code. Similarly, the Wyner-Ziv coding based on TCQ and LDPC codes has been studied. Assuming 256-state TCQ and "ideal" Slepian-Wolf coding (the one which achieves $H(Q(X)|Y)$) results in only 0.2 dB away from the theoretical Wyner-Ziv limit at high rates [128, 55]. With practical Slepian-Wolf coder this gap can be as small as 0.5 dB at high rates. The performance of practical design can be even closer to theoretical limit. For instance, 8192-state TCQ with irregular LDPC and optimum *nonlinear estimation* at decoder has a gap of 0.2 dB away from the theoretical limit at a rate of 3.83 bps [129, 55]. It should be however highlighted that the decoder complexity is very high; it uses an LDPC code of length 10^6 with 300 iterations and nonlinear estimation for reconstruction.

2.3.3 Berger-Tung Coding

In a more general context, we may consider the problem of distributed lossy source coding design in which, at the decoder, neither of the sources can be reconstructed without loss. This means that side information is no longer perfect, at the decoder. This problem, that arises in practical distributed applications, is also a source-channel coding problem. Wagner *et al.*, in their theoretical work [120] derived the rate-distortion region for such a problem in the quadratic Gaussian setup. Moreover, they proved that vector quantization along with Slepian-Wolf is optimal for this problem when there are only two terminals. Inspired by this leading result, practical multi-terminal (MT) source coding was studied employing Slepian-Wolf-coded quantization (SWCQ). SWCQ explicitly separates vector quantization from Slepian-Wolf coding. This allows designing a good source code as well as a good channel code "individually," and make it possible to separately evaluate their contribution to total loss. Assuming ideal source coding and ideal channel coding any point on the sum rate bound is achievable for both direct and indirect MT source coding problems [55]. Similar to practical Winer-Zive coding, practical MT coding can be realized by TCQ and turbo/LDPC codes, and achieves reasonably low gap.

2.4 Advanced Techniques

In practice, the statistical dependency between source and side information varies in time in a fashion that might be unknown in advance. To overcome this problem, the encoder can adapt its coding rate on-the-fly, to handle different levels of statistical dependency between the source and side information; this is referred to as *rate adaptation* [103]. Rate-adaptive schemes are popular in transmission of non-ergodic data (e.g., video), to adjust to compression ratio variations [119].

In parity-based DSC using systematic convolutional and turbo codes, *puncturing* can be used to generate variable rates. Rate adaptation in syndrome-based DSC using convolutional and turbo codes are performed both for the asymmetric and symmetric cases in [91] and [103], respectively. Removing syndrome bits of LDPC codes leads to performance degradation; rate-adaptive LDPC codes are presented in [119]. Other advanced techniques like side information adaptation and multilevel DSC are considered in the literature.

2.5 Applications

In the previous sections we focused on the theory and techniques to implement DSC. The driving force behind recent practical implementation of the theory which dates back to the 1970's, has been a set of emerging applications of DSC in multi-terminal communication networks e.g., wireless sensor networks, ad hoc wireless networks, multi-terminal video coding, cooperative and relay communications.

With possible uses in environmental control, industrial automation, intelligent homes, health, military, etc., wireless sensor networks are expected to play an important role in sensor technology. The sensor nodes are required to be energy efficient, particularly because the sensors usually run on batteries and are required remain functional for a long period of time, if not for the entire lifetime of the sensor. The main sources of power consumption in sensor networks are sensing, communication, and data processing [4]. Reducing the energy spent on communication is possible through reducing the amount of information that needs to be transmitted. DSC comes in handy in this context if two nodes have a high correlation between their readings.

Another set of applications of DSC appears in distributed video coding [77]. DVC is a new paradigm for video coding which has several advantages to the conventional video coding. One important benefit of DVC is to shift the burden of motion estimation from the encoder to the decoder, compared to the popular video compression standards such as H.263 and MPEG [82]. Flexible allocation of the overall video codec complexity, improved error resilience, and codec independent scalability are among other advantages of DVC [77]. There are many other application scenarios for which DVC may bring major advantages, however it should be mentioned that the coding efficiency of current DVC systems is still below that of the latest conventional video codecs, such as H.264/AVC.

2.6 Challenges and Open Issues

In block channel coding, there is a fundamental interplay between the rate, block length, and probability of error. The analysis of the interplay between these three parameters has been one of the main questions of coding and information theory for decades. Motivated by practical applications in which limited delay is a key design constraint, there has recently been a lot of interest in non-asymptotic (finite block length) data compression and data

transmission, even in the information theory community. For example, [79, 78] address the fundamental limits of channel coding at finite block length. In fact, a viable path and an active research area in physical layer is to find short, powerful codes with reasonably simple decoding algorithms [30].

Despite tremendous advances in practical DSC techniques, this problem is still demanding in terms of processing power, bandwidth, and delay in various applications. Here, we address some open issues such as coding efficiency, complexity, and delay, as well as problems in modeling the correlation channel and finding the corresponding rate distortion performance.

2.6.1 Correlation Channel Model

Accurate modeling of the correlation between the sources plays a crucial role in the performance and efficiency of the DSC systems. Existing work on DSC model the correlation between the continuous-valued sources after quantization. Some of them, mainly theoretical ones, assume that this statistical dependency can be modeled by a binary symmetric channel (BSC) [81, 50, 128, 2, 8]. Many practical works, such as [82, 25], however, consider non-binary correlation models. Motivated by many applications, in which the exact correlation information is not available, there has been a flurry of recent activity on improving the correlation model and/or estimating it. Specifically, most DVC modeling approaches e.g., [59, 19, 38, 10, 35] assume the correlation noise as independent Laplacian or Gaussian-Bernoulli-Gaussian (GBG).

Some recent results even report that a better model for correlation noise is obtained considering the dependency between noise and side information [29]. Not surprisingly, simulation results indicate that finer a noise model improves the rate distortion performance [59]. Thus, the performance of DSC strongly depends on accurate knowledge of correlation between sources. We will address this issue in Chapter 3 of this dissertation.

2.6.2 Rate Distortion Performance

The rate distortion (RD) theory [28, Chapter 10] provides an analytical expression for the amount of compression that can be achieved using lossy compression methods. The Wyner-Ziv theorem provides the RD function for Gaussian sources; however, in many applications, like video, data statistics is known to be non-Gaussian. The RD region is not known for

non-Gaussian models including the Laplacian and the GBG. Further, the RD function is different for the sources with and without memory. The latter, also called temporal correlation, typically is naturally found in many sources.

Apart from this, motivated by practical systems especially by real-time applications, there is a renewed interest in understanding the fundamental limitations of finite block length codes [78, 30]. In this line of research, the RD performance of DSC systems with limited delay (i.e., finite block length) is another interesting open issue.

2.6.3 Complexity

Evaluating the rate distortion (RD) performance addresses only one side of the problem; the complexity to achieve such an RD performance is the other side of the problem. Low-complexity encoding is one of the most widely cited advantages of DVC with respect to conventional coding schemes because DVC provides a framework that eliminates high computational burden motion estimation at the encoder as well as the corresponding memory to store reference frames. Indeed, it is shown that DVC encoding complexity (see, e.g., DISCOVER codec [8, 1]) provides a substantial speed-up when compared to conventional H.264/AVC in terms of software execution time [34]. Despite reducing the encoding complexity compared to the conventional video coding approaches, the complexity is still high since one has to utilize capacity achieving channel codes which require very large block lengths. Although this approach may be acceptable for video applications, it is problematic for sensor network applications, which tend to be delay sensitive and have lower overall information rate.

Besides, most existing work on distributed source coding focused on memoryless sources. To exploit the temporal correlation typically found in many sources, usually a discrete cosine transform (DCT) or prediction filters is applied to both the source and the side information in order to improve efficiency. This will add some extra complexity to the system.

2.6.4 Delay

If the sensors are measuring delay-sensitive signals such as audio, in addition to low power and bandwidth constraints, delay becomes another constraint. Current methods that approach theoretical rate distortion of DSC, require prohibitively large block lengths which,

in turn, translate into a large system delay and an increased memory requirement at the sensors nodes even for low-dimensional (or scalar) quantizers. Therefore, it may not be plausible to implement these methods in real-time scenarios [77], e.g., surveillance for security or military applications, where low delay is crucial. The other extreme case, i.e., zero delay distributed source-channel coding, has been investigated in [6,5,7,23] through the use of analog mapping. Although these schemes have lower complexity, they are far from the theoretical limits and they do not benefit from the advantages of digital communications as they use analog communications.

In this dissertation, we introduce the use of short analog codes to do binning in the lossy DSC; transmission is still through a digital communication channel. This can address the complexity and delay to a large extent; it also helps model the correlation noise more accurately. However, due to very short block length, the rate distortion performance is expected to be far from the asymptotic RD function.[2]

2.7 Summary

In this chapter, we reviewed different variants of the problem of separate compression of two statistically dependent sources, both in theory and practice. Although the fundamental results date back to the 1970's, only in the past decade and half have emerging applications like sensor networks motivated techniques to design practical codes. It has been shown that the *source coding* problem of Slepian-Wolf is in fact a *channel coding* problem and state-of-the-art channel codes, like turbo and LDPC codes, have been exploited for this purpose. The Wyner-Ziv coding is then implemented by cascading quantization and Slepian-Wolf coding. We also highlighted a number of advanced techniques as well as applications of the DSC. Finally, motivated by the needs of the applications, we discussed recent advances in research, in the areas of rate adaptation, correlation noise modeling, rate distortion performance, and low-complexity and/or low-delay code design. Our contributions in Chapters 3 - 5 of this dissertation build on and unify elements of this literature. Throughout this dissertation, particularly in Chapters 3 and 5, we revisit the above issues and apply our novel framework and techniques.

[2] This highlights the need for the non-asymptotic RD function, but is out of the scope of this dissertation. In practice, using finite block lengths results in a performance loss which can be sizeable [34].

Chapter 3

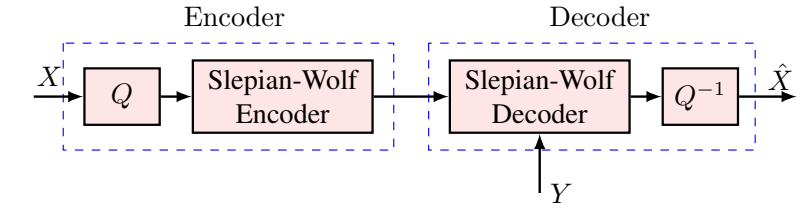
Distributed Source Coding Using BCH-DFT Codes

As described in Chapter 2, the current approach to the DSC of continuous-valued sources is to first convert them to discrete-valued sources and then apply *lossless* (Slepian-Wolf) coder [50, 128, 81]. Similarly, a practical Wyner-Ziv encoder consists of a quantizer and Slepian-Wolf encoder. There are, hence, *quantization* and *binning* losses for the source coder. Despite this, *rate-distortion theory* promises that block codes of sufficiently large length are asymptotically optimal, and they can be seen as vector quantizers followed by fixed-length coders [32]. Therefore, practical *Slepian-Wolf* coders have been realized using different binary channel codes, e.g., LDPC and turbo codes [69,9,2]. These codes, however, are out of the question if low delay is imposed on the system as they may introduce excessive delay when the desired probability of error is very low. The other extreme case, i.e., zero delay source-channel coding, can be achieved through the use of analog mapping [6,5,7,23]. These schemes have lower complexity but they do not benefit from the advantages of digital communications as they use analog communications; they are also far from the theoretical limits.

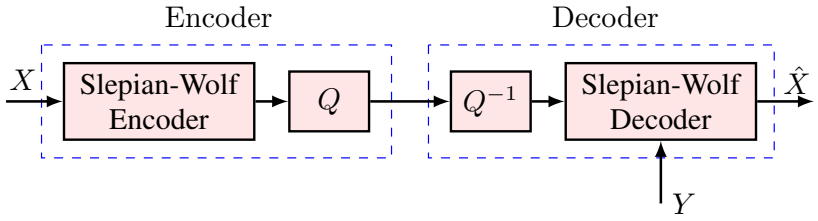
In this chapter, [1] we establish a new framework for distributed lossy source coding over digital communication channels, in which we propose to first compress the continuous-valued sources and then quantize them, as opposed to the conventional approach.[2] The

[1] The material in this chapter is based on [111, 113, 117, 115].

[2] The proposed framework can be exploited in analog communication systems simply by removing the quantization block. In sensor networks, analog communication can be optimal in some specific cases [48,49].



(a) Using binary codes



(b) Using real-number codes

Figure 3.1 The Wyner-Ziv coding based on binary and real-number codes. Both schemes can be straightforwardly extended to distributed source coding.

new framework is compared against the existing one in Figure 3.1. It introduces the use of *real-number codes* (see, e.g., [72, 124, 76, 73, 24]), to represent correlated sources with fewer samples, in the real field. That is, the compression is in the real field, aiming at representing the source with fewer samples. To achieve compression, we send *syndrome* or *parity* samples of the input sequence using a real-number channel code, similar to what is done to compress a binary sequence of data using binary channel codes. Then, we quantize the syndrome or parity samples and transmit them. There are still binning and quantization losses; however, since coding is performed before quantization, error correction is accomplished in the real field; and, in the extreme case where the two sources are the same over a block of code the quantization error can also be mitigated. A second and more significant advantage of this approach is the fact that the correlation channel model can be more realistic as it captures the dependency between the continuous-valued sources rather than quantized sources. In the conventional approach, it is implicitly assumed that quantization of correlated signals results in correlated sequences in the binary domain; this may not necessarily be precise due to the nonlinearity of the quantization operation. To avoid any loss due to the inaccuracy of correlation model, we exploit the correlation between the continuous-valued sources before quantization. The new approach is also capable of alleviating the quantization error. This

is possible because coding precedes quantization. Specifically, we use the *Bose-Chaudhuri-Hocquenghem* (BCH) DFT codes [72,14,86,41,100,113] for compression, and owing to them the loss due to quantization can be decreased by a factor of $\frac{k}{n}$ for an (n, k) code [51,84,113]. Additionally, reconstruction loss becomes zero if the two sources are the same over one short codeword of a DFT code. This is achieved in view of modeling the correlation between the two sources in the continuous domain. Moreover, the new framework is suitable for low-delay communications since, by using short DFT codes, a reasonably low reconstruction error is attainable.

Section 3.1 motivates the new framework for lossy DSC. We also present the correlation channel model that is used throughout this dissertation, as well as two simple information-theoretic bounds for the correlation channel. In Section 3.2, we study the encoding and decoding of BCH-DFT codes, and we leverage subspace-based decoding (e.g., MUSIC) to improve the decoding. Then, in Section 3.3, we present the encoder and decoder for the Wyner-Ziv coding based on the BCH-DFT codes. The proposed system is extended to rate-adaptive coding in Section 3.4. Section 3.5 presents the simulation results and confirms that the proposed system is well suited for low-delay communications, as the mean-squared reconstruction error (MSE) is shown to be reasonably low for very short block length.

3.1 Real-Number Codes for DSC

The problem of error correction in the real field using *real-number* codes was first considered by Marshall [72] and Wolf [124]; they proposed the discrete Fourier transform (DFT) for this purpose. Marshall also introduced an important subclass of DFT codes, the BCH-DFT codes. DFT codes find applications in a wide range of areas including wireless communications [121,57], image transforms [40], joint source-channel coding [41], distributed source coding [111], and compressive sensing [33]. Looking from the *frame theory* perspective, these codes are used to provide robustness against packet erasure in wireless networks [51,84,16]. Similar to error correction in finite fields, the basic idea of error correcting codes in the *real field* is to insert *redundancy* to a message vector to convert it to a longer vector, called the codeword. However, the insertion of redundancy is done in the real field, i.e., before quantization and entropy coding [72, 124]. One main advantage of *soft redundancy* (real field codes) over *hard redundancy* (binary field codes) is that by using soft redundancy one can go beyond the quantization error level and thus reconstruct continuous-valued signals

more accurately.

We introduce the use of real-number codes in lossy compression of correlated signals. The proposed system for the Wyner-Ziv coding is depicted in Figure 3.1. Although it consists of the same blocks as the existing Wyner-Ziv coding scheme [50, 128], the order of these blocks is changed. Hence, binning is performed before quantization and can benefit from soft redundancy; we use DFT codes for this purpose.

3.1.1 Motivations

The following paragraphs summarize the motivation for the proposed change in the order of binning and quantization blocks and the use of DFT codes for binning.

Realistic correlation model

In the existing framework for lossy DSC, correlation between two sources is modeled after quantization, either using binary [81, 50, 128, 2, 8] or non-binary [82, 25] correlation models. Admittedly though, due to the nonlinearity of the quantization operation, correlation model between the quantized signals cannot be as accurate as that of the continuous-valued signals. In addition, with this new framework one can consider symbol-by-symbol correlation while in the existing approach bit-by-bit correlation is investigated. In the latter case, to have a decent compression, it is necessary to consider the dependency between different bits of each sample [112], which can complicate the system. On the contrary, a symbol-by-symbol correlation does not suffer from such an issue. These motivate us to perform DSC by investigating a method that exploits correlation between continuous-valued sources rather than binary sources.

Alleviating the quantization error

In lossy data compression with side information at the decoder, soft redundancy, added by DFT codes, can be used to correct both quantization errors and (correlation) channel errors. Thus, the loss due to quantization can be recovered, at least partly if not wholly. In fact, if the two sources are exactly the same over a codeword,[3] the quantization error can

[3] Note that this can happen in practice especially if the code length is short; for example, it is possible when the readings of two closely-located sensors are the same for a short period of time, i.e., over one block of code.

be compensated for. That is, perfect reconstruction is attainable over the corresponding samples. The loss due to the quantization error is decreased by a factor of code rate ($\frac{k}{n}$ for an (n, k) code) even if correlation is not perfect, i.e., when errors exist. This is because DFT codes are *tight* frames; hence, they minimize the MSE [51, 84, 113].

Low-delay communications

Limited delay is a key constraint in many modern applications, and low-delay coding has recently drawn a lot of attention. If communication is subject to low-delay constraints, the performance of turbo or LDPC codes may not be satisfactory due to the imposed short code length. Early works on DSC are mainly based on turbo and LDPC codes [9, 2, 75, 45, 69]. Low-delay systems can be realized by mapping short source blocks into channel blocks, in a linear or non-linear fashion [6, 122, 22]. Whether the low-delay requirement exists or not depends on the specific applications. However, even in applications in which low-delay transmission is not imperative, it is sometimes useful to consider low-dimensional systems for their lower computational complexity. We use DFT codes with short block length and scalar quantization which is suitable if limited delay is required.

3.1.2 Correlation Channel Model

Accurate modeling of the correlation between the sources plays a crucial role in the performance evaluation and efficiency of the DSC systems. Existing works on DSC model the correlation between the continuous-valued sources after quantization. Some of them, mainly theoretical ones, assume that this statistical dependency can be modeled by a binary symmetric channel (BSC) [81, 50, 128, 2, 8, 112]. Some other works, such as [82, 25, 59, 19, 10, 35], consider non-binary correlation models, such as Laplacian or Gaussian, and transform them to the binary domain. However, in any of those cases, the correlation is effectively modeled after quantization and, due to the nonlinearity of quantization operation, such a correlation model may not be as accurate as a model in the continuous domain. This issue can be dealt with by exploiting the correlation between the continuous-valued sources before quantization.[4]

[4] To be precise, the term "correlation channel" is used to represent a statistical dependency rather than a correlation. With this view, it is natural to consider the statistical dependency of analog sources in the analog domain, e.g., as in (3.1).

The correlation between the analog sources X and Y, in general, can be defined by

$$Y = X + E, \tag{3.1}$$

where E is a real-valued random variable. This model, in which the correlation noise E is independent from X, will be referred to as the *forward correlation channel*.[5] Particularly, the above model represents some well-known models motivated in video coding and sensor networks. Let

$$E \sim \begin{cases} \mathcal{N}(0, \sigma_0^2) & \text{w.p.} \quad p_0, \\ \mathcal{N}(0, \sigma_1^2) & \text{w.p.} \quad p_1, \\ 0 & \text{w.p.} \quad 1 - p_0 - p_1, \end{cases} \tag{3.2}$$

in which $\sigma_1^2 = \sigma_i^2 + \sigma_0^2$, $\sigma_i^2 \gg \sigma_0^2$, and $p_0 + p_1 \leq 1$. This is a mixture of Gaussian *impulses* with power σ_i^2 and a *background noise* with power σ_0^2. Then, for $p_0 = 1$ or $p_1 = 1$ the Gaussian correlation is obtained. Further, for $p_0 + p_1 = 1$ the Gaussian-Bernoulli-Gaussian (GBG) and for $p_0 + p_1 < 1$, $p_0 p_1 = 0$ the Gaussian-Erasure (GE) models are realized. The Gaussian model is broadly used in the sensor networks literature whereas the latter two models are more suitable for video applications.

The GBG model can be considered as an extension of the jointly Gaussian model. The rate-distortion region of this model is not known; thus, its characterization can be obtained through *lower and upper bounds*. To find such bounds, which will be later used in this chapter, one can make use of the rate-distortion function of related, simpler coding problems. We know that when side information Y is available both at the encoder and decoder, the rate-distortion function for jointly Gaussian sources is given by (2.4) in which $\sigma_{X|Y}^2 = \frac{\sigma_X^2 \sigma_E^2}{\sigma_X^2 + \sigma_E^2}$ for forward correlation channel and $\sigma_{X|Y}^2 = \sigma_E^2$ for reverse correlation channel. This is known as *conditional* source coding problem, and its rate region is obviously a lower bound for lossy source coding with side information available "only" at the decoder (the Wyner-Ziv problem); i.e., $R_{X|Y}^{\text{WZ}}(D) - R_{X|Y}(D) \geq 0$. Wyner and Ziv [126] proved that for jointly Gaussian memoryless sources and mean-squared error (MSE) distortion

[5] Alternatively, one may use $X = Y + E$ to show the dependency [132]. In such a case, the source sample X can be interpreted as the sum of the side information (Y) and of an innovation component (E). This model which is preferred in the distributed video coding literature will be referred to as the *reverse correlation channel*.

$R^{\mathrm{WZ}}_{X|Y}(D) = R_{X|Y}(D)$.

In the following, $R_{X|Y}(D)$ is used to develop lower and upper bounds for $R^{\mathrm{GBG}}_{X|Y}(D)$. Assuming that the position of impulses are revealed both to the encoder and decoder, the rate-distortion function can be obtained in a time division manner. Thus, for $D \leq \sigma_0^2$ we have

$$R^{\mathrm{GBG}}_{X|Y}(D) \geq \underline{\mathrm{R}}^{\mathrm{GBG}}_{X|Y}(D) = \sum_j p_j R_{X|Y,s_j}(D), \tag{3.3}$$

where $\sigma_{X|Y,s_j} = \sigma_j$, and $j \in \{0,1\}$.[6] It is also straightforward to see that

$$R^{\mathrm{GBG}}_{X|Y}(D) \leq \bar{R}^{\mathrm{GBG}}_{X|Y}(D) = R_{X|Y,s_1}(D). \tag{3.4}$$

By limiting the knowledge of the position of impulses to the decoder only, one may come up with tighter bounds. This is an interesting topic per se, but it is out of the scope of this work. Yet, to provide another benchmark level, we will evaluate the performance of the proposed systems when the position of impulses are known at the decoder.

We also study the Gauss-Markov sources as in many applications there is dependency between source samples. The rate distortion function of a Gauss-Markov process with a variance σ^2 and a correlation coefficient $0 \leq \rho < 1$ is given by [11, 123]

$$R(D) = \frac{1}{2}\log_2\left(\frac{(1-\rho^2)\sigma^2}{D}\right) \qquad \text{if} \quad D \leq \frac{1-\rho}{1+\rho}\sigma^2, \tag{3.5}$$

and, similar to (3.3) and (3.4), we can develop lower and upper bounds for the case where X is a Gauss-Markov source [43]. Note that for $D > \frac{1-\rho}{1+\rho}\sigma^2$ the rate distortion function can be evaluated numerically [123]. For that range, which corresponds to $R < \log_2(1+\rho)$, distortion is always less than what is evaluated by (3.5). Obviously, $\rho = 0$ gives a Gaussian source and (3.5) becomes identical to its Gaussian counterpart.

[6] For target distortion $D > \sigma_0^2$ we do not need to transmit in the absence of impulse and $\underline{\mathrm{R}}^{\mathrm{GBG}}_{X|Y} = p_1 R_{X|Y,s_1}(D_i)$ in which $D_i = p_0\sigma_0^2 + p_1 D$.

3.2 BCH-DFT Codes: Construction and Decoding

In this section, we study a class of real-number codes that are employed for binning throughout this dissertation, investigate some properties of their syndrome, and adapt their decoding algorithm to the Slepian-Wolf coding setup. These codes are a family of BCH codes in the real field whose parity-check matrix H and Generator matrix G are defined based on the DFT matrix; they are known as BCH-DFT codes, or simply DFT codes.

BCH-DFT codes [72] are linear block codes over the *real* or *complex* fields. Similar to other BCH codes, the spectrum of any codeword generated by an (n, k, t) code[7] is zero in a block of $d \triangleq n - k$ cyclically adjacent components, where $d + 1$ is the designed distance of the code [14]. The error correction capability of the code is, hence, given by $t = \lfloor \frac{d}{2} \rfloor$. We define DFT codes and introduce their basic properties in the following.

3.2.1 Encoding

Definition 3.1. An $n \times n$ DFT matrix is defined by

$$W_n \triangleq \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix}, \tag{3.6}$$

in which $\omega = e^{-j\frac{2\pi}{n}}$ [74]. Let W_n^H denote *conjugate transpose* of the DFT matrix. Since W_n is unitary, $W_n^H = W_n^{-1}$; i.e., W_n^H gives the inverse DFT (IDFT) matrix.

The generator matrix of an (n, k) DFT code [72], in general, consists of any k columns of the IDFT matrix of order n; the remaining $n-k$ columns of this matrix are used to build the *parity-check matrix*, as we will see shortly in this section. These codes are a family of *cyclic codes* over the complex field. Thus, their codewords satisfy certain spectral properties in the frequency domain [13, Chapter 4]. Within the class of DFT codes, there are BCH codes in the *complex* and *real* fields. The generator matrix of an (n, k) real BCH-DFT code

[7] For simplicity of notation, throughout this dissertation, we will use (n, k) instead of (n, k, t).

is commonly defined by [114, 41]

$$G = \sqrt{\frac{n}{k}} W_n^H \Sigma W_k, \tag{3.7}$$

where W_n and W_k are the DFT matrices of size n and k, and

$$\Sigma = \begin{pmatrix} I_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_\beta \end{pmatrix} \tag{3.8}$$

is an $n \times k$ matrix with $\alpha = \lceil \frac{n}{2} \rceil - \lfloor \frac{n-k}{2} \rfloor$[8] and $\alpha + \beta = k$ [116, 86, 100, 41]. I_α represents the identity matrix of size α and alpha is the frequency offset required to obtain a real matrix G. In other words, it identifies the indices in which zeros must be inserted in order to get the conjugacy constraint required to have a real-valued G [13, Fig. 4.2], [113]. A complex BCH-DFT code is obtained by removing W_k from (3.7). We can also remove the first constraint on α; that is, for a complex code, α can be any integer between 1 and k. Then, for any $\boldsymbol{u}$, this enforces the spectrum of the codeword

$$\boldsymbol{c} = G\boldsymbol{u}, \tag{3.9}$$

to have $n - k$ consecutive zeros, which is required for any BCH code [14].

The *parity-check matrix* H, both for real and complex BCH-DFT codes, is constructed by using the $n - k$ columns of W_n^H corresponding to the $n - k$ zero rows of Σ, and is then defined as

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \omega^{\alpha} & \omega^{\alpha+1} & \cdots & \omega^{n-\beta-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{\alpha(n-1)} & \omega^{(\alpha+1)(n-1)} & \cdots & \omega^{(n-\beta-1)(n-1)} \end{pmatrix}^H . \tag{3.10}$$

Obviously, H is an $(n - k) \times n$ matrix and, by virtue of the unitary property of W_n^H it is

[8] Knowing that n and k cannot be simultaneously even for a real DFT code [72], one can show that $\alpha = \lceil \frac{k+1}{2} \rceil$.

the null space of G, i.e.,

$$HG = \mathbf{0}_{n-k\times k} \tag{3.11}$$

both for real and complex codes. Each codeword of an (n, k) BCH-DFT code has

$$d \triangleq n - k \tag{3.12}$$

cyclically adjacent zeros in the frequency domain. These codes are *maximum distance separable* (MDS) codes with *minimum Hamming distance* $d_{\min} = d + 1$. They are, hence, capable of correcting up to $t = \lfloor \frac{d}{2} \rfloor$ errors.

In the rest of this chapter, we use the term "DFT code" in lieu of "real BCH-DFT code."

3.2.2 Decoding

Before introducing the decoding algorithm, we define some notation and basic concepts. Let $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ be the received vector (a noisy version of $\boldsymbol{c}$), where $\boldsymbol{c}$ is a codeword generated by (3.9). Suppose that $\boldsymbol{e}$ is an error vector with ν nonzero elements at positions $i_1, \ldots, i_\nu$; the magnitude of error at position i_p is e_{i_p}. Then, we can compute

$$\boldsymbol{s} = H\boldsymbol{r} = H(\boldsymbol{c} + \boldsymbol{e}) = H\boldsymbol{e}, \tag{3.13}$$

where $\boldsymbol{s} = [s_1,\ s_2, \ldots, s_d]^T$ is a complex vector with

$$s_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell}, \quad \ell = 1, \ldots, d, \tag{3.14}$$

in which $\alpha = \lceil \frac{k+1}{2} \rceil$ as defined in (3.8), $X_p = e^{\frac{-j2\pi i_p}{n}}$, and $p = 1, \ldots, \nu$. Next, we define the syndrome matrix

$$S_m = \begin{bmatrix} s_1 & s_2 & \ldots & s_{d-m+1} \\ s_2 & s_3 & \ldots & s_{d-m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_m & s_{m+1} & \ldots & s_d \end{bmatrix}, \tag{3.15}$$

for $\nu + 1 \leq m \leq d - \nu + 1$ [86]. Also, we define the covariance matrix as

$$R = S_m S_m^H. \tag{3.16}$$

For decoding, the extension of the well-known Peterson-Gorenstein-Zierler (PGZ) algorithm to the real field [14] can be used. This *coding-theoretic* decoding, aimed at detecting, localizing, and calculating the errors, works based on the syndrome of error. The algorithm is elaborated in Appendix A. We summarize the main steps of this algorithm, adapted for a DFT code of length n, in the following.

- **Error detection:** Determine the number of errors ν by constructing a syndrome matrix and finding its rank.
- **Error localization:** Find the coefficients $\Lambda_1, \ldots, \Lambda_\nu$ of the *error-locating polynomial* $\Lambda(x) = \prod_{i=1}^{\nu}(1 - xX_i^{-1})$ whose roots $X_1, \ldots, X_\nu$ are used to determine error locations; the errors are then in the locations $i_1, \ldots, i_\nu$ such that $X_1 = \omega^{i_1}, \ldots, X_\nu = \omega^{i_\nu}$ and $\omega = e^{-j\frac{2\pi}{n}}$.
- **Error calculation:** Finally, calculate the error magnitudes by solving a set of linear equations whose constants coefficients are powers of X_i.

In practice however, the received vector is distorted because of quantization. Let $\hat{\boldsymbol{c}}$ and $\boldsymbol{q}$ denote the quantized codeword and quantization noise so that $\hat{\boldsymbol{c}} = \boldsymbol{c} + \boldsymbol{q}$. Therefore, $\boldsymbol{r} = \hat{\boldsymbol{c}} + \boldsymbol{e}$ and its syndrome is no longer equal to the syndrome of error because

$$H\boldsymbol{r} = H(\boldsymbol{c} + \boldsymbol{q} + \boldsymbol{e}) = \boldsymbol{s}_q + \boldsymbol{s} = \tilde{\boldsymbol{s}}, \tag{3.17}$$

where $\boldsymbol{s}_q \equiv H\boldsymbol{q}$ and $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T$ is the quantization error. The elements of the distorted syndrome vector $\tilde{\boldsymbol{s}}$ can be written as

$$\tilde{s}_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell} + \frac{1}{\sqrt{n}} \sum_{p'=1}^{n} q_{i_{p'}} X_{p'}^{p'-1}. \tag{3.18}$$

The distorted syndrome matrix $\tilde{S}_m$ and the corresponding covariance matrix $\tilde{R} = \tilde{S}_m \tilde{S}_m^H$ are defined similar to (5.6) and (5.7) but for the distorted syndrome samples.

While the exact value of the error is determined neglecting quantization, the decoding becomes an estimation problem in the presence of quantization. Then, it is imperative to modify the PGZ algorithm to decode the errors reliably [14, 86, 100, 41, 113]. An alternative approach is to use the *subspace-based* decoding. In the remainder of this section, we discuss this problem and also improve the error detection and localization, by introducing a slightly different version of the existing methods.

3.2.3 Modified Subspace-Based Decoding

The subspace-based error localization outperforms the coding theoretic one [86, 115]; it can be integrated into DSC in a straightforward fashion once the syndrome of error is found [109]. In addition to doing that, in this section, we introduce a new method for error detection.

Error detection

For a given DFT code, we first fix an empirical threshold θ based on eigendecomposition of $\tilde{R}$ when the codewords are error-free, i.e. when only the quantization error exist. This threshold is on the magnitude of eigenvalues, rather than the determinant of $\tilde{R}$. Let $\lambda_{\max}$ denote the largest eigenvalue of $\tilde{R}$ for $m = t + 1$. We find θ such that, for a desired probability of correct detection p_d,

$$\Pr(\lambda_{\max} < \theta) \geq p_d. \tag{3.19}$$

Note that $\lambda_{\max}$ is a random variable and we need to estimate its probability distributed function (pdf) for this purpose. In practice, when errors can occur, we estimate the number of errors by the number of eigenvalues of $\tilde{R}$ greater than θ, as illustrated in Section 3.5. This one step estimation is better than the original estimation in the PGZ algorithm [14, 100], where the last row and column of S_t are removed until we come up with a non-singular matrix. The improvement comes from incorporating all syndrome samples, rather than some of them, for the decision making.

Ideally, we should set different thresholds depending on channel error powers; however, we choose one θ for all ranges to make the decoder simpler. Numerical results with different codes show that one suitable θ can be used for a wide range of error powers at the expense

of a slight MSE performance degradation. When we bring down the threshold θ, effectively we let the decoder detect more errors. On the contrary, if we increase θ we allow less error detection.

Error localization

The *subspace* or *coding-theoretic* error localizations can be used to find the coefficients $\Lambda_1, \ldots, \Lambda_\nu$ of the error-locating polynomial [86]. The subspace approach is, however, more general than the coding-theoretic approach in the sense that it can use up to $t+1-\nu$ degrees of freedom to localize ν errors, compared to just one degree of freedom in the coding-theoretic approach. This is because, the eigen-decomposition of the covariance matrix $\tilde{R} = \tilde{S}_m \tilde{S}_m^H$ results in two orthogonal subspaces, namely the *error* and *noise* subspaces. There are $m-\nu$ vectors in the noise subspace; these are used to localize errors leveraging a line spectral estimation method, e.g., the multiple signal classification (MUSIC) algorithm [96]. Effectively, each vector corresponds to one error-locating polynomial [115], and by using the MUSIC approach we are averaging these $m-\nu$ polynomials to reduce the effect of quantization noise. Hence, in the subspace method, we get a better error localization compared with the coding-theoretic approach which is solely based on one polynomial. The best result is then obtained for $m = t+1$ [86] for which the size of $\tilde{S}_m$ is either $(t+1)\times(t+1)$ or $(t+1)\times t$.

We apply the subspace-based error localization to the DSC, similar to that in channel coding. However, one should note that in the DSC the syndrome of error is computed in a different manner from that in channel coding; this will be elaborated in Section 3.3.

Error calculation

This last step is rather simple. Let H_e be the matrix consisting of the columns of H corresponding to error indices. The errors magnitude $\boldsymbol{E} = [e_{i_1}, e_{i_2}, \ldots, e_{i_\nu}]^T$ can be determined by solving

$$H_e \boldsymbol{E} = \tilde{\boldsymbol{s}}, \tag{3.20}$$

in a *least squares* sense, for example. This completes the error correction algorithm by calculating the error vector.

3.2.4 Performance Compared to Binary Codes

DFT codes by construction are capable of decreasing the quantization error. When there is no error, an (n, k) DFT code brings down the MSE below the quantization error level with a factor of $R_c = k/n$ [84,51]. This is also shown to be valid for channel errors [113], as long as the channel can be modeled as an additive noise. To appreciate this, one can consider the generator matrix of a DFT code as *analysis frame operator* of a tight frame [84]; it is known that frames are resilient to any additive noise, and tight frames reduce the MSE k/n times [65]. Hence, DFT codes can result in a MSE even less than the quantization error level whereas the MSE in a binary code is obviously lower-bounded by the quantization error level.

3.3 Wyner-Ziv Coding Using DFT Codes

In this section, we use DFT codes to do Wyner-Ziv coding in the real field. This is accomplished by using DFT codes for binning and transmitting the compressed signal, in the form of *syndrome* or *parity* samples, in a digital communication system. This methods will be called as syndrome-based and parity-based approaches, respectively. We study the syndrome-based method in this chapter; the parity-based method will be discussed in Chapter 4.

Let $\boldsymbol{x}$ be a sequence of real random variables $x_1 x_2 \dots x_n$, and $\boldsymbol{y}$ be a noisy version of $\boldsymbol{x}$ such that $y_i = x_i + e_i$, where e_i is continuous, i.i.d., and independent of x_i, as described in (3.2). The lower-case letters x, y, and e, respectively, are used to show the realization of the random variables X, Y, and E. Since $\boldsymbol{e}$ is continuous, this model precisely captures any variation of $\boldsymbol{x}$, so it can be used to model the dependency between $\boldsymbol{x}$ and $\boldsymbol{y}$ accurately. This correlation model is important, for example, in video coders that exploit Wyner-Ziv concepts, e.g., when the decoder builds side information via extrapolation of previously decoded frames or interpolation of key frames [10].

In this chapter, we study the GBG (and GE) correlation model and, for the sake of analysis, we assume that $\boldsymbol{e}$ contains up to t spikes (big errors with variance σ_i^2 in (3.2)) in each codeword.[9] For simplicity of presentation, the non-spiked errors are assumed to be zero in this section; this is as if we were using the GE model. However, when doing simulation

[9] When doing simulation, to make the assumptions more realistic, we drop the constraint that a codeword cannot have more that t spikes.

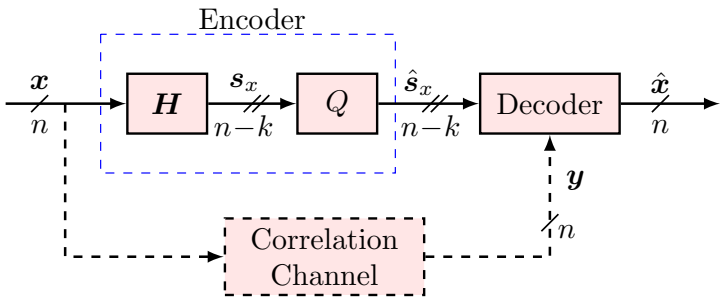


Figure 3.2 The Wyner-Ziv coding using DFT codes: Syndrome approach.

for the GBG model, we use (3.2) with $p_0 + p_1 = 1$ and a reasonably small p_1 so that, for $\sigma_i^2 \gg \sigma_0^2$, with high probability there are t or less spikes in each codeword. Obviously, the remaining elements of $\boldsymbol{e}$ are non-zero for the GBG model, as they are samples of a Gaussian distribution.

3.3.1 Syndrome Approach

Encoding

To compress an arbitrary sequence of data samples, we multiply it with the parity-check matrix H, defined in (3.10), to find the corresponding syndrome samples $\boldsymbol{s}_x = H\boldsymbol{x}$. The syndrome is then quantized ($\hat{\boldsymbol{s}}_x = \boldsymbol{s}_x + \boldsymbol{q}$) and transmitted over a noiseless digital communication system, as shown in Fig. 3.2. Note that $\boldsymbol{s}_x$, $\hat{\boldsymbol{s}}_x$ are both complex vectors of length $n - k$. Thus, it seems that to transmit each sample we need to send two real numbers, one for the real part and one for the imaginary part, which halves the compression ratio. However, we observe that the syndrome of a DFT code is symmetric, as stated below.

Lemma 3.1. *The syndrome of an (n, k) DFT code satisfies*

$$s_m = \begin{cases} s^*_{d-m+1}, & \text{if } k \text{ is odd,} \\ s^*_{d-m}, & \text{if } k \text{ is even,} \end{cases} \tag{3.21}$$

for $m = 1, \ldots, d$ and $d \triangleq n - k$.

Proof. The proof is straightforward; we show this for odd k and leave the other case to the

reader. Since $\alpha = \lceil \frac{k+1}{2} \rceil$ and $d = n - k$, using (5.2), for odd k we can write

$$\begin{aligned} s_{d-m+1} &= \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\frac{k-1}{2}+n-k-m+1} \\ &= \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\frac{-k+1}{2}-m} = s_m^*. \end{aligned} \tag{3.22}$$

Note that $X_p^n = 1$, for any p. □

The above lemma implies that, for any d, it suffices to know the first $\lceil \frac{d}{2} \rceil$ syndrome samples. We know that syndromes are complex numbers in general; however, from $s_{d-m+1} = s_m^*$ it is clear that if d is an odd number, s_m is real for $m = \lceil \frac{d}{2} \rceil$. Therefore, for an (n, k) code with odd k, transmitting $n - k$ real numbers suffices. This results in a compression ratio of $\eta_s = \frac{n-k}{n}$ for the binning step and thus for the encoder. Yet, one can check that for even k, we have to transmit $n - k + 1$ real samples, which incurs a slight loss in compression. It is however negligible for large n.

Decoding

The decoder estimates the input sequence from the received syndrome and side information $\boldsymbol{y}$. To this end, it needs to evaluate the syndrome of (correlation) channel errors. This can be simply done by subtracting the received syndrome from the syndrome of the side information. Then, neglecting the quantization error, we obtain,

$$\boldsymbol{s}_e = \boldsymbol{s}_y - \boldsymbol{s}_x, \tag{3.23}$$

and $\boldsymbol{s}_e$ can be used to precisely estimate the error vector, as described in Section 3.2.2. In practice, however, the decoder knows $\hat{\boldsymbol{s}}_x = \boldsymbol{s}_x + \boldsymbol{q}$ rather than $\boldsymbol{s}_{\boldsymbol{x}}$. Therefore, only a distorted syndrome of error is available, i.e.,

$$\tilde{\boldsymbol{s}}_e = \boldsymbol{s}_y - \hat{\boldsymbol{s}}_x = \boldsymbol{s}_e - \boldsymbol{q}. \tag{3.24}$$

Hence, using the PGZ algorithm, error correction is accomplished based on (3.24). Note that, having computed the syndrome of error, decoding algorithm in a DSC using DFT codes is exactly the same as that in the channel coding problem. This is different from

DSC techniques in the binary field which usually require a slight modification in the corresponding channel coding algorithm to be customized for DSC.

3.4 Rate-Adaptive DSC

The proposed DSC schemes is suitable for low-delay coding as by using short DFT codes the MSE between the transmitted and reconstructed signals can be reasonably small, even less than quantization error [117]. On the other hand, a code with short block length is vulnerable to the variations of channel. Expectedly, the performance of the proposed systems degrades when the correlation between the sources changes. When the statistical dependency between the sources varies or is not known at the encoder, a *rate-adaptive* system with feedback is an appealing solution; these systems are popular in the transmission of non-ergodic data, like video [119]. Rate-adaptive DSC based on binary codes, e.g., puncturing the parity or syndrome bits of turbo and LDPC codes, have been proposed in [119, 103]. Although puncturing the syndrome samples can be used for rate-adaption in our system, it severely affects the decoding algorithm and substantially increases the MSE.

We propose an alternative, more efficient method to perform DSC in a rate-adaptive fashion. This algorithm, which works based on transmitting additional syndrome samples to decoder, is based on a straightforward extension of the subspace decoding, proposed in [115] to enhance the error localization of quantized DFT codes. Having access to more syndrome samples makes it possible to form a bigger syndrome matrix, which in turn enlarges the dimension of the noise subspace and thus increases the number of error-locating polynomials. Then, instead of using one polynomial for decoding, one can superimpose many polynomials to diminish the effect of quantization error and improve error localization to a large extent. We will discuss this algorithm in more details in Chapter 5. The extended subspace algorithm is naturally suitable for rate adaption in the syndrome-based DSC.[10] This is because the decoder is able to compute extra syndromes of error upon receiving some new syndrome samples of the data vector. To this end, the encoder and decoder are required to agree on another parity check matrix $\bar{H}$, which we call *extended parity check matrix* and form it based on those k columns of W_n^H (the IDFT matrix of order n) that are

[10] It can, however, be applied to the the parity-based DSC in the same way.

not used to build H. More precisely, similar to (3.10), we have

$$\bar{H} \triangleq \begin{pmatrix} 1 & \cdots & 1 \\ \omega^{n-\beta} & \cdots & \omega^{n+\alpha-1} \\ \vdots & \ddots & \vdots \\ \omega^{(n-\beta)(n-1)} & \cdots & \omega^{(n+\alpha-1)(n-1)} \end{pmatrix}^{H}. \tag{3.25}$$

Then, similar to the syndrome vector $\boldsymbol{s}$, the extended syndrome vector $\bar{\boldsymbol{s}}$ is defined as

$$\bar{\boldsymbol{s}} = \bar{H}\boldsymbol{r} = \bar{H}\boldsymbol{c} + \bar{H}\boldsymbol{e}. \tag{3.26}$$

But $\bar{H}\boldsymbol{c}$ is not necessarily zero and should be compensated for. This can be done naturally, based on the following rate adaptation algorithm:

1. The decoder requests some extra syndrome samples based on the estimated number of errors, e.g. when $\hat{\nu} > t$
2. The encoder computes $\bar{\boldsymbol{s}}_x = \bar{H}\boldsymbol{x}$ and transmits it to the decoder sample by sample
3. The decoder computes $\bar{\boldsymbol{s}}_y = \bar{H}\boldsymbol{y} = \bar{\boldsymbol{s}}_x + \bar{\boldsymbol{s}}_e$ to find $\bar{\boldsymbol{s}}_e = \bar{\boldsymbol{s}}_y - \bar{\boldsymbol{s}}_x$ and append it to $\boldsymbol{s}_e$ in order to use the extended subspace decoding algorithm.

Therefore the encoder transmits a short syndrome based on an (n, k) code and augments it with additional samples, if more samples are required. The algorithm is incremental so that there is no need to re-encode the sources when extra syndrome samples are requested.

In the above algorithm, the decision to request more syndrome samples is based on the estimated number of errors and the decoder does not need to decode the whole block. On the contrary, the decision to request extra syndrome samples in a rate-adaptive DSC systems based on LDPC codes, and any binary codes in general, is based on the decoding of the whole long block which is computationally less efficient and requires more time. Another importance of the rate adaptation is to find out whether or not the value of the error detection threshold θ is appropriate, especially if we have no estimation of the CEQNR. We know that by bringing the threshold down, we can increase the probability of error detection. On the contrary, if we increase the threshold fewer errors will be detected and thus rate adaptation is required to a lesser extent. With this in mind, if there are too many requests for extra syndromes we may increase θ; conversely, if there are not many

requests for rate adaptation we can bring the threshold down as long as it improves the end-to-end distortion.

3.5 Simulation Results

We evaluate the performance of the proposed systems using a Gauss-Markov source $\boldsymbol{x}$ with mean zero, variance one, and correlation coefficient $0 \leq \rho < 1$ generated by

$$x_i = \sqrt{1-\rho^2} z_i + \rho x_{i-1}, \tag{3.27}$$

in which $\boldsymbol{z} = \{z_i\}$ is a zero-mean, unit-variance i.i.d. Gaussian process. The side-information is generated using the forward correlation model as presented in Section 3.1.2. That is, for the GBG model there are both background noise and impulsive errors while for the GE model only the impulsive errors exist. The background noise, generated by $\mathcal{N}(0, \sigma_0^2)$, affects each and every sample of $\boldsymbol{x}$ while the impulsive error is added to a fraction of samples of $\boldsymbol{x}$. The amplitude of impulses is generated based on $\mathcal{N}(0, \sigma_i^2)$ and their position is also selected randomly. To compress the data, $\boldsymbol{x}$ is binned using a (n, k) DFT code. The compressed vector, i.e., the syndrome samples, is then quantized with a b-bit uniform quantizer ($4 \leq b \leq 6$) and transmitted over a communication channel. The quantizer step size Δ depends on the effective range of the compressed data (i.e., the syndrome or parity) and thus it can be different for the syndrome and parity approaches. It also depends on the percent of the data we would like to be in the range of quantizer.[11] For an input range of $[-m\sigma, m\sigma]$ we get $\Delta = \frac{2m\sigma}{2^b}$. The decoder detects, localizes, and decodes errors. We compute the MSE between the transmitted and reconstructed data, to measurer end-to-end distortion. We define channel-error-to-quantization-noise ratio (CEQNR) as

$$\text{CEQNR} \triangleq \sigma_e^2/\sigma_q^2, \tag{3.28}$$

where $\sigma_q^2 = \frac{\Delta^2}{12}$ and $\sigma_e = \sigma_i$. Thus, a larger CEQNR corresponds to a larger variation of the impulses amplitude than that of the quantization noise power. In all simulations, we use 10^6 input samples for each CEQNR, which corresponds to $\frac{10^6}{n}$ code blocks. We vary the CEQNR from 10dB to 40dB and plot the resulting MSE.

[11] For one thing, $[-4\sigma_x, 4\sigma_x]$ contains more than 99.99% of the input data, where σ_x^2 is the variance of x.

Before showing the simulation results for the reconstruction distortion, we elaborate how to choose the parameter θ and the way it affects the MSE. As we explained in Section 3.2.3, θ is used to estimate the number of errors, and from (3.19) we need to have the pdf of $\lambda_{\max}$ to find θ for specific p_d. So, the first step is to evaluate the pdf of $\lambda_{\max}$; this is done based on the eigendecomposition of $\tilde{R}$ for the quantized code where there is only quantization error. We then need to fix the value of p_d to effectively estimate the number of errors. Numerical results show that $p_d = 90 - 99\%$ is a good initial value for θ. Increasing θ will decrease $\hat{\nu}$, the estimated number of errors, whereas decreasing θ will increase that. Meanwhile, by decreasing θ the probability of false detection increases. So there is a trade off between correct detection and false alarm. By changing θ in reasonably small steps and evaluating the end-to-end distortion, one can find its optimal value so as to minimize the MSE.[12] The optimal value of θ varies based on the CEQNR, even though this variation is small for a small range of CEQNR. Despite that, in our simulations, for each code, we use one θ at all CEQNRs so that the decoder does not need to know the value of CEQNR. If CEQNR is know at the decoder, we can assign a more accurate θ to get a slightly better MSE.

At very low CEQNRs, although error localization is poor, the MSE is still very low because compared to the quantization error, the errors can be so small that the algorithm does not detect (and localize) them. Instead, it may occasionally localize and correct quantization errors. Note that, even if no errors are localized and corrected, the MSE is still very small as the errors are negligible at very low CEQNRs. Additionally, recall that the MSE is always reduced with a factor of $R_c = \frac{k}{n}$, in an (n, k) DFT code.

To familiarize the reader with the decoding steps, in Figure 3.3, we detail the decoding steps for a $(10, 5)$ DFT code for the GE correlation model. First, based on Figure 3.3(a), the threshold $\theta_0 = 0.0024$ is found for $p_d = 99\%$. Next, this is used to estimate ν in Figure 3.3(b). The estimated ν is subsequently used to find the location of errors, both for the PGZ and subspace-based error localization, in Figure 3.3(c). Then, the output of Figure 3.3(c), for the subspace method, is fed to the last step to find the magnitude of errors and correct them. The resulting MSE is depicted in Figure 3.3(d). In the remainder of this section, we will focus only on the MSE performance without plotting the results for the intermediate steps (i.e., error detection and localization). We should point out that $\theta_0 = 0.0024$ gives an initial value and needs to be optimized depending on the CEQNR.

[12] Note that finding the optimal value of θ can be done off-line or using pilot data.

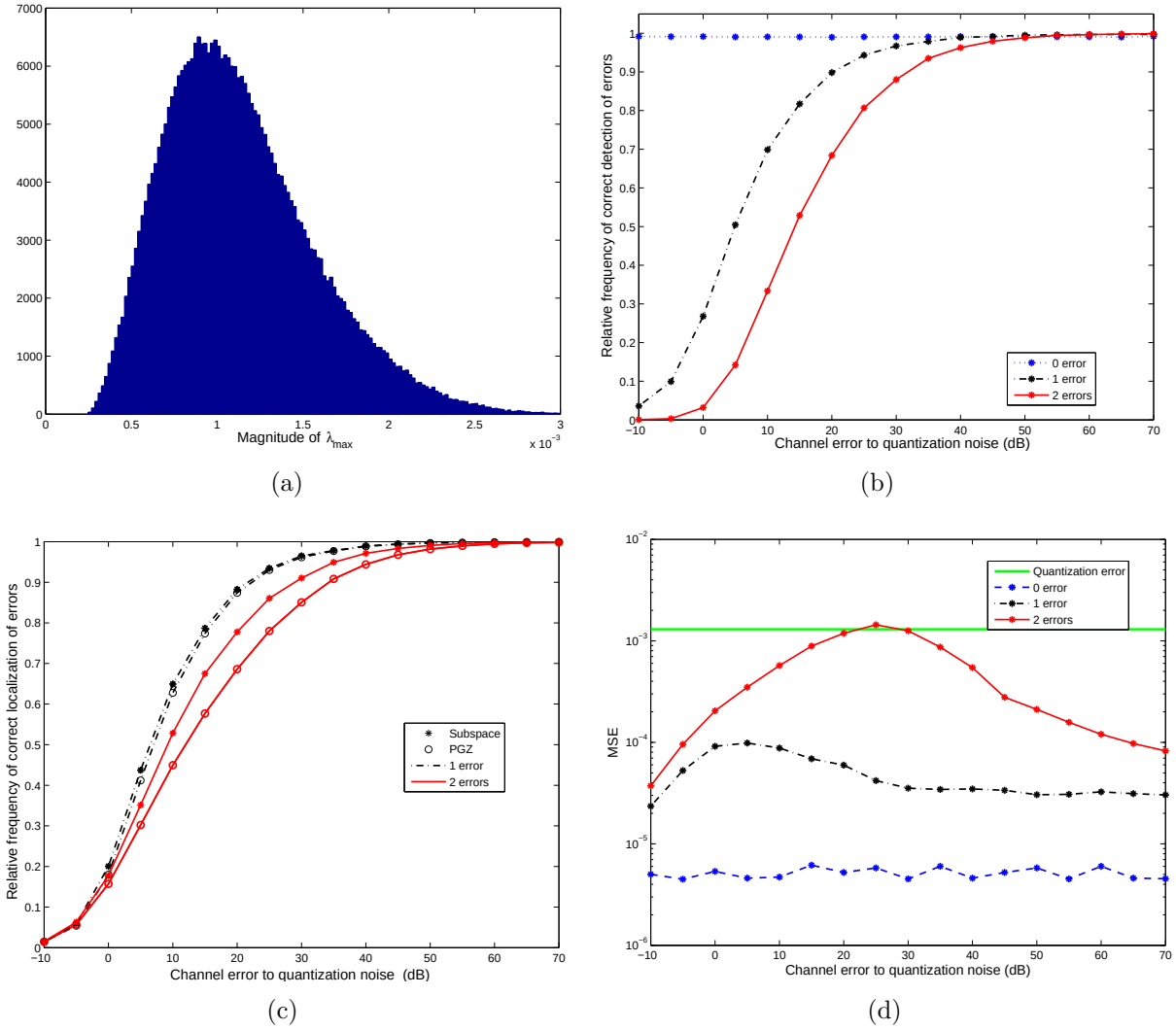


Figure 3.3 Performance evaluation of the syndrome-based DSC proposed in Figure 3.2, for a $(10, 5)$ DFT code and different number of errors in each block of the code. The results are based on a Gauss-Markov source with $\rho = 0.9$ and a quantizer with $m = 4$ and $b = 6$. (a) Histogram of $\lambda_{\max}(\tilde{R})$ for the quantized code. This is used to set a threshold for detection. (b) Probability of correct detection of errors for $\theta = 0.0024$. (c) Probability of correct localization of errors based on Figure 3.3(b). (d) The end-to-end distortion for subspace-based error localization given in Figure 3.3(c).

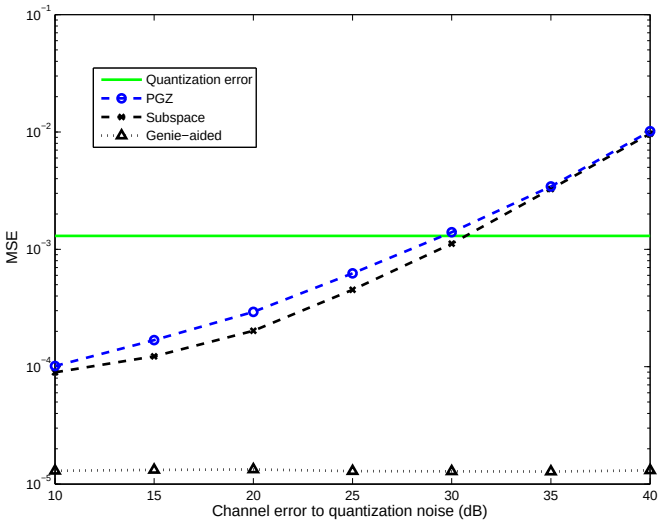


Figure 3.4 The performance of the PGZ and subspace (MUSIC) decoding for a $(10, 5)$ DFT code and the GE correlation model with $p_1 = 0.04$. The plots are based on a Gauss-Markov source with $\rho = 0.9$, a 6-bit quantizer, and $\theta = 0.0064$.

So we change θ_0 by a step of 0.001 and check the resulting MSE. Numerical results show that $\theta = 0.0064$, which corresponds to $pd = 0.9995$, result in minimum average MSE at CEQNR between 10dB to 40dB. Clearly, if we change the range of CEQNR the optimal θ will differ. However, its impact on the average MSE is very small. We begin with the syndrome approach and probe its performance by varying the CEQNR for different codes with different compression ratios. In Figure 3.4, we compare the performance of decoding based on the PGZ and subspace error localization. To put our results in perspective, we also calculate the MSE for the *genie-aided* error localization in which the true location of errors is given to the decoder. The genie-aided decoding reflects the ideal performance of the proposed DSC system, when least square decoding is used in the last step of the decoding (i.e., the error calculation). It can be seen that the performance of the subspace-based decoding is better than that of coding-theoretic one. This is because subspace-based error localization performs better than the coding-theoretic method. However, as CEQNR increases this gain gradually decreases because the performance of the two methods becomes very much the same, as it can be seen in Figure 3.3(c). A similar pattern is observed for other codes and in general the subspace-based error localization outperforms the coding-theoretic one [86, 115]. Therefore in the remaining plots we only represent the MSE based

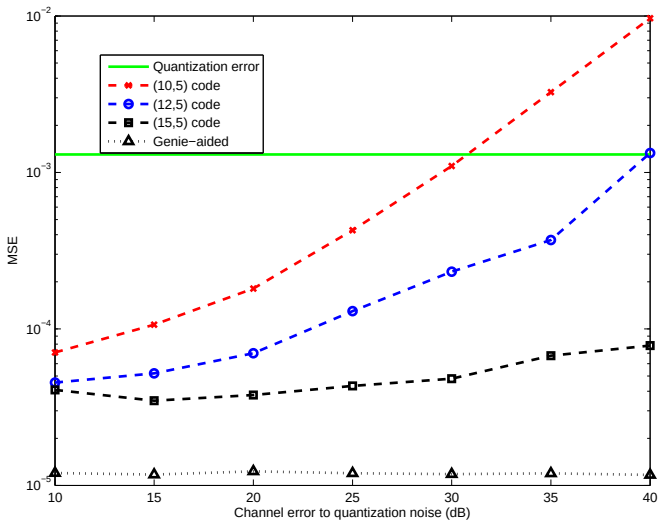


Figure 3.5 The MSE performance of the syndrome-based DSC, with subspace-based decoding, using three different DFT codes. The other parameters are the same as those in Figure 3.4.

on the subspace method.

Figure 3.5 presents the MSE performance for three different codes, namely $(10, 5)$, $(12, 5)$ and $(15, 5)$ codes. Expectedly, as the code rate increases, which implies a lower compression ratio, the MSE goes down. Next we evaluate the performance of rate adaptation on the MSE in Figure 3.6. For rate adaptation, the decoder requests extra syndrome samples if $\hat{\nu} \geq t$. Upon receiving the new syndrome samples, the decoder applies the extended subspace method for error localization and uses this new result for error correction. As it can be seen from Figure 3.6, rate adaptation can noticeably improve the performance at the expense of negligible increase in the effective code rate η_e, which is defined as the ratio of total number of transmitted syndrome and extended samples to total number of input samples (number of codewords multiplied by n). This is particularly important when the codeword length n is short. With a small n, the probability of having more than t errors in one block can be relatively high. By requesting more syndrome samples and using the extended subspace decoding algorithm, a rate-adaptive system effectively increases the error correction capability of the code, and thus improves the overall performance. Another important use of rate adaptation is to compensate for the fixed threshold at different CEQNR. As explained earlier, ideally, we should have different thresholds at each

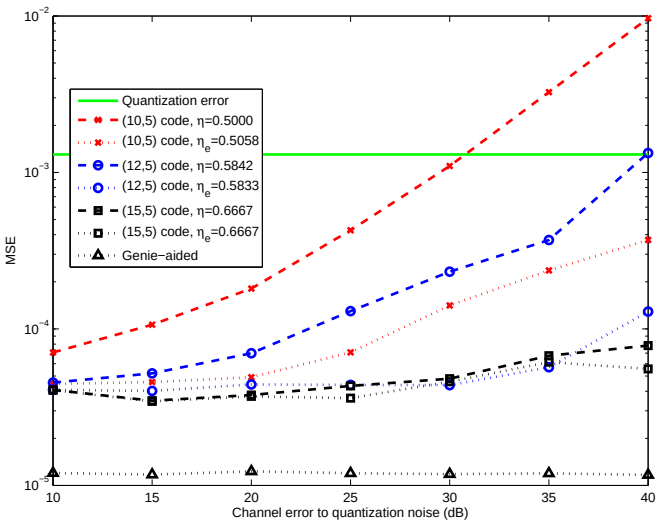


Figure 3.6 The performance of rate-adaptive DSC versus non-adaptive approach for three different DFT codes with the GE correlation model. The effective rates (η_e) for the rate-adaptive approach are a little more than their original code rates (η) while the gain in the MSE is noticeably high. For the $(15, 5)$ code $\eta_e = 0.66669$ and $\eta = 0.66667$, thus there is not much gain from rate-adaption.

CEQNR even though, for simplicity of decoding, we use one θ for all ranges of CEQNR. When a fixed threshold is used at different CEQNRs, effectively we let the decoder detect more errors at higher CEQNRs rather than the lower ones. Then, rate adaptation comes in handy when there are more than t errors in one block. Even if there are t errors in one block, one might use a few more syndrome samples to enhance the decoding performance. This improves the end-to-end distortion at the expense of increasing the rate. Finally, the most important application of rate adaptation is for the case where the statistics of the channel vary. In all above situations it sounds reasonable to use an aggressive code and exploit rate adaptation when required.

We now evaluate the performance of the syndrome-based DSC for the GBG model and $(10, 5)$, $(12, 5)$ codes. The results are shown in Figure 3.7 for $\sigma_0 = 0.01\sigma_i$ at different CEQNRs. It can be seen that rate adaptation decreases the MSE in both cases and the average rate increase due to that is still very small. Without further estimation block after Slepian-Wolf coding, the MSE performance of DSC based on binary codes is limited to

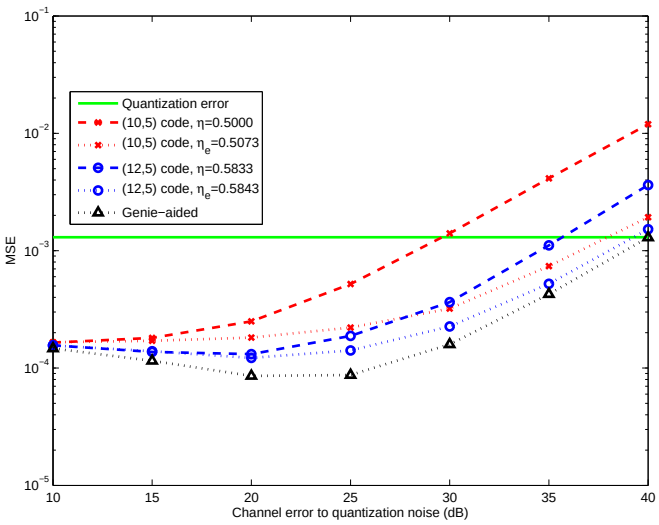


Figure 3.7 The MSE performance of the rate-adaptive DSC versus non-adaptive approach for a Gauss-Markov source with $\rho = 0.9$ and the GBG correlation model with $\sigma_0 = 0.01\sigma_e, p_1 = 0.04$. The average rate increases due to the rate-adaption are still very small while reduction in the MSE is considerable.

the quantization error level whereas the proposed DSC scheme break through this limit. However, if the final MSE is based on a joint estimation error between the quantized source and the decoder side information, it can be smaller than the quantization error in the former cases. Thus, the quantization error may not be a fair benchmark in those cases. To evaluate the effectiveness of the new framework, the MSE can be compared against the asymptotic information-theoretic bounds available in the literature.

Finally, we plot the distortion rate region for the syndrome-based DSC and the GBG model in Figure 3.8 and Figure 3.9, for $b = 6$ and $b = 4$ respectively. The goal is to compare the rate distortion performance of the system with the asymptotic bounds. Since the rate distortion function for the GBG model is not known we compare the results with the upper and lower bounds introduced in Section 3.1.2 in (3.3) and (3.4). It should be emphasized that the lower bound is based on the assumption that both encoder and decoder know the true position of the errors (impulses); this can make the lower bound very loose. In other words, when the encoder and/or decoder does not have this knowledge the lower bound is expected to be tighter; i.e., it will shift upward and get closer to the achievable points. In both figures we have applied rate-adaptation to find the achievable points. To visualize the encouraging potential of the proposed framework, in Figure 3.9 we have also shown the

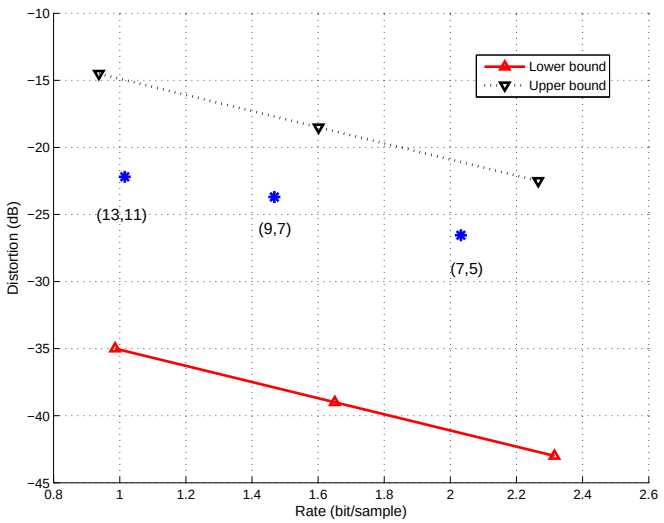


Figure 3.8 Comparison of distortion-rate function with asymptotic bounds for coding a Gauss-Markov source X with $\sigma_X = 1$, $\rho = 0.9$ quantized with a 6-bit quantizer where the GBG correlation model with $p_1 = 0.04$, $\sigma_0 = 0.05\sigma_e$ at CEQNR= 25dB (or $\sigma_0 = 0.0321$ and $\sigma_e = 0.6412$ for $b = 6$) is used. The asterisks show the achievable points based on $(13, 11)$, $(9, 7)$, $(7, 5)$ codes, respectively from left to right; we have also applied rate-adaptation so that the corresponding rate for each code is $\eta_e b$ bits/sample.

performance of our system for the genie-aided (perfect) error localization.

Seeing that we do not use the ideal Slepian-Wolf coding assumption ($n \to \infty$), the gap between performance of our schemes and theoretical rate-distortion function is expected to be more than that for capacity-approaching codes. However, it should be noted that capacity-approaching channel codes introduce significant delay if one strives to approach the capacity of the channel with a very low probability of transmission error. Hence, those are out of the question for delay-sensitive systems. In that case, it would be best to use channel codes of low rate and focus on achieving a very low probability of error. The system we introduced is a low-delay system which works well with reasonably high-rate codes. This is because the block length in our system is nb bits (n samples) and n is usually much smaller than the code length in LDPC and turbo codes. For example, in Figure 3.9 all points are for $n \leq 12$ and $b = 4$ which results in block length $L \leq 48$ at the worst case, while it is common to have codes of length 10^5 in DSC based on LDPC codes.

In Figure 3.9 we also envisage the performance of our system for the genie-aided (perfect)

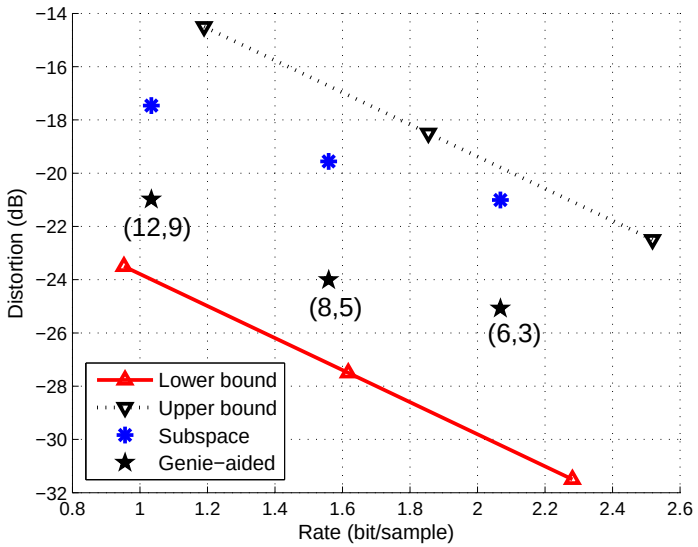


Figure 3.9 The distortion-rate performance and bounds for coding a Gauss-Markov source X with $\sigma_X = 1$, $\rho = 0.9$ quantized with a 4-bit quantizer. The graph is based on the GBG correlation model with $\sigma_0 = 0.05\sigma_e$ at CEQNR= 25dB, or equivalently $\sigma_0 = 0.1282$ and $\sigma_e = 2.5647$ for $b = 4$. The asterisks show the achievable points based on $(12, 9)$, $(8, 5)$, $(6, 3)$ codes, respectively from left to right, while the stars show the achievable points for the same codes assuming a perfect error localization. Rate-adaptation has been applied in both cases.

error localization. It indicates the great potential of the proposed framework to become much closer to the theoretical lower bound and encourages investigation on better error localization algorithms. In addition, we know that for sources with memory there is dependency between the samples and vector quantization (VQ) can exploit this dependency. The gain resulting from the memory advantage can be considerable at high rates [123]; therefore, it is reasonable to expect that VQ would improve the rate distortion performance of our system.

Remark 3.1. The simulation results, and in general the framework we developed in this chapter, are based on the assumption that the measured signals ($\boldsymbol{x}$ and $\boldsymbol{y}$) are continuous-valued. One might argue that the measurements are anyhow discrete to a certain level, due to the floating point precision, and question the performance of the system for discrete-valued signals. From the decoding algorithm of BCH-DFT codes it is evident that these codes work for discrete input data, as this is a special case of continuous (real-valued) data.

Similarly, the framework we developed in this chapter considers real inputs so it should work for discrete-valued signals, as the latter is a subset of the former. Simulations results, with different levels of quantization of $\boldsymbol{x}$ and $\boldsymbol{y}$, proves the efficacy of the proposed framework for discrete-valued signals as well. It is worth mentioning that by decreasing b, the number of quantization bits per sample, the mean-squared reconstruction error decreases slightly. However, the difference is not noticeable for large values of b. Specifically, it is not significant when we quantize the signals with $b = 32$, $b = 16$, and even $b = 10$.

3.6 Comparison with DSC Based on Binary Codes

There are several ways to compare the efficiency and performance of the proposed system with that of DSC based on binary codes. This comparison can be carried out for equal encoding/decoding computational complexity, equal encoding/decoding processor complexity, equal decoding delay and memory requirement, and equal end-to-end reconstruction performance. While the last criterion has been the main comparison method in this chapter, and this dissertation in general, in the following we address some other metrics. We will compare our results with DSC based on LDPC codes.

Let N be the length of one LDPC block encoded by an (N, K) code. Similarly, let n be the length of one block generated by an (n, k) BCH-DFT code. We assume each sample is quantized by a b-bit quantizer. By now, the reader should have notice that usually $N \gg n$.[13] For ease of comparison, we assume $N = mbn$, where m is an integer number.

3.6.1 Decoding Delay

Let T_s denote the time between the arrival of successive samples; that is, the sample rate is $\frac{1}{T_s}$ and the bit rate is $\frac{b}{T_s}$. Depending on the data rate and processing speed, different cases can arise in decoding of block codes. At the best case, each block is decoded by the time of the next block arrival [27]. In such a scenario, observing that we use $N - K$

[13] This is because the performance of LDPC codes is poor for small block lengths; it improves as N increases. On the contrary, BCH-DFT codes work well with short block lengths, and their performance deteriorates for very large n, as error localization is challenging for large n. To see this, recall that the error locator polynomial has ν roots out of n possible roots ($X_1, \ldots, X_n$, where $X_{i_p} = \omega^{i_p}$, $i_p \in \{1, \ldots, n\}$, and $\omega = e^{-j\frac{2\pi}{n}}$), and this n roots are uniformly placed over the unit circle. Obviously, as n goes up the distance between two successive roots decreases; for quantized codes, this increases the probability of wrong decision during error localization. To improve error localization, one should use a finer quantizer; i.e, to increase the number of bits/sample which results in a higher CEQNR.

syndrome bits to decode a block of LDPC code, the decoding delay for an LDPC-based DSC is $\Delta_{\text{io}}^{\text{LDPC}} = (N-K)T_b$. Similarly, if we assume that each block of a BCH-DFT code is decoded before the arrival of the next block, it can be seen that $\Delta_{\text{io}}^{\text{BCH-DFT}} = (n-k)T_s$; observe that the BCH-DFT decoder works based on samples rather than bits. For $R = \frac{1}{2}$ we get $\Delta_{\text{io}}^{\text{LDPC}} = \frac{N}{bn}\Delta_{\text{io}}^{\text{BCH-DFT}}$. That is, for $N > bn$ a DSC system based on LDPC codes impose more delay than the proposed scheme in this chapter. Note that the difference is typically very high as $N \gg n$.

3.6.2 Computational Complexity

It is possible to measure the encoding complexity in different ways. In numerical linear algebra, the cost of an algorithm is often expressed by total number of floating-point operations (flops[14]) required to carry it out. Multiplication of an $m \times n$ matrix with an $n \times p$ matrix costs $mp(2n-1) \approx 2mnp$ flops [18]. Here we compare the complexity based on the operations cost. Alternatively, one can get a good feeling of relative complexities by means of the encoding time for a sequence of data [8].

Encoder

The complexity of computing the syndrome of the input vector of length n is $2n(n-k)$ flops. Therefore, for a block of N bits we need $\frac{N}{bn}2n(n-k) = \frac{2N}{b}(n-k)$ flops. This is equal to $\frac{Nn}{b}$ flops for codes of $R = \frac{1}{2}$, which has $\mathcal{O}(N)$ complexity for small n. The encoding complexity of LDPC codes is quadratic in the code length ($\mathcal{O}(N^2)$), if it is performed by matrix multiplication, though this can be reduced to $\mathcal{O}(N)$ with other algorithms [62, 71]. The complexity of quantization should be also considered in the complexity analysis of the encoder. Given the same quantizer, this complexity is less for the proposed framework as we quantize less amount of data (a compressed version of that) compared to the conventional approach in which quantization is before compression.

Decoder

The complexity of decoding is based on the computing the syndrome of error, eigendecomposition, and least square solution. To find the syndrome of error, we need to compute

[14] A flop is defined as one addition, subtraction, multiplication, or division of two floating-point numbers [18].

the syndrome of side information and subtract it from the syndrome of input data. The first operation costs $2n(n-k)$ flops and the second one is $n-k$ flops; thus the total complexity is approximately $2n(n-k)$ flops. Then forming the covariance matrix R in (3.16) (for $m = \lceil \frac{d}{2} \rceil = \lceil \frac{n-k}{2} \rceil$) and its eigendecomposition respectively require $2d^3$ and $\frac{1}{3}d^3$ flops. Finally, the flops required for pseudo-inters (least squares solution) of H_e in (A.13) is very small compared to d^3 because this size of the matrix is $(n-k) \times \nu$ and ν is usually very small. Hence,

$$C_{\mathrm{D}}^{\mathrm{BCH-DFT}} \approx 2n(n-k) + \frac{7}{3}(\lceil \frac{n-k}{2} \rceil)^3.$$

Then, for a block of N bits we need $\frac{N}{bn} C_{\mathrm{D}}^{\mathrm{BCH-DFT}}$ flops.

On the other hand, the complexity of decoding for a block of LDPC code, $C_{\mathrm{D}}^{\mathrm{LDPC}}$, is equal to [27]

$$C_{\mathrm{D}}^{\mathrm{LDPC}} = ((1-R)J_c + J_v)IN,$$

in which J_c and J_v, respectively, are the are the number of computations required for a check node and variable node and I is the number of iterations. Next, one can compare the decoding complexity of the above algorithms for N bits (one block of LDPC code). It can be seen that for typical values of parameters (e.g., $10^4 \leq N \leq 10^6, 50 \leq I \leq 300$ [69, 129]) and $n < 30$ it can be seen that the complexity of the system we have proposed in this chapter is less than that of LDPC-based DSC.

3.7 Summary

In this Chapter, we have introduced a new framework for the distributed lossy source coding, in general, and the Wyner-Ziv coding, in particular. The idea is to do binning before quantizing the continuous-valued signal, as opposed to the conventional approach where binning is done after quantization. By doing binning in the real field, the virtual correlation channel can be modeled more accurately and the quantization error can be compensated for when there is no error. In the new paradigm, Wyner-Ziv coding is realized by cascading a Slepian-Wolf encoder with a quantizer. We employed BCH-DFT codes to do binning in the real field and we introduced the syndrome-based approach in this chapter. This scheme maps short source blocks into channel blocks, and thus it is appropriate for low-delay coding. Finally, we introduced rate adaptation in the new framework to compensate for the variation in the statistical dependency between the sources and increase

the error correction capability of the code if required. Due to rate adaptation, the MSE can decrease significantly for very small compression loss. A more accurate algorithm for error localization is a key to further improve the reconstruction error. From simulation results, we conclude that the proposed system can improve the reconstruction error even using short codes, so it can become viable in real-world scenarios where low-delay communication is required.

In the next chapter, we will study an alternative approach for distributed source coding based on BCH-DFT codes in which the parity samples are used to represent the compressed signal. A parity-based approach works even if the transmission medium is noisy, as we show in Chapter 4.

Chapter 4

Distributed Joint Source and Channel Coding

We introduced the syndrome-based lossy distributed source coding and Wyner-Ziv coding based on DFT codes in Chapter 3, in which syndrome samples are quantized and transmitted over a *noiseless* channel. Alternatively, to achieve compression in a DSC setting, one can encode a signal using a real DFT code and transmit only the parity samples. This chapter [1] considers parity-based distributed source coding and its extension to distributed joint source-channel coding (DJSCC). To do DJSCC, we use a single DFT code both to *compress* the signal and *protect* it against channel variations. To do parity-based DSC and DJSCC we introduce the notion of systematic DFT frames (or codes.)

In Section 4.1, we present parity-based distributed source coding. DSC, in general, implies a *separate* source and channel coding. Motivated by transmission over non-ergodic channels and/or real-time communication, for which the separation theorem breaks down, in Section 4.2 we extend the parity-based DSC to carry out source and channel coding jointly, and in a distributed manner. Developing parity-based DSC and DJSCC we introduce the notion of systematic DFT codes (frames), in this chapter. Looking from a frame theory perspective, an (n, k) systematic DFT code is a frame that includes the identity matrix of size k as a subframe. We show that for an (n, k) DFT code there exist $\binom{n}{k}$ systematic codes, each of which may result in a different reconstruction error. With this in mind, in the remainder of this chapter, we will be looking for systematic DFT frames with

[1] The material in this chapter has been presented in [114, 109, 111] and published in [116].

the "best" performance, from the minimum mean-squared reconstruction error sense. To do so, in Section 4.3, we first present the basic definitions and a few fundamental lemmas that will be used later in the chapter. We also introduce DFT frames and set the ground to study the extreme eigenvalues of their subframes, in this section. Section 4.4 motivates the work in this chapter by introducing systematic DFT frames and their application. Some fundamental results on the the extreme eigenvalues of DFT frames and their subframes are presented in Section 4.5. Sections 4.6 and 4.7 are devoted to the evaluation of reconstruction error and to the classification of systematic frames based on reconstruction performance. Simulation results, both for DSC and DJSCC, are presented in Section 4.8. We conclude in Section 4.9.

4.1 Parity-Based DSC Using DFT Codes

The syndrome-based Wyner-Ziv coding is straightforward, but it is not clear how we can use it for noisy transmission. In the sequel, we explore an alternative approach for Wyner-Ziv coding, namely parity-based approach, which as we will see in the following, is much more suited to distributed source coding over noisy channels.

4.1.1 Encoding

To compress $\boldsymbol{x}$, the encoder generates the corresponding parity sequence $\boldsymbol{p}$ with $n-k$ samples. The parity is then quantized and transmitted, as shown in Figure 4.1, instead of transmitting the input data. To this end, we need to find a systematic generator matrix G_{sys}, as G in (3.7) is not in the systematic form.

A first approach is to find H_{sys} and build G_{sys} based on that [111]. Let H be partitioned as $H = [H_{n-k\times k} \mid H_{n-k}]$. Since H_{n-k} is a submatrix of the Vandermonde matrix W_n, H_{n-k}^{-1} exists and we can write

$$H_{\text{sys}} = H_{n-k}^{-1} H = [-P \mid I_{n-k}], \tag{4.1}$$

in which $P \triangleq -H_{n-k}^{-1} H_{n-k\times k}$ is an $(n-k)\times k$ matrix, and I_{n-k} is an identity matrix of

size $n-k$. The systematic generator matrix corresponding to H_{sys} is given by

$$G_{\text{sys}} = \begin{bmatrix} I_k \\ P \end{bmatrix}. \tag{4.2}$$

By inspection, $H_{\text{sys}}G_{\text{sys}} = \mathbf{0}$. It is also easy to check that

$$HG_{\text{sys}} = \mathbf{0}. \tag{4.3}$$

Therefore, we do not need to calculate H_{sys} and the same parity-check matrix H can be used for decoding in the parity approach.

Another, simpler, way is to obtain a systematic generator matrix directly from G. Let G_k be a square matrix of size k composed of k arbitrary rows of G. We see that G_k is invertible because using (3.7) any $k \times k$ submatrix of G can be represented as product of a Vandermonde matrix and the DFT matrix W_k. This is also proven using a different approach in [84], where it is shown that any subframe of G is a frame, and its rank is equal to k. Hence, a systematic generator matrix is given by

$$G_{\text{sys}} = GG_k^{-1}. \tag{4.4}$$

Besides, from $HG = \mathbf{0}$, it is clear that

$$HG_{\text{sys}} = \mathbf{0}. \tag{4.5}$$

Therefore, we do not need to calculate H_{sys}, and the same parity-check matrix H can be used for decoding in the parity approach. It is also obvious that G_{sys} is a real matrix. The question that remains to be answered is whether G_{sys} corresponds to a BCH code. To generate a BCH code, G_{sys} must have $n-k$ consecutive zeros in the transform domain. The Fourier transform of this matrix $W_nG_{\text{sys}} = (W_nG)G_k^{-1}$ satisfies the required condition because W_nG, the Fourier transform of original matrix, satisfies that. That is, G_{sys} corresponds to a BCH code in the real field.

It should be emphasized that one can arbitrarily choose the rows of G_k in (4.4); this yields $\binom{n}{k}$ systematic matrices for each (n,k) DFT code. Although any of the corresponding systematic codes can be used for encoding, the dynamic range of parity samples depends on

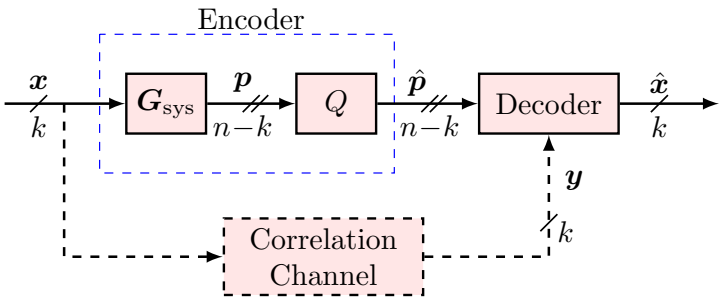


Figure 4.1 The Wyner-Ziv coding using DFT codes: Parity approach.

the relative position of the rows chosen to constitute G_k [116]. In Section 4.7, we prove that when using these systematic frames for error correction, the mean-squared reconstruction error is minimized when the systematic rows are chosen as evenly spaced as possible. In the extreme scenario, where the systematic rows are equally spaced, the systematic frame is also *tight*. This is realized only when n is an integer multiple of k. Such a frame lends itself well to minimizing the reconstruction error [51,64,84,20]. Since the parity samples are error-free, the optimal G_{sys} reduces the occurrence of successive errors in one codeword. A similar idea is used in [33] to inhibits closely spaced sinusoids in oversampled DFT frames to increase spectral compression.

Finally, seeing that parity samples are real numbers, using an (n, k) DFT code, a compression ratio of $\eta_p = \frac{n-k}{k}$ is achieved. Obviously, a compression ratio of $\frac{n-k}{n}$ is achievable if we use a $(2n-k, n)$ DFT code.

4.1.2 Decoding

A parity decoder estimates the input sequence from the received parity and side information $\boldsymbol{y}$. Similar to the syndrome approach presented in Section 3.3.1, at the decoder, we need to find the syndrome of (correlation) channel errors. To do so, we append the parity to the side information and form a vector of length n whose syndrome, neglecting quantization, is equal to the syndrome of errors. That is,

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{p} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{0} \end{bmatrix} = G_{\text{sys}}\boldsymbol{x} + \boldsymbol{e}', \tag{4.6}$$

and $\boldsymbol{e}' = [\boldsymbol{e} \mid \boldsymbol{0}]^T$. Hence,

$$\boldsymbol{s}_z = \boldsymbol{s}_{e'}. \tag{4.7}$$

Similarly, when quantization is involved ($\hat{\boldsymbol{p}} = \boldsymbol{p} + \boldsymbol{q}$), we get

$$\tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{y} \\ \hat{\boldsymbol{p}} \end{bmatrix} = \boldsymbol{z} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q} \end{bmatrix} = G_{\text{sys}}\boldsymbol{x} + \boldsymbol{e}' + \boldsymbol{q}', \tag{4.8}$$

and

$$\boldsymbol{s}_{\tilde{z}} = \boldsymbol{s}_{e'} + \boldsymbol{s}_{q'}, \tag{4.9}$$

where, $\boldsymbol{q}' = [\boldsymbol{q} \mid \boldsymbol{0}]^T$, and $\boldsymbol{s}_{q'} \equiv H\boldsymbol{q}'$. Therefore, we obtain a distorted version of error syndrome. In both cases, the rest of the algorithm, which is based on the syndrome of error, is similar to that in the channel coding problem using DFT codes, as explained in Section 3.2.3.

Error localization algorithms for the parity-based DSC [111] can be further improved using the fact that parity samples are error-free. As parity samples are transmitted over a noiseless channel, the error locations, in the codewords, are restricted to the systematic samples. Therefore, we can exclude the set of roots corresponding to the location of the parity samples. We call this method *adapted* error localization. Furthermore, it makes sense to use a code with evenly-spaced parity samples so as to maximize the location of error-free and error-prone samples in the codewords. Such a code maximizes the distance between the error-prone roots of the code; hence, it helps decrease the probability of incorrect decision.

4.1.3 Comparison Between the Parity- and Syndrome-Based DSC

Rate

As it was shown earlier, using an (n, k) code the compression ratio in the syndrome and parity approaches is $\eta_s = \frac{n-k}{n}$ and $\eta_p = \frac{n-k}{k}$, respectively. Hence, for a given code, the parity approach is $\frac{k}{n} < 1$ times less efficient than the syndrome approach. Conversely, we can find two different codes that result in the same compression ratio η, say $\frac{n-k}{n}$. We mentioned that in the parity approach, a $(2n-k, n)$ code can be used for this matter,

whereas an (n, k) DFT code gives the desired compression ratio in the syndrome approach. Thus, for a given compression ratio the syndrome approach implies a code with smaller rate compared to the code required in the parity approach.

Delay

Assuming the delay imposed by a system to decode each block of code depends on the number of samples to be transmitted, the delay in the parity approach is larger than in the syndrome approach. More precisely, for the compression ratio of $\eta = \frac{n-k}{n}$, the delay in the former approach is proportional to $n - k$ while the it is proportional to n in the latter approach. These are the length of syndrome and parity vectors, respectively.

Performance

From frame theory, we know that DFT frames are tight, and an (n, k) tight frame reduces the quantization error with a factor of $R_c = \frac{k}{n}$ [65, 84, 51]. This result is extended to errors, given that channel can be modeled by an additive noise [113]. The MSE performance of systematic DFT frames also linearly depends on the code rate, though they are not necessarily tight [116]. Therefore, for codes with the same error correction capability, the lower the code rate the better the error correction performance. This implies a better performance for syndrome-based DSC. Further, a $(2n - k, n)$ code has $n - k$ roots more than an (n, k) code on the unit circle; hence, the roots are closer to each other and the probability of incorrect localization of errors increases.

Additionally, from rate-distortion theory we know that the rate required to transmit a Gaussian source logarithmically increases with the source variance [28]. Thus, in a system that uses a real-number code for encoding, since coding is performed before quantization, the variance of transmitted sequence depends on the behavior of the encoding matrix. In the syndrome-based DSC we transmit $\boldsymbol{s} = H\boldsymbol{x}$. One can check that the standard deviations of elements of s and x are equal, i.e., $\sigma_{\boldsymbol{s}} = \sigma_{\boldsymbol{x}}$. Unlike that, in the parity-based DSC, the variance of the parity samples is larger than that of the inputs. More precisely, in an (n, k) systematic DFT code, if $\boldsymbol{c} = G_{\text{sys}}\boldsymbol{x}$, then $\sigma^2_{\boldsymbol{c}} = \gamma\sigma^2_{\boldsymbol{x}}$ where $\gamma = \frac{1}{n}\operatorname{tr}\left(G^H_{\text{sys}}G_{\text{sys}}\right) \geq 1$ [116]. Since we can write $\boldsymbol{c} = [\boldsymbol{x} \mid \boldsymbol{p}]^T$, we have

$$\sigma^2_{\boldsymbol{p}} = \frac{\gamma n - k}{n - k}\sigma^2_{\boldsymbol{x}} \geq \sigma^2_{\boldsymbol{x}}. \tag{4.10}$$

From [116, Theorem 7], we know that the smallest $\sigma_{\boldsymbol{p}}$ for a given DFT code is achieved when the parity samples, in the corresponding codewords, are located as "evenly" as possible. Furthermore, the equality in (4.10) can be achieved only when $n = Mk$ where M is an integer greater than or equal to 2. Thus, such a code cannot be used in the parity-based DSC as $M \geq 2\,(n \geq 2k)$ result in signal "expansion" rather than compression. Obviously though, we can use such a code when expansion is allowed, for example in joint source-channel coding, as we will see the next section.

Considering the above arguments, it is reasonable to expect the syndrome-based DSC perform better than the parity-based one, for a given code or a fixed compression ratio. This is verified numerically in Section 4.8. The parity-based DSC, however, has other advantages. For example, it can be easily extended to distributed joint source-channel coding, as explained in the following section.

4.2 Distributed Joint Source and Channel Coding

The concept of lossy DSC and Wyner-Ziv coding using DFT codes was explained both for the syndrome and parity approaches in Sections 3.3 and 4.1, respectively, where syndrome and parity samples are quantized and transmitted over a *noiseless* channel. This implies *separate* source and channel coding. Although simple, the *separation theorem* is based on several assumptions, such as the source and channel coders not being constrained in terms of complexity and delay, which do not hold in many situations. It breaks down, for example, for non-ergodic channels and real-time communication. In such cases, it makes sense to integrate the design of the source and channel coder systems, because *joint source-channel coding* (JSCC) can perform better given a fixed complexity and/or delay constraints. Likewise, distributed JSCC (DJSCC) has been shown to outperform separate distributed source and channel coding in some practical cases [83]. DJSCC has been addressed in [2, 83, 70, 47, 133, 44], using different binary codes.

In this section, we extend the parity-based Wyner-Ziv coding of analog sources to the case where errors in the transmission can happen. Thus, we introduce distributed JSSC of analog correlated sources in the analog domain. Specifically, we consider transmission corrupted by *impulsive noise*. The usage of this model in our work is mainly motivated by implementation of wireless sensor networks in power substations [54, 94]. The impulsive noise is prevalent in power substations since it is created by partial discharges, corona noise

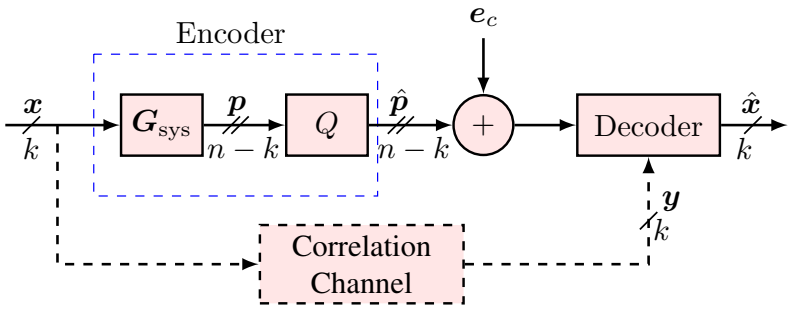


Figure 4.2 Joint source-channel coding (JSCC) with side information at the decoder based on DFT codes. This scheme can be straightforwardly extended to distributed JSCC.

and electrical arcs, hosted by high-voltage equipment such as transformers, bushings, power lines, circuit breakers and switch-gear [94]. Impulsive noise is also found in many other urban areas, e.g. in hospitals, homes, and offices, as it is generated by microwave ovens, fluorescent lights, photocopier machines, and automobile ignition [15, 130]. Therefore, sensor networks implemented in various applications, like in health and home, also experience an impulsive transmission medium.

The magnitude of the impulses is assumed to have a Gaussian distribution; hence, the Gaussian-Erasure channel is used to model the transmission channel, as well. To do this, we use a single DFT code both to *compress* $\boldsymbol{x}$ and *protect* it against channel variations; this gives rise to a new framework for DJSCC, in which quantization is performed after doing JSCC in the analog domain. This scheme directly maps short source blocks into channel blocks, and thus it is well suited to low-delay coding.

4.2.1 Coding and Compression

To compress and protect $\boldsymbol{x}$, the encoder generates the parity sequence $\boldsymbol{p}$ of $n-k$ samples, with respect to a good systematic DFT code. The parity is then quantized and transmitted over a noisy channel, as shown in Figure 4.2. To keep the dynamic range of parity samples as small as possible, we make use of optimal systematic DFT codes, proposed in [116]. This increases the efficiency of the system for a fixed number of quantization levels. Using an (n, k) DFT code a total compression ratio of $\eta = \frac{n-k}{k}$ is achieved. Obviously, if $n < 2k$ compression is possible ($\eta < 1$). However, since there is little redundancy the end-to-end

distortion could be high. Conversely, a code with $n > 2k$ ($\eta > 1$) expands input sequence by adding *soft redundancy* to protect it in a noisy channel.

4.2.2 Decoding

Let $\tilde{\boldsymbol{p}} = \hat{\boldsymbol{p}} + \boldsymbol{e}_c$ be the received parity vector which is distorted by quantization error $\boldsymbol{q}$ ($\hat{\boldsymbol{p}} = \boldsymbol{p} + \boldsymbol{q}$) and channel error $\boldsymbol{e}_c$. Also, let $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}_v$ denote side information where $\boldsymbol{e}_v$ represents the error due to the "virtual" correlation channel. The objective of the decoder is to estimate the input sequence from the received parity and side information. Although we only need to determine $\boldsymbol{e}_v$, effectively it is required to find both $\boldsymbol{e}_v$ and $\boldsymbol{e}_c$. From an error correction point of view, this is equal to finding the error vector $\boldsymbol{e} = [\boldsymbol{e}_v \quad \boldsymbol{e}_c]^T$ that affects the codeword $[\boldsymbol{x} \quad \boldsymbol{p}]^T$. Hence, to find the syndrome of error at the decoder, we append the parity $\tilde{\boldsymbol{p}}$ to the side information $\boldsymbol{y}$ and form $\tilde{\boldsymbol{z}}$, a valid codeword perturbed by quantization and channel errors,

$$\tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{p} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_v \\ \boldsymbol{e}_c \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q} \end{bmatrix} = G_{\text{sys}}\boldsymbol{x} + \boldsymbol{e} + \boldsymbol{q}'. \tag{4.11}$$

Multiplying both sides by H, we obtain

$$\boldsymbol{s}_{\tilde{z}} = \boldsymbol{s}_e + \boldsymbol{s}_{q'}, \tag{4.12}$$

where $\boldsymbol{s}_e \equiv H\boldsymbol{e}$ and $\boldsymbol{s}_{q'} \equiv H\boldsymbol{q}'$. Again, we use the GE model with $q_2 = 0$ in (3.2) to generate $\boldsymbol{e}$. It should be emphasized that for $\boldsymbol{q} = \boldsymbol{0}$, error vector can be determined exactly, as long as the number of errors is not greater than t. In practice, quantization is also involved, and we obtain only a distorted version of error syndrome. Knowing the syndrome of error, we use the error detection and localization algorithm, explained in Section 3.2.2, to find and correct error.

Although the extension of parity-based DSC to DJSCC is straightforward, it is not clear how to do this for syndrome-based DSC. This is because, in the syndrome-based DSC with a noisy transmission, the decoder can only form a noisy version of syndrome. More precisely, it can only obtain $\boldsymbol{s}_e + \boldsymbol{e}_c$, where $\boldsymbol{s}_e$ is the difference between the transmitted syndrome and syndrome of side information, i.e., $\boldsymbol{s}_e = \boldsymbol{s}_y - \boldsymbol{s}_x$, and $\boldsymbol{e}_c$ is the transmission channel error. However, with $\boldsymbol{s}_e + \boldsymbol{e}_c$, and $\boldsymbol{e}_c \neq 0$, the rank of the syndrome matrix S_t

is not equal to ν, even if there is no quantization error. Therefore, both the PGZ and subspace-based methods fail to correctly find the number and location of errors.

Before presenting numerical result for parity-based DSC and DJSCC in Section 4.8, we study some properties of the systematic DFT frames using a frame theory notion. By examining their extreme eigenvalues, we show that, unlike DFT frames, systematic DFT frames are not necessarily *tight*. Then, we come up with conditions for which these frames can be tight, and thus, minimize the mean squared reconstruction error. in either case, the best and worst systematic frames are established in the minimum mean-squared reconstruction error sense. Eigenvalues of DFT frames and their subframes play a pivotal role in this work. Particularly, we derive some bounds on the extreme eigenvalues DFT subframes which are used to prove most of the results.

4.3 Frame Theory and DFT Codes

Frames, "redundant" set of vectors used for signal representation, are increasingly found in signal processing applications. Frames are more general than bases as frames are complete but not necessarily linearly independent. A *basis*, on the contrary, is a set of vectors used to "uniquely" represent a vector as a linear combination of basis elements. Frames are generally motivated by applications requiring some level of *redundancy*, and they offer flexibility in design, resilience to additive noise (including quantization error), robustness to erasure (loss), and numerical stability of reconstruction.

From the frame theory perspective, DFT codes are harmonic tight frames. In the absence of erasure, tight frames minimize the mean-squared error (MSE) between the transmitted and received signals [51, 84, 64]. The MSE is the ultimate measure of performance in many digital communication systems where a quantized analog signal is transmitted. Frames are naturally robust to transmission loss since they provide an overcomplete expansion of signals [51, 20, 64, 84, 65].

Although it is straightforward to construct systematic DFT frames, we prove that systematic "tight" DFT frames exist only for specific frame sizes. More precisely, we show that a systematic frame is tight if and only if data (systematic) samples are circularly equally spaced, in the codewords generated by that frame. When such a frame does not exist, we will be looking for systematic DFT frames with the "best" performance, from the minimum mean-squared reconstruction error sense. We also demonstrate which systematic

frames are the "worst" in this sense. In addition, we show that circular shift and reversal of the vectors in a DFT frame does not change the eigenvalues of the frame operator. We use these properties to categorize different systematic frames, generated from an (n, k) DFT frame, based on their performance. We first introduce the definitions and some basic results which are frequently used in the remainder of this chapter.

4.3.1 Definitions and preliminaries

Definition 4.1. A spanning family of n vectors $F = \{\boldsymbol{f}_i\}_{i=1}^n$ in a k-dimensional complex vector space $\mathbb{C}^k$ is called a *frame* if there exist $0 < a \leq b$ such that for any $\boldsymbol{x} \in \mathbb{C}^k$

$$a\|\boldsymbol{x}\|^2 \leq \sum_{i=1}^{n} |\langle \boldsymbol{x}, \boldsymbol{f}_i \rangle|^2 \leq b\|\boldsymbol{x}\|^2, \tag{4.13}$$

where $\langle \boldsymbol{x}, \boldsymbol{f}_i \rangle$ denotes the inner product of $\boldsymbol{x}$ and $\boldsymbol{f}_i$ and gives the ith coefficient for the frame expansion of $\boldsymbol{x}$ [64,20,65]. a and b are called *frame bounds*; they, respectively, ensure that the vectors span the space, and the basis expansion converges. A frame is *tight* if $a = b$. *Uniform* or *equal-norm* frames are frames with same norm for all elements, i.e., $\|\boldsymbol{f}_i\| = \|\boldsymbol{f}_j\|$, for $i, j = 1, \ldots, n$.

Definition 4.2. An $n \times n$ Vandermonde matrix with unit complex entries is defined by

$$W \triangleq \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ e^{j\theta_1} & e^{j\theta_2} & \cdots & e^{j\theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(n-1)\theta_1} & e^{j(n-1)\theta_2} & \cdots & e^{j(n-1)\theta_n} \end{pmatrix}, \tag{4.14}$$

in which $\theta_p \in [0, 2\pi)$ and $\theta_p \neq \theta_q$ for $p \neq q$, $1 \leq p, q \leq n$. If $\theta_p = \frac{2\pi}{n}(p-1)$, W becomes the well-known IDFT matrix [74]. For this Vandermonde matrix we can write [106], [107]

$$\det(WW^H) = |\det(W)|^2 = \frac{1}{n^n} \prod_{1 \leq p < q \leq n} |e^{i\theta_p} - e^{i\theta_q}|^2. \tag{4.15}$$

Central to this work are the properties of the *eigenvalues* of V^HV or VV^H, in which V is a submatrix of a DFT matrix.[2] Hence, we recall some bounds on the eigenvalues of

[2]Note that eigenvalues of V^HV and VV^H are equal for a square V; also, V^HV and VV^H have the same

Hermitian matrices which are used in this chapter. Let A be a Hermitian $k \times k$ matrix with real eigenvalues $\{\lambda_1(A), \ldots, \lambda_k(A)\}$ which are collectively called the *spectrum* of A, and assume $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_k(A)$. Schur-Horn inequalities show to what extent the eigenvalues of a Hermitian matrix constraint its diagonal entries.

Proposition 4.1. *Schur-Horn inequalities [97]*
Let A be a Hermitian $k \times k$ matrix with real eigenvalues $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_k(A)$. Then, for any $1 \leq i_1 < i_2 < \cdots < i_l \leq k$,

$$\lambda_{k-l+1}(A) + \cdots + \lambda_k(A) \leq a_{i_1 i_1} + \cdots + a_{i_l i_l} \leq \lambda_1(A) + \cdots + \lambda_l(A), \tag{4.16}$$

where $a_{11}, \ldots, a_{kk}$ are the diagonal entries of A. Particularly, for $l = 1$ and $l = k$ we obtain

$$\lambda_k(A) \leq a_{11} \leq \lambda_1(A), \tag{4.17}$$

$$\sum_{i=1}^{k} \lambda_i(A) = \sum_{i=1}^{k} a_{ii}. \tag{4.18}$$

Another basic question in linear algebra asks the degree to which the eigenvalues of two Hermitian matrices constrain the eigenvalues of their sum. Weyl's theorem gives an answer to this question in the following set of inequalities.

Proposition 4.2. *Weyl inequalities [97]*
Let A and B be two Hermitian $k \times k$ matrices with spectrums $\{\lambda_1(A), \ldots, \lambda_k(A)\}$ and $\{\lambda_1(B), \ldots, \lambda_k(B)\}$, respectively. Then, for $i, j \leq k$, we have

$$\lambda_i(A + B) \leq \lambda_j(A) + \lambda_{i-j+1}(B) \qquad \text{for} \quad j \leq i, \tag{4.19}$$

$$\lambda_i(A + B) \geq \lambda_j(A) + \lambda_{k+i-j}(B) \qquad \text{for} \quad j \geq i. \tag{4.20}$$

Corollary 4.1. *If $A + B = \gamma I_k, \gamma > 0$, where A and B are Hermitian matrices, then $\lambda_j(A) + \lambda_{k-j+1}(B) = \gamma$.*

Proof. It suffices to set $i = k$ and $i = 1$ respectively in (4.19) and (4.20), and use $\lambda_k(A + B) = \lambda_1(A + B) = \gamma$ which is obtained from $A + B = \gamma I_k$. □

nonzero eigenvalues for a non-square V.

Lemma 4.1. *Let A and B be two Hermitian $k \times k$ matrices and suppose that, for every $1 \leq i, j \leq k$, $A_{i,j} = e^{j\theta_i} B_{i,j}$; then $A^H A$ and $B^H B$ have the same spectrum.*

Proof. The proof is immediate using Lemma 3 in [107] since $(A^H A)_{i,j} = \frac{e^{j\theta_i}}{e^{j\theta_i}} (B^H B)_{i,j}$; i.e., $A^H A = B^H B$. □

4.3.2 Connection to Frame Theory

BCH-DFT codes are *linear block codes* over the complex field whose parity-check matrix H is defined based on the DFT matrix. The generator matrix of an (n, k) *real BCH-DFT code*[3] is defined in (3.7) in which W_l represents the DFT matrix of size l, and Σ, as defined in (3.8), inserts $n - k$ consecutive zeros to each codeword in the frequency domain which ensures having a BCH code [14, 72]. One can check that $\Sigma^H \Sigma = I_k$, and $\Sigma\Sigma^H$ is an $n \times n$ matrix given by

$$\Sigma\Sigma^H = \begin{pmatrix} I_\alpha & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_\beta \end{pmatrix}. \tag{4.21}$$

Remark 4.1. Removing W_k from (3.7) we end up with a complex G, representing a *complex BCH-DFT code*. In such a code, α and β can be any nonnegative integers such that $\alpha + \beta = k$.

The generator matrix G in (3.7) can be viewed as an *analysis frame operator*. In this view, a real BCH-DFT code is a rotation of the well-known *harmonic frames* [20, 65], and a complex BCH-DFT code is basically a harmonic frame. The latter can be understood by removing W_k from (3.7) which results in a complex BCH-DFT code, on the one hand, and the analysis frame operator of a harmonic frame, on the other hand. The former is then evident as W_k is a rotation matrix. Further, it is easy to see that the *frame operator* $G^H G$ and *Gramian* GG^H are equal to

$$G^H G = \frac{n}{k} I_k, \tag{4.22}$$

$$GG^H = \frac{n}{k} W_n^H \Sigma\Sigma^H W_n. \tag{4.23}$$

[3] Real BCH-DFT codes do not exist when n and k are both even [72].

The following lemma presents some properties of the frame operator and relevant matrices which are crucial for our results in this chapter.

Lemma 4.2. *Let $G_{p\times k}$ be a matrix consisting of p arbitrary rows of G defined by* (3.7)*. Then, the following statements hold:*
i. GG^H is a Toeplitz and circulant matrix
ii. $G_{p\times k}G_{p\times k}^H, 1 < p < n$ is a Toeplitz matrix
iii. All principal diagonal entries of $G_{p\times k}G_{p\times k}^H, 1 \leq p \leq n$ are equal to 1*.*

Proof. Let $a_{r,s}$ be the (r, s) entry of the matrix GG^H then it can readily be shown that

$$a_{r,s} = \frac{1}{k}\sum_{m=0}^{\alpha-1} e^{jm(\theta_r-\theta_s)} + \frac{1}{k}\sum_{m=n-\beta}^{n-1} e^{jm(\theta_r-\theta_s)}, \tag{4.24}$$

in which $\theta_x = \frac{2\pi}{n}(x-1)$. From this equation, it is clear that $a_{r,s} = a_{r+i,s+i}$; that is, the elements of each diagonal are equal, which means that GG^H is a *Toeplitz* matrix. In addition, we can check that $a_{r,n} = a_{r+1,1}$, i.e., the last entry in each row is equal to the first entry of the next row. This proves that the Toeplitz matrix GG^H is *circulant* as well [53]. Also, a quick look at (4.24) reveals that the elements of the principal diagonal ($r = s$) are equal to 1. Similarly, one can see that for any $1 < p < n$, the square matrix $G_{p\times k}G_{p\times k}^H$ is also a Toeplitz matrix; it is not necessarily circulant, however. □

Considering Remark 4.1, one can check that (4.24) is also valid for complex BCH-DFT codes. Note that, α and β are less constrained for these codes, as mentioned in Remark 4.1.

Remark 4.2. Lemma 4.2 also holds for complex BCH-DFT codes.

Further, in a DFT frame, in general, the $n-k$ zero rows of Σ are not required to be successive if they are not designed for error correction. That is any matrix that can be rearranged as $[I_k \mid \mathbf{0}_{k\times n-k}]^T$ may represent Σ. Then, $\Sigma\Sigma^H$ is not necessarily in the form given in (4.21); it can be any square matrix of size n with k nonzero elements equal to 1, arbitrary located on the main diagonal. Then, again Lemma 4.2 holds because $a_{rs} = \frac{1}{k}\sum_{i=0}^{k-1} e^{jm_i(\theta_r-\theta_s)}$ and $m_i \in \{1, \ldots, n\}$.

Remark 4.3. Lemma 4.2 holds for all DFT frames.

4.4 Systematic DFT Frames

In general, every sample in the codewords of a DFT frame is a linear combination of all data samples of the input block, i.e., the data samples do not appear explicitly in the codewords. A specific method of encoding, known as *systematic encoding*, leaves the data samples unchanged. These unchanged samples can be exhibited in any component of the codeword, therefore:

Definition 4.3. An (n, k) frame is said to be systematic if its analysis frame operator includes I_k as a subframe.

4.4.1 Motivation and Applications

In the context of channel coding, there is a special interest in *systematic codes* [14] since the input data is embedded in the encoded output which simplifies the encoding and decoding algorithms. For example, in systematic convolutional codes data can be read directly if no errors are made, or in case only parity bits are affect in an *erasure channel*. Systematic codes are also used in parity-based distributed source coding (DSC) techniques, e.g., DSC that uses turbo codes for compression [9, 45, 2].

The new framework that we have developed for DSC exploits BCH-DFT codes for DSC [111]. There are syndrome- and parity-based approaches to implement DSC, as we explained in Sections 3.3 and 4.1; the compression is achieved by representing the input data with fewer samples, which are a linear combination of the input samples. To do so, in the latter approach the encoder generates parity samples with respect to a systematic DFT code. The parity is then quantized and transmitted over a noiseless channel. Assuming the asymmetric DSC [126], where one source is available at the decoder as side information, the decoder looks for the closest vector to the side information, among the vectors whose parity is equal to the received one. The parity-based approach is worthwhile as the parity of a real DFT code is a real vector contrary to its syndrome which is complex. More importantly, to accomplish DJSCC only the parity-based approach is known to be applicable [117]. On the other hand, the parity-based approach mandates systematic DFT codes and is the main motivation for what we present in the following.

4.4.2 Construction

In view of Definition 4.3, the systematic generator matrix for a real BCH-DFT code can be obtained by (4.4) in which G_k is a submatrix (subframe [84]) of G including k arbitrary rows of G. Note that G_k is invertible since it can be represented as

$$G_k = \sqrt{\frac{n}{k}} W_{k\times n}^H \Sigma W_k = V_k^H W_k, \tag{4.25}$$

in which $V_k^H \triangleq \sqrt{\frac{n}{k}} W_{k\times n}^H \Sigma$ and W_k are invertible as they are Vandermonde and DFT matrices, respectively. Obviously, this argument is valid if W_k is removed and/or when the $n-k$ zero rows of Σ are not successive. This indicates that any k rows of a DFT frame make a *basis* of $\mathbb{C}^k$ and proves that G_k^{-1} and thus systematic DFT frames exist for any DFT frame.

Remark 4.4. From the above discussion and Remark 4.3 one can see that what we prove in the remainder of this chapter is valid for "any" DFT frame, not just for real BCH-DFT codes.

The construction in (4.4) suggests that for each DFT frame there are many (but, a finite number of) systematic frames since the rows of G_k can be arbitrarily chosen from those of G. This will be discussed in detail later in Section 4.7.3. The codewords generated by these systematic frames differ in the "position" of systematic samples (i.e., input data). This implies that parity (data) samples are not restricted to form a consecutive block in the associated codewords. Such a degree of freedom is useful in the sense that one can find the most suitable systematic frames for specific applications (e.g., the one with the smallest reconstruction error.)

4.4.3 Optimality Condition

From rate-distortion theory [28, Chapter 10], it is well known that the rate required to transmit a source, with a given distortion, increases as the variance of the source becomes larger. Particularly, as we can see from (2.4) and (3.5), for Gaussian sources this relation is logarithmic with variance, under the mean-squared error (MSE) distortion measure. In DSC that uses real-number codes [111], since coding is performed before quantization, the variance of transmitted sequence depends on the behavior of the encoding matrix. In syndrome approach, $\boldsymbol{s} = H\boldsymbol{x}$ and it can be checked that $\sigma_{\boldsymbol{s}} = \sigma_{\boldsymbol{x}}$, that is, the variance

is preserved.[4] However, as we show shortly, this is not valid in parity approach and the variance of parity samples depends on the behavior of encoding matrix G_{sys}. In view of rate-distortion theory, it makes a lot of sense to keep this variance as small as possible. Not surprisingly, we will show that using a tight frame (tight G_{sys}) for encoding is optimal.

Let $\boldsymbol{x}$ be the message vector, a column vector whose elements are i.i.d. random variables with variance σ_x^2, and let $\boldsymbol{c} = G_{\text{sys}}\boldsymbol{x}$ represent the codeword generated using the systematic frame. The variance of $\boldsymbol{c}$ is then given by

$$\begin{aligned}\sigma_c^2 &= \frac{1}{n}\mathbb{E}\{\boldsymbol{c}^H\boldsymbol{c}\} = \frac{1}{n}\mathbb{E}\{\boldsymbol{x}^H G_{\text{sys}}^H G_{\text{sys}}\boldsymbol{x}\} \\ &= \frac{1}{n}\sigma_x^2 \operatorname{tr}(G_{\text{sys}}^H G_{\text{sys}}),\end{aligned} \tag{4.26}$$

and

$$\begin{aligned}\operatorname{tr}\left(G_{\text{sys}}^H G_{\text{sys}}\right) &= \operatorname{tr}\left(G_k^{-1H} G^H G G_k^{-1}\right) \\ &= \frac{n}{k}\operatorname{tr}\left((G_k G_k^H)^{-1}\right) \\ &= \frac{n}{k}\operatorname{tr}\left((V_k^H V_k)^{-1}\right) \\ &= \frac{n}{k}\sum_{i=1}^{k}\frac{1}{\lambda_i},\end{aligned} \tag{4.27}$$

in which $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$ are the eigenvalues of $G_k G_k^H$ (or $V_k^H V_k$ equivalently).

This shows that the variance of codewords, generated by a systematic frame, depends on the submatrix G_k which is used to create G_{sys}. G_k, in turn, is fully known once the position of systematic samples is fixed in the codewords. In other words, the "position" of systematic samples determines the variance of the codewords generated by a systematic DFT frame. To minimize the effective range of transmitted signal, from (4.26) and (4.27), we need to do the following optimization problem

$$\begin{aligned}\underset{\lambda_i}{\text{minimize}} \quad & \sum_{i=1}^{k}\frac{1}{\lambda_i} \\ \text{s.t.} \quad & \sum_{i=1}^{k}\lambda_i = k,\ \lambda_i > 0,\end{aligned} \tag{4.28}$$

[4] In general, any unitary matrix U preserves norms, i.e., for any complex vector $\boldsymbol{x}$, $\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$. Note that H is not unitary because it is not a square matrix; however, its rows are selected from a unitary matrix and are orthonormal. This lead to $HH^H = I_{n-k}$, and $\operatorname{tr}(H^H H) = n - k$.

where, the constraint $\sum_{i=1}^{k} \lambda_i = k$ is achieved in consideration of Lemma 4.2 and (4.18).

By using the Lagrangian method [18], we can show that the optimal eigenvalues are $\lambda_i = 1 \ \forall i$; this implies a tight frame [51]. In the sequel, we analyze the eigenvalues of $G_{p\times k}G_{p\times k}^H$, $p \leq n$, that helps us characterize tight systematic frames, so as to minimize the variance of transmitted codewords.

4.5 Main Results on the Extreme Eigenvalues

In this section we investigate some bounds on the eigenvalues of $G_{p\times k}G_{p\times k}^H$ where G is defined in (3.7). These bounds play an important role in the performance evaluation of the systematic DFT frames. We also determine the exact values of some eigenvalues in certain cases.

Theorem 4.1. *Let $G_{p\times k}, 1 \leq p \leq n$ be any $p \times k$ submatrix of G. Then, the smallest eigenvalue of $G_{p\times k}G_{p\times k}^H$ is no more than one, and the largest eigenvalue of $G_{p\times k}G_{p\times k}^H$ is at least one.*

Proof. From Lemma 4.2, we know that all principal diagonal entries of $G_{p\times k}G_{p\times k}^H$ are unity. As a result, using the Schur-Horn inequality in (4.17), we obtain $\lambda_{\min}(G_{p\times k}G_{p\times k}^H) \leq 1 \leq \lambda_{\max}(G_{p\times k}G_{p\times k}^H)$. This proves the claim. □

Note that $\lambda_1(G_{p\times k}G_{p\times k}^H) = \lambda_1(G_{p\times k}^H G_{p\times k})$ for any $G_{p\times k}$. Nevertheless, this is not correct for $\lambda_{\min}$ in general. A tighter bound on λ_1 can be achieved when $G_{p\times k}$ is a tall[5] matrix.

Theorem 4.2. *Given a tall (short) $G_{p\times k}$, the largest (smallest) eigenvalue of $G_{p\times k}^H G_{p\times k}$ is lower (upper) bounded by p/k.*

Proof. Let $p > k$. Since all diagonal entries of $G_{p\times k}G_{p\times k}^H$ are unity, from (4.18) we have $\sum_{i=1}^{p} \lambda_i(G_{p\times k}G_{p\times k}^H) = p$. On the other hand, since the nonzero eigenvalues of $G_{p\times k}G_{p\times k}^H$

[5] An $m \times n$ matrix A is called to be tall if $m > n$. Similarly, if $m < n$, then A is a short matrix.

and $G_{p\times k}^H G_{p\times k}$ are equal, $G_{p\times k}G_{p\times k}^H$ has k nonzero eigenvalues and we get

$$\begin{aligned} p &= \sum_{i=1}^{p} \lambda_i(G_{p\times k}G_{p\times k}^H) \\ &= \sum_{i=1}^{k} \lambda_i(G_{p\times k}^H G_{p\times k}) \\ &\leq k\lambda_1(G_{p\times k}^H G_{p\times k}). \end{aligned} \tag{4.29}$$

Thus, for any tall $G_{p\times k}$, $\lambda_1(G_{p\times k}^H G_{p\times k}) = \lambda_1(G_{p\times k}G_{p\times k}^H) \geq \frac{p}{k} > 1$. Following a similar line of proof, for a short submatrix ($p < k$) we obtain $\lambda_{\min}(G_{p\times k}^H G_{p\times k}) \leq \frac{p}{k} < 1$.

Obviously the same bounds are valid for the extreme eigenvalues of $G_{p\times k}G_{p\times k}^H$. What is more, since p/k is the average value of eigenvalues, considering that $\lambda_{\min}(G_{p\times k}G_{p\times k}^H) = 0$ for $p > k$, and $\lambda_{\min}(G_{p\times k}^H G_{p\times k}) = 0$ for $p < k$, from (4.29) we conclude that corresponding bounds on the largest eigenvalues are strict. □

It is worth noting that in (4.29) the equality is achieved when $p = n$; it can also be achieved for "specific" submatrices only in the case of integer oversampling, i.e., when $n = Mk$, as we discuss later in this chapter.

We use the above results to find better bounds for the extreme eigenvalues of $G_k G_k^H$ in the following theorem.

Theorem 4.3. *For any G_k, a square submatrix of G in (3.7) in which $n \neq Mk$, the smallest (largest) eigenvalue of $G_k G_k^H$ is strictly upper (lower) bounded by 1.*

Proof. See Appendix B.1. □

Theorem 4.3 implies that for $n \neq Mk$ we cannot have "tight" systematic frames. Because, for a frame with frame operator $F^H F$, the tightest possible frame bounds are, respectively, $a = \lambda_{\min}(F^H F)$ and $b = \lambda_{\max}(F^H F)$ [21]. In other words, for a tight frame $\lambda_{\min}(F^H F) = \lambda_{\max}(F^H F)$; i.e., the eigenvalues of $F^H F$ are equal [51].

Corollary 4.2. *Tight systematic DFT frames can exist only if $n = Mk$, where M is a positive integer.*

Note that systematic DFT frames are not necessarily tight for $n = Mk$. In Section 4.7, we prove that tight systematic DFT frames exist for $n = Mk$ and show how to construct such frames.

In the remainder of this section, we shall find exact values, rather than bounds, for some of the eigenvalues of $G_k^H G_k$ when $k < n \leq 2k$. This range of n is specifically important in parity-based DSC [111], where $n-k$ parity samples are used to represent k samples and so for compression, $n-k < k$.

Theorem 4.4. *For any G_k, a square submatrix of G in (3.7), where $k < n < 2k$, the $2k-n$ largest eigenvalues of $G_k G_k^H$ are equal to n/k.*

Proof. From Corollary 4.1 we know that if two Hermitian matrices sum up to a scaled identity matrix, their eigenvalues add up to be fixed. Thus, if A and B have the same spectrum we obtain

$$\lambda_j(A) + \lambda_{k-j+1}(A) = \gamma. \tag{4.30}$$

Now, let G be partitioned as $G = \begin{bmatrix} G_k \\ \bar{G}_{p\times k} \end{bmatrix}$ where $p = n-k$. Let $A = G_k^H G_k$ and $B = \bar{G}_{p\times k}^H \bar{G}_{p\times k}$, then $A + B = G^H G = \frac{n}{k} I_k$. Clearly, Corollary 4.1 holds with $\gamma = \frac{n}{k}$. Also, note that when $p < k$ then $\bar{G}_{p\times k}^H \bar{G}_{p\times k}$ has only p nonzero eigenvalues. Therefore, in such a case, $k-p$ largest eigenvalues of $G_k^H G_k$ are equal to n/k. □

Another interesting case arises when $n = 2k$. Numerical results shows that under this condition, A and B have the same set of eigenvalues. We prove this when G_k either includes successive or every other rows of G. In such cases, one can verify that $(\bar{G}_k)_{i,j} = e^{j\theta}(G_k)_{i,j}$; thus, Lemma 4.1 holds and A and B have the same eigenvalues. Hence, from (4.30) we get

$$\lambda_j(G_k^H G_k) + \lambda_{k-j+1}(G_k^H G_k) = \frac{n}{k} = 2. \tag{4.31}$$

This further implies that for odd values of k the middle eigenvalue of $G_k^H G_k$ is 1.

We close this section with an example illustrating some of the above properties. Consider an (n,k) DFT frame and the the following two cases. First, the rows of G_k are evenly spaced rows of G (i.e., either odd rows or even rows). This is the "best" submatrix in the sense that it minimizes the MSE. For such a submatrix, all eigenvalues are known to be equal, as it is a DFT matrix. For example, for $n = 10, k = 5$, the best square submatrix results in $\lambda = 1$ with multiplicity of 5. The other extreme case, which maximizes the MSE,

happens when the rows of G_k are circularly consecutive rows of G. Again, for the above example, $\lambda = \{0.0011, 0.1056, 1, 1.8944, 1.9989\}$. With these examples in mind, we will explore the best and worst frames in Section 4.7. We shall now discuss signal reconstruction for systematic frames.

4.6 Performance Analysis

In this section, we analyze the performance of quantized systematic DFT codes using the quantization model proposed in [51], which assumes that noise components are uncorrelated and each noise component q_i has mean 0 and variance σ_q^2, i.e., for any i, j,

$$\mathbb{E}\{q_i\} = 0, \qquad \mathbb{E}\{q_i q_j\} = \sigma_q^2 \delta_{ij}. \tag{4.32}$$

For example, q can be uniformly distributed on $[-\Delta/2, \Delta/2]$, where $\sigma_q^2 = \Delta^2/12$. We assume the quantizer range covers the dynamic range of all codewords encoded using the systematic DFT code in (4.4).

Let $\boldsymbol{x}$ be the signal (message) to be transmitted. The corresponding codeword is generated by

$$\boldsymbol{c} = G_{\text{sys}}\boldsymbol{x}. \tag{4.33}$$

This is then quantized to $\hat{\boldsymbol{c}}$ and transmitted. Assuming the quantization model in (4.32), transmitted codeword can be modeled by

$$\hat{\boldsymbol{c}} = G_{\text{sys}}\boldsymbol{x} + \boldsymbol{q}, \tag{4.34}$$

where $\boldsymbol{q}$ represents quantization error. This also models the received codvector provided that there is no error or erasure in channel. Now, suppose we want to estimate $\boldsymbol{x}$ from (4.34). This can be done through the use of linear or nonlinear operations.

4.6.1 Linear Reconstruction

We first consider *linear reconstruction* of $\boldsymbol{x}$ form $\hat{\boldsymbol{c}}$ using the pseudoinverse [51] of G_{sys}, which is defined by

$$G_{\text{sys}}^{\dagger} = (G_{\text{sys}}^H G_{\text{sys}})^{-1} G_{\text{sys}}^H = \frac{k}{n} G_k G^H. \tag{4.35}$$

The linear reconstruction is hence given by

$$\hat{\boldsymbol{x}} = \frac{k}{n} G_k G^H \hat{\boldsymbol{c}} = \boldsymbol{x} + \frac{k}{n} G_k G^H \boldsymbol{q}, \tag{4.36}$$

where $\boldsymbol{q}$ represents quantization error.

Let us now evaluate the reconstruction error. The mean-squared reconstruction error, due to the quantization noise, using a systematic frame can be written as

$$\begin{aligned}
\text{MSE}_{\text{q}} &= \frac{1}{k}\mathbb{E}\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2\} = \frac{1}{k}\mathbb{E}\{\|G_{\text{sys}}^{\dagger}\boldsymbol{q}\|^2\} \\
&= \frac{1}{k}\mathbb{E}\{\boldsymbol{q}^H G_{\text{sys}}^{\dagger H} G_{\text{sys}}^{\dagger}\boldsymbol{q}\} = \frac{1}{k}\sigma_q^2 \,\text{tr}\left(G_{\text{sys}}^{\dagger H} G_{\text{sys}}^{\dagger}\right) \\
&= \frac{k}{n^2}\sigma_q^2 \,\text{tr}\left(G G_k^H G_k G^H\right) \\
&= \frac{k}{n^2}\sigma_q^2 \,\text{tr}\left(G_k^H G_k G^H G\right) \\
&= \frac{1}{n}\sigma_q^2 \,\text{tr}\left(G_k^H G_k\right) = \frac{k}{n}\sigma_q^2,
\end{aligned} \tag{4.37}$$

where the last step follows because of Lemma 4.2. This shows that DFT codes reduce quantization error.

The fact that the MSE is inversely proportional to the *redundancy* of the frame is a well-known result for *tight* frames [64, 113, 51, 84]. The above analysis, however, indicates that the MSE is the same for all systematic DFT frames of the same size, no matter they are tight or not. This is yet assuming that the effective range of the codewords generated by different G_{sys} is equal, which implies the same σ_q^2 for a given number of quantization levels. However, from (4.26) it is known that, for a fixed number of quantization levels, σ_q^2 depends on the variance of transmitted codewords (σ_c^2) if the quantizer is designed to cover the entire effective range of codewords. Obviously, though, σ_c^2 can vary from one systematic frame to another, as shown in (4.27).

Theorem 4.5. *When encoding with a systematic DFT frame in* (4.4) *and decoding with linear reconstruction, for the noise model* (4.32) *and given a same number of quantization levels, the MSE is minimum if and only if the systematic frame is tight.*

Proof. All systematic DFT frames amount to a same quantization error provided that the effective range of codewords are fully covered, as shown in (4.37). Nevertheless, for a fixed number of quantization levels more codewords are within the range of quantizer if the systematic frame is tight. This is clear from (4.27) and (4.28), recalling that (4.28) is minimized by the tight frames. Moreover, any frame that minimizes (4.28) is required to be tight. This will be proved in Section 4.7.1. □

The problem we are considering in Theorem 4.5 is somewhat the *dual* of Theorem 3.1 in [64]. Note that in [64, Theorem 3.1] "uniform" frames are used for encoding which implies the same variance for all samples of codewords whereas the reconstruction error is proportional to $\sum_{i=1}^{k} \lambda_i$. On the other hand, the frames used in Theorem 4.5 are not uniform in general; this result in a codeword variance proportional to $\sum_{i=1}^{k} \lambda_i$ while having a fixed, minimum reconstruction error.

4.6.2 Consistent Reconstruction

Linear reconstruction is not always the best one can estimate $\boldsymbol{x}$ from $\hat{\boldsymbol{c}}$. Although *linear reconstruction* is more tractable, *consistent reconstruction* is known to give significant improvement over linear reconstruction in overcomplete expansions [101, 102, 52]. Asymptotically, the MSE is $O(r^{-2})$ for consistent reconstruction, where $r = n/k$ is the frame redundancy [102]. As it can be seen from (4.37), for linear reconstruction this is $O(r^{-1})$. The improvement, in consistent reconstruction, is due to using deterministic properties of quantization rather than considering quantization as an independent noise as in (4.32).

Although the MSE in consistent reconstruction is approximated by $c_0 r^{-2}$, where the constant c_0 depends on the source and quantization, this is verified only if the oversampling ratio r is very high [52]. In some practical applications of frames, e.g., channel coding, this ratio cannot be high, though. Particularly, in the context of interest, i.e., DSC, r is limited to two [111]. Besides, consistent reconstruction methods do not provide a guidance on how to design the frame, as they do not point out how to compute the constant c_0. More importantly, (4.37) proves to be predictive of the performance of consistent recon-

struction [51]; therefore, it can be convincingly used as a design criterion regardless of the reconstruction method.

4.6.3 Reconstruction with Error and Erasure

In the context of channel coding, DFT codes are primarily used to provide robustness against channel impairments which can be errors or erasures. Likewise, in DSC these codes play the role of channel codes to combat the errors due to the virtual correlation channel [111]. Thus, it makes sense to evaluate the performance of these codes in the presence of error. To this end, let $\hat{\boldsymbol{c}} = G\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{\eta} = \boldsymbol{q} + \boldsymbol{e}$. Assuming that the quantization and channel errors are independent, we will have

$$\begin{aligned} \mathbb{E}\{\boldsymbol{\eta}^T\boldsymbol{\eta}\} &= \mathbb{E}\{\boldsymbol{q}^T\boldsymbol{q} + \boldsymbol{q}^T\boldsymbol{e} + \boldsymbol{e}^T\boldsymbol{q} + \boldsymbol{e}^T\boldsymbol{e}\} \\ &= n\sigma_q^2 + \nu\sigma_e^2, \end{aligned} \tag{4.38}$$

where ν is the average number of errors in each codeword and $\mathbb{E}\{\boldsymbol{e}^T\boldsymbol{e}\} \triangleq \nu\sigma_e^2$. Note that $\mathbb{E}\{\boldsymbol{e}^T\boldsymbol{q}\} = \mathbb{E}\{\boldsymbol{q}^T\boldsymbol{e}\} = 0$, because q and e are independent and q has mean equal to zero. Finally, following a similar analysis as in (4.37), we obtain

$$\mathrm{MSE}_{\mathrm{q+e}} = \frac{k}{n}\sigma_\eta^2 = \frac{k}{n}\left(\sigma_q^2 + \frac{\nu}{n}\sigma_e^2\right). \tag{4.39}$$

From (4.39) it is clear that reconstruction error has two distinct parts caused by the quantization and channel errors. It also proves that DFT codes decrease both channel and quantization errors by a factor of frame redundancy $r = n/k$. The above results is for the case when no error correction is done. It is worth noting that, even without correcting errors, the MSE can be smaller than quantization error.

As another extreme case, let us consider the case when error localization is perfect, i.e., errors are in the *erasure* form. Then, we remove the corrupted samples and do reconstruction using the error-free samples. This approach does not require error correction in order to reconstruct the message; however, it is shown to be equal to the coding theoretic approach [84]. Let $\hat{\boldsymbol{c}}_R$ and $\boldsymbol{\eta}_R$ denote remaining rows of $\hat{\boldsymbol{c}}$ and $\boldsymbol{\eta}$, respectively. Obviously, $\boldsymbol{\eta}_R$ includes only quantization error, hence we represent $\boldsymbol{\eta}_R$ with $\boldsymbol{q}_R$. Also, let F denote

the rows of G_{sys} corresponding to $\boldsymbol{q}_R$. Then, we can write

$$\hat{\boldsymbol{c}}_R = F\boldsymbol{x} + \boldsymbol{q}_R, \tag{4.40}$$

$$\hat{\boldsymbol{x}} = F^{\dagger}\hat{\boldsymbol{c}}_R, \tag{4.41}$$

where $F^{\dagger} = (F^H F)^{-1} F^H$. Thus, similar to (4.37) we will have

$$\begin{aligned} \text{MSE}_{\text{q}+\rho} &= \frac{1}{k}\mathbb{E}\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2\} = \frac{1}{k}\mathbb{E}\{\|F^{\dagger}\boldsymbol{q}_R\|^2\} \\ &= \frac{1}{k}\sigma_q^2 \operatorname{tr}\left(F^{\dagger H}F^{\dagger}\right) = \frac{1}{k}\sigma_q^2 \operatorname{tr}\left(F^H F\right)^{-1} \\ &= \frac{1}{k}\sigma_q^2 \sum_{i=1}^{k} \frac{1}{\mu_i}, \end{aligned} \tag{4.42}$$

where subscript ρ denotes erasure and $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_k > 0$ represent the eigenvalues of $F^H F$. We assume at least k samples are intact which implies $\mu_k > 0$.

One nice property of systematic frames is that reconstruction error cannot be more than quantization error as long as systematic samples are intact. This holds even if consecutive samples are erased. We know that consecutive erasures can increase the MSE very fast (e.g., see [84, Table I]). This can be understood from (4.42) since F contains I_k as a subframe and in the worst case we can use this subframe for reconstruction which leads to $\text{MSE}_{\text{q}+\rho} = \sigma_q^2$. Adding any other row (sample) will decrease the MSE. To show this, let $F^H = [I_k \,|\, E^H]$. Then, $F^H F = I_k + E^H E$ and, from (4.20), for $i = j$, we get $\mu_i \geq 1 + \xi_k$ for $i = 1, \ldots, k$, where ξ_k is the smallest eigenvalue of $E^H E$. Clearly, $\xi_k \geq 0$ since $E^H E$ is a positive semidefinite matrix. Further, at least $\mu_1 > 0$ since otherwise E must be zero. Hence, $\sum_{i=1}^{k} \frac{1}{\mu_i}$ decreases by adding new rows.

Finally, with consistent reconstruction, we can further decrease the MSE. To do so, we check if reconstructed values $\hat{x}_i$ for systematic samples in (4.41) are consistent with their values before reconstruction or not, i.e., for any systematic sample, we must have $Q(\hat{x}_i) = Q(\hat{c}_{Ri})$. Otherwise, we replace $\hat{x}_i$ with

$$\hat{\hat{x}}_i = Q(\hat{c}_{Ri}) - \text{sign}(Q(\hat{c}_{Ri}) - \hat{x}_i)\frac{\Delta}{2}. \tag{4.43}$$

4.7 Characterization of Systematic Frames

4.7.1 The Best and Worst Systematic Frames

As we discussed in Section 4.5, the optimal G_{sys} is achieved from the optimization problem (4.28). Similarly, to find the worst G_{sys}, we can *maximize* (4.28) instead of minimizing it. The optimal eigenvalues are known to be $\lambda_i = 1, 1 \leq i \leq k$. But, how can we find the corresponding G_{sys}, or G_k equivalently? More importantly, if a G_k with $\lambda_i = 1$ does not exist, is there any suggestion for the best matrix?

We approach this problem by studying another optimization problem. To this end, we first prove the following theorem for the eigenvalues of $G_k G_k^H$.

Theorem 4.6. *Let $\{\lambda_i\}_{i=1}^k$ be the eigenvalues of $G_k G_k^H$, where G_k includes k arbitrary rows of G, then we have*

$$\underset{\{\lambda_i\}}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{\lambda_i} = \underset{\{\lambda_i\}}{\operatorname{argmax}} \prod_{i=1}^{k} \lambda_i. \tag{4.44}$$

Proof. Let $\{\lambda_i\}_{i=1}^k$ be the eigenvalues of $G_k G_k^H$. From Lemma 4.2, we know that $\sum_{i=1}^k \lambda_i (G_k^H G_k) = k$. Then, subject to this constraint, by using the Lagrangian method [18], its is straightforward to see that the optimal values of the optimization problems in both sides of (4.44) are $\lambda_i = 1, i = 1, \ldots, k$. □

Now, in view of Theorem (4.6), the optimal arguments of the optimization problem in (4.28) are equal to those of

$$\begin{aligned} \underset{\{\lambda_i\}}{\text{maximize}} \quad & \prod_{i=1}^{k} \lambda_i \\ \text{s.t.} \quad & \sum_{i=1}^{k} \lambda_i = k, \ \lambda_i > 0, \end{aligned} \tag{4.45}$$

in which $\{\lambda_i\}_{i=1}^k$ are the eigenvalues of $G_k G_k^H$ (or $V_k^H V_k$). By using the Lagrangian method, one can check that (4.45) has the maximum of 1 and infimum of 0. Then, considering that

$$\prod_{i=1}^{k} \lambda_i = \det(V_k^H V_k) = \det(G_k G_k^H), \tag{4.46}$$

we conclude that the "best" submatrix is the one with the largest determinant (possibly 1) and the "worst" submatrix is the one with smallest determinant.

Next, we evaluate the determinant of $V_k^H V_k$ so as to find the matrices corresponding to the extreme cases. To this end, we first evaluate the determinate of WW^H where W is the Vandermonde matrix with unit complex entries as defined in (5.29). From (4.15) we can write

$$\begin{aligned}\det(WW^H) &= \frac{1}{n^n}\prod_{1\le p<q\le n}|e^{i\theta_p}-e^{i\theta_q}|^2 \\ &= \frac{1}{n^n}\prod_{1\le p<q\le n}4\sin^2\frac{\pi}{n}(q-p) \\ &= \frac{2^{n(n-1)}}{n^n}\prod_{r=1}^{n-1}\left(\sin^2\frac{\pi}{n}r\right)^{n-r},\end{aligned} \tag{4.47}$$

in which $\theta_x = \frac{2\pi}{n}(x-1), r = q-p$, and $n(n-1)/2$ is the total number of terms that satisfy $1 \le p < q \le n$. But, we see that W is a DFT matrix, and thus, its determinant must be 1. Therefore, we have

$$\prod_{r=1}^{n-1}\left(\sin^2\frac{\pi}{n}r\right)^{n-r} = \frac{n^n}{2^{n(n-1)}}. \tag{4.48}$$

The above analysis helps us evaluate the determinant of V_k or G_k, defined in (4.25). Let $\mathcal{I}_{r_k} = \{i_{r_1}, i_{r_2}, \ldots, i_{r_k}\}$ be those rows of G used to build G_k. Also, without loss of generality, assume $i_{r_1} < i_{r_2} < \cdots < i_{r_k}$. Clearly, $i_{r_1} \ge 1, i_{r_k} \le n$, and we obtain

$$\begin{aligned}\det(V_k V_k^H) &= \frac{1}{k^k}\prod_{\substack{1\le p<q\le n\\ p,q\in\mathcal{I}_{r_k}}}|e^{i\theta_p}-e^{i\theta_q}|^2 \\ &= \frac{1}{k^k}\prod_{\substack{1\le p<q\le n\\ p,q\in\mathcal{I}_{r_k}}}4\sin^2\frac{\pi}{n}(q-p).\end{aligned} \tag{4.49}$$

Then, since $\sin\frac{\pi}{n}u = \sin\frac{\pi}{n}(n-u)$, one can see that this determinant depends on the circular distance between rows in $\mathcal{I}_{r_k}$. For a matrix with n rows, we define the *circular distance* between rows p and q as $\min\{|q-p|, n-|q-p|\}$. In this sense, for example, the distance between rows 1 and n is one. Now, it is reasonable to believe that (4.49) is minimized

when the selected rows are *circularly successive*.[6] Note that $\sin u$ is strictly increasing for $u \in [0, \pi/2]$, and the circular distance cannot be greater than $n/2$ in this problem.

In such circumstances where all rows in $\mathcal{I}_{r_k}$ are (circularly) successive, (4.49) is minimal and reduces to

$$\det(V_k V_k^H) = \frac{2^{k(k-1)}}{k^k} \prod_{r=1}^{k-1} \left(\sin^2 \frac{\pi}{n} r\right)^{k-r}. \tag{4.50}$$

The other extreme case comes up when $n = Mk$ (M is a positive integer) provided that G_k consists of every Mth row of G. In such a case, (4.49) simplifies to 1, because

$$\begin{aligned} \det(V_k V_k^H) &= \frac{2^{k(k-1)}}{k^k} \prod_{r=1}^{k-1} \left(\sin^2 \frac{\pi}{n} Mr\right)^{k-r} \\ &= \frac{2^{k(k-1)}}{k^k} \prod_{r=1}^{k-1} \left(\sin^2 \frac{\pi}{k} r\right)^{k-r} \\ &= 1, \end{aligned} \tag{4.51}$$

where the last step is because of (4.48). Recall that this gives the best V_k (and equivalently G_k), in light of (4.45). For such a G_k, it is easy to see that G_{sys} stands for a "tight" systematic frame and minimizes the MSE for a given number of quantization levels. Effectively, such a frame is performing *integer oversampling*. There are M such frames; they all have the same spectrum, though.

Recall that, from (4.44)–(4.46) and Theorem 4.3, $\det(V_k V_k^H) < 1$ for $n \neq Mk$. For such an (n, k) frame, the systematic rows cannot be equally spaced in the corresponding systematic frame; instead, we may explore a systematic frame in which the circular distance between successive systematic samples is as evenly as possible. Then, the circular distance between each successive systematic rows is either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$. More precisely, if l and m, respectively, represent the number of systematic rows with circular distance equal to

[6] A set of J rows $\{i_{r_1} < i_{r_2} < \ldots < i_{r_J}\}$ of a matrix are successive if they are one after the other, i.e., $i_{r_j} = i_{r_{j-1}+1}$. A set of rows are circularly successive if they or their complement set of rows are successive, where the complement of a set of rows includes all rows except that set of rows.

$\lceil n/k \rceil$ and $\lfloor n/k \rfloor$, they must satisfy

$$\begin{cases} l + m = k, \\ l\lceil \frac{n}{k} \rceil + m\lfloor \frac{n}{k} \rfloor = n. \end{cases} \tag{4.52}$$

In the following theorem, we prove that the best performance is achieved when the systematic rows are as equally spaced as possible, i.e., when (4.52) is satisfied.

Theorem 4.7. *When encoding with an (n, k) systematic DFT frame in* (4.4) *and decoding with linear reconstruction, for the noise model* (4.32) *and given a same number of quantization levels, the MSE is minimum when there are $l = n - \lfloor n/k \rfloor k$ systematic rows with successive circular distance $\lceil n/k \rceil$ and the remaining $m = k - l$ systematic rows have a successive circular distance equal to $\lfloor n/k \rfloor$.*

Proof. See Appendix B.2. □

Effectively, the above theorem is generalizing Theorem 4.5. Note that when $n = Mk$, $\lfloor n/k \rfloor = \lceil n/k \rceil = M$ and there exist k systematic rows with equal distance; in this case, Theorem 4.7 reduces to Theorem 4.5 and the corresponding systematic frame is tight. The optimality of this case was proved in (4.51). When $n \neq Mk$, we cannot have a systematic frame with equally spaced systematic rows; however, the best performance is still achieved when the circular distance between the systematic (parity) rows is as evenly as possible, as detailed above. Note that in either case $d_{\min}$, the minimum distance between the systematic rows, is $\lfloor n/k \rfloor$. This is a necessary condition for an optimal systematic frame, as shown in the proof of Theorem 4.7. Further, to satisfy Theorem 4.7, the minimum distance between the parity rows must be $\bar{d}_{\min} = \lfloor n/(n-k) \rfloor$.

4.7.2 Numerical Examples

Numerical calculations confirm that "evenly" spaced data samples gives rise to systematic frames with the best performance. When a systematic frame is doing integer oversampling, we end up with tight systematic frames. The first and last codes in Table 4.1 are examples of this case. When $n \neq Mk$, data samples cannot be equally spaced; however, as it can be seen from the second code in Table 4.1, still the best performance is achieved when they are as equally spaced as possible. In this table, "$\times$'s" and "$-$'s" represent data

Table 4.1 Eigenvalues structure for two systematic DFT frames with different codeword patterns. A "×" and "−" respectively represent data (systematic) and parity samples.

Code	Codeword patern	$\lambda_{\min}$	$\lambda_{\max}$	$\sum_{i=1}^{k} 1/\lambda_i$	$\prod_{i=1}^{k} \lambda_i$
$(6,3)$	$\times\times\times---$	0.0572	1.9428	19	0.1111
	$\times\times-\times--$	0.2546	1.7454	5.5	0.4444
	$\times\times--\times-$	0.2546	1.7454	5.5	0.4444
	$\times-\times-\times-$	1	1	3	1
$(7,5)$	$\times\times\times\times\times--$	0.0396	1.4	28.70	0.0827
	$\times\times\times\times-\times-$	0.1506	1.4	10.32	0.2684
	$\times\times-\times\times-\times$	0.3110	1.4	7.40	0.4173
	$\times-\times\times\times-\times$	0.3110	1.4	7.40	0.4173
$(10,5)$	$\times\times\times\times\times----$	0.0011	1.9989	908.21	4.46×10^{-4}
	$\times\times\times\times-\times---$	0.0041	1.9959	249.94	0.0047
	$\times\times\times\times--\times--$	0.0110	1.9890	96.09	0.0122
	$\times\times\times-\times---\times-$	0.0202	1.9798	53	0.0400
	$\times\times\times--\times\times--$	0.0496	1.9504	25.64	0.0489
	$\times\times\times-\times-\times--$	0.0310	1.9690	35.73	0.0611
	$\times\times\times--\times--\times-$	0.0512	1.9488	23.41	0.0838
	$\times\times\times--\times-\times--$	0.0835	1.9165	16	0.1280
	$\times\times-\times\times--\times--$	0.1056	1.8944	13.79	0.1436
	$\times\times--\times\times--\times-$	0.2497	1.7503	9.56	0.2193
	$\times\times-\times-\times--\times-$	0.1902	1.8098	8.86	0.3351
	$\times-\times-\times-\times--\times$	0.2377	1.7623	7.77	0.4189
	$\times-\times-\times-\times-\times-$	1	1	5	1

and parity samples, respectively. Moreover, we observe that circularly shifted codeword patterns behave the same (e.g., in the $(7,5)$ code, frames with pattern $\times - \times \times \times - \times$ and $\times \times - \times \times - \times$ have the same performance). Also, reversal of a codeword pattern yields a codeword with the same performance (e.g., $\times \times - \times --$ is shifted version of reversed $\times \times - - \times -$ in the $(6,3)$ code). These properties hold in general, as stated below.

Property 4.1. *Circular shift of $\mathcal{I}_{r_k}$, the systematic rows of a systematic frame with analysis frame G_{sys}, does not change the spectrum of $G_{\text{sys}}^H G_{\text{sys}}$.*

Property 4.2. *Reversal of $\mathcal{I}_{r_k}$ yields a systematic frame with the same spectral properties.*

Proof. From (4.4) we obtain

$$\lambda_i(G_{\text{sys}}^H G_{\text{sys}}) = \frac{n/k}{\lambda_i(G_k G_k^H)}, \tag{4.53}$$

for $i = 1, \ldots, k$. But $G_k G_k^H$ is invariant to the circular shift of rows of G that make $\mathcal{I}_{r_k}$, as long as all rows are shifted the same amount in the same direction. This can be seen from the proof of Lemma 4.2 in (4.24) by defining $r' = r + c$ where r' represent the shifted rows by a constant c and $r \in \mathcal{I}_{r_k}$. This proves Property 4.1. Likewise, let $r'' = n + 1 - r$ be the reversed row indices. Again, from (4.24), it is clear that Property 4.2 holds. □

These properties together show that the frame operators of systematic frames $(G_{\text{sys}}^H G_{\text{sys}})$, in which the "relative" circular distance among the systematic rows are the same, inherit the same spectrum and thus show the same performance.

4.7.3 Number of Systematic Frames

From previous sections, its is obvious that there are many systematic frames for a given (n,k) frame. The performance of these systematic frames depends on the position of the systematic rows, or equivalently, the position of data (or parity) samples in the associated codewords, and can be the same for different systematic frames. In what follows, we derive an upper and lower bound on the number of systematic frames with different spectrum. In other words, we categorize these frames based on their performance. To this end, we observe that the problem of finding $k \times k$ submatrices of an $n \times k$ matrix can be viewed as finding

different k-subsets of a set with n elements. This is given by the binomial coefficient $\binom{n}{k}$ and is also equivalent to the number of systematic frames. As stated earlier in Property 4.1, circular shift of a codeword pattern does not change its spectrum, and so its performance. We define a *coset* as square submatrices that result in a same performance. Each coset has at least n elements (k-subsets), as shown in Table 4.2. To find these elements, it suffices to circularly shift a subset n times. Equivalently, for a given k-subset, we simply add up 1 to each element of a subset. Note that, the subsets elements are k row indices of $G_{n\times k}$ and thus cannot be greater than n. Therefore, once a shifted index x becomes greater than n, we replace it with $\langle\langle x\rangle\rangle_n$ where $\langle\langle x\rangle\rangle_n \triangleq x - dn$ if $dn + 1 \leq x \leq dn + n, d \in \mathbb{Z}$. Obviously, each coset has at least n subsets since $n - 1$ circular shifts of a given subset are distinct; all these subsets have the same relative distance, though. This can be seen in Table 4.2. Thus, it is clear that the number of cosets is the bounded by

$$n_c \leq u = \frac{1}{n}\binom{n}{k}. \tag{4.54}$$

Let $\mathcal{I}^r_{r_k}$ denote the reversal of $\mathcal{I}_{r_k} = \{i_{r_1}, i_{r_2}, \ldots, i_{r_k}\}$ where

$$\mathcal{I}^r_{r_k} \triangleq \langle\langle n + 1 - \mathcal{I}_{r_k}\rangle\rangle_n. \tag{4.55}$$

This operation is performed on every element of $\mathcal{I}_{r_k}$. One can see that reversal of a subset does not change its distance and spectrum, owing to Property 4.2. This can reduce the number of cosets. For example, in Table 4.2, the reversal of $\{1, 2, 4\}$, which is the *coset leader* in C_2, is $\{7, 6, 4\}$ which belongs to C_5. This indicates C_2 and C_5 are essentially one coset. The bound in (4.54) is tight if and only if there are u self-reversal cosets. Trivial examples of such a code are achieved when $k = n - 1$ or $k = 1$. A self-reversal coset is a coset that the reversal of its elements belong to itself, e.g., C_1, C_3, and C_4 in Table 4.2.

On the other hand, $n_c \geq u/2$ is a lower bound because there cannot be more than one reversal for a given coset. It can be further seen that the coset with smallest weight (C_1) is always self-reverse, i.e., the reversal of each element of C_1 is its own element for any (n, k) frame. This implies that the lower bound is not achievable. Therefore,

$$\frac{1}{2n}\binom{n}{k} < n_c \leq \frac{1}{n}\binom{n}{k}. \tag{4.56}$$

Table 4.2 Different cosets of $(7, 3)$ DFT frame and their corresponding relative distances and spectrums. The Coset leaders are in boldface.

	C_1	C_2	C_3	C_4	C_5
Leader	**1 2 3**	**1 2 4**	**1 2 5**	**1 3 5**	**1 3 4**
	2 3 4	2 3 5	2 3 6	2 4 6	2 4 5
	3 4 5	3 4 6	3 4 7	3 5 7	3 5 6
	4 5 6	4 5 7	1 4 5	1 4 6	4 6 7
	5 6 7	1 5 6	2 5 6	2 5 7	1 5 7
	1 6 7	2 6 7	3 6 7	1 3 6	1 2 6
	1 2 7	1 3 7	1 4 7	2 4 7	2 3 7
Distance	1 1 2	1 2 3	1 3 3	2 2 3	1 3 2
Weight	4	6	7	7	6
λ_1	2.1558	1.7539	1.9066	1.2673	1.7539
λ_2	0.8150	1.1133	0.8424	1.1601	1.1133
λ_3	0.0292	0.1328	0.2510	0.5726	0.1328

One can check that the first two frames in Table 4.1 reach the upper bound $\lfloor \frac{1}{n}\binom{n}{k} \rfloor$ whereas the third one satisfies the lower bound $\lceil \frac{1}{2n}\binom{n}{k} \rceil$.

4.8 Simulation Results: Parity-Based DSC and DJSCC

In this section we focus on numerical results for the parity-based DSC and its extension to the noisy channel setting, i.e., DJSCC. We do simulations both for a Gauss-Markov source with $\rho = 0.9$ and Gaussian sources, i.e., a Gauss-Markov source with $\rho = 0$. We first compare the performance of the parity- and syndrome- based approaches for two codes with the same compression ratio using a Gauss-Markov source with $\rho = 0.9$. We use a $(5, 1)$ code for the syndrome approach and a $(9, 5)$ code for the parity approach; thus, the compression ratio for both codes is $\eta = 0.8$. The results are presented in Figure 4.3; it can be seen that syndrome-based DSC performs a little better than the parity-based DSC. As we explained in Section 4.1.3 the performance of the parity-based system is not as good as that of the syndrome-based system, mainly because $\sigma_{\boldsymbol{p}} > \sigma_{\boldsymbol{s}}$. This implies a that the rate required to transmit the parity samples is more than that of the syndrome samples, given

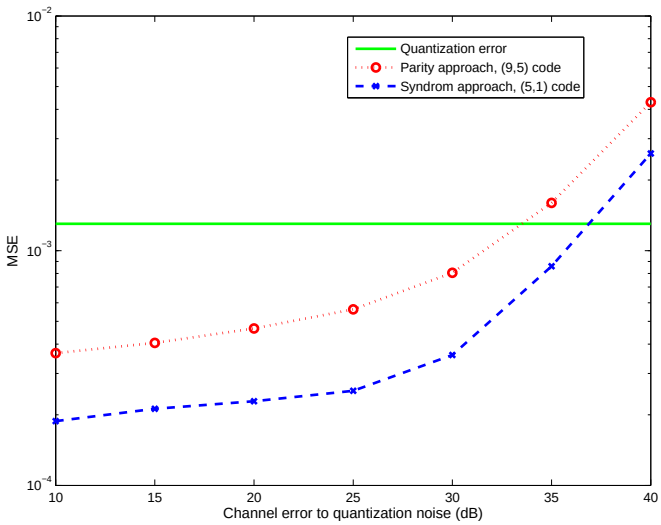


Figure 4.3 The MSE performance of syndrome- parity-based DSC for a Gauss-Markov source with $\rho = 0.9$ and the GBG correlation model with $\sigma_0 = 0.01\sigma_e$, $p_1 = 0.04$. The compression ratio for both approaches is 0.8 as the corresponding codes are $(5, 1)$ and $(9, 5)$ for the syndrome and parity approaches, respectively.

a same distortion.

We next plot the distortion-rate function for coding a Gauss-Markov source with $\rho = 0$ (Gaussian source) based on the parity-based DSC and compare its performance against the theoretical limits in Figure 4.4. The points are based on $(19, 17)$, $(9, 7)$, $(8, 5)$ codes, respectively. As we mentioned earlier, the performance of the parity-based DSC depends on the position of the parity samples. To find the rate distortion pairs in the above figure, the position of parity samples were chosen as even as possible to minimize the MSE.

Seeing that we do not use the ideal Slepian-Wolf coding assumption ($n \to \infty$), the gap between performance of the proposed schemes and the Wyner-Ziv rate-distortion function is more than usual. However, It should be noted that capacity-approaching channel codes may introduce significant delay if one strives to approach the capacity of the channel with very a low probability of error. Therefore, those are out of the question for delay-sensitive systems. In that case, it would be best to use channel codes of low rate and focus on achieving a very low probability of error. The system we introduced is a low-delay system which works well with reasonably high-rate codes. Finally, by puncturing some parity samples, rate-adaptive schemes are realized for the proposed DJSCC and parity-based DSC.

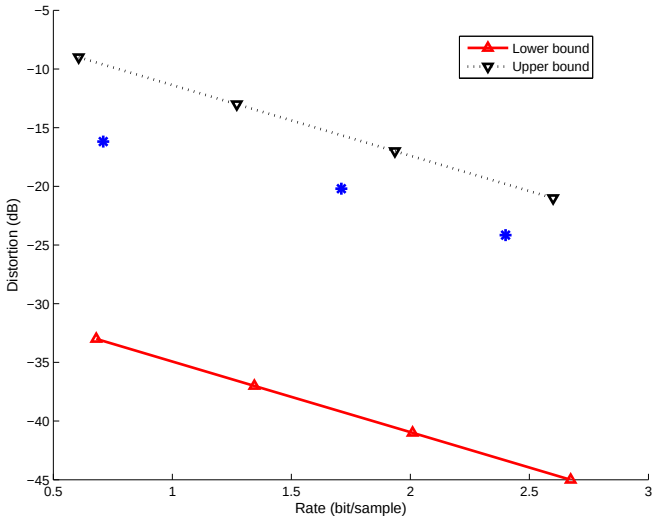


Figure 4.4 The distortion-rate function and bounds for coding a Gaussian source X with $\sigma_X = 1$ and the GBG correlation model with $\sigma_0 = 0.05\sigma_e$ at CEQNR= 25dB and $b = 6$. The achievable points are based on $(19, 17)$, $(9, 7)$, $(8, 5)$ codes, respectively.

Rate-adaptive systems are popular in the transmission of non-ergodic data, like video [119].

Next, we evaluate the performance of the JSCC with side information at the decoder, illustrated in Figure 4.2. By using a systematic $(10, 5)$ DFT code, we generate, quantize, and transmit parity samples over a noisy channel. Note that, for this code the best systematic code [116] achieves the lower bound in (4.10); i.e., it results in $\sigma_p = \sigma_x$. The correlation channel and transmission channel altogether insert up to t errors generated by $\mathcal{N}(0, \sigma_e^2)$. Simulation results are plotted in Figure 4.5. First, based on Figure 4.5(a), the threshold $\theta = 0.0064$ is fixed for $p_d = 90\%$. Next, this is used to estimate ν in Figure 4.5(b). The estimated number is subsequently used to find the location of errors, both for the PGZ and subspace methods, in Figure 4.5(c).[7] Then, the output of Figure 4.5(c), for the subspace method, is fed to the last step to find the magnitude of errors and correct them. Finally, in Figure 4.5(d), the MSE is compared against the quantization error level To put our results in perspective, we also calculate the MSE assuming perfect error localization; it gives 0, 6.5×10^{-5}, and 1.8×10^{-4} respectively for 0, 1, and 2 errors, in every CEQNR. This implies

[7] It is worth mentioning that if the amplitude of errors is fixed, as assumed in [86], the results improve considerably for both methods. For one thing, at the CEQNR of 20dB the probability of correct localization becomes 1.

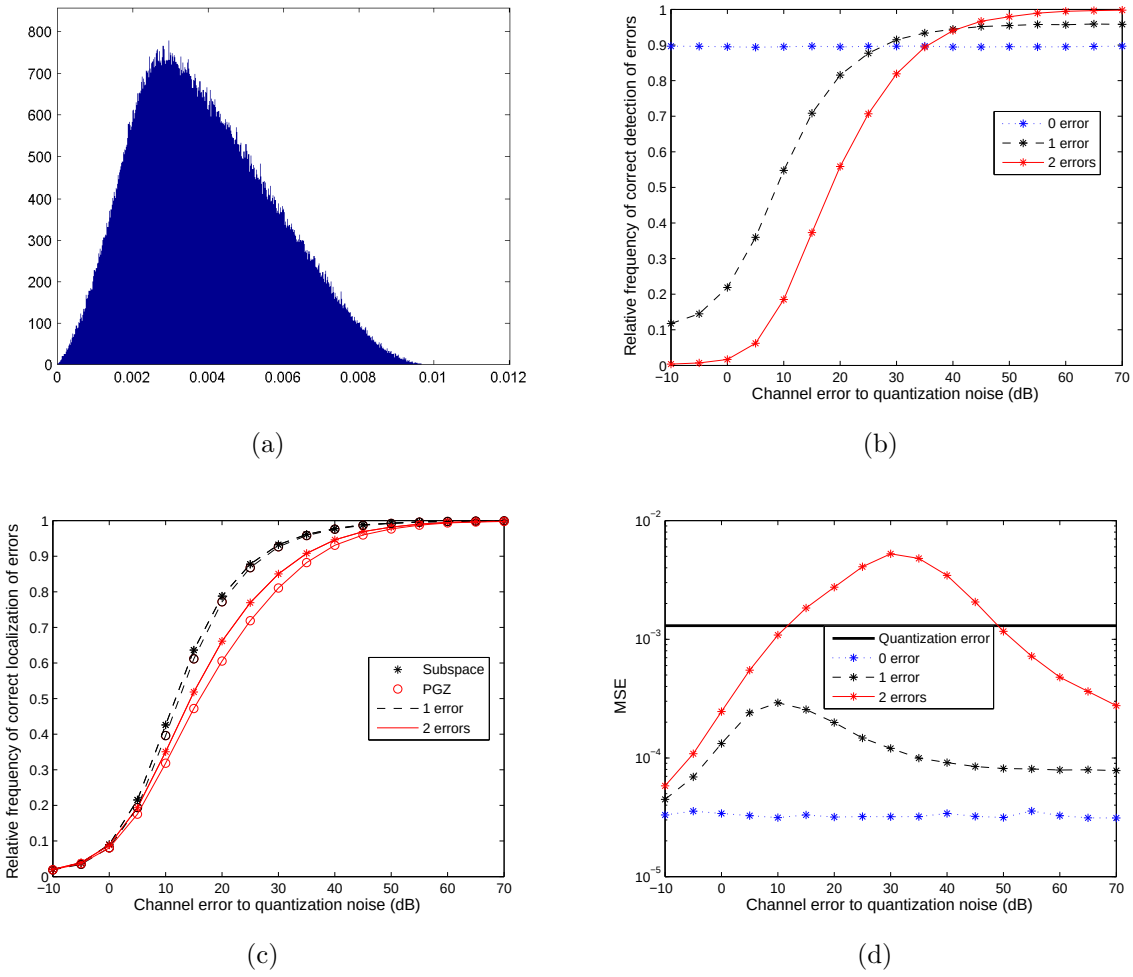


Figure 4.5 Performance evaluation of joint source-channel coding with side information at the decoder, proposed in Figure 4.2, for $(10, 5)$ DFT code. (a) Histogram of $\lambda_{\max}(\tilde{R})$ for the quantized code. This is used to set a threshold for detection. (b) Probability of correct detection of errors for $\theta = 0.0064$. (c) Probability of correct localization of errors based on Figure 4.5(b). (d) The end-to-end distortion for subspace-based error localization given in Figure 4.5(c).

that there is still room to improve the MSE performance of the proposed system, given a better solution for the error localization. Expectedly, for the same number of errors, high rate codes have better performance. As an example, in [109, Figure 6] we show the MSE performance of a systematic $(12, 5)$ code, which is better than that in Figure 4.5(d).

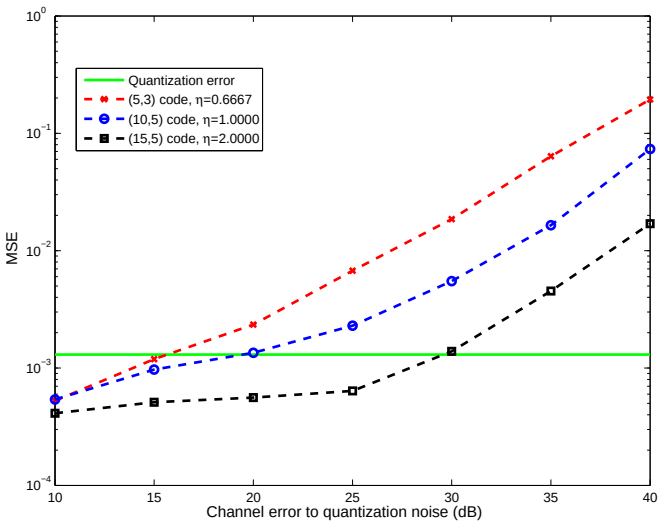


Figure 4.6 The MSE performance of the DJSCC for the GBG correlation model with $\sigma_0 = 0.05\sigma_e, p_1 = 0.03$ and $\rho = 0$. The compression ratio for the codes are equal to $\frac{2}{3}$, 1, and 2, implying compression for the first code and expansion for the third code. For the $(10, 5)$ code there is neither compression nor expansion since $\eta = 1$.

Finally, we compare the performance of the proposed system for DJSCC using three codes $(5, 3)$, $(10, 5)$, and $(15, 5)$, with compression ratios equal to $\frac{2}{3}$, 1, and 2, respectively. Again, for each code we use a G_{sys} with the best MSE performance, in light of Theorem 4.7. Specifically, rows $\{1, 3, 5\}$ are chosen as the systematic rows of G_{sys} for the $(5, 3)$. For the $(10, 5)$ and $(15, 5)$ codes the optimal solution is to choose, respectively, every second and third rows as the systematic rows. The remaining rows correspond to the parity samples. As expected, in Figure 4.6 it can be seen that when code rate decreases the MSE decreases. Note that, in this simulation 3% of parity samples are affected by an impulsive noise whose power is the same as the power of impulses in the correlation channel. The $(10, 5)$ code neither compresses nor expands the input source as it transmit 5 parity samples in lieu of every 5 source samples; nevertheless, it can combat the transmission noise without adding any redundancy.

4.9 Summary

In this chapter, we introduced a parity-based approach to perform DSC based on DFT codes, which provides an alternative to the syndrome-based DSC, presented in Chapter 3.

The proposed scheme is then extended to the case where transmission channel is noisy, which gives rise to DJSCC. Then, motivated by its use in the parity-based DSC and DJSCC, we proposed the construction of systematic DFT frames in this chapter. Numerous systematic DFT frames can be made out of one DFT frame; the performance of these frames differs depending on the relative position of the systematic and parity samples in the codeword. We proved that evenly spaced systematic (or parity) samples result in the minimum mean-squared reconstruction error, whereas the worst performance is expected when the parity samples are circularly consecutive. We also proved that a tight, systematic DFT frame can be realized only if the frame is performing integer oversampling. Finally, for each DFT frame, we classified systematic DFT frame based on their performance. In light of this classification, we are able to use the best systematic BCH-DFT code in the context of DSC and DJSCC. Simulation results are used to compare the efficiency of the proposed systems with syndrome-based DSC and asymptotic bounds, respectively.

Chapter 5

Rate-Adaptive Systems

In this chapter,[1] we take a fresh look at the error localization of BCH-DFT codes. We first analyze how the *subspace-based error localization* outperforms the coding-theoretic one. We then propose an extension of the subspace-based error localization, based on additional syndrome, that improves the existing one and is naturally suitable for *rate-adaptive* DSC. We also propose a new generic subspace-based algorithm to decode BCH-DFT codes that generalizes the encoding and decoding of this important class of DFT codes. It introduces many different decoding matrices for a DFT code; this *diversity* is then used to diminish the effect of the quantization noise and improve the decoding. Finally, the *extended* and *generalized* approaches are combined to maximize the decoding gain.

5.1 Introduction

In all applications of DFT codes where error correction is required, see for example [121, 41, 111, 33], *error localization* is a crucial step of the decoding algorithm. Error localization in BCH-DFT codes can be done by extending the coding-theoretic method of binary BCH codes to the real field [14]. By adopting the MUSIC [96] and ESPRIT [92] based methods of multiple frequency component estimation, Rath and Guillemot [86] proposed *subspace-based* error localization algorithms which perform better than the *coding-theoretic* one [110]; they extended these methods to error and erasure decoding in [87]. The subspace-based approaches have been successfully applied to other real codes such as the *discrete cosine*

[1] The material in this chapter has been partly presented in [115] and published in [110].

transform (DCT) and *discrete sine transform* (DST) codes [88, 66], which are based on orthogonal transform matrices. Between the above subspace-based algorithms, the MUSIC-like approach performs slightly better than the ESPRIT-like approach and it is the most accurate error localization method to date.

Our main contribution, in this chapter, is to further improve the error localization of quantized DFT codes. To put our results in perspective, in Section 5.2, we analyze why the subspace-based error localization outperforms the coding-theoretic one. The key is that a subspace-based method is capable of providing more than one error locating polynomial. These polynomials have the same set of roots, if the code is not quantized; however, they can result in different roots for quantized codes. Even so, by averaging the coefficients of polynomials, one can diminish the effect of the quantization error. With this insight, and based on additional syndrome samples, in Section 5.3 we extend the subspace error localization to further increase the number of polynomials and improve error localization. The proposed algorithm is naturally suitable for rate-adaptive distributed source coding; it is also applicable to channel coding in specific cases. It can be used with DCT and DST codes, too.

A second major advance towards improving the subspace error localization is achieved by generalizing this algorithm based on various decoding matrices, in Section 5.4. We prove that for an (n, k) DFT code there are $\phi(n)$ syndrome matrices for decoding, where $\phi(n)$ is the number of positive integers less than n that are *relatively prime* to n. We then exploit this *diversity* to improve the error localization, when quantization comes into play, by combining the error localization polynomials corresponding to different matrices. This algorithm is referred to as the *generalized subspace* method; it is then combined with the *extended subspace* method to further improve the decoding, in Section 5.5.

Apart from the diversity in decoding, the generalized subspace method brings another novelty at the encoder side; that is, the parity frequencies of BCH-DFT codes (equivalently, the zeros of codewords in the frequency domain) are not required to be cyclically adjacent.[2] This provides substantial flexibility in constructing BCH-DFT codes. Finally, in addition to improving the error localization, both extended and generalized subspace algorithms can be applied to increase the accuracy of the error detection step.

[2]Strictly speaking, these codes are not BCH codes as the spectra of a BCH code has consecutive zeros. However, in Section 5.4, we prove the existence of codes with non-consecutive zeros in the frequency domain which have the same properties as of BCH-DFT.

As elaborated in Section 5.6 by means of a few examples, the main application of the algorithms we develop in this chapter is in rate-adaptive DSC; they can be also applied to channel coding in special cases. The proposed algorithms are based on syndrome augmentation and can be applied both to the syndrome- and parity-based DSC. Simulation results in Section 5.7 demonstrate the capability of the proposed algorithms to perform significantly better than the existing subspace-based error localization, in the presence of quantization noise.

5.2 Error Localization in DFT Codes: A Review

Let the $n \times 1$ vector $\boldsymbol{c}$ represent a codeword generated by an (n, k) DFT code. Also, let $\boldsymbol{r} = \boldsymbol{c}+\boldsymbol{e}$ be a noisy version of $\boldsymbol{c}$ and suppose that the error vector $\boldsymbol{e}$ has $\nu \leq t$ nonzero elements. Let $1 \leq i_1, \ldots, i_\nu \leq n$ and $e_{i_1}, \ldots, e_{i_\nu}$ denote, respectively, the locations and magnitudes of the nonzero elements of $\boldsymbol{e}$. The decoding algorithm in DFT codes is composed of three main steps [14]: *error detection* (to determine ν), *error localization* (to find $i_1, \ldots, i_\nu$), and *error calculation* (to calculate $e_{i_1}, \ldots, e_{i_\nu}$). This chapter is mainly focused on improving the error localization step. Thus, for the moment during the development, we assume that the number of errors (i.e., ν) is known at the decoder. Later, it will be apparent how to determine the true value of ν at the decoder. This is part of the spectral decomposition and will be briefly discussed in Section 5.2.2.

The syndrome of $\boldsymbol{e}$, which is the key input of the decoding algorithm, is computed as

$$\boldsymbol{s} = H\boldsymbol{r} = H(\boldsymbol{c} + \boldsymbol{e}) = H\boldsymbol{e}, \tag{5.1}$$

where $\boldsymbol{s} = [s_1,\ s_2, \ldots, s_d]^T$ is a complex vector with

$$s_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell}, \quad \ell = 1, \ldots, d, \tag{5.2}$$

in which α is defined in (3.8) and

$$X_p = \omega^{-i_p}, \qquad p = 1, \ldots, n, \tag{5.3}$$

where $\omega = e^{-j\frac{2\pi}{n}}$. It is worth noting that $X_1, \ldots, X_n$ are the nth roots of unity and

$X_1, \ldots, X_\nu$ indicate those roots which correspond to the error indices, as we have assumed $i_1, \ldots, i_\nu$ to be the error indices. There are two main approaches to find the error indices; we describe them in the following.

5.2.1 Coding-Theoretic Approach

The classical approach to the error localization is to identify an *error-locator polynomial* whose roots correspond to error locations. The error-locator polynomial can be defined as

$$\Lambda(x) = \prod_{i=1}^{\nu}(1 - xX_i^{-1}) = 1 + \Lambda_1 x + \cdots + \Lambda_\nu x^\nu, \tag{5.4}$$

and its roots $X_1, \ldots, X_\nu$ correspond to the error locations i_p, $p \in [1, \ldots, \nu]$, as from (5.3) we have $i_p = \frac{\log X_p}{\log \omega^{-1}} = \frac{\arg X_p}{\arg \omega^{-1}}$. Note that for a complex z we have $\log z = \ln |z| + j \arg z$ and we take $\arg z \in (0, 2\pi]$. The coefficients $\Lambda_1, \ldots, \Lambda_\nu$ can be found by solving the following set of *consistent* equations [14]

$$s_j\Lambda_\nu + s_{j+1}\Lambda_{\nu-1} + \cdots + s_{j+\nu-1}\Lambda_1 = -s_{j+\nu}, \tag{5.5}$$

for $j = 1, \ldots, d - \nu$. To put it differently, as the IDFT of $\mathbf{\Lambda}_n = [1, \Lambda_1, \ldots, \Lambda_\nu, \mathbf{0}_{1\times(n-\nu-1)}]^T$ becomes zero at the error locations, the circular convolution of $\mathbf{\Lambda}_n$ with the DFT of the error vector is a zero vector [14, 86].

5.2.2 Subspace-Based Approach: A Fresh Look

Alternatively, one can use a *subspace-based* method for error localization [86]. We elaborate the subspace-based method which is along the lines of the MUSIC [96], since it is shown to perform slightly better than the ESPRIT-like [92] method, in [86]. To this end, let $\nu + 1 \leq m \leq d - \nu + 1$ and define the following syndrome matrix

$$S_m = \begin{bmatrix} s_1 & s_2 & \ldots & s_{d-m+1} \\ s_2 & s_3 & \ldots & s_{d-m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_m & s_{m+1} & \ldots & s_d \end{bmatrix}, \tag{5.6}$$

whose elements are given by (5.1). Also, define the covariance matrix as

$$R_m = S_m S_m^H. \tag{5.7}$$

Subspace-based error localization is based on eigendecomposition of R_m. Before proceeding to the details of the algorithm, it is important to point out that S_m can be decomposed as

$$S_m = V_m D V_{d-m+1}^T, \tag{5.8}$$

in which V_m is a *Vandermonde* matrix defined as

$$V_m = \begin{bmatrix} 1 & 1 & \dots & 1 \\ X_1 & X_2 & \dots & X_\nu \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{m-1} & X_2^{m-1} & \dots & X_\nu^{m-1} \end{bmatrix}, \tag{5.9}$$

and D is a diagonal matrix of size ν

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_\nu \end{bmatrix}, \tag{5.10}$$

with nonzero diagonal elements $d_p = \frac{1}{\sqrt{n}} e_{i_p} X_p^\alpha, p = 1, \dots, \nu$. V_m is called the *error-locator matrix* of order m [86], and its columns are the *error-locator vectors* of order m. Since $m > \nu$, the columns of V_m define a ν-dimensional subspace of the m-dimensional vector space, which is referred to as the *channel-error subspace*. Its orthogonal complement subspace is called the *noise subspace* and has dimension $m - \nu$.

One can verify that the rank of S_m is ν. To check this, from linear algebra, we know that $\text{rank}(AB) = \text{rank}(A)$ if B has a full column rank [97]. Then, by twice applying this to (5.8) and recalling that $\nu + 1 \leq m \leq d - \nu + 1$ its is easy to see that $\text{rank}(S_m) = \text{rank}(V_m) = \nu$.

From this, it is obvious that the rank of R_m is ν;[3] thus, it can be eigendecomposed as

$$R_m = [U_\mathrm{e}\ U_\mathrm{q}] \begin{bmatrix} \Delta_\mathrm{e} & \mathbf{0} \\ \mathbf{0} & \Delta_\mathrm{q} \end{bmatrix} [U_\mathrm{e}\ U_\mathrm{q}]^H, \tag{5.11}$$

where the square matrices Δ_e and Δ_q contain the ν largest and $m-\nu$ smallest eigenvalues, and U_e and U_q contain the eigenvectors corresponding to Δ_e and Δ_q, respectively.[4] The sizes of U_e and U_q are $m \times \nu$ and $m \times (m-\nu)$. The columns in U_e span the channel-error subspace [86, Proposition 1] spanned by V_m. Thus, from the fact that $U_\mathrm{e}^H U_\mathrm{q} = \mathbf{0}$, we conclude that

$$V_m^H U_\mathrm{q} = \mathbf{0}. \tag{5.12}$$

Now, let $\boldsymbol{v} = [1, x, x^2, \ldots, x^{m-1}]^T$ where x is a variable that can take on any of $X_1, \ldots, X_n$. We define the function

$$F(x) \triangleq \sum_{j=1}^{m-\nu} \boldsymbol{v}^H \boldsymbol{u}_{q,j} = \sum_{j=1}^{m-\nu} \sum_{k=0}^{m-1} f_{jk} x^k, \tag{5.13}$$

where $\boldsymbol{u}_{q,j}$ represents the jth column of U_q. $F(x)$ can be considered as sum of $m-\nu$ polynomials $\{f_j\}_{j=1}^{m-\nu}$ of order $m-1$; each polynomial is derived from a column of U_q. Let $\mathcal{F}$ denote this set of polynomials restricted to coefficients from U_q. In light of (5.12), each one of these polynomials vanishes for $x = X_1, \ldots, X_\nu$, i.e., $F(x) = 0$ for $X_1, \ldots, X_\nu$. These are the only common roots of $\{f_j\}$ over the nth roots of unity [86][5]; thus, the errors location can be determined by finding the zeros of $F(x)$ over the set of nth roots of unity. It should be mentioned that the above algorithm is based on the *noise subspace*. One may, equivalently, use the *signal subspace* to find the error locations [63]. Also, the above approach is along the lines of the MUSIC [96]; it is shown to perform slightly better than

[3] From the above argument one can see that the number of errors can be found by evaluating the rank of S_m or R_m for any $\nu + 1 \leq m \leq d - \nu + 1$. Thus, since we do not know ν, it is better to choose $m = \lfloor \frac{d}{2} \rfloor$ or $m = \lceil \frac{d}{2} \rceil$, to detect as many errors as possible.

[4] Clearly, since no noise (or quantization error) is considered at this stage, $\Delta_\mathrm{q} = \mathbf{0}$ and Δ_e contains the ν nonzero eigenvalues of R_m.

[5] Assume, for the sake of contradiction, that there are $\nu + 1$ common roots of unity. This implies that V_m has another column (corresponding to $X_{\nu+1}$) for which (5.12) holds; i.e., there are more than ν errors which is contradicting.

the ESPRIT-like [92] method, in [86].

The subspace method outperforms the coding-theoretic error localization. To see this, we can see that $\Lambda(x)$ is the smallest degree polynomial that has roots in $X_1, \ldots, X_\nu$ and lies in the noise subspace; it is achieved for $m = \nu + 1$ in (5.13). As m increases, the degree of polynomials $\{f_j\}$ goes up which gives more *degrees of freedom* (DoF) and helps improve the estimation of the roots, and the error locations consequently. An even more important factor that affects location estimation is the number of polynomials $\{f_j\}$ whose coefficients come from *linearly independent* columns of U_{q}. The more there are such polynomials, the better the estimation can be as the variation due to noise (quantization) is reduced by adding such independent polynomials in $F(x)$.

Although the number of polynomials increases with m, their coefficients may not be independent. The latter depends on the number of nonzero eigenvalues in the noise subspace which is, in turn, related to the rank of S_m and is limited by

$$\operatorname{rank}(S_m) \leq \max_m \min(m, d - m + 1) = \left\lceil \frac{d}{2} \right\rceil . \tag{5.14}$$

This suggests that the optimum value for m is $\lceil \frac{d}{2} \rceil$. Then, based on (5.13), one can expect that, in the presence of noise, the subspace approach will result in an error localization better than the coding-theoretic approach, except when $\nu = t$ and d is even; this is confirmed by simulation results in [86]. In the last case where $\nu = t$ we have $m = \nu + 1$ and there is just one polynomial and its degree is ν, the same as (5.4) in the coding-theoretic approach. In general, $F(x)$ is composed of

$$\mathcal{N}(m) = \min(m, d - m + 1) - \nu \leq \left\lceil \frac{d}{2} \right\rceil - \nu \tag{5.15}$$

independent polynomials where the upper bound is obtained for $m = \lceil \frac{d}{2} \rceil$. We will use $\mathcal{N}(m)$ as a measure of the gain introduced by the subspace-based error localization compared with the coding-theoretic approach.

5.2.3 Quantization Effect

In practice, where quantization comes into play, the received vector is distorted both by the error vector $\boldsymbol{e}$ and quantization noise $\boldsymbol{q}$. Therefore $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} + \boldsymbol{q}$, and its syndrome is

only a perturbed version of $\boldsymbol{s}$ because

$$H\boldsymbol{r} = H(\boldsymbol{c}+\boldsymbol{q}+\boldsymbol{e}) = \boldsymbol{s}_{\mathrm{q}} + \boldsymbol{s} = \tilde{\boldsymbol{s}}, \tag{5.16}$$

where $\boldsymbol{s}_{\mathrm{q}} \equiv H\boldsymbol{q}$ and $\boldsymbol{q} = [q_1,\, q_2, \ldots, q_n]^T$ is the quantization error. The distorted syndrome samples can be written as

$$\begin{aligned}\tilde{s}_\ell &= \frac{1}{\sqrt{n}}\sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell} + \frac{1}{\sqrt{n}}\sum_{p'=1}^{n} q_{i_{p'}} X_{p'}^{\alpha-1+\ell} \\ &= s_\ell + \frac{1}{\sqrt{n}}\sum_{r=1}^{n} q_r \omega^{(\alpha-1+\ell)r}, \quad 1 \le \ell \le d,\end{aligned} \tag{5.17}$$

where $i_{p'}$ shows the index for quantization error. The distorted syndrome matrix $\tilde{S}_m$ and its corresponding covariance matrix $\tilde{R} = \tilde{S}_m \tilde{S}_m^H$ are defined similar to (5.6) and (5.7) but for the distorted syndrome samples and we can write

$$\tilde{R}_m = [\tilde{U}_{\mathrm{e}}\ \tilde{U}_{\mathrm{q}}] \begin{bmatrix} \tilde{\Delta}_e & \mathbf{0} \\ \mathbf{0} & \tilde{\Delta}_{\mathrm{q}} \end{bmatrix} [\tilde{U}_{\mathrm{e}}\ \tilde{U}_{\mathrm{q}}]^H, \tag{5.18}$$

where $\tilde{U}_{\mathrm{e}}$ and $\tilde{U}_{\mathrm{q}}$ span the "estimated" channel error and quantization noise subspaces, respectively. Due to the quantization error, these estimated subspaces are perturbations of the channel and noise subspaces, that is $\tilde{U}_{\mathrm{e}} = U_{\mathrm{e}} + U_{\mathrm{qe}}$ and $\tilde{U}_{\mathrm{q}} = U_{\mathrm{q}} + U_{\mathrm{qq}}$. Consequently, unlike (5.12), $V_m^H \tilde{U}_{\mathrm{q}} = V_m^H U_{\mathrm{qq}} \triangleq \Delta_{\tilde{F}} \neq \mathbf{0}$, or equivalently, the resulting polynomial

$$\tilde{F}(x) \triangleq \sum_{j=1}^{m-\nu} \boldsymbol{v}^H \tilde{\boldsymbol{u}}_{q,j} = \sum_{j=1}^{m-\nu}\sum_{k=0}^{m-1} \tilde{f}_{jk} x^k, \tag{5.19}$$

does not necessarily have roots at the nth roots of unity. Hence, among those n roots, ν roots that result in the smallest $|\tilde{F}(x)|$ are used to estimate error locations.

5.3 Extended Subspace Approach

Error localization is a crucial step in the decoding algorithm of DFT codes. Numerical results show [113, Fig. 2 and 3] that if the location of errors are known at the de-

coder, reconstruction error can be less than quantization error. It is known that an (n, k) DFT code decreases the mean-squared reconstruction error (MSE) by a factor of code rate $r = \frac{k}{n}$ [51, 84, 113]. This motivates the search for methods that can further improve the error localization in DFT codes. In particular, we are interested to know whether error localization can be improved without or with extra syndrome samples. Since our objective is to improve the error localization performance, we assume that the number of errors ν is known at the decoder.

The main idea behind the extended subspace approach is to try increase the number of vectors in noise subspace such that the number of polynomials with linearly independent coefficients and/or their degree grow, in (5.13). We observe that if we are able to construct a syndrome matrix S'_m such that it can be decomposed as

$$S'_m = V_m D V^T_{d'-m+1}, \tag{5.20}$$

for $d' > d$ and $\nu + 1 \leq m \leq d' - \nu + 1$, where V_m and D are defined in (5.9), then we can expect a better estimation for the location of errors. This is because following the same argument that led to (5.13) and (5.14) it is easy to see that, if S'_m is used for error localization, the optimal m in this case is $\lceil \frac{d'}{2} \rceil$ which results in $\lceil \frac{d'}{2} \rceil - \nu \geq \lceil \frac{d}{2} \rceil - \nu$ error-locator polynomials. Then, as explained in Section 5.2.2, this can improve the error localization.

The challenge is to find the entries of the extended syndrome matrix S'_m. That is, we need to find s'_ℓ for all $1 \leq \ell \leq d'$ so as to build

$$S'_m = \begin{bmatrix} s'_1 & s'_2 & \dots & s'_{d'-m+1} \\ s'_2 & s'_3 & \dots & s'_{d'-m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s'_m & s'_{m+1} & \dots & s'_{d'} \end{bmatrix}. \tag{5.21}$$

From (5.20), it can be verified that

$$s'_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell}, \quad \ell = 1, \dots, d'. \tag{5.22}$$

Comparing with (5.2), it is clear that $s'_\ell = s_\ell$ for $1 \leq \ell \leq d$. Thus, we only need to

determine the entries for $d < \ell \leq d'$.

With this in mind, similar to the syndrome vector $\boldsymbol{s}$, we can define the extended syndrome vector $\bar{\boldsymbol{s}}$ as

$$\bar{\boldsymbol{s}} = \bar{H}\boldsymbol{e}, \tag{5.23}$$

where $\bar{H}$ consists of those k columns of the IDFT matrix of order n used to build G. In other words, $\bar{H}$ is the complement of H. More precisely, similar to (3.10),

$$\bar{H} \triangleq \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & \cdots & 1 \\ \omega^{n-\beta} & \cdots & \omega^{n+\alpha-1} \\ \vdots & \ddots & \vdots \\ \omega^{(n-\beta)(n-1)} & \cdots & \omega^{(n+\alpha-1)(n-1)} \end{pmatrix}^H. \tag{5.24}$$

Note that $\bar{H}$ is a $k \times n$ matrix and, from (5.23), for $\ell = 1, \ldots, k$, we have

$$\bar{s}_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{d+\alpha-1+\ell}. \tag{5.25}$$

Now, we can see that

$$s'_\ell = \begin{cases} s_\ell, & 1 \leq \ell \leq d, \\ \bar{s}_{\ell-d}, & d < \ell \leq d', \end{cases} \tag{5.26}$$

where $d' - d$, $d' - d \leq k$, is the number of extra syndromes.

So far we have shown that if we are able to compute (5.23) then we can form the extended syndrome matrix in (5.21), and benefit from the larger number of polynomials it gives as compared to (5.6). But how can we compute $\bar{\boldsymbol{s}}$ at the receiver? Obviously, we do not know $\boldsymbol{e}$ at the decoder; instead, we have $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. Seeing that

$$\bar{H}\boldsymbol{r} = \bar{H}\boldsymbol{e} + \bar{H}\boldsymbol{c} = \bar{\boldsymbol{s}} + \bar{H}\boldsymbol{c} \tag{5.27}$$

we will get $\bar{\boldsymbol{s}}$ provided that $\bar{H}\boldsymbol{c}$, the second term on the right-hand side of (5.27), is either removed or becomes zero. Observe that despite the fact that $H\boldsymbol{c} = \boldsymbol{0}$, $\bar{H}\boldsymbol{c}$ is not necessarily zero. In Section 5.6, we accomplish this for source coding with side-information available at

the decoder, or more generally for DSC. To this end, for the source $\boldsymbol{x}$, the encoder computes and transmits $\bar{\boldsymbol{s}}_x = \bar{H}\boldsymbol{x}$ to the decoder. At the decoder, we have access to side information $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$ and we can compute $\bar{\boldsymbol{s}}_y = \bar{H}\boldsymbol{y} = \bar{\boldsymbol{s}}_x + \bar{\boldsymbol{s}}_e$. From this $\bar{\boldsymbol{s}} = \bar{\boldsymbol{s}}_e = \bar{\boldsymbol{s}}_y - \bar{\boldsymbol{s}}_x$. Note that here $\boldsymbol{x}$ plays the role of $\boldsymbol{c}$.

Finally, considering quantization, $\boldsymbol{c}$ will be replaced by $\boldsymbol{c} + \boldsymbol{q}$; i.e., the new syndrome $\tilde{\bar{s}}_\ell$ will contain a term related to quantization error, similar to $\tilde{s}_\ell$ in (5.17). Likewise, $\tilde{s}'_\ell$ is built upon $\tilde{s}_\ell$ and $\tilde{\bar{s}}_\ell$, that is

$$\tilde{s}'_\ell = \begin{cases} \tilde{s}_\ell, & 1 \le \ell \le d, \\ \tilde{\bar{s}}_{\ell-d}, & d < \ell \le d', \end{cases} \tag{5.28}$$

where $\tilde{\boldsymbol{s}} = \boldsymbol{s}_e + \boldsymbol{s}_\mathrm{q}$, $\tilde{\bar{\boldsymbol{s}}} = \bar{\boldsymbol{s}}_e + \bar{\boldsymbol{s}}_\mathrm{q}$, and $\bar{\boldsymbol{s}}_\mathrm{q} = \bar{H}\boldsymbol{q}$. The new $\tilde{R}'_m = \tilde{S}'_m \tilde{S}'^H_m$ is then used for error localization as detailed in Section 5.2. Then, similar to (5.14) it can be seen that the new optimal m is $\lceil \frac{d'}{2} \rceil$; i.e., we can improve error localization as there are $\lceil \frac{d'}{2} \rceil - \nu \ge \lceil \frac{d}{2} \rceil - \nu$ error-locator polynomials. Hence, the presence of quantization does not really affect the method.

Remark 5.1. Similar to the subspace approach [88], the extended subspace approach can be applied to the DCT and DST codes; further, it can be used even for the non-BCH DCT and DST codes [66].[6]

Remark 5.2. Knowing that $\tilde{R}_m$ can also be used to determine the number of errors ν [117], where the extended error localization is applicable, $\tilde{R}'_m$ can be used for this purpose and it improves the results reasonably.

Remark 5.3. Once the location of errors are determined, it is rather simple to find their amplitude. Let H_e denote the matrix consisting of the columns of H corresponding to error indices, then the errors magnitudes $\boldsymbol{E} = [e_{i_1}, \ldots, e_{i_\nu}]^T$ can be computed from $H_e \boldsymbol{E} = \boldsymbol{s}$, using a least squares method, for example. More details are available in [88, Section 5.3], [113], so we will not discuss it more in depth in this chapter.

Before moving on to the next section, we look at extended subspace error localization for a special, yet important class of DFT codes in channel coding where $n = 2k$. For such a code, $d = k$ and X_p^d is $+1$ (-1) for errors in the even (odd) positions in the codeword. Then, if all errors are in the even (odd) positions, we can simply replace $\bar{\boldsymbol{s}}$ with

[6] The idea of syndrome extension has been recently applied for error correction of DCT codes, with a different algorithm in [90].

$\boldsymbol{s}$ ($-\boldsymbol{s}$). Thus, using (5.26) and (5.21) we can form $S'_m(\tilde{S}'_m)$ and the corresponding $R'_m(\tilde{R}'_m)$ for $d' = 2d$. Subsequently, the eigendecomposition of $\tilde{R}'_m$ for $m = \lceil d'/2 \rceil$ increases the number of polynomials in $\mathcal{F}$ and their degree. This can lead to substantial improvement in error localization; for one thing, Figure 1 in [115] represents the merit of the extended error localization to the existing one in a $(10, 5)$ code. Such a significant improvement in error localization is achieved by using the same d syndrome samples but forming a larger syndrome matrix which allows a larger noise subspace.

5.4 Generalized Subspace Decoding

This section is primarily focused on the generalization of the subspace-based decoding of BCH-DFT codes. Meanwhile, the proposed algorithm gives rise to a more general encoding for this class of codes. Let $V_m^{[i]}$ be the matrix whose columns are the ith powers of the *error-locator vectors* of order m, i.e.,

$$V_m^{[i]} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ X_1^i & X_2^i & \dots & X_\nu^i \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{i(m-1)} & X_2^{i(m-1)} & \dots & X_\nu^{i(m-1)} \end{bmatrix}. \tag{5.29}$$

Also, let $\mathcal{P}_n$ be the set of positive integers less than or equal to n that are *relatively prime* to n. The cardinality of $\mathcal{P}_n$, also called the *Euler phi function* $\phi(n)$, is obviously upper bounded by $|\mathcal{P}_n| = \phi(n) \leq n - 1$ where the upper bound is achieved when n is a *prime number*. [7]

Proposition 5.1. *For any $i \in \mathcal{P}_n$ and $m \geq \nu$, the rank of $V_m^{[i]}$ is equal to the number of errors ν.*

Proof. Since $V_m^{[i]}$ has the form of the Vandermonde matrix with elements $X_1^i, X_2^i, \dots, X_\nu^i$, to prove the claim it suffices to show that the above elements are distinct for any $i \in \mathcal{P}_n$. Suppose for contradiction that $X_p^i = X_{p'}^i$ for some $p \neq p'$, $p, p' \in [1, \dots, \nu]$, and $i \in \mathcal{P}_n$; thus $e^{\frac{j2\pi i}{n}(i_p - i_{p'})} = 1$ which implies $n = \frac{i}{l}(i_p - i_{p'})$ for some integer l. This means that n and i are not relatively prime, which is contradicting. □

[7] Interested readers may refer to [118, Section 6.6] for Euler's function and nth roots of unity.

Next, for any $i \in \mathcal{P}_n$ and $m > \nu$, the columns of $V_m^{[i]}$ define a ν-dimensional subspace of the m-dimensional vector space which we will refer to as the channel error subspace. We will shortly prove that, for an (n, k) code, this generalizes the notion of [86] by introducing $\phi(n)$ different sets of *spanning basis* for channel error subspace rather than just one. Recall from linear algebra that there are infinitely many different sets of basis which can span the same subspace. Clearly, the channel error subspace of (5.9) is attained for $i = 1$. Initially, similar to (5.6), for $\nu + 1 \leq m \leq d - \nu + 1$, we define the ith syndrome matrix by

$$S_m^{[i]} = V_m^{[i]} D V_{d-m+1}^{[i]\,T}. \tag{5.30}$$

It can be seen that $S_m^{[i]}$, for each i, is composed of d syndrome samples[8] at most; these samples can be identified in the first row and last column. Before proceeding, we should determine the entries of $S_m^{[i]}$. Through simple algebra, one can show that the right-hand side of (5.30) simplifies to

$$S_m^{[i]} = \begin{bmatrix} s_{\lceil\!\lceil 0 \rceil\!\rceil_n} & s_{\lceil\!\lceil i \rceil\!\rceil_n} & \cdots & s_{\lceil\!\lceil i(d-m) \rceil\!\rceil_n} \\ s_{\lceil\!\lceil i \rceil\!\rceil_n} & s_{\lceil\!\lceil 2i \rceil\!\rceil_n} & \cdots & s_{\lceil\!\lceil i(d-m+1) \rceil\!\rceil_n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\lceil\!\lceil i(m-1) \rceil\!\rceil_n} & s_{\lceil\!\lceil im \rceil\!\rceil_n} & \cdots & s_{\lceil\!\lceil i(d-1) \rceil\!\rceil_n} \end{bmatrix}, \tag{5.31}$$

in which $i \in \mathcal{P}_n$ and the subscripts are interpreted modulo n such that $\lceil\!\lceil a \rceil\!\rceil_n \triangleq b + 1$ where $a \equiv b \pmod{n}$ and $0 \leq b < n$. To see the rationale behind the modulo operation recall that $X_p^n = 1$ for any p. Again, it is easy to see that $S_m^{[1]} = S_m$. However, note that for $i > 1$ we will have a different syndrome matrix than (5.8), where the syndrome samples and their order varies based on i. Note that, the elements of $S_m^{[i]}$, in general, are given by

$$s_\ell = \frac{1}{\sqrt{n}} \sum_{p=1}^{\nu} e_{i_p} X_p^{\alpha-1+\ell}, \quad \ell = 1, \ldots, n. \tag{5.32}$$

Thus, they are available from (5.2) for $\ell = 1, \ldots, d$, and for $\ell = d + 1, \ldots, n$ they become equal to $\bar{s}_{\ell-d}$, where $\bar{s}_\ell$ is the extended syndrome sample defined in (5.25). Once the

[8]With a little abuse of notation, we use the term syndrome samples both for syndrome and extended syndrome samples. Thus, we assume we have s_ℓ in which ℓ can take any values from 1 to n, whereas originally this range was $[1, \ldots, d]$. Such a case is plausible, for instance, in rate-adaptive systems based on DFT codes, as discussed in Section 5.6.

elements of $S_m^{[i]}$ are properly set, we can define the ith covariance matrix

$$R_m^{[i]} = S_m^{[i]} S_m^{[i]\,H}, \tag{5.33}$$

and eigendecompose it as

$$R_m^{[i]} = [U_{\mathrm{e}}^{[i]}\ U_{\mathrm{q}}^{[i]}] \begin{bmatrix} \Delta_{\mathrm{e}}^{[i]} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathrm{q}}^{[i]} \end{bmatrix} [U_{\mathrm{e}}^{[i]}\ U_{\mathrm{q}}^{[i]}]^H. \tag{5.34}$$

The matrices $U_{\mathrm{e}}^{[i]}, U_{\mathrm{q}}^{[i]}, \Delta_{\mathrm{e}}^{[i]}$, and $\Delta_{\mathrm{q}}^{[i]}$, respectively, have the same sizes as $U_{\mathrm{e}}, U_{\mathrm{q}}, \Delta_{\mathrm{e}}$, and Δ_{q} and hold similar properties. Specifically, we have

Proposition 5.2. *The columns of $U_{\mathrm{e}}^{[i]}$ span the channel error subspace.*

Proof. On the one hand, we observe that the rank of $R_m^{[i]}$ is ν since the rank of $S_m^{[i]}$ is so by construction. Therefore, $\Delta_{\mathrm{q}}^{[i]} = \mathbf{0}$ and $R_m^{[i]}$ can be expressed as $R_m^{[i]} = U_{\mathrm{e}}^{[i]} \Delta_{\mathrm{e}}^{[i]} U_{\mathrm{e}}^{[i]\,H}$. On the other hand, from (5.33) and (5.30), it can be seen that $R_m^{[i]} = V_m^{[i]} D V_{d-m+1}^{[i]\,T} V_{d-m+1}^{[i]\,*} D^H V_m^{[i]\,H}$. Hence, $U_{\mathrm{e}}^{[i]} = V_m^{[i]} M^{[i]}$ where $M^{[i]} = D V_{d-m+1}^{[i]\,T} (\Delta_{\mathrm{e}}^{[i]})^{\frac{-1}{2}}$; i.e., the columns of $U_{\mathrm{e}}^{[i]}$ can be expressed as linear combinations of the columns of $V_m^{[i]}$, and vice versa; this completes the proof. □

An immediate implication of the above proposition is that the columns in $U_{\mathrm{q}}^{[i]}$ span the noise subspace. More importantly, we have the following theorem.

Theorem 5.1. *For an (n, k) BCH-DFT code defined by* (3.7)*, there exist $\phi(n)$ syndrome matrices for decoding, where $\phi(n)$ is the Euler phi function.*

Proof. On the one hand, from eigendecomposition in (5.44) we have $U_{\mathrm{e}}^{[i]\,H} U_{\mathrm{q}}^{[i]} = \mathbf{0}$. On the other hand, from Proposition 5.2 we know that the columns in $U_{\mathrm{e}}^{[i]}$ and $V_m^{[i]}$ span the same subspace. Therefore,

$$V_m^{[i]\,H} U_{\mathrm{q}}^{[i]} = \mathbf{0} \quad \forall\, i \in \mathcal{P}_n. \tag{5.35}$$

Hence, since i can take $\phi(n)$ different values, one can make $\phi(n)$ different syndrome matrices $S_m^{[i]}$, as defined in (5.31), and utilize them for error localization and detection. □

The *diversity* introduced in Theorem 5.1 brings in two main novelties which are discussed in the remainder of this section.

5.4.1 Improved Decoding

Since (5.35) holds for any $i \in \mathcal{P}_n$, for a complex variable x we define $\boldsymbol{v}^i = [1, x^i, x^{2i}, \ldots, x^{i(m-1)}]^T$ and form

$$F^{[i]}(x) \triangleq \sum_{j=1}^{m-\nu} \boldsymbol{v}^{i\,H} \boldsymbol{u}_{q,j}^{[i]} = \sum_{j=1}^{m-\nu} \sum_{k=0}^{m-1} f_{jk}^{[i]} x^{ki}. \tag{5.36}$$

For each i, the function $F^{[i]}(x)$ can be considered as sum of $m-\nu$ polynomials $\{f_j^{[i]}\}_{j=1}^{m-\nu}$ of order $m-1$; each polynomial corresponds to one column of $U_{\mathrm{q}}^{[i]}$. Let $\mathcal{F}^{[i]}$ denote this set of polynomials. In view of (5.35), $F^{[i]}(x) = 0$ for $X_1, \ldots, X_\nu$, and these are the only common roots of $\{f_j^{[i]}\}$ over the nth roots of unity; thus, the error locations can be determined by finding the zeros of $F^{[i]}(x)$ over the set of nth roots of unity. That is, each $F^{[i]}(x)$ individually can be employed for error localization. Therefore, we can equivalently define

$$\Gamma(x) \triangleq \sum_{i \in \mathcal{P}_n} F^{[i]}(x) = \sum_{i \in \mathcal{P}_n} \sum_{j=1}^{m-\nu} \sum_{k=0}^{m-1} f_{jk}^{[i]} x^{ki}, \tag{5.37}$$

and use it for error localization. That is, the zeros of $\Gamma(x)$ over the set of nth roots of unity give $X_1, \ldots, X_\nu$.

Comparing with (5.13), it can be seen that (5.37) combines $\phi(n)$ subspace-based decoding functions; this is referred to as *diversity* in this chapter. Hence, the decoding polynomial $\Gamma(x)$ introduces both *diversity* and *degrees of freedom*[9] in comparison with the error-locator polynomial $\Lambda(x)$, whereas $F(x)$ provides degrees of freedom only. Obviously, $1 \leq \phi(n) \leq n-1$ and diversity gain $\phi(n)$ is maximized when n is a *prime number*. It should be mentioned that the diversity is achieved at the expense of an increase in the number of transmitted syndromes, for a given code, which implies a lower transmission rate.

It should be emphasized that when there is no quantization error, utilizing (5.37) presents no gain over using (5.13) for the same reason that there is no difference in using (5.13) over (5.4). In other words, the coding-theoretic, subspace, and generalized subspace approaches all have the same performance, and result in the exact location of errors, as long as the number of channel errors are within the capacity of the code. Nevertheless,

[9] *Diversity* and *degrees of freedom* are well-established terms in wireless communication [105]; we use them in this context for the similarity of concepts.

when quantization error comes into play, the generalized subspace approach outperforms the subspace approach and the subspace approach does better than the coding-theoretic one.

Now let us analyze the effect of quantization error. The distorted syndrome matrix can be represented as $\tilde{S}_m^{[i]} = S_m^{[i]} + Q_m^{[i]}$. Then, the eigendecomposition of the corresponding covariance matrix $\tilde{R}^{[i]} = \tilde{S}_m^{[i]} \tilde{S}_m^{[i]\,H}$ results in

$$\tilde{R}_m^{[i]} = [\tilde{U}_{\mathrm{e}}^{[i]}\ \tilde{U}_{\mathrm{q}}^{[i]}] \begin{bmatrix} \tilde{\Delta}_e^{[i]} & \mathbf{0} \\ \mathbf{0} & \tilde{\Delta}_{\mathrm{q}}^{[i]} \end{bmatrix} [\tilde{U}_{\mathrm{e}}^{[i]}\ \tilde{U}_{\mathrm{q}}^{[i]}]^H. \tag{5.38}$$

Let $\tilde{U}_{\mathrm{e}}^{[i]} = U_{\mathrm{e}}^{[i]} + U_{\mathrm{qe}}^{[i]}$ and $\tilde{U}_{\mathrm{q}}^{[i]} = U_{\mathrm{q}}^{[i]} + U_{\mathrm{qq}}^{[i]}$. Following the same line of arguments as for (5.35) we obtain

$$V_m^{[i]\,H} \tilde{U}_{\mathrm{q}}^{[i]} = V_m^{[i]\,H} (U_{\mathrm{q}}^{[i]} + U_{\mathrm{qq}}^{[i]}) = V_m^{[i]\,H} U_{\mathrm{qq}}^{[i]} \triangleq \Delta_{\tilde{F}}^{[i]}, \tag{5.39}$$

and similar to (5.37) we get

$$\tilde{\Gamma}(x) = \sum_{i \in \mathcal{P}_n} \tilde{F}^{[i]}(x) = \sum_{i \in \mathcal{P}_n} \sum_{j=1}^{m-\nu} \sum_{k=0}^{m-1} \tilde{f}_{jk}^{[i]} x^{ki}. \tag{5.40}$$

Our goal is to reduce the effect of $\Delta_{\tilde{F}}^{[i]}$ and this is done through adding polynomials for different i since the entries of

$$\Delta_{\tilde{\Gamma}} = \frac{1}{\phi(n)} \sum_{i \in \mathcal{P}_n} \Delta_{\tilde{F}}^{[i]} \tag{5.41}$$

diminish as the cardinality of $\mathcal{P}_n$ increases. Intuitively, this is because the perturbation caused by the quantization error is reduced by averaging.

5.4.2 Generalized Encoding

Another characteristic of the generalized syndrome matrix is that, except for $i = 1$, the syndrome samples used to build (5.31) are not successive samples. This suggests that $n - k$ zeros in the frequency domain, padded by Σ in (3.7) to make a BCH code, are not constrained to be consecutive. In others words, we can have BCH-DFT codes with

non-consecutive zeros in the frequency domain, or equivalently, the rows of H are the not consecutive powers of the first n powers of ω. This is because for $\xi = \omega^i$ where i is relatively prime to n, $\{\omega^0, \omega^1, \omega^2, \ldots, \omega^{n-1}\}$ and $\{\xi^0, \xi^1, \xi^2, \ldots, \xi^{n-1}\}$ both are the set of roots of unity. This was originally observed by Marshall [72], however in (5.31) we present the decoding algorithm as well. Hence, unlike BCH codes which are constructed by selecting a sequence of $n-k$ cyclically adjacent frequencies as the parity frequencies, as a corollary of Theorem 5.1, we have

Corollary 5.1. *An (n,k) BCH-DFT code in the complex field can be constructed by selecting any $n-k$ frequencies, spaced by $i < n$, as the parity frequencies, as long as* $\gcd(n,i) = 1$. *To decode such a code, one can use $S_m^{[i]}$ in* (5.31).

The fact that the parity frequencies (equivalently, the zeros of Σ) are not required to be cyclically adjacent provides substantial flexibility in constructing real/complex DFT codes. The position of these zeros are determined by the indices of syndrome samples used to build up $S_m^{[i]}$. Specifically, for an (n,k) BCH-DFT code, $\Sigma^{[i]}$ has k nonzero elements given by $\Sigma^{[i]}_{\ell_1,\ell_2} = 1$, $\ell_1 = \lceil i(n-\ell_2) + \alpha \rceil_n$ and $\ell_2 = 1, \ldots, k$. The parity-check matrix $H^{[i]}$ is comprised of the columns of the IDFT matrix W_n^H corresponding to these d zeros.

5.5 Generalized-Extended Subspace Error Localization

To maximize the diversity gain, we can combine the extended algorithm of Section 5.3 with the generalized algorithm of Section 5.4. Suppose the total number of syndrome samples at the decoder is $\bar{d} \in [d, n]$. Then, for each d', $d' = d, \ldots, \bar{d}$, $i \in \mathcal{P}_n$, and $\nu+1 \le m \le d'-\nu+1$ we define $S_m^{'[i]}$ similar to $S_m^{[i]}$ in (5.31) as

$$S_m^{'[i]} = \begin{bmatrix} s_{\lceil 0 \rceil_n} & s_{\lceil i \rceil_n} & \cdots & s_{\lceil i(d'-m) \rceil_n} \\ s_{\lceil i \rceil_n} & s_{\lceil 2i \rceil_n} & \cdots & s_{\lceil i(d'-m+1) \rceil_n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\lceil i(m-1) \rceil_n} & s_{\lceil im \rceil_n} & \cdots & s_{\lceil i(d'-1) \rceil_n} \end{bmatrix}, \tag{5.42}$$

with the optimal value of $m = \lceil d'/2 \rceil$. Next, we can compute

$$R_m^{'[i]} = S_m^{'[i]} S_m^{'[i]\,H}, \tag{5.43}$$

and eigendecompose it as

$$R_m^{'[i]} = [U_{\mathrm{e}}^{'[i]}\ U_{\mathrm{q}}^{'[i]}] \begin{bmatrix} \Delta_{\mathrm{e}}^{'[i]} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathrm{q}}^{'[i]} \end{bmatrix} [U_{\mathrm{e}}^{'[i]}\ U_{\mathrm{q}}^{'[i]}]^H. \tag{5.44}$$

Again $U_{\mathrm{e}}^{'[i]}$ and $\Delta_{\mathrm{e}}^{'[i]}$ have the same sizes as U_{e} and Δ_{e}, and like Proposition 5.2 we have

Proposition 5.3. *The columns of $U_{\mathrm{e}}^{'[i]}$ span the channel error subspace.*

Proof. The proof is very similar to that of Proposition 5.2, so we omit it for brevity. Just note that similar to (5.30), $S_m^{'[i]} = V_m^{[i]} D V_{d'-m+1}^{[i]\,T}$ and thus the rank of $S_m^{'[i]}$ and $R_m^{'[i]}$ is ν. □

Note that for $i = 1$ we can always form $S_m^{'[i]}$ whereas for other $i > 1$ we may not be able to form $S_m^{'[i]}$, depending on the availability of the corresponding syndrome samples. One can see that we need $s_{\llbracket i(d'-j) \rrbracket_n}$ for any $j = 1, \ldots, d'$ in order to form $S_m^{'[i]}$. Then, since we have assumed that there are $\bar{d}$ syndrome samples at the decoder, we can form $S_m^{'[i]}$ if and only if $\llbracket i(d'-j) \rrbracket_n \leq \bar{d}$ for any $j = 1, \ldots, d'$. Let $\mathcal{I}_m$ be the set of those $i \in \mathcal{P}_n$ for which all elements of $S_m^{'[i]}$ are available, for a given $\bar{d}$. It can be seen that $1 \leq |\mathcal{I}_m| \leq \phi(n)$, where the upper bound is attainable if and only if $\bar{d} = n$. Note that $\mathcal{I}_m$ is non-empty as it always contains $i = 1$. Since for each $i \in \mathcal{I}_m$ and each $d' = d, \ldots, \bar{d}$ we have one $S_m^{'[i]}$, we can form the corresponding $F^{'[i]}(x)$, define

$$\psi(x) \triangleq \sum_{m=\lceil \frac{d}{2} \rceil}^{\lceil \frac{\bar{d}}{2} \rceil} \sum_{i \in \mathcal{I}_m} F^{'[i]}(x) = \sum_{m=\lceil \frac{d}{2} \rceil}^{\lceil \frac{\bar{d}}{2} \rceil} \sum_{i \in \mathcal{I}_m} \sum_{j=1}^{m-\nu} \sum_{k=0}^{m-1} f_{jkm}^{'[i]} x^{ki}, \tag{5.45}$$

and use it to find the location of errors. For each d' we use the corresponding optimal $m = \lceil d'/2 \rceil$. Then, comparing (5.45) with (5.13), we define the total decoding diversity as

$$\mathcal{D}(d, \bar{d}) = \sum_{m=\lceil d/2 \rceil}^{\lceil \bar{d}/2 \rceil} |\mathcal{I}_m|, \tag{5.46}$$

considering the two extreme cases, i.e., $\bar{d} = d$ and $\bar{d} = n$, it can be checked that, $1 \leq \mathcal{D}(d, \bar{d}) \leq (\lceil \frac{n}{2} \rceil - \lceil \frac{n-k}{2} \rceil + 1)\phi(n)$. The upper bound can be achieved only if n syndrome samples are available at the decoder. In such an extreme case, $\mathcal{I}_m = \mathcal{P}_n$ thus $|\mathcal{I}_m| = \phi(n)$ and Theorem 5.1 can be generalized as

Theorem 5.2. *In addition to $\phi(n)$ syndrome matrices, one can define $(\lceil \frac{n}{2} \rceil - \lceil \frac{n-k}{2} \rceil)\phi(n)$ extended syndrome matrices of different sizes for decoding an (n, k) BCH-DFT code.*

Finally, it is worth mentioning that, by combining the extended and generalized approaches, when there are $\bar{d}$ syndrome samples to decode ν errors, the total number of error-locator polynomials is given by

$$\mathcal{G} = \sum_{m=\lceil d/2 \rceil}^{\lceil \bar{d}/2 \rceil} |\mathcal{I}_m|(m - \nu). \tag{5.47}$$

What we discussed in this section was for unquantized DFT codes. For quantized codes we will have $\tilde{S}_m^{\prime[i]}, \tilde{R}_m^{\prime[i]}, \tilde{F}^{\prime[i]}(x)$ and $\tilde{\psi}(x)$. For such codes, by adding many $(\mathcal{G})$ polynomials each of which may have different roots due to quantization error, $\tilde{\psi}(x)$ can result in more accurate roots and improve error localization.[10] Then $\mathcal{G}$ can be an indicator of the generalized-extended subspace-based error localization gain with respect to the coding-theoretic approach. In the extreme case where $\bar{d} = n$ and n is a prime number $\mathcal{G}$ is huge.

5.6 Application and Examples

In this section we develop a algorithm for rate adaptation for both syndrome- and parity-based DSC based on DFT codes, which were developed in Sections 3.3 and 4.1, respectively. Rate adaptation is a well-known idea, both in channel coding and DSC context, in which the encoder switches flexibly among different coding rates. Such a capability, combined with a feedback channel from the decoder, means that the encoder can adapt itself to the degree of statistical dependence between sources. Rate adaptation becomes more important when codeword lengths are short, as such codes are more vulnerable to variations of the (virtual) channel. Hence, it is a useful technique to improve the rate-distortion performance when short source blocks are encoded [117, 119].

5.6.1 Rate-Adaptive Distributed Lossy Source Coding

When the statistical dependency between the sources varies or is not known at the encoder, a *rate-adaptive* system with feedback is an appealing solution [119,127]. Rate-adaptive DSC

[10] Thus far, it should be clear that any of the error locator polynomials in (5.4), (5.13), (5.37), and (5.45), give the exact location of errors; i.e., they have the same performance.

based on binary codes, e.g., *puncturing* the parity or syndrome bits of turbo and LDPC codes, have been proposed in [119,103,60,36]. In the sequel, we extend DSC based on DFT codes [111] to perform DSC in a rate-adaptive fashion. We consider two continuous-valued correlated sources $\boldsymbol{x}$ and $\boldsymbol{y}$ where x_i and y_i are statistically dependent by $y_i = x_i + e_i$, and e_i is continuous, i.i.d., and independent of x_i. The goal is to compress $\boldsymbol{x}$ given that $\boldsymbol{y}$ is known at the decoder, only.

Rate-adaption using puncturing is not natural for syndrome-based DSC systems [103]. Instead, the encoder can transmit a short syndrome based on an aggressive code and augment it with additional syndrome samples, if decoding fails. This process loops until the decoder gets sufficient samples for successful decoding. This approach is viable only for feedback channels with reasonably short round-trip time [119].

Syndrome-based approach

In the syndrome-based DSC based on DFT codes, as one can find in Figure 3.2, the encoder transmits $\boldsymbol{s}_x = H\boldsymbol{x}$ to the decoder. At the decoder, we have access to the side information $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$ and we can compute its syndrome so as to find $\boldsymbol{s}_e = \boldsymbol{s}_y - \boldsymbol{s}_{\boldsymbol{x}}$. Then based on this syndrome samples, one can form the syndrome matrix and correct the errors, as we explained in Section 5.2.

For rate adaptation, if required, the encoder transmits $\bar{\boldsymbol{s}}_x = \bar{H}\boldsymbol{x}$ sample by sample, in which $\bar{H}$ is defined in (5.24); the receiver also can compute $\bar{\boldsymbol{s}}_y = \bar{H}\boldsymbol{y} = \bar{\boldsymbol{s}}_x + \bar{\boldsymbol{s}}_e$ and evaluate $\bar{\boldsymbol{s}}_e = \bar{\boldsymbol{s}}_y - \bar{\boldsymbol{s}}_x$. After that, we can form the extended or generalized syndrome matrices where $\boldsymbol{s} = \boldsymbol{s}_e$, $\bar{\boldsymbol{s}} = \bar{\boldsymbol{s}}_e$, and

$$s_\ell = \begin{cases} s_\ell, & 1 \le \ell \le d, \\ \bar{s}_{\ell-d}, & d < \ell \le n. \end{cases} \tag{5.48}$$

and use it for decoding. In short, the rate adaptation algorithm can be summarized as:

- The decoder requests some extra syndrome samples based on the estimated number of errors, for example when $\hat{\nu} > t$ where $\hat{\nu}$ is the estimated number of errors.
- The encoder computes $\bar{\boldsymbol{s}}_x = \bar{H}\boldsymbol{x}$ and transmits it to the decoder sample by sample.
- The decoder computes $\bar{\boldsymbol{s}}_y = \bar{H}\boldsymbol{y}$ and finds $\bar{\boldsymbol{s}}_e = \bar{\boldsymbol{s}}_y - \bar{\boldsymbol{s}}_x$. It then can form R'_m and use the extended subspace decoding algorithms to find the location of errors.

As usual, when quantization is considered this equation needs to be updated as

$$\tilde{s}'_\ell = \begin{cases} \tilde{s}_\ell, & 1 \le \ell \le d, \\ \bar{\tilde{s}}_{\ell-d}, & d < \ell \le n, \end{cases} \tag{5.49}$$

in which $\tilde{\boldsymbol{s}} = \boldsymbol{s}_e + \boldsymbol{s}_\mathrm{q}$, $\bar{\tilde{\boldsymbol{s}}} = \bar{\boldsymbol{s}}_e + \bar{\boldsymbol{s}}_\mathrm{q}$, and $\bar{\boldsymbol{s}}_\mathrm{q} = \bar{H}\boldsymbol{q}$. The new $\tilde{R}'_m = \tilde{S}'_m \tilde{S}'^H_m$ then is used for error localization as detailed in Sections 5.2–5.4. Note that the code is incremental, so the encoder does not need to re-encode the sources when more samples are requested. It buffers and transmits the extra syndrome samples to the decoder sample by sample.

Parity-based approach

In the parity-based DSC based on DFT codes, as shown in Figure 4.1, for an input sequence $\boldsymbol{x}$, the encoder computes the codeword $\boldsymbol{c} = [\boldsymbol{x} \mid \boldsymbol{p}]^T$ with respect to a systematic DFT code and transmits only $\boldsymbol{p}$, for the sake of compression. At the decoder, we have access to $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$ (the noisy version of $\boldsymbol{x}$) in addition to $\boldsymbol{p}$; we form $\boldsymbol{z} = [\boldsymbol{y} \mid \boldsymbol{p}]^T$ and compute its syndrome. Since $\boldsymbol{z} = \boldsymbol{c} + \boldsymbol{e}'$ where $\boldsymbol{e}' = [\boldsymbol{e} \mid \boldsymbol{0}]^T$ and $\boldsymbol{s}_c = 0$, we have $\boldsymbol{s}_z = \boldsymbol{s}_{e'}$. Thus we have the syndrome of error and we can perform decoding.

Although parity- and syndrome-based DSC systems present somewhat different methods for binning, the technique we use for rate-adaptation is the same. For rate adaptation in a parity-based system, once requested by decoder, the encoder computes and transmits $\bar{\boldsymbol{s}}_c = \bar{H}\boldsymbol{c}$; the decoder also computes $\bar{\boldsymbol{s}}_z = \bar{H}\boldsymbol{z} = \bar{\boldsymbol{s}}_c + \bar{\boldsymbol{s}}_{e'}$ and evaluate $\bar{\boldsymbol{s}}_{e'} = \bar{\boldsymbol{s}}_z - \bar{\boldsymbol{s}}_c$. The remainder of the algorithm is similar to the syndrome-based rate-adaptive DSC. Observe that even if the parity samples are not error-free, the above algorithm works [117], which gives rate-adaptive distributed joint source-channel coding.

5.6.2 Rate-Adaptive Channel Coding

An approach similar to the one used for parity-based rate-adaptive DSC can be used to make rate-adaptive DFT channel codes. There are however a few differences: First, there is no need to use a systematic code for encoding, any DFT code generated by (3.7) or, in general, as stated in Section 5.4.2 can be used for encoding. Second, since the whole codeword is transmitted over a noisy channel, parity samples are no longer error-free. On the other hand, similar to DSC the extended syndrome samples $\bar{\boldsymbol{s}}_c$ are to be sent

over a noiseless channel, if an improvement is expected by virtue of the extended and/or generalized subspace decodings.

5.6.3 Examples

In order to fully utilize the extended and generalized subspace decoding algorithms, in this subsection, we assume that the decoder knows n syndrome samples of error, for every codeword of an (n, k) code. If there are fewer samples, we cannot build up $S_m^{[i]}$ and $S'^{[i]}_m$ for some i because we may not have all samples corresponding to those matrices. It is worth noting that, once the syndrome samples of error are known, there is now difference in decoding algorithm used for DSC (syndrome- or parity-based) and channel coding problems.

Example 1

Consider the $(10, 5)$ code, for which $\mathcal{P}_n = \{1, 3, 7, 9\}$. From (5.31) we have

$$S_3^{[1]} = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \\ s_3 & s_4 & s_5 \end{bmatrix}, S_3^{[3]} = \begin{bmatrix} s_1 & s_4 & s_7 \\ s_4 & s_7 & s_{10} \\ s_7 & s_{10} & s_3 \end{bmatrix},$$

$$S_3^{[7]} = \begin{bmatrix} s_1 & s_8 & s_5 \\ s_8 & s_5 & s_2 \\ s_5 & s_2 & s_9 \end{bmatrix}, S_3^{[9]} = \begin{bmatrix} s_1 & s_{10} & s_9 \\ s_{10} & s_9 & s_8 \\ s_9 & s_8 & s_7 \end{bmatrix}.$$

It is seen that different matrices have several samples in common but they differ in some others. The latter implies an increase in the code rate if we wish to exploit more than one matrix for decoding at the same time.

Interestingly, the same syndrome samples are used to form $S_m^{[i]}$ for $i \in \{2, 4, 6, 8\}$. The

only difference is that the position of those samples differ for each i, as is seen below

$$S_3^{[2]} = \begin{bmatrix} s_1 & s_3 & s_5 \\ s_3 & s_5 & s_7 \\ s_5 & s_7 & s_9 \end{bmatrix}, S_3^{[4]} = \begin{bmatrix} s_1 & s_5 & s_9 \\ s_5 & s_9 & s_3 \\ s_9 & s_3 & s_7 \end{bmatrix}, \tag{5.50}$$

$$S_3^{[6]} = \begin{bmatrix} s_1 & s_7 & s_3 \\ s_7 & s_3 & s_9 \\ s_3 & s_9 & s_5 \end{bmatrix}, S_3^{[8]} = \begin{bmatrix} s_1 & s_9 & s_7 \\ s_9 & s_7 & s_5 \\ s_7 & s_5 & s_3 \end{bmatrix}. \tag{5.51}$$

This means that without any increase in code rate, i.e., merely by changing the arrangement of the syndrome samples and summing up the resulting polynomials (averaging the results), the accuracy of the decoding can be improved. Nevertheless, it should be noted that for $i \in \{2, 4, 6, 8\}$ Proposition 5.1 does not hold because $X_1^i, \ldots, X_\nu^i$ are not distinct anymore. In fact, for such an i, by using $S_3^{[i]}$ the algorithm cannot differentiate the errors in locations i_p and $i_{p'}$ when $i_p \equiv i_{p'} \pmod 5$, because $X_p^i = X_{p'}^i$. That is, the above matrices can be used only if we have some specific side information about errors. For instance, when we know that the first (last) half of the samples in each codeword are error-free.

Finally, let us examine

$$S_3^{[5]} = \begin{bmatrix} s_1 & s_6 & s_1 \\ s_6 & s_1 & s_6 \\ s_1 & s_6 & s_1 \end{bmatrix}, \tag{5.52}$$

which is built upon only two syndrome samples. Clearly, the last row and columns can be removed as they are redundant. With $S_3^{[5]}$ one can only tell if the errors are in odd or even locations, since $i_p \equiv i_{p'} \pmod 2 \Rightarrow X_p^5 = X_{p'}^5$.

Remark 5.4. The last two cases were not included in our generic algorithm in Section 5.4 since they cannot be used to determine the location of errors uniquely. However, we can use them to remove the ambiguity partly, as we explained above.

Example 2

Here, using the $(11, 3)$ code, we explain how we can make use of extending and generalizing the syndrome matrices at the same time. Since $n = 11$ is a prime number, $\mathcal{P}_n = \{1, \ldots, 10\}$

and we can have 10 syndrome matrices for each $d' \in [8, \ldots, 11]$. For the extreme case of $d' = 11$, these matrices share the same elements with different arrangements. For instance,

$$S_6^{'[2]} = \begin{bmatrix} s_1 & s_3 & s_5 & s_7 & s_9 & s_{11} \\ s_3 & s_5 & s_7 & s_9 & s_{11} & s_2 \\ s_5 & s_7 & s_9 & s_{11} & s_2 & s_4 \\ s_7 & s_9 & s_{11} & s_2 & s_4 & s_6 \\ s_9 & s_{11} & s_2 & s_4 & s_6 & s_8 \\ s_{11} & s_2 & s_4 & s_6 & s_8 & s_{10} \end{bmatrix},$$

and

$$S_6^{'[9]} = \begin{bmatrix} s_1 & s_{10} & s_8 & s_6 & s_4 & s_2 \\ s_{10} & s_8 & s_6 & s_4 & s_2 & s_{11} \\ s_8 & s_6 & s_4 & s_2 & s_{11} & s_9 \\ s_6 & s_4 & s_2 & s_{11} & s_9 & s_7 \\ s_4 & s_2 & s_{11} & s_9 & s_7 & s_5 \\ s_2 & s_{11} & s_9 & s_7 & s_5 & s_3 \end{bmatrix}.$$

As we proved in Section 5.5, both of these matrices, and any of those 10 matrices on the whole, result in the exact location of errors based on (5.45). In practice, due to quantization, we have $\tilde{S}_m^{'[i]} = S_m^{'[i]} + Q_m^{'[i]}$. Hence,

$$\begin{aligned} \tilde{R}_m^{'[i]} &= \tilde{S}_m^{'[i]} \tilde{S}_m^{'[i]H} \\ &= R_m^{'[i]} + \underbrace{S_m^{'[i]} Q_m^{'[i]H} + Q_m^{'[i]} S_m^{'[i]H} + Q_m^{'[i]} Q_m^{'[i]H}}_{T_m^{'[i]}(Q_m^{'[i]})} \end{aligned} \tag{5.53}$$

From Proposition (5.3) we know that $R_m^{'[i]}$, for any $i = 1, \ldots, 10$, results in the exact location of errors. However, for quantized DFT codes, $\tilde{R}_m^{'[i]}$ is used for decoding and it yields slightly different results for each i, due to different combinations of the quantization error at the decoding process which is represented by $T_m^{'[i]}$. This brings in some sort of decoding *diversity*, and we exploit this diversity to improve the decoding accuracy, as explained in Sections 5.5 and 5.4.

5.7 Simulation Results

To evaluate the performance of the proposed algorithms we perform simulations using a Gauss-Markov source X, with a mean zero, a variance one, and a correlation coefficient $\rho = 0.9$. The Gauss-Markov process $\{X_i\}$ is generated based on the following recursion

$$X_i = \sqrt{1-\rho^2} Z_i + \rho X_{i-1}, \tag{5.54}$$

in which $\{Z_i\}$ is a zero-mean i.i.d. Gaussian process with variance 1, and $0 \leq \rho < 1$ is the correlation coefficient [123]. Given an (n, k, t) code, the encoder divides X into blocks $\boldsymbol{x}$ of length n and generates the syndrome $\boldsymbol{s}_x$ and extended syndrome $\bar{\boldsymbol{s}}_x$; these are then quantized by a 3-bit uniform quantizer with step size $\Delta = 0.25$. At the decoder, we have the quantized syndromes and the side information $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e}$ is the error vector. The number of errors in each block is ν, $\nu \leq t$, where t is the error correction capacity of the code. Similar to [86], we assume that the error components are fixed. We plot the relative frequency of correct localization of errors[11] versus CEQNR, which is the ratio of channel error power to the quantization noise power, as defined in (3.28). The simulation results are for 10^4 input blocks for each CEQNR.

In Figure 5.1, we compare the frequency of correct localization of errors for the subspace and extended subspace approaches given a $(10, 5)$ code for different errors. The gain due to the extended subspace method is remarkable both for one and two errors; it is more significant for two errors. In fact, as discussed in Section 5.2.2, for $\nu = t$ the subspace approach loses its degrees of freedom (DoF) and its performance drops to that of the coding-theoretic approach. Providing some extra DoF, at the expense of a higher code rate, the extended subspace approach significantly improves the error localization. The gain caused by the extended subspace method increases for codes with higher capacity. For instance, simulation results for a $(17, 9)$ DFT code, presented in Figure 5.2, show a significant gain in any CEQNR between 10 to 40 dB; this is achieved by sending 5 additional

[11] The probability of error localization is defined as relative frequency of correct localization of all ν errors in each block (codeword) to the number of blocks. With this definition, correct error localization will guarantee a decent error correction also. In [86], this parameter is defined as the total number of correctly identified locations to the total number of errors. If such a criterion is used, all curves corresponding to $\nu > 1$ will shift up and we will get better probability of error localization. Fore one thing, simulation results for this latter case can be found in Figure 5.9.

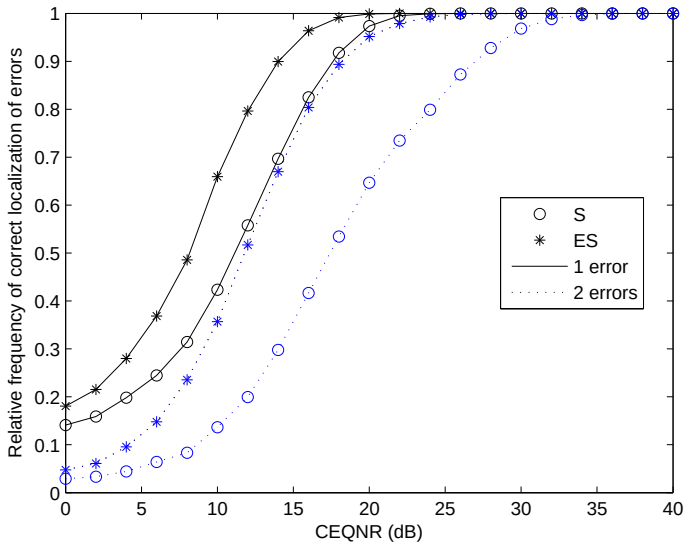


Figure 5.1 Probability of error localization in the subspace (S) and extended subspace (ES) approaches at different CEQNRs for a $(10, 5)$ DFT code. The curves for the extended case are based on 2 additional syndrome samples, implying that the code rate is increased from 0.5 to 0.7.

syndrome samples.

Next, we evaluate the performance of the *generalized subspace* method. To begin with, in Figure 5.3 we show the merit of the generalized subspace error localization with respect to the subspace method. An important question is whether the generalized subspace decoding performs better than the extended subspace method. To answer this question, we compare the two algorithms for different codes and various numbers of extended syndrome. In Figure 5.4, we compare the frequency of correct localization of errors for the extended and generalized subspace approaches for a $(11, 5)$ code with different number of errors; both methods use 3 syndrome samples more than the subspace method. The same simulation is done for 4 extra samples and the results are presented in Figure 5.5. It is seen that, for 3 errors, the extended subspace approach produces better results than the generalized subspace method in the first case while the performance gain swaps in the second case. In general, extensive simulations show that the performance of the extended subspace

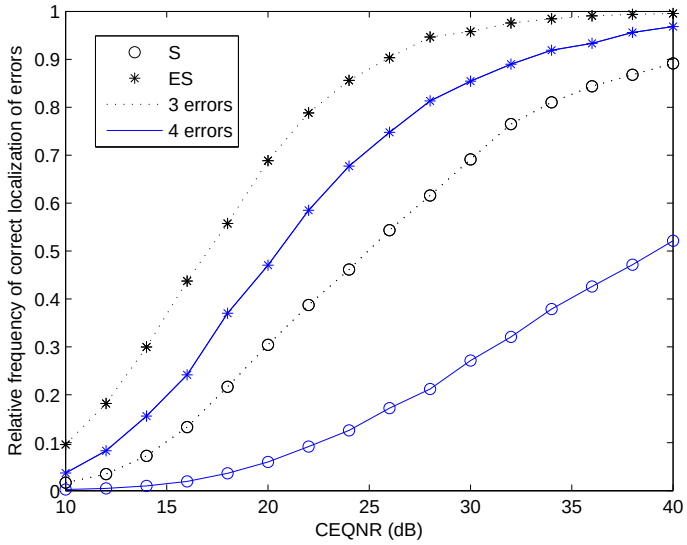


Figure 5.2 Probability of error localization in the subspace and extended subspace methods at different CEQNRs for a $(17, 9)$ DFT code. The curves for the extended case are based on 5 additional syndrome samples.

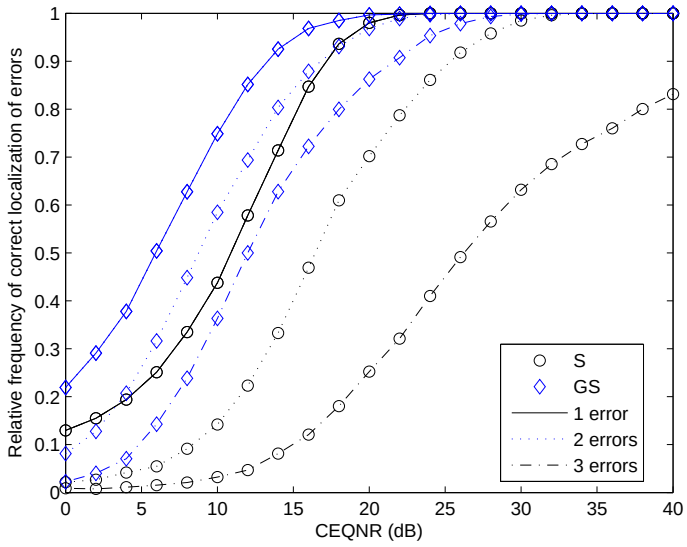


Figure 5.3 Probability of correct localization of 1 to 3 errors for a $(11, 5)$ DFT code using the subspace (S) and generalized subspace (GS) methods with 5 additional syndrome samples.

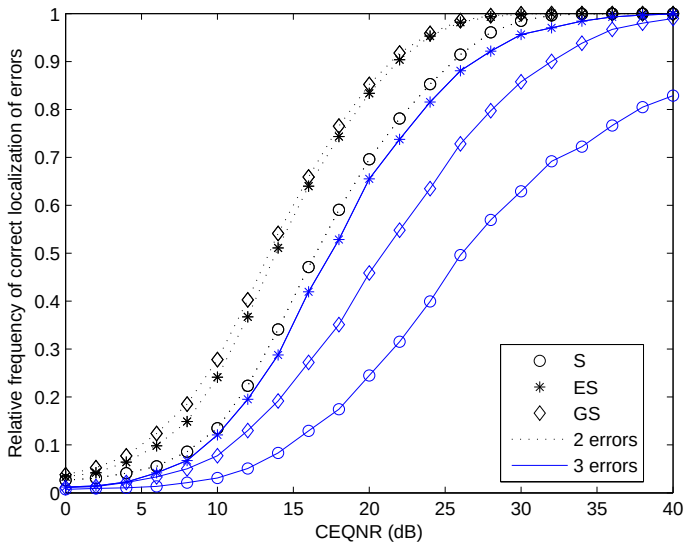


Figure 5.4 Probability of correct localization of 2 and 3 errors for a $(11, 5)$ DFT code using the subspace (S), extended subspace (ES), and generalized subspace (GS) methods with 3 additional syndrome samples.

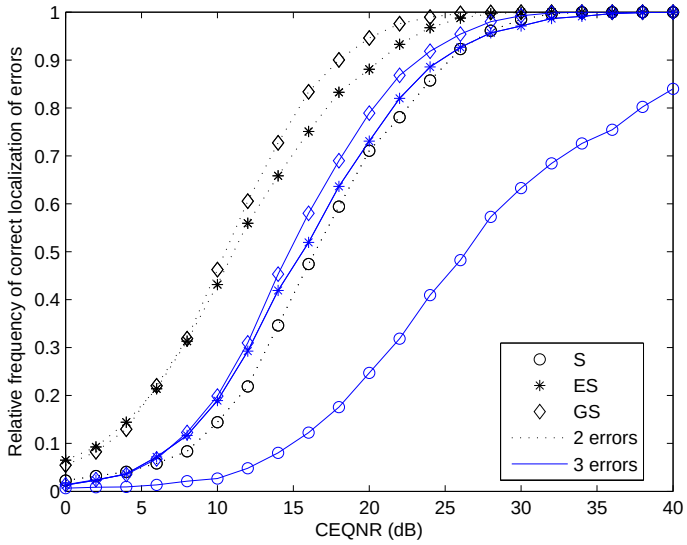


Figure 5.5 Probability of correct localization of 2 and 3 errors for a $(11, 5)$ DFT code using the subspace, extended subspace, and generalized subspace methods with 4 additional syndrome samples.

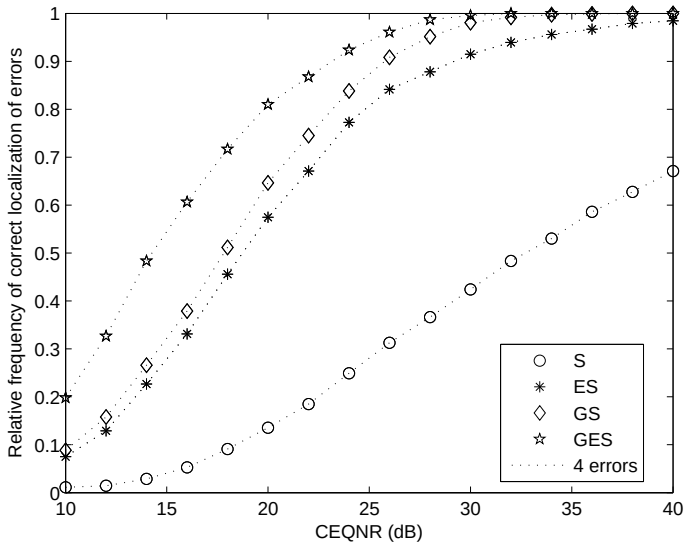


Figure 5.6 Probability of correct localization of 4 errors for a $(13, 5)$ DFT code using the subspace, extended subspace, generalized subspace, and generalized-extended subspace (GES) methods with 4 additional syndrome samples.

error localization is better than the generalized subspace if just a few extra syndrome samples are available. By increasing the extra samples the gain caused by the generalized subspace method increases sharply such that it can catch up with and even outperform the extended subspace method. In the extreme case, when there are k extra samples, generalized subspace method outperforms its opponent distinctly. The performance of the generalized subspace localization obviously depends on the number of matrices (or equivalently, polynomials); therefore, its performance gain is noticeable when n is prime number.

Finally, let us evaluate the performance of the generalized-extended approach for a couple of codes. We consider a $(13, 5)$ which is capable of correcting up to 4 errors. Since $n = 13$ is a prime number, we expect that the generalized and generalized-extended perform better than the other approaches. This is true as shown in Figure 5.6; the gain from generalization is considerably high especially when $\nu \to t$. A similar pattern is seen for other codes, e.g., $(11, 5)$, $(14, 5)$, and $(17, 9)$ codes, to name a few.

In summary, we conclude that no one approach is superior in every situation. The

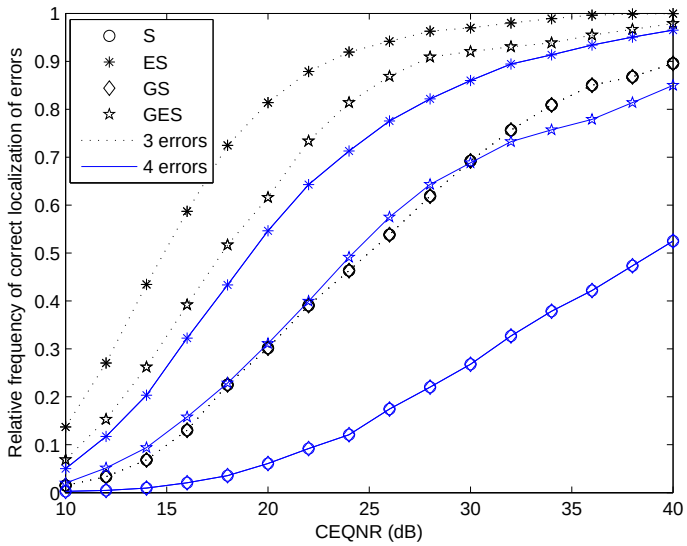


Figure 5.7 Probability of correct localization of 3 and 4 errors for a $(17, 9)$ DFT code using different subspace-based error localizations with 4 additional syndrome samples.

performances of the proposed algorithms vary with $\bar{d}$, ν, t, and CEQNR. However, based on extensive simulation, some of which presented in the chapter, the following patterns are observed.

- The generalized and generalized-extended approaches markedly outperform the other ones in different CEQNRs as $\bar{d} \to n$, specially when n is prime (See Figure 5.6, for example). This is because $\mathcal{D}$ and thus $\mathcal{G}$ are very large, which brings a huge gain.

- When $\bar{d} \to d$, the generalized approach loses its gain as $|\mathcal{I}_m| = 1$. In such cases, the extended subspace is the best. The rationale behind the generalized approach is to use multiple syndrome matrices with smaller sizes compared with the extended approaches that use up all syndrome samples in a large syndrome matrix. However, it should be noted that when there are few new syndromes ($\bar{d} \to d$), it is not possible to build up more than one syndrome matrix. In such cases, the generalized approach fails to use extra syndromes while the extended approach uses them up and thus outperforms its opponent. This is examined for a $(17, 9)$ code in Figure 5.7

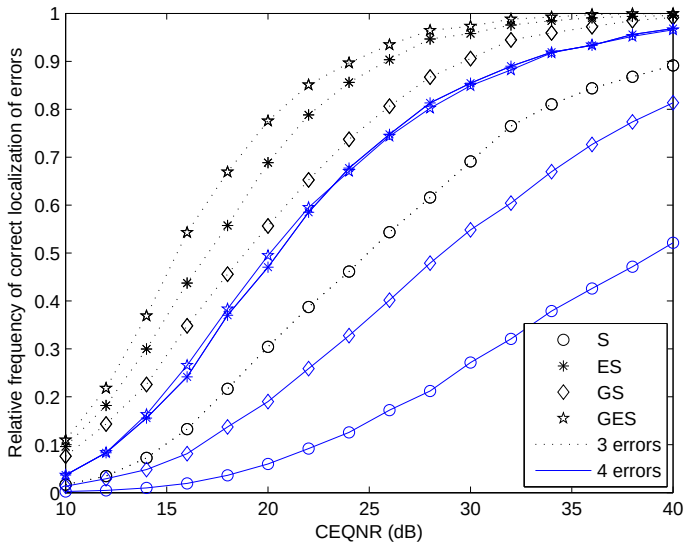


Figure 5.8 Probability of correct localization of 3 and 4 errors for a $(17, 9)$ DFT code using different subspace-based error localizations with 5 additional syndrome samples.

and Figure 5.8. As it can be seen, with 4 extra syndrome samples (Figure 5.7) the performance of generalized subspace is the same as subspace approach. This is because with that many syndromes we cannot form any generalized syndrome matrix for $i > 1$, therefore there is no gain over the subspace method; in such a case the extended subspace gives the best results, as explained above. But, in Figure 5.8, when only one more sample is added the gain from generalization comes in and the generalize-extended method outperforms the extended one.

- The generalized-extended method seems to be the best choice when there are enough extra syndrome samples to build several or more syndrome matrices. This is because it averages as many polynomials as possible to improve the results. As the CEQNR goes up, its dominance is less because the other approaches work well, also.

Finally, as we mentioned at the beginning of this section, if we define probability of correct localization as the total number of correctly identified locations to the total number of errors all curves will shift up and we will get better probability of error localization in any error localization algorithm. To visualize the difference between the output of the two definitions, in Figure 5.9 we plot the probability of error localization based on the new

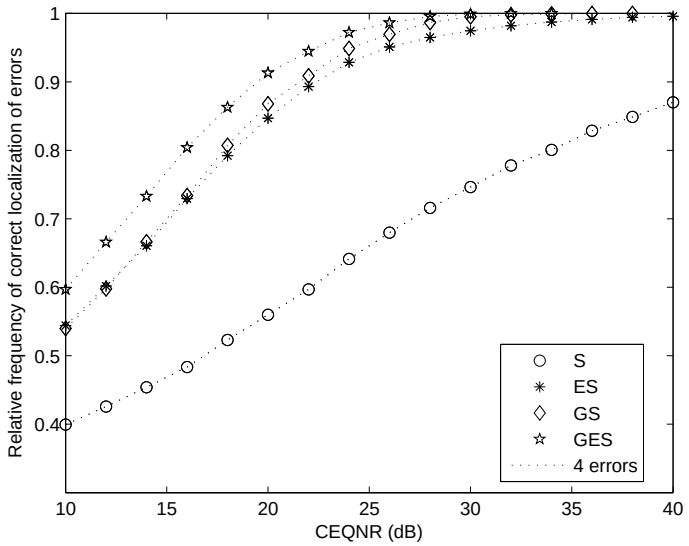


Figure 5.9 Probability of correct localization of 4 errors for a $(13, 5)$ DFT code using for methods with 4 additional syndrome samples. Note that the definition of correct error localization is based on the total number of errors. Compare this figure with Figure 5.6.

definition, for a $(13, 5)$ DFT code. Note that in Figure 5.6 and Figure 5.9 the simulations are exactly for the same setting except that in Figure 5.9 we calculate the probability of correct localization based on the total number of correctly identified locations to the total number of errors. In other words, Figure 5.6 shows correct block localization while Figure 5.9 shows correct sample localization. It can be seen that, for example at CEQNR=10dB, the difference between the probability of error localization in these figures is huge.

It is also worth mentioning that numerical results proves the superiority of using $\tilde{R}'_m$, instead of $\tilde{R}_m$ for finding the number of errors. Finally, knowing that a better error localization implies a lower MSE between the sources [117], rate-adapted DFT codes with extended subspace decoding can be used both to adapt the channel variations and decrease the MSE in DSC.

5.8 Summary

In this chapter, we have developed three subspace-based algorithms that substantially improve the existing subspace error localization of quantized DFT codes, in favor of extra syndrome samples. The first approach, named extended subspace, simply extends and improves the existing subspace-based algorithms by increasing the number of vectors in the quantization noise subspace, or equivalently, the number of polynomials obtainable for error localization. This is followed by another method that generalizes the decoding, and also the encoding, of the DFT codes. We proved that many syndrome matrices, each of which uses d syndromes, can be utilized to decode DFT codes. This diversity is exploited for increasing the decoding accuracy since by averaging the corresponding error locating polynomials the effect of the quantization error diminishes. The third approach, i.e., the generalized-extended subspace decoding, combines the aforementioned algorithms to further increase the decoding gain. The extended decoding can be applied to DCT and DST codes, whereas the extension of the generalized subspace decoding to these classes of codes is not straightforward. The proposed algorithms are primarily useful for rate-adaptation in a DSC system that uses DFT codes for binning; they can also be used in the channel coding. Rate-adaptation is realized by augmenting the syndrome samples and does not need to re-encode the sources.

Chapter 6

Conclusions and Future Work

In this thesis, we have laid the groundwork for several promising directions in source and channel coding based on a class of real/complex Bose-Chaudhuri-Hocquenghem (BCH) codes, called discrete Fourier transform (DFT) codes, and signal processing tools. Here, we summarize these contributions and discuss some open problems and research directions that can be pursued in the future.

6.1 Thesis Summary

In Chapter 3, we have established a new framework for lossy distributed source coding, in which real-number codes, rather than binary codes, are used to compress statistically dependent distributed sources. In this new framework, compression is done in the real field and is followed by quantization; that is, we quantize compressed sources instead of compressing quantized sources. This change in the order of binning and quantization blocks makes it possible to model the correlation between continuous-valued sources more realistically and also to partially compensate for the quantization error. The encoding and decoding procedures are described in detail for a class of real/complex BCH codes called DFT codes. We leverage subspace-based techniques to improve the decoding performance of quantized DFT codes. Moreover, by extending the subspace-based decoding, we are able to perform distributed source coding in a rate-adaptive fashion to further improve the decoding performance when there is uncertainty in the statistical dependence. The mean-squared reconstruction error is shown to be reasonably low, even for very short code length [111, 117].

While Chapter 3 focuses on syndrome-based approach, in Chapter 4 we study parity-based distributed source coding and we extend it to the case where the transmission channel is noisy and thus we introduce distributed joint source-channel coding based on DFT codes. The proposed system is well suited for low-delay communications, as it directly maps short source blocks to channel blocks. Furthermore, we introduce the notion of *systematic DFT codes* (frames), in this chapter. We show that for an (n, k) DFT code there are $\binom{n}{k}$ systematic codes, and we find the conditions for which a systematic frame is *tight* and thus minimizes reconstruction error. Furthermore, in Theorem 4.7 we find the "best" systematic DFT frames, from the minimum mean-squared reconstruction error sense.

Chapter 5 is devoted to developing algorithms that improve the encoding and decoding of quantized DFT codes with applications in rate-adaptive coding. We first have a fresh look on how the subspace-based error localization outperforms the coding-theoretic one. With this insight, we propose an extension of the subspace-based error localization, based on additional syndrome, that improves the existing one and is naturally suitable for rate-adaptive distributed source coding. The algorithm is incremental so that there is no need to re-encode the sources when more syndrome samples are requested. Next, as a generalization of this work, we propose a new subspace-based algorithm to decode BCH-DFT codes. The proposed approach introduces many, rather than one, decoding matrices for a BCH-DFT code; this diversity is then used to diminish the effect of the quantization noise and improve the decoding. To be specific, we prove that for an (n, k) DFT code there are $\phi(n)$ syndrome matrices for decoding, where $\phi(n)$ is the number of positive integers less than n that are relatively prime to n. Apart from the diversity in decoding, the generalized subspace method brings another novelty at the encoder side; that is, the parity frequencies of BCH-DFT codes (or, the zeros of codewords in the frequency domain) are not required to be cyclically adjacent. This provides substantial flexibility in constructing BCH-DFT codes, as one can select the parity frequencies in many different ways. Finally, in addition to improving the error localization, both extended and generalized subspace algorithms can be applied to increase the accuracy of the error detection step. Simulation results demonstrate the capability of the proposed algorithms to perform significantly better than the existing subspace-based error localization, in the presence of quantization noise.

Another contribution of this chapter is the construction of rate-adaptive DFT codes that uses syndrome augmentation to increase the error correction capacity of the code and to create rate flexibility. Rate adaptation is an existing idea that the encoder switches

among coding rates. This capability, combined with a feedback channel from the decoder, means that the encoder need not know in advance the degree of statistical dependency between the sources.

6.2 Future Directions

There are several avenues for future work. Many of them revolve around improving the decoding algorithm for DFT codes or extending the developed algorithms to other real codes such as the discrete cosine transform (DCT) and discrete sine transform (DST) codes, or to other fields like source localization. Further, the models contributed in this thesis can be leveraged for signal compression through filter banks, as well as for compressive sensing and signal processing.

6.2.1 Improving Rate-Distortion Performance

Improving the rate-distortion performance of the proposed distributed source coding is a valuable contribution. To this end, a more accurate algorithm for error localization is a key, as we showed in Figure 3.9. One promising direction is to explore integrating subspace-based decoding into more powerful *greedy* and *iterative* recovery algorithms. To begin with, one can use recursively applied and projected MUSIC to decode DFT codes. These methods can be applied to the basic and generalized subspace-based error localization algorithms. The decoding diversity we introduced in this thesis was used in the error localization step; it can be useful in the other steps of decoding a DFT code; i.e., when finding the number of errors and their amplitudes. In addition to improving the rate-distortion performance, exploring the rate-distortion function for finite block codes is very important.

6.2.2 Generalized Decoding for DCT and DST Codes

The proposed algorithms are primarily useful for rate-adaptation in a DSC system that uses DFT codes for binning; one can use DCT and DST codes for this purpose. DCT codes can be specially useful when sources have temporal correlation; for example, in many distributed video coding systems a DCT block is used exploit the temporal correlation before binning. By using DCT codes one can combine these two steps and look for the best possible performance with constrained computational resources. The extended decoding

can be applied to DCT and DST codes, whereas the extension of the generalized subspace decoding to these classes of codes is not straightforward. One possible direction to pursue is to investigate applying this generalized algorithm to DCT and DST codes. Furthermore, constructing any other set of spanning basis V_m, for instance (5.29), such that the entries of $V_m D V_{d-m+1}^T$ are syndrome samples, with any permutation, is valuable as it can be used, in conjunction with the other spanning matrices, for diminishing the effect of quantization error, in DFT, DCT, and DST codes.

6.2.3 Lossy DSC Using Oversampled Filter Banks

A more general way of inserting "structured," soft redundancy in signals is possible through the use of oversampled filter banks [68]. It would be interesting to extend our work to oversampled DFT filter banks, an infinite-dimension of DFT frames, since oversampled filter banks can be used for error correction. A filter bank, in general, comprises n channels each with a filter of length l, in which each channel is subsampled by a factor $k \leq n$. For $k = n$ a *critically sampled* filter bank is achieved; it implements a nonredundant (basis) expansion of the input signal. An *oversampled* filter banks (OFB), as opposed to a critically samples filter bank, is realized when $k < n$; it implements a redundant (frame) expansion of the input signal. As a result, subband signals in an OFB contain more than enough information for reconstruction of the original signal. This inherent redundancy can be exploited to correct signal transmission errors. In view of the fact that an (n, k) DFT code is a special case of an OFB in which the number of channels n is equal to the length of filters, i.e., $l = n > k$, the extension of our lossy DSC system to lossy DSC by means of OFBs is straightforward. Also, this helps integrate error correction in filter banks, as an important tool for image and signal processing. In a relevant work, systematic wavelet subcodes, a form of real-number convolutional codes, has been proposed for data protection [89].

6.2.4 Parametric Frequency Estimation

In a somewhat different direction, observing that the error-localization of DFT codes can be viewed as a problem of estimation of sinusoids in background noise, one can apply the generalized subspace-based algorithm to the other related literature such as source localization. Parametric estimation of sinusoids in noise is a mature subject in the area of array signal processing [67]. Among various parameter estimation techniques, MUSIC and

ESPRIT are the most widely studied techniques for direction-of-arrival (DOA) estimation. Despite similarity of the DOA estimation and error localization in DFT codes, in the DOA estimation there are many sets of observations available over some given time interval, whereas in the error-correction problem there is only one set of syndrome coefficients for each received vector. Hence, the proposed generalized subspace error-localization of DFT codes needs to be adapted to the DOA estimation problem.

6.2.5 Spectral Compressive Sensing

Compressive sensing (CS) is a promising approach to simultaneous sensing and compression by having sparse or compressible representation of signals in some basis. CS algorithms seek the sparsest signal in some discrete basis or frame that agrees with the measurements. A lot of applications feature frequency-sparse signals that can be modeled as a sum of a small number of sinusoids; the DFT basis is a natural choice for CS recovery of such signals. Line spectral estimation methods (e.g., MUSIC) have already been applied to improve the state-of-the-art CS algorithms. Seeing that there is a close connection between spectral CS and error correction in DFT codes, it seems plausible to leverage recursive MUSIC and diversity in decoding to further improve the performance of spectral CS recovery. Also, it might be possible to combine distributed source coding and compressive sensing to introduce distributed compressive sensing in the context of DFT codes.

Appendix A

BCH-DFT Codes

A.1 Introduction

The problem of error correction in the real field using *real-number* codes was first considered by Marshall [72] and Wolf [124]; they proposed the discrete Fourier transform (DFT) for this purpose. Marshall also introduced an important subclass of DFT codes, the Bose-Chaudhuri-Hocquenghem (BCH) [58, 17] DFT codes. The ideas of *coding theory* can be described within the *signal processing* realm by virtue of this class of complex or real BCH codes [72], [14, Chapter 6]. Apart from being used for error and erasure correction in the real field [85, 76, 100, 84, 16], DFT codes find applications a wide range of areas including wireless communications [121, 57, 56], image transforming [40], joint source-channel coding [41], distributed source coding [111, 117], distributed joint source-channel coding [109], and compressive sensing [33, 3].

Compared with the finite-field channel codes, DFT codes are preferred for their unconstrained codeword length, fast and easy implementation with floating-point operations, and ability in alleviating the quantization error besides correcting the errors (erasures) introduced by the transmission channel. On the other hand, there is a fundamental difference between the syndrome decoding of binary BCH codes and that of BCH-DFT codes. To be transmitted over a digital communication channel, the codewords of any BCH-DFT code have to be quantized; this adds noise to each sample of every codeword, in channel coding. Similarly, in distributed source coding based on BCH-DFT codes [111], the parity or syndrome samples have to be quantized before transmission. In any of those cases, the syndrome formed at the decoder is affected by quantization noise, and thus the *syndrome*

decoding of errors is feasible only if the *quantization error* is very small compared with the impulse noise.

Similar to error correcting codes in finite fields, error correcting codes in the real field insert *redundancy* into a message vector consisting of k samples to convert it to a codevector of n samples ($n > k$). But this redundancy is inserted in the *analog* domain, i.e., before quantization and entropy coding. In such settings, we are dealing with *soft redundancy* rather than *hard redundancy* in the binary field error correcting codes. By using structured, soft redundancy, one can go beyond quantization error [113], and thus reconstruct continuous-valued signals more accurately. In general, error correction using real-number codes benefits from the advantage that the insertion of redundancy at the transmitter is done before quantization, making it possible to alleviate part of quantization error at the receiver side. The challenge is that quantization error is not distinguishable from channel error. This means that, from the perspective of the decoder, there are always n errors in a quantized codevector of length n, regardless of the number of channel errors. In the absence of quantization noise it is straightforward to detect, localize, and correct the errors introduced by the channel, by means of algebraic real codes [14]. However, the problem becomes more challenging considering quantization error. This problem has been investigated in [86] and [100].

The objective of this appendix is to familiarize the reader with the concept of real-number codes, and their convenience in lossy data transmission. More precisely, we focus on real BCH-DFT codes and show how they are used to encode and decode data in *digital* communication systems. We introduce and improve the Peterson-Gorenstein-Zierler (PGZ) decoding algorithm to reconstruct the input signal with small MSE, for quantized DFT codes. The improvement is based on the observation that in the PGZ algorithm we encounter *overdetermined* systems in different steps of error decoding, i.e., detection, localization, and estimation of the errors. While neglecting quantization error theses overdetermined systems are consistent, they are not so when quantization is introduced. In the latter case, there is no exact solution. Instead, we look for the solution with the smallest 2-norm error vector, that is the least squares (LS) solution. Thanks to the LS estimation, DFT codes can result in a MSE lower than quantization error, even when several errors occur during transmission. This is one of the main advantages of real-number error correcting codes over binary codes that motivates further study in this field. It paves the way for introduction of DFT codes in the context of lossy distributed source coding [111, 117].

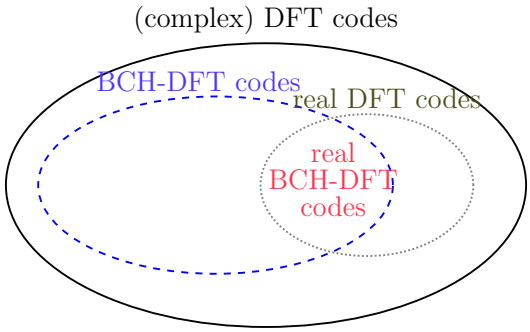


Figure A.1 The class of DFT codes.

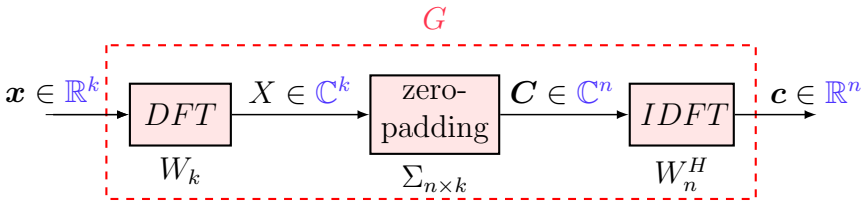


Figure A.2 The typical real BCH-DFT encoding scheme.

A.2 Real BCH-DFT Codes

Discrete Fourier transform (DFT) codes are linear block codes over the complex field. A DFT code is defined by its generator and parity-check matrices, which are based on a IDFT matrix. The parity-check matrix of an (n, k) DFT code is comprised of any $n-k$ columns from the IDFT matrix of size n (W_n^H); the remaining k columns of W_n^H can be used to construct the corresponding generator matrix; thus, $HG = 0$. This code has a blocklength n and designed minimum distance $d+1$ where $d \triangleq n-k$. DFT codes are cyclic codes in complex and real fields and their codewords satisfy certain spectral properties in the frequency domain [72], [13, Chapter 4]. More specifically, the spectrum of each codeword of a BCH code must be zero in a block of d cyclicly adjacent components [14]. Within the class of DFT codes, there are BCH codes in the *complex* and *real* fields. Figure A.1 shows the family of DFT codes. A real DFT code has a generator matrix with real entries, making it suitable for encoding of real data. A real BCH-DFT code has these properties

altogether. Figure A.2 represents the typical encoding scheme for an (n, k) real BCH-DFT code [1]. The generator matrix of this code is given by

$$G = \sqrt{\frac{n}{k}} W_n^H \Sigma W_k, \tag{A.1}$$

in which W_k and W_n^H respectively are the DFT and IDFT matrices of size k and n, and Σ is an $n \times k$ matrix [86, 40].

The code generated by (A.1), as illustrated in Fig. A.2, is a real BCH code provided that Σ inserts $n - k$ successive zeros in X, while keeping the conjugacy constraint [13, Chapter 4], [74]. Real BCH-DFT codes exist for any $n > k \geq 1$ except when both n, k are even [72]. Particularly, for odd k, Σ has exactly k nonzero elements given as $\Sigma_{00} = 1$, $\Sigma_{i,i} = \Sigma_{n-i,k-i} = 1$, $i = 1 : \frac{k-1}{2}$ [86], [100]. The parity-check matrix H is then comprised of the columns of the IDFT matrix W_n^H corresponding to those $n - k$ zeros. Because of the unitary property of the IDFT matrix, $HG = \mathbf{0}$. Throughout this appendix, an (n, k) DFT code refers to a code generated by (A.1) using the zero-padding matrix Σ as specified above; thus, it is a BCH code in the real field. Also, k is assumed to be an odd number while n can be any integer greater than k. Furthermore, we assume $\nu \leq t$ is the number of errors where $t = \lfloor \frac{n-k}{2} \rfloor$ represents the maximum number of errors that can be corrected by the employed DFT code. In the rest of this appendix, we show how real BCH-DFT codes can be decoded, without and with quantization.

A.3 Decoding Algorithm for BCH-DFT Codes

Although any general decoding technique for cyclic codes can be used to decode BCH codes, there are much better algorithms that have been developed specifically for decoding BCH codes. We use the Peterson-Gorenstein-Zierler (PGZ) decoding algorithm; the Peterson algorithm finds the locations of the errors, and the Gorenstein-Zierler algorithm finds their magnitudes. More precisely, we explain the extension of the binary PGZ algorithm to the real field [13], [14], in two different setting: 1) When quantization does not exist 2) When quantization come into play. These are elaborated in following sections.

[1] In general, the DFT matrix can be replaced by an IDFT matrix of the same size. Likewise, the IDFT matrix can be replaced by a DFT matrix of the same size. Thus, four different combinations are plausible; in each case, the resulting code is a real BCH-DFT but possibly with a differen G.

A.3.1 Neglecting Quantization

Consider an (n, k) DFT code with parity-check matrix H and generating matrix G as described in Section A.2. Let the $n \times 1$ vector $\boldsymbol{c}$ denote the transmitted codevector over some noisy channel. The received vector is corrupted version of $\boldsymbol{c}$ by noise vector $\boldsymbol{e}$. The syndrome samples of the received vector $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ can be expressed as

$$\boldsymbol{s} = H\boldsymbol{r} = H(\boldsymbol{c} + \boldsymbol{e}) = H\boldsymbol{e}, \tag{A.2}$$

and $\boldsymbol{s}$ is a complex vector of length $n-k$. From (A.2), it is evident that the syndrome does not depend on the transmitted codevector $\boldsymbol{c}$; it only depends on the error pattern $\boldsymbol{e}$. Thus, $\boldsymbol{s} \neq \boldsymbol{0}$ indicates that one or more errors have occurred, and we need to correct them. Note that the syndrome is a sufficient statistics for determining the error pattern, because any two vectors yield the same syndrome if and only if they belong to the same coset of code. Suppose there are $\nu \leq t$ errors in locations $i_1, i_2, \ldots, i_\nu$ with corresponding magnitudes $e_{i_1}, e_{i_2}, \ldots, e_{i_\nu}$. Then, the error polynomial is

$$e(x) = e_{i_1}x^{i_1} + e_{i_2}x^{i_2} + \ldots + e_{i_\nu}x^{i_\nu}. \tag{A.3}$$

For DFT codes[2], the partial syndromes[3] are *complex* values $s_i = e(\alpha^i)$ where $\alpha = e^{-j\frac{2\pi}{n}}$ and $1 \leq i \leq 2t$; these are calculated in (A.2). Using the change of variables $X_1 = \alpha^{i_1}, \ldots, X_\nu = \alpha^{i_\nu}$ for *error locators* and renaming *error magnitudes* $E_1 = e_{i_1}, \ldots, E_\nu = e_{i_\nu}$, from (A.2) we get

$$\begin{aligned} s_1 &= E_1X_1 + \ldots + E_\nu X_\nu, \\ s_2 &= E_1X_1^2 + \ldots + E_\nu X_\nu^2, \\ &\vdots \\ s_{2t} &= E_1X_1^{2t} + \ldots + E_\nu X_\nu^{2t}. \end{aligned} \tag{A.4}$$

That is, the partial syndromes constitute a system of $2t$ equations with 2ν unknowns, i.e.,

[2] In general, for a BCH code over $GF(q)$, partial syndromes are values in *decoder* alphabet ($GF(q^m)$), and α is an element of $GF(q^m)$. Each row of H is a row of finite field Fourier transform matrix of size n. For a t-error correcting, primitive narrow-sense BCH code, the rows of H are the first $2t$ powers of consecutive powers of α [14].

[3] "Partial syndromes" and "syndrome samples" are used interchangeably, in this appendix.

error locators (X_p) and error magnitudes (E_p), where $1 \leq p \leq \nu$. The goal is to reduce this system of equations to a one-variable polynomial with ν solutions. The (syndrome) decoding algorithm of DFT codes, and BCH codes in general, has the following major steps:

- *Detection* (to find the number of errors)
- *Localization* (to determine the location of errors)
- *Calculation* (to calculate the magnitude of errors)

These steps are elaborated in the following.

Error Detection

Let $\nu \leq t$ be the number of errors where $t = \lfloor \frac{n-k}{2} \rfloor$ is the maximum number of errors that can be corrected by the DFT code; ν can be found by forming the syndrome matrix

$$M_t = \begin{bmatrix} s_1 & s_2 & \dots & s_t \\ s_2 & s_3 & \dots & s_{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_t & s_{t+1} & \dots & s_{2t-1} \end{bmatrix}, \tag{A.5}$$

and evaluating its rank. The entries of this matrix are the partial syndromes, picked from the syndrome vector $\boldsymbol{s} = [s_1, s_2, \dots, s_{2t}]^T$, which is calculated by (A.2). Now, if M_μ is nonsingular for $\mu = \nu < t$ but it is singular for $\mu = \nu + 1$, then ν indicates the number of channel errors [14, Chapter 6]. Equivalently, the rank of M_t gives the number of errors ν; this is formally proved in Chapter 5.

Error Localization

The error locator polynomial $\Lambda(x)$ for a BCH code is defined as a polynomial whose roots are the reciprocals of error locators, which are of interest. That is,

$$\Lambda(x) = \prod_{i=1}^{\nu} (1 - xX_i^{-1}) = \Lambda_0 + \Lambda_1 x + \ldots + \Lambda_\nu x^\nu. \tag{A.6}$$

The decoder requires to find the roots of $\Lambda(x)$, i.e., $X_1, \dots, X_\nu$, which are the error locators. Meanwhile, the decoder needs to find the degree of $\Lambda(x)$ as well. In fact, the PGZ algorithm

first determines ν, the degree of $\Lambda(x)$ or equivalently the number of errors, as explained earlier under "error detection."
The coefficients of $\Lambda(x)$ can be found by solving the following set of equations [14]

$$s_j\Lambda_\nu + s_{j+1}\Lambda_{\nu-1} + \cdots + s_{j+\nu-1}\Lambda_1 = -s_{j+\nu}, \tag{A.7}$$

for $j = 1, \ldots, 2t-\nu$, $\nu \leq t$. This set of equations are *consistent* and can be written in the following matrix equation

$$\underbrace{\begin{bmatrix} s_1 & s_2 & \ldots & s_\nu \\ s_2 & s_3 & \ldots & s_{\nu+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_\nu & s_{\nu+1} & \ldots & s_{2\nu-1} \end{bmatrix}}_{\mathbf{M}_\nu} \begin{bmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = - \begin{bmatrix} s_{\nu+1} \\ s_{\nu+2} \\ \vdots \\ s_{2\nu} \end{bmatrix}. \tag{A.8}$$

Thus, the coefficients of $\Lambda(x)$ are obtained by solving (A.8) for $\Lambda_1, \ldots, \Lambda_\nu$. Finally, to find the error locations, we evaluate $\Lambda(\alpha^i)$ for $i = 1, 2, \ldots, n$, where $\alpha = e^{-j\frac{2\pi}{n}}$. Let $i_1, i_2, \ldots, i_\nu$ be those indices for which $\Lambda(\alpha^i) = 0$. Thus, the location of errors is known and the error polynomial can be defined once the magnitude of errors is determined.

Error Calculation

The last step is to find the magnitude of errors. This step is rather simple. Let H_e denote the matrix consisting of the columns of H corresponding to error indices, then the errors magnitude $\boldsymbol{E} = [E_1,\ E_2, \ldots, E_\nu]^T = [e_{i_1},\ e_{i_2}, \ldots, e_{i_\nu}]^T$ can be determined by solving

$$H_e\boldsymbol{E} = \boldsymbol{s}, \tag{A.9}$$

which gives a set of $n-k$ consistent equations with ν errors. One can choose any ν arbitrary equations and solve them for $\boldsymbol{E}$. Hence, without loss of generality, we can write

$$H_{e,\nu}\boldsymbol{E} = \boldsymbol{s}_\nu, \tag{A.10}$$

where $\boldsymbol{s}_\nu = [s_1, s_2, \ldots, s_\nu]^T$ contains the first ν syndrome samples, and $H_{e,\nu}$ includes those rows of H_e corresponding to $\boldsymbol{s}_\nu$. This completes the error correction algorithm by determining the error vector.

Consider that, with this algorithm, we obtain the exact value of channel errors as long as the number of errors is not greater than the error correction capability of the code. Admittedly, we cannot expect such an exact result in the presence of quantization error, as quantization error affects each and every sample of the codewords. Since quantization error is random, the decoding becomes an estimation problem. This issue is dealt with in the following sections.

A.3.2 Quantized BCH-DFT Codes

The transmission of continuous-valued signals in digital communication systems is subject to quantization; therefore, it is necessary to modify the decoding algorithm stated in Section A.3.1 to take into account the error introduced by quantization. This problem has been considered in [86] and [100]. The former focuses on error localization, assuming the number of channel errors is known. The latter, on the other hand, focuses on determining the number of errors with the assumption that amplitude of quantization noise is lower than that of channel errors. We briefly explain the whole decoding process in what follows.

Let $\hat{\boldsymbol{c}}$ be the quantized version of the codevector $\boldsymbol{c}$, and $\boldsymbol{q}$ denote the associated quantization error, i.e., $\hat{\boldsymbol{c}} = \boldsymbol{c} + \boldsymbol{q}$. The received vector, which is affected by channel noise as well as quantization error, is given by $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{q} + \boldsymbol{e}$. As a result, the syndrome samples will be distorted and the new syndrome is given by

$$\tilde{\boldsymbol{s}} = H\boldsymbol{r} = H(\boldsymbol{q} + \boldsymbol{e}) = \boldsymbol{s}_q + \boldsymbol{s}_e, \tag{A.11}$$

where $\boldsymbol{s}_q \equiv H\boldsymbol{q}$, and $\boldsymbol{s}_e \equiv H\boldsymbol{e}$. However, we can use this distorted syndrome to perform decoding, particularly if quantization noise is much smaller than channel errors [86], [100]. The new syndrome matrix $\tilde{S}_t$, is the same as S_t except that its entries are distorted syndrome samples, as given in (A.11). Obviously, the rank of $\tilde{S}_t$ is not necessarily equal to the number of errors, since it is unlikely to get a singular matrix. It is thus common to set a threshold, either theoretically [100] or empirically [40], to determine the rank of $\tilde{S}_t$. This is usually accomplished by doing singular value (eigenvalue) decomposition and estimating the number of nonzero diagonal elements (eigenvalues).

The rest of decoding, namely, error localization and calculation, is similar to what we discussed in Section A.3.1, except that syndrome samples are replaced by distorted syndrome samples in (A.7)-(A.10). With resulting distorted error locating polynomial, it is difficult to reliably localize errors, unless quantization noise is much smaller than channel errors. The last step in decoding algorithm is also affected by quantization error, and the problem of computing the magnitude of errors also becomes an estimation problem, as the syndrome samples are random.

A.4 LS Decoding Algorithms for Quantized BCH-DFT Codes

To somewhat alleviate the effect of quantization noise, one can use the least squares (LS) solution to estimate the number of errors [113], coefficients of the error locating polynomial and magnitude of errors [86,113]. This is based on the observation that every decoding step in Section A.3 is using only a limited number of available syndrome samples. More precisely, without quantization, ν syndrome samples are enough to exactly determine the magnitude of ν errors in (A.10) and 2ν samples are sufficient for detection and localization of ν errors. There is no benefit in using more samples. Nevertheless, in quantized codevectors, one can utilize the remaining $2t-\nu$ and $2t-2\nu$ samples to, respectively, improve the estimation of error magnitudes as well as the quantity and location of errors.

A.4.1 Error Detection and Localization

Consider the error locating polynomial in (A.7) for quantized DFT codes, i.e., with distorted syndrome. To have a better visualization, we rewrite it in the following matrix form

$$\underbrace{\begin{bmatrix} \tilde{s}_1 & \tilde{s}_2 & \dots & \tilde{s}_\nu \\ \tilde{s}_2 & \tilde{s}_3 & \dots & \tilde{s}_{\nu+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_\nu & \tilde{s}_{\nu+1} & \dots & \tilde{s}_{2\nu-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_{2t-\nu} & \tilde{s}_{2t-\nu+1} & \dots & \tilde{s}_{2t-1} \end{bmatrix}}_{L_{\nu,t}} \begin{bmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = - \begin{bmatrix} \tilde{s}_{\nu+1} \\ \tilde{s}_{\nu+2} \\ \vdots \\ \tilde{s}_{2\nu} \\ \vdots \\ \tilde{s}_{2t} \end{bmatrix}. \tag{A.12}$$

Now, it is easy to see that for $\nu < t$ the system is *overdetermined*, i.e., there are more equations than unknowns. Precisely speaking, there are $2t-\nu$ equations with ν unknowns. Thus, the estimation of $\mathbf{\Lambda} = [\Lambda_\nu, \Lambda_{\nu-1}, \ldots, \Lambda_1]^T$ becomes more accurate if we find the LS solution. The accuracy of the LS estimation depends on the the number of equations per unknowns, which is $\frac{2t-\nu}{\nu}$. Thus, it increases as the number of errors decreases.

The question that remains is how to determine the number of errors. In other words, with this arrangement, can we also estimate ν more accurately than what we did in Section A.3.2? The answer is positive and the new arrangement of syndrome samples in matrix $L_{\nu,t}$ in (A.12) also gives rise to an improved estimation of the number of errors in the presence of quantization error. Similar to what we discussed in Section A.3.2, to find the number of errors we begin with evaluating the rank of $L_{t,t}$, which is essentially the square matrix $\tilde{S}_t$, i.e., $L_{t,t} = \tilde{S}_t$. As we mentioned earlier, with distorted syndrome samples, the rank of $\tilde{S}_t$ is not necessarily equal to the number of errors, and we need to set a threshold to determine its rank. Then, similar to what we did in Section A.3.2, we can determine if $L_{t,t}$ is a full rank matrix or not. But unlike that, if this matrix is singular, we evaluate the singularity of $L_{t-1,t}$ rather than $\tilde{S}_{t-1} = L_{t-1,t-1}$ in the next step. In general, if $L_{\mu,t}$ is nonsingular for $\mu = \nu < t$ but it is singular for $\mu = \nu + 1$, then ν indicates the number of channel errors. Observe that, for $\mu = \nu + 1 \leq t$, $L_{\mu,t}$ is a tall matrix which makes use of $2t-1$ syndrome samples while the square matrix $\tilde{S}_\mu$ includes only part of them $(2\mu - 1)$; thus, a better estimation is attainable in the first case.

A.4.2 Error Estimation

Although a reliable localization is necessary for proper decoding, with conventional estimation method, presented in Section A.3.2, even perfect error localization does not imply a small estimation error. We show that the LS solution can largely overcome this problem and reduce the MSE between reconstructed and original sequences. The gain comes from the exploitation of more syndrome samples by

$$H_e \boldsymbol{E} = \tilde{\boldsymbol{s}}, \tag{A.13}$$

which engages all $2t$ syndrome samples to estimate $\nu \leq t$ errors. The accuracy of estimation depend on the number of equations per input sample, which is a function of code rate $(\frac{n-k}{k} = \frac{1}{R} - 1)$. The lower the code-rate, the more accurate the error estimation.

A.5 Performance Analysis and Reconstruction Error

Any continuous-valued signals is required to be approximated to a set of discrete levels before being transmitted in digital communication systems. This approximation from continuous to discrete space is realized by quantization which is a non-invertible process. In this section, we analyze the effect of *soft redundancy* inserted by DFT codes on the quantization error, and we show that the MSE introduced by quantization error reduces on account of this redundancy. Owing to its efficacy in measuring signal fidelity, particularly in conjunction with Gaussian random variables and linear systems, we use the MSE between transmitted and reconstructed codevectors to measurers end to end distortion.

A.5.1 Quantization Model

In order to be able to analyze and compare the performance of quantized DFT codes, we need to model quantization noise stochastically. We assume that the range of the quantizer covers the dynamic range of all codevectors of the DFT code. Let $\boldsymbol{q}$ denote the quantization error vector. Our model is based on the following assumptions [51]:

- Each noise component q_i has mean equal to zero and variance σ_q^2.
- The noise components q_i and q_j are uncorrelated for $i \neq j$.

Therefore, for any $0 \leq i, j \leq n$ we have

$$\mathbb{E}\{q_i\} = 0, \qquad \mathbb{E}\{q_i q_j\} = \sigma_q^2 \delta_{ij}, \tag{A.14}$$

where δ denotes the Kronecker delta function. The design of quantizer depends on the dynamic range and statistical distributions of the source sequence [51]. For simplicity, we consider a *uniform quantizer* with a fixed quantization step size Δ. Although the choice of Δ is arbitrary, decreasing Δ decreases quantization error but increases the amount of data to be transmitted. In order for the quantizer to satisfy the criteria in (A.14), we assume that Δ is small enough such that the probability distribution of the quantization noise vector is uniform on the interval $[-\Delta/2, \Delta/2]$. Therefore, $\sigma_q^2 = \Delta^2/12$.

A.5.2 Reconstruction

The codevectors in a DFT code are generated by $\boldsymbol{c} = G\boldsymbol{x}$ where G is defined in (A.1). Since G is not a square matrix, one possibility to linearly reconstruct $\boldsymbol{x}$ form $\boldsymbol{c}$ is to use the pseudoinverse of G [51], which is defined by $G^{\dagger} = (G^T G)^{-1} G^T$. It is easy to check that $G^{\dagger} G = I_k$, hence $G^{\dagger}(G\boldsymbol{x}) = \boldsymbol{x}$. Since $G^T G = \frac{n}{k} I_k$ [86], the pseudoinverse $G^{\dagger}$ is further simplified and the linear reconstruction can be written as

$$\boldsymbol{x} = G^{\dagger}\boldsymbol{c} = (G^T G)^{-1} G^T \boldsymbol{c} = \frac{k}{n} G^T \boldsymbol{c}. \tag{A.15}$$

Let $\boldsymbol{q}$ denote the quantization error that satisfies the conditions in Section A.5.1. Suppose we want to estimate $\boldsymbol{x}$ from $\hat{\boldsymbol{c}} = G\boldsymbol{x} + \boldsymbol{q}$. From (A.15), we obtain

$$\hat{\boldsymbol{x}} = \frac{k}{n} G^T \hat{\boldsymbol{c}} = \boldsymbol{x} + \frac{k}{n} G^T \boldsymbol{q}, \tag{A.16}$$

thus reconstruction error due to quantization is $\frac{k}{n} G^T \boldsymbol{q}$ and the mean squared reconstruction error is [86], [51],

$$\begin{aligned} \mathrm{MSE_q} &= \frac{1}{k}\mathbb{E}\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2\} = \frac{1}{k}\mathbb{E}\{\|\frac{k}{n} G^T \boldsymbol{q}\|^2\} \\ &= \frac{k}{n^2}\mathbb{E}\{\boldsymbol{q}^T G G^T \boldsymbol{q}\} = \frac{k}{n^2}\mathrm{Var}\{\boldsymbol{q}\}\,\mathrm{tr}\left(GG^T\right) \\ &= \frac{k}{n}\sigma_q^2 \end{aligned} \tag{A.17}$$

where tr(.) represent the trace function. Note that the second line follows from $\mathbb{E}[\boldsymbol{x}^T A \boldsymbol{x}] = \mathrm{tr}[\mathrm{Var}[\boldsymbol{q}] A A^T] + \mathbb{E}[\boldsymbol{q}]^T A \mathbb{E}[\boldsymbol{q}]$, the third line results from $G^T G = \frac{n}{k} I_k$. Since $k < n$, (A.17) proves that DFT codes decrease quantization error when there is no channel error [86], [51].

We show that this is correct even if channel errors exist. To see this, let $\boldsymbol{e}$ denote channel errors, then received codevector is affected both by quantization and channel errors, that is, $\hat{\boldsymbol{c}} = G\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{\eta} = \boldsymbol{q} + \boldsymbol{e}$. Assuming that the quantization and channel errors are independent, we have [113]

$$\begin{aligned}\mathrm{MSE_{q+e}} &= \frac{1}{k}\mathbb{E}\{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|^2\} = \frac{1}{k}\mathbb{E}\{\|\frac{k}{n}G^T\boldsymbol{\eta}\|^2\} \\ &= \frac{k}{n}\sigma_\eta^2 = \frac{k}{n^2}\mathbb{E}\{\boldsymbol{q}^T\boldsymbol{q}+\boldsymbol{q}^T\boldsymbol{e}+\boldsymbol{e}^T\boldsymbol{q}+\boldsymbol{e}^T\boldsymbol{e}\} \\ &= \frac{k}{n}\left[\sigma_q^2+\frac{\nu}{n}\sigma_e^2\right],\end{aligned} \tag{A.18}$$

where ν is the number of errors and $\mathbb{E}\{\boldsymbol{e}^T\boldsymbol{e}\} \triangleq \nu\sigma_e^2$. Also, note that $\mathbb{E}\{\boldsymbol{e}^T\boldsymbol{q}\} = \mathbb{E}\{\boldsymbol{e}^T\}\mathbb{E}\{\boldsymbol{q}\} = 0$ based on the assumption in (A.14). Similarly, $\mathbb{E}\{\boldsymbol{q}^T\boldsymbol{e}\} = 0$.

From (A.18), it is evident that reconstruction error has two distinct parts, one due to the quantization error and another one due to the channel error. It also proves that DFT codes decrease both channel error and quantization error by a factor of k/n, which is equal to the code rate R_c. Moreover, we also conclude that the MSE is monotonically increasing with the number of errors as well as their power. It is also worth noting that, even without correcting errors, the MSE using DFT codes with linear reconstruction can be smaller than quantization error. More precisely, $MSE_{q+e} \leq \sigma_q^2$ for

$$\frac{\sigma_e^2}{\sigma_q^2} \leq \frac{n}{k}\frac{n-k}{\nu} \simeq \frac{n}{k}\frac{2t}{\nu}, \tag{A.19}$$

without error correction but merely using linear reconstruction. Then, while $\sigma_e^2 \leq \frac{2}{R_c}\sigma_q^2$, a reconstruction error better than quantization error is guaranteed if the number of errors is within the error correction capability of code ($\nu \leq t$). Eventually, in the extreme case, when all samples in a codevector are corrupted by channel errors ($\nu = n$), reconstruction error is less than quantization error as long as

$$\sigma_e^2 \leq (\frac{1}{R_c}-1)\sigma_q^2. \tag{A.20}$$

This simply proves the superiority of DFT (real-number) to binary channel coding, for a given code rate, when the distribution of channel errors is such that (A.20) holds.

The above results are valid when no error correction is done. With error correction, the MSE mainly depends on the accuracy of estimation at localizing the errors and finding their magnitude. The former improves when the number of errors is small compared to t, whereas the latter depends on the code rate and substantially improves for low-rate codes.

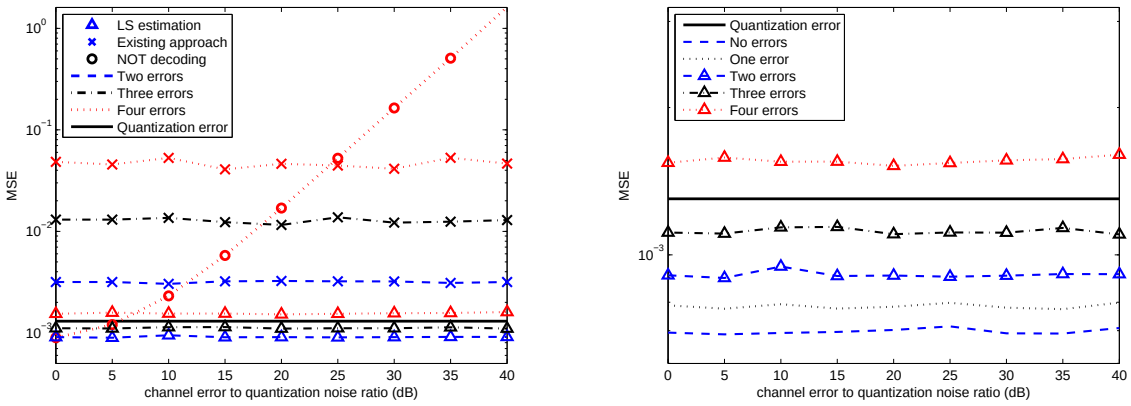
A.6 Numerical Results

To evaluate the performance of the LS decoding, simulations are carried out for transmitting both memoryless source with uniform distribution and Gauss-Markov source with zero mean, unit variance, and correlation coefficient 0.9, over an impulsive channel for a range of channel-error-to-quantization-noise ratio (CEQNR), defined as σ_e^2/σ_q^2. The generated sequences are encoded using a DFT code. The codevectors are then quantized with a 6-bit quantizer, and transmitted over a noisy channel that randomly inserts $\nu \leq t$ errors, generated by $\mathcal{N}(0, \sigma_e^2)$. For each setting, we evaluate the effect of LS estimation assuming perfect or imperfect localization of errors, and compare the MSE of received and decoded codevectors with respect to the input signal.

Perfect Localization

To evaluate the effect of the LS solution in the estimation of errors, first we assume that the location of errors are perfectly known to the decoder. Effectively, this gives rise to an erasure channel which has applications in packet networks such as the internet where some packets are lost during transmission [51].

In Fig. A.3(a), for a $(17, 9)$ DFT code, we compare the LS estimation method with the conventional error estimation as well as the case where error correction is not performed. The latter corresponds to (A.18) and referred to as "not decoding." The performance gap between the LS estimation and the existing method is remarkably high, and almost constant, in any CEQNR. At very low CEQNR "not decoding" results in a MSE lower than quantization error, for any $\nu \leq t$; the plot in Fig. A.3(a) is for $\nu = t = 5$. This is due to the fact that DFT codes, and tight frames in general, minimize the MSE [51]. This elegant property, which is achieved in light of soft redundancy, is one of the main reason to use these codes in the context of error correction. To get a better visibility, we compare the result for the LS method with quantization noise in a separate figure. Figure A.3(b) clearly shows that the MSE between transmitted and linearly reconstructed signals is less than quantization error for several error patterns. The LS estimation substantially decreases the estimation error even when code length goes up. Moreover, for low-rate codes, the MSE is better than quantization for any error pattern [113]. This indicates that, in DFT codes the MSE can go under quantization error level even though there are many errors.



(a) (b) LS estimation

Figure A.3 The MSE performance of a $(17, 9)$ DFT code with perfect error localization for different error patterns. (a) The merit of LS estimation to the existing approach. (b) The LS estimation: This figure shows DFT codes can result a MSE better than quantization error, even when several errors occur.

Considering Localization Error

When error localization is not perfect, the LS estimation still performs much better than the existing approach. This improvement, however, is not the same for different CEQNRs since error localization depends on CEQNR. It is noticeably high at low CEQNRs but gradually comes down as CEQNR increases. This loss is due to the fact that as CEQNRs becomes larger even one localization error can lead to a poor estimation and severely increase the MSE. Fascinatingly, at low CEQNRs, even with very poor probability of error localization, the LS estimation gives an acceptable MSE. In this range of CEQNR it is very challenging to reliably localize errors, as it is hard to distinguish between channel and quantization errors. Previous works have ignored this region either by limiting their study to the case that channel errors are larger than quantization error [100] or by excluding the results for this range [86]. Considering conventional localization approaches, either coding theoretic or subspace-based approaches, in such a range of CEQNR, not decoding would be better than decoding if we use the existing estimation method. The LS estimation, however, overcomes this deficiency [113]. This becomes more important noting that without the LS estimation, even perfect error localization cannot guarantee relatively low MSE, as shown in [113].

Appendix B

Proofs of Chapter 4

B.1 Proof of Theorem 4.3

Proof. Let $n = Mk + l, 0 < l < k$, then G can be partitioned as

$$G = [G_k^H \mid G_k^{1H} \mid \cdots \mid G_k^{(M-1)H} \mid G_{k\times l}^{MH}]^H.$$

where $G_k, G_k^1, \ldots, G_k^{M-1}$ are square submatrices and G^M is an $l \times k$ submatrix of G. In general, $G_k, G_k^1, \ldots, G_k^{M-1}$ and $G_{k\times l}^M$ include arbitrary rows of G, hence they have different spectrums, i.e., different sets of eigenvalues. Suppose, for the purpose of contradiction, that $\lambda_k(G_k^H G_k) = 1$; this can occur only if G_k consist of the rows of G such that the distance between each two successive rows is at least M.[1] Such an arrangement guarantees the existence of $G_k^1, \ldots, G_k^{M-1}$ so that $G_k^{mH} G_k^m$, for any $1 \le m \le M-1$, has the same spectrum as $G_k^H G_k$. To find the row indices corresponding to G_k^m, we can simply add m to each row index of G_k. Then, to show these matrices have the same spectrum, we use Lemma 4.1. Given a G_k, one can verify that $(G_k^m)_{i,j} = e^{j\frac{2\pi m}{n}}(G_k)_{i,j}$ and thus $(G_k^m)_{i,j}^H = e^{-j\frac{2\pi m}{n}}(G_k)_{i,j}^H$. Therefore, $G_k^{mH} G_k^m$ and $G_k^H G_k$ have the same spectrum for any $1 \le m \le M-1$. Next, we see that $G^H G = A + B$ in which $A = G_k^H G_k + \cdots + G_k^{(M-1)H} G_k^{M-1}$ and $B = G_{k\times l}^{MH} G_{k\times l}^M$. Then, in consideration of the above discussion, $\lambda_i(A) = M\lambda_i(G_k^H G_k)$ for any $1 \le i \le k$.

[1] $\lambda_k(G_k^H G_k) = 1$ is the optimal solution for (4.28) and necessitate $d_{\min} = M$, as discussed in Theorem 4.7.

Hence, from (4.20), for $i = 1, j = k$, we will have

$$\begin{aligned} &\lambda_k(A) + \lambda_1(B) \leq \lambda_1(A + B) \\ \Leftrightarrow\ &M\lambda_k(G_k^H G_k) \leq \frac{n}{k} - \lambda_1(B) \\ \Leftrightarrow\ &\lambda_k(G_k^H G_k) \leq \frac{\frac{n}{k} - 1}{M} = \frac{\frac{n}{k} - 1}{\lfloor \frac{n}{k} \rfloor} < 1, \end{aligned} \tag{B.1}$$

where the last line follows using $\lambda_1(B) \geq 1$ from Theorem 4.1. But this is contradicting our assumption $\lambda_k(G_k^H G_k) = 1$, and thus completes the proof that, for $n \neq Mk$, the largest possible $\lambda_k(G_k^H G_k)$ is strictly less than 1, for any G_k.[2] The proof of the other bound ($\lambda_1(G_k^H G_k) > 1$) is then immediate because

$$\sum_{i=1}^{k} \lambda_i(G_k^H G_k) = \sum_{i=1}^{k} a_{ii} = k$$

. □

B.2 Proof of Theorem 4.7

Proof. Consider an (n, k) DFT frame, let $M = \lfloor n/k \rfloor$, and assume that all rows in $\mathcal{I}_{r_k}$, except the first and last rows, are equally spaced with distance M (without loss of generality, we assume $i_{r_1} = 1$, then $i_{r_j} = (j-1)M + 1$, $j \leq k$). Hence $d_{\min} = M$, where the minimum distance $d_{\min}$ is defined as the smallest circular distance among the selected rows. In such a setting, from (4.49) and similar to (4.51), we can write

$$\det(V_k V_k^H) = \frac{2^{k(k-1)}}{k^k} \prod_{r=1}^{k-1} \left(\sin^2 \frac{\pi}{n} Mr \right)^{k-r}. \tag{B.2}$$

We prove that, in view of (4.45), the systematic frame corresponding to the above arrangement has better performance than any other arrangement in which $d_{\min}$ among the systematic rows is less than M. To this end, we first assume that all selected rows in $\mathcal{I}_{r_k}$ remain the same except one row which is shifted one unit in a way that $d_{\min}$ decreases.

[2] Note that when $n = Mk$, B is an empty matrix and we must plug $\lambda_1(B) = 0$ into (B.1) which result in $\lambda_k(G_k^H G_k) \leq 1$ and does not guarantee a bound strictly less than 1.

For example, without loss of generality, consider $\mathcal{I}'_{r_k}$ for which $i'_{r_1} = 2$, $i'_{r_j} = i_{r_j}, 1 < j \leq k$; hence $d_{\min} = M - 1$. Then, from (4.49), we obtain

$$\frac{\det(V_k V_k^H)|_{\mathcal{I}'_{r_k}}}{\det(V_k V_k^H)|_{\mathcal{I}_{r_k}}} = \frac{\prod_{r=1}^{k-1} \sin^2 \frac{\pi}{n}(Mr - 1)}{\prod_{r=1}^{k-1} \sin^2 \frac{\pi}{n} Mr} < 1. \tag{B.3}$$

To prove the inequality, equivalently, we show that

$$\frac{\sin \frac{(M-1)\pi}{n} \sin \frac{(2M-1)\pi}{n} \cdots \sin \frac{((k-1)M-1)\pi}{n}}{\sin \frac{M\pi}{n} \sin \frac{2M\pi}{n} \cdots \sin \frac{(k-1)M\pi}{n}} < 1. \tag{B.4}$$

We break up this inequality into $\lfloor k/2 \rfloor$ inequalities, each of which strictly less than one. First, consider the first and last terms in the numerator and denominator. We can write

$$\begin{aligned} \frac{\sin \frac{(M-1)\pi}{n} \sin \frac{((k-1)M-1)\pi}{n}}{\sin \frac{M\pi}{n} \sin \frac{(k-1)M\pi}{n}} &= \frac{\cos \frac{(k-2)M\pi}{n} - \cos \frac{(kM-2)\pi}{n}}{\cos \frac{(k-2)M\pi}{n} - \cos \frac{kM\pi}{n}} \\ &< 1, \end{aligned} \tag{B.5}$$

where the inequality follows since $\cos \frac{(kM-2)\pi}{n} > \cos \frac{kM\pi}{n}$, as $\frac{kM}{n}\pi \leq \pi$. Likewise, for the second and penultimate terms we have

$$\begin{aligned} \frac{\sin \frac{(2M-1)\pi}{n} \sin \frac{((k-2)M-1)\pi}{n}}{\sin \frac{2M\pi}{n} \sin \frac{(k-2)M\pi}{n}} &= \frac{\cos \frac{(k-4)M\pi}{n} - \cos \frac{(kM-2)\pi}{n}}{\cos \frac{(k-4)M\pi}{n} - \cos \frac{kM\pi}{n}} \\ &< 1. \end{aligned} \tag{B.6}$$

A similar reasoning can be used for other terms that are equally spaced from the two ends.

Clearly, the same argument is valid when $2 < i'_{r_1} < M$ and the other rows are the same, i.e., $i'_{r_j} = i_{r_j}, 1 < j \leq k$ and $d_{\min} = M - i'_{r_1}$. Moreover, when more than one row index is changed, in a way that two or more selected rows have a distance less than M, the above argument is valid and we can show that new determinant is even less than the case with one changed index. In fact, in such a case, it is easier to compare the new one with its parent; i.e., to compare the case with two changes with the case with one change. As a result, we can see that any combination of rows with $d_{\min} < M$ performs worse than the case with $d_{\min} = M$, on account of (37); that is, $d_{\min} = M$ is necessary condition for optimality. In other words, that optimal systematic frame must satisfy $d_{\min} = M$.

Next, we show that among systematic frames with $d_{\min} = M$ the one that satisfies (4.52) is the best. That is, the optimal systematic frame has $l = n - \lfloor n/k \rfloor k$ systematic rows with successive circular distance of $\lceil n/k \rceil$ and $m = k - l$ systematic rows with successive circular distance of $\lfloor n/k \rfloor$. To prove this, again we compare $\det(V_k V_k^H)$ in (4.49) for this case and the other cases with $d_{\min} = M$. The arguments are very similar to what we used above. Before moving on, we should mention that for $l \in \{0, 1, k-1\}$ the proof in the first part is sufficient.

Let $\mathcal{I}_{r_k}^o$ denote the set of rows satisfying the constraints in (4.52); obviously, $d_{\min} = M$. We claim that any other selection of systematic rows, for which $d_{\min}$ is M, results in a smaller $\det(V_k V_k^H)$; that is, $\det(V_k V_k^H)|_{\mathcal{I}_{r_k}} < \det(V_k V_k^H)|_{\mathcal{I}_{r_k}^o}$. Let us evaluate the case where only the row index for one of those l rows varies, provided that $d_{\min} = M$ is kept.[3] We then have

$$\frac{\det(V_k V_k^H)|_{\mathcal{I}_{r_k}}}{\det(V_k V_k^H)|_{\mathcal{I}_{r_k}^o}} = \frac{\prod_{r=1}^{k-1} \sin^2 \frac{\pi}{n} Mr}{\prod_{r=1}^{k-1} \sin^2 \frac{\pi}{n}(Mr+1)} < 1. \tag{B.7}$$

Again it suffice to prove that

$$\frac{\sin \frac{M\pi}{n} \sin \frac{2M\pi}{n} \cdots \sin \frac{(k-1)M\pi}{n}}{\sin \frac{(M+1)\pi}{n} \sin \frac{(2M+1)\pi}{n} \cdots \sin \frac{((k-1)M+1)\pi}{n}} < 1, \tag{B.8}$$

and this can be done by the same divide and conquer approach, used in the first part of this proof. For instance, for the first and last terms in the numerator and denominator we have

$$\begin{aligned} \frac{\sin \frac{M\pi}{n} \sin \frac{(k-1)M\pi}{n}}{\sin \frac{(M+1)\pi}{n} \sin \frac{((k-1)M+1)\pi}{n}} &= \frac{\cos \frac{(k-2)M\pi}{n} - \cos \frac{kM\pi}{n}}{\cos \frac{(k-2)M\pi}{n} - \cos \frac{(kM+2)\pi}{n}} \\ &< 1, \end{aligned} \tag{B.9}$$

where the inequality follows for $\cos \frac{(kM+2)\pi}{n} < \cos \frac{kM\pi}{n}$. Finally, the other cases, where two or more rows change, can be proved comparing their determinant with their ancestor's with a similar reasoning. This completes the proof that a systematic frame with the most evenly

[3] Note that, with this shift of row, we are looking for an arrangement of a systematic frame that does not satisfy (4.52); otherwise, $\det(V_k V_k^H)$ will not vary, as the frame properties has not changed essentially. More specifically, a new, different arrangement will introduce a new distance equal to $\lceil n/k \rceil + 1$.

spaced systematic rows, or equivalently data samples in the corresponding codewords, is the best in the minimum MSE sense.

□

References

[1] The DISCOVER codec: Complexity performance evaluation. [Online]. Available: `http://www.img.lx.it.pt/~discover/complexity.html` [Accessed October 2013].

[2] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proc. IEEE Data Compression Conference*, pages 252–261, 2002.

[3] M. Akçakaya and V. Tarokh. A frame construction and a universal distortion bound for sparse representations. *IEEE Transactions on Signal Processing*, 56(6):2443–2450, 2008.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[5] E. Akyol and K. Rose. Optimized analog mappings for distributed source-channel coding. In *Proc. IEEE Data Compression Conference (DCC)*, pages 159–168, 2010.

[6] E. Akyol, K. Rose, and T. Ramstad. Optimal mappings for joint source channel coding. In *Proc. IEEE Information Theory Workshop (ITW)*, pages 1–5, 2010.

[7] E. Akyol, K. Viswanatha, K. Rose, and T. Ramstad. On zero delay source-channel coding. *arXiv preprint arXiv:1302.3660*, 2013.

[8] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The DISCOVER codec: Architecture, techniques and evaluation. In *Proc. Picture Coding Symposium*, 2007.

[9] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 1400–1404, 2001.

[10] F. Bassi, M. Kieffer, and C. Weidmann. Source coding with intermittent and degraded side information at the decoder. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2941–2944, 2008.

[11] T. Berger. *Rate distortion theory: A mathematical basis for data compression*. Prentice-Hall, 1971.

[12] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *Proc. IEEE International Conference on Communications (ICC 93)*, volume 2, pages 1064–1070. IEEE, 1993.

[13] R. E. Blahut. *Algebraic Methods for Signal Processing and Communications Coding*. New York: Springer-Verlag, 1992.

[14] R. E. Blahut. *Algebraic Codes for Data Transmission*. New York: Cambridge Univ. Press, 2003.

[15] T. K. Blankenship and T. S. Rappaport. Characteristics of impulsive noise in the 450-MHz band in hospitals and clinics. *IEEE Transactions on Antennas and Propagation*, 46(2):194–203, 1998.

[16] B. G. Bodmann and P. K. Singh. Burst erasures and the mean-square error for cyclic Parseval frames. *IEEE Transactions on Information Theory*, 57(7):4622–4635, July 2011.

[17] R. C. Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, 1960.

[18] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[19] Catarina Brites and Fernando Pereira. Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1177–1190, 2008.

[20] P. G. Casazza and J. Kovačević. Equal-norm tight frames with erasures. *Advances in Computational Mathematics*, 18(2):387–430, February 2003.

[21] T. Chen, E. Serpedin, and D. Rajan. *Mathematical Foundations for Signal Processing, Communications, and Networking*. CRC Press, 2011.

[22] X. Chen and E. Tuncel. Low-delay prediction-and transform-based Wyner–Ziv coding. *IEEE Transactions on Signal Processing*, 59(2):653–666, February 2011.

[23] X. Chen and E. Tuncel. Zero-delay joint source-channel coding for the Gaussian Wyner-Ziv problem. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 2929–2933, 2011.

[24] Z. Chen and J. Dongarra. Numerically stable real-number codes based on random matrices. *Computational Science–ICCS 2005*, pages 115–122, 2005.

[25] N. M. Cheung, H. Wang, and A. Ortega. Sampling-based correlation estimation for distributed source coding under rate and complexity constraints. *IEEE Transactions on Image Processing*, 17(11):2122–2137, 2008.

[26] M. H. M. Costa. Writing on dirty paper. *IEEE Transactions on Information Theory*, 29(3):439–441, May 1983.

[27] D. J. Costello Jr, A. E. Pusane, S. Bates, and K. S. Zigangirov. A comparison between LDPC block and convolutional codes. In *Proc. Information Theory and Applications Workshop*, pages 6–10, 2006.

[28] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, 2006.

[29] N. Deligiannis, J. Barbarien, M. Jacobs, A. Munteanu, A. Skodras, and P. Schelkens. Side-information-dependent correlation channel estimation in hash-based distributed video coding. *IEEE Transactions on Image Processing*, 21(4):1934–1949, 2012.

[30] M. Dohler, R. W. Heath, A. Lozano, C. B. Papadias, and R. A. Valenzuela. Is the PHY layer dead? *IEEE Communications Magazine*, 49(4):159–165, 2011.

[31] A. Dragland. Big data - for better or worse. [Online]. `http://www.sintef.no/home/Press-Room/Research-News/Big-Data--for-better-or-worse` [Accessed October 2013].

[32] P. L. Dragotti and M. Gastpar. *Distributed Source Coding: Theory, Algorithms, and Applications*. Academic Press, 2009.

[33] M. F. Duarte and R. G. Baraniuk. Spectral compressive sensing. *Applied and Computational Harmonic Analysis*, 35(1):111–129, 2013.

[34] F. Dufaux, W. Gao, S. Tubaro, and A. Vetro. Distributed video coding: Trends and perspectives. *Journal on Image and Video Processing*, 2009:10, 2009.

[35] E. Dupraz, F. Bassi, T. Rodet, and M. Kieffer. Distributed coding of sources with bursty correlation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2973–2976, 2012.

[36] A. W. Eckford and W. Yu. Rateless Slepian-Wolf codes. In *Proc. 39th Asilomar Conf. Signals, Syst. Comput.*, pages 1757–1761, 2005.

[37] A. El Gamal and Y. H. Kim. *Network information theory.* Cambridge University Press, 2011.

[38] A. Elamin, V. Jeoti, and S. Belhouari. Best fit models test for the virtual channel in distributed video coding. *Journal of Engineering Science and Technology*, 6(4):481–492, 2011.

[39] Y. Fang and J. Jeong. Correlation parameter estimation for LDPC-based Slepian-Wolf coding. *IEEE Communications Letters*, 13(1):37–39, Jan. 2009.

[40] A. Gabay, P. Duhamel, and O. Rioul. Real BCH codes as joint source channel codes for satellite images coding. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 820–824, 2000.

[41] A. Gabay, M. Kieffer, and P. Duhamel. Joint source-channel coding using real BCH codes for robust image transmission. *IEEE Transactions on Image Processing*, 16(6):1568–1583, June 2007.

[42] R. G. Gallager. *Low-density parity-check codes.* Ph.D. dissertation, MIT, 1963.

[43] Y. Gao and M. R. Soleymani. Rate-distortion function for Gauss-Markov source with side information at the decoder. In *25th Biennial Symposium on Communications*, pages 44–48, 2010.

[44] J. Garcia-Frias and Z. Xiong. Distributed source and joint source-channel coding: From theory to practice. In *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 1093–1096, 2005.

[45] J. Garcia-Frias and Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, 5(10):417–419, Oct. 2001.

[46] J. Garcia-Frias and W. Zhong. Ldpc codes for compression of multi-terminal sources with hidden markov correlation. *IEEE Communications Letters*, 7(3):115–117, 2003.

[47] Javier Garcia-Frias. Joint source-channel decoding of correlated sources over noisy channels. In *Proc. Data Compression Conference*, pages 283–292. IEEE, 2001.

[48] M. Gastpar. Uncoded transmission is exactly optimal for a simple Gaussian sensor network. *IEEE Transactions on Information Theory*, 54(11):5247–5251, 2008.

[49] M. Gastpar, M. Vetterli, and P. L. Dragotti. Sensing reality and communicating bits: A dangerous liaison. *IEEE Signal Processing Magazine*, 23(4):70–83, 2006.

[50] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, Jan. 2005.

[51] V. K. Goyal, J. Kovaevic, and J. A. Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10(3):203–233, 2001.

[52] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\mathbb{R}^N$: Analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, January 1998.

[53] R. M. Gray. *Toeplitz and Circulant Matrices: A Review*. Now Publishers, 2006.

[54] V. C. Gungor, B. Lu, and G. P. Hancke. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, 57(10):3557–3564, 2010.

[55] S. Haykin and K. J. R. Liu. *Handbook on Array Processing and Sensor Networks*. Wiley-IEEE Press, 2010.

[56] W. Henkel. RS codes, OFDM, and MIMO. In *Proc. the 15th International OFDM Workshop*, 2010.

[57] W. Henkel and F. Hu. OFDM and analog RS/BCH codes. In *Proc. the 9th International OFDM Workshop*, 2005.

[58] A. Hocquenghem. Codes correcteurs derreurs. *Chiffres*, 2(2):147–56, 1959.

[59] X. Huang and S. Forchhammer. Improved virtual channel noise model for transform domain Wyner-Ziv video coding. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 921–924. IEEE, 2009.

[60] J. Jiang, D. He, and A. Jagmohan. Rateless Slepian-Wolf coding based on rate adaptive low-density-parity-check codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 1316–1320, 2007.

[61] S. J. Johnson. *Iterative error correction: turbo, low-density parity-check and repeat-accumulate codes*. Cambridge University Press, 2009.

[62] S. J. Johnson and S. R. Weller. A family of irregular LDPC codes with low encoding complexity. *IEEE Communications Letters*, 7(2):79–81, 2003.

[63] S. M. Kay. *Modern Spectral Estimation*. Englewood Cliffs, N.J.: Prentice-Hall, 1988.

[64] J. Kovacevic and A. Chebira. Life beyond bases: The advent of frames (Part I). *IEEE Signal Processing Magazine*, 24(4):86–104, 2007.

[65] J. Kovacevic and A. Chebira. *An introduction to frames*. Now Publishers, 2008.

[66] A. Kumar and A. Makur. Improved coding-theoretic and subspace-based decoding algorithms for a wider class of DCT and DST codes. *IEEE Transactions on Signal Processing*, 58(2):695–708, February 2010.

[67] J. Kusuma. Parametric frequency estimation: ESPRIT and MUSIC. 2002. [Online]. Available: `http://cnx.org/content/m10588/latest`.

[68] F. Labeau, J. C. Chiang, M. Kieffer, P. Duhamel, L. Vandendorpe, and B. Macq. Oversampled filter banks as error correcting codes: Theory and impulse noise correction. *IEEE Transactions on Signal Processing*, 53(12):4619–4630, Dec. 2005.

[69] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6(10):440–442, Oct. 2002.

[70] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Joint source-channel coding of binary sources with side information at the decoder using IRA codes. In *Proc. IEEE Workshop on Multimedia Signal Processing*, pages 53–56, 2002.

[71] J. Lu and J. Moura. Linear time encoding of LDPC codes. *IEEE Transactions on Information Theory*, 56(1):233–249, 2010.

[72] T. Marshall Jr. Coding of real-number sequences for error correction: A digital signal processing problem. *IEEE Journal on Selected Areas in Communications*, 2(2):381–392, March 1984.

[73] F. Marvasti, M. Hasan, M. Echhart, and S. Talebi. Efficient algorithms for burst error recovery using FFT and other transform kernels. *IEEE Transactions on Signal Processing*, 47(4):1065–1075, April 1999.

[74] S. K. Mitra and Y. Kuo. *Digital Signal Processing: A Computer-Based Approach*. New York: McGraw-Hill, 2006.

[75] P. Mitran and J. Bajcsy. Turbo source coding: A noise-robust approach to data compression. In *Proc. IEEE Data Compression Conference*, page 465, 2002.

[76] V. S. S. Nair and J. A. Abraham. Real-number codes for fault-tolerant matrix operations on processor arrays. *IEEE Transactions on Computers*, 39(4):426–435, April 1990.

[77] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp. Distributed video coding: Selecting the most promising application scenarios. *Signal Processing: Image Communication*, 23(5):339–352, 2008.

[78] Y. Polyanskiy. *Channel coding: Non-asymptotic fundamental limits*. Ph.D. dissertation, Princeton University, 2010.

[79] Y. Polyanskiy, H. V. Poor, and S. Verdú. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, 2010.

[80] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Proc. IEEE Data Compression Conf.*, pages 158–167, 1999.

[81] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643, March 2003.

[82] R. Puri and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proc. the Annual Allerton Conference on Communication Control and Computing*, 2002.

[83] V. Stankovic Q. Xu and Z. Xiong. Distributed joint source-channel coding of video using raptor codes. *IEEE Journal on Selected Areas in Communications*, 25(4):851–861, May 2007.

[84] G. Rath and C. Guillemot. Frame-theoretic analysis of DFT codes with erasures. *IEEE Transactions on Signal Processing*, 52(2):447–460, February 2004.

[85] G. Rath and C. Guillemot. Recent advances in DFT codes based quantized frame expansions for erasure channels. *Digital Signal Processing*, 14(4):332–354, July 2004.

[86] G. Rath and C. Guillemot. Subspace algorithms for error localization with quantized DFT codes. *IEEE Transactions on Communications*, 52(12):2115–2124, December 2004.

[87] G. Rath and C. Guillemot. Subspace-based error and erasure correction with DFT codes for wireless channels. *IEEE Transactions on Signal Processing*, 52(11):3241–3252, November 2004.

[88] G. Rath and C. Guillemot. Characterization of a class of error correcting frames for robust signal transmission over wireless communication channels. *EURASIP Journal on Advances in Signal Processing*, 2005(2):229–241, 2005.

[89] G. Redinbo. Systematic wavelet subcodes for data protection. *IEEE Transactions on Computers*, 60(6):904–909, June 2011.

[90] G. Redinbo. Correcting DCT codes with Laurent Euclidean algorithm and syndrome extension. *IEEE Transactions on Signal Processing*, 61(9):2308–2318, May 2013.

[91] A. Roumy, K. Lajnef, and C. Guillemot. Rate-adaptive turbo-syndrome scheme for Slepian-Wolf coding. In *Proc. the forty-first Asilomar Conference on Signals, Systems and Computers*, pages 545–549, 2007.

[92] R. Roy and T. Kailath. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(7):984–995, 1989.

[93] W. E. Ryan. An Introduction to LDPC Codes, April 2003. [Online]. Available: `http://tuk88.free.fr/LDPC/ldpcchap.pdf`.

[94] F. Sacuto, F. Labeau, and B. L. Agba. Wide band time-correlated model for wireless communications under impulsive noise within power substation. [Online]. Available: `http://arxiv.org/abs/1305.4095`.

[95] M. Sartipi and F. Fekri. Distributed source coding in wireless sensor networks using LDPC coding: The entire slepian-wolf rate region. In *Proc. IEEE Wireless Communications and Networking Conference*, volume 4, pages 1939–1944, 2005.

[96] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, March 1986.

[97] G. A. F. Seber. *A Matrix Handbook for Statisticians*. New Jersey: John Wiley & Sons, 2008.

[98] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[99] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19(4):471–480, July 1973.

[100] G. Takos and C. N. Hadjicostis. Determination of the number of errors in DFT codes subject to low-level quantization noise. *IEEE Transactions on Signal Processing*, 56(3):1043–1054, March 2008.

[101] N. T. Thao and M. Vetterli. Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates. *IEEE Transactions on Signal Processing*, 42(3):519–531, March 1994.

[102] N. T. Thao and M. Vetterli. Reduction of the MSE in R-times oversampled A/D conversion $O(1/R)$ to $O(1/R^2)$. *IEEE Transactions on Signal Processing*, 42(1):200–203, January 1994.

[103] V. Toto-Zarasoa, A. Roumy, and C. Guillemot. Rate-adaptive codes for the entire Slepian-Wolf region and arbitrarily correlated sources. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2965–2968, 2008.

[104] V. Toto-Zarasoa, A. Roumy, and C. Guillemot. Maximum likelihood BSC parameter estimation for the Slepian-Wolf problem. *IEEE Communications Letters*, 15(2):232–234, February 2011.

[105] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge Univ. Press, 2005.

[106] G. H. Tucci and P. A. Whiting. Asymptotic results on generalized Vandermonde matrices and their extreme eigenvalues. In *Proc. the 49th Annual Allerton Conference on Communication, Control, and Computing*, pages 1816–1823, 2011.

[107] G. H. Tucci and P. A. Whiting. Eigenvalue results for large scale random Vandermonde matrices with unit complex entries. *IEEE Transactions on Information Theory*, 57(6):3938–3954, 2011.

[108] S. Y. Tung. *Multiterminal source coding*. Ph.D. thesis, Cornell University, Ithaca, NY, 1978.

[109] M. Vaezi, A. Combernoux, and F. Labeau. Low-Delay Joint Source-Channel Coding with Side Information at the Decoder. In *Proc. IEEE Digital Signal Processing Workshop (DSP/SPE)*, 2013.

[110] M. Vaezi and F. Labeau. Generalized and extended subspace algorithms for error correction with quantized DFT codes. *IEEE Transactions on Communications*, December 2013. accepted for publication.

[111] M. Vaezi and F. Labeau. Distributed lossy source coding using real-number codes. In *Proc. the 76th IEEE Vehicular Technology Conference, VTC Fall*, pages 1–5, 2012.

[112] M. Vaezi and F. Labeau. Improved modeling of the correlation between continuous-valued sources in LDPC-based DSC. In *Proc. the 46th Asilomar Conference on Signals, Systems and Computers*, pages 1659–1663, 2012.

[113] M. Vaezi and F. Labeau. Least squares solution for error correction on the real field using quantized DFT codes. In *Proc. the 20th European Signal Processing Conference (EUSIPCO)*, pages 2561–2565, 2012.

[114] M. Vaezi and F. Labeau. Systematic DFT frames: Principle and eigenvalues structure. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 2436–2440, 2012.

[115] M. Vaezi and F. Labeau. Extended subspace error localization for rate-adaptive distributed source coding. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 2174–2178, 2013.

[116] M. Vaezi and F. Labeau. Systematic DFT frames: Principle, eigenvalues structure, and applications. *IEEE Transactions on Signal Processing*, 61(15):3774–3885, August 2013.

[117] M. Vaezi and F. Labeau. Distributed source-channel coding based on real-field BCH codes. *IEEE Transactions on Signal Processing*, 62(5):1171–1184, March 2014.

[118] B. L. van der Waerden, E. Artin, and E. Noether. *Algebra. Vol. 1.* Springer, 1966.

[119] D Varodayan, A. Aaron, and B. Girod. Rate-adaptive codes for distributed source coding. *Signal Processing*, 86(11):3123–3130, Nov. 2006.

[120] A. B. Wagner, S. Tavildar, and P. Viswanath. Rate region of the quadratic Gaussian two-encoder source-coding problem. *IEEE Transactions on Information Theory*, 54(5):1938–1961, 2008.

[121] Z. Wang and G. B. Giannakis. Complex-field coding for OFDM over fading wireless channels. *IEEE Transactions on Information Theory*, 49(3):707–720, March 2003.

[122] N. Wernersson and M. Skoglund. Nonlinear coding and estimation for correlated data in wireless sensor networks. *IEEE Transactions on Communications*, 57(10):2932–2939, October 2009.

[123] T. Wiegand and H. Schwarz. *Source Coding: Part I of Fundamentals of Source and Video Coding*. Now Publishers, 2010.

[124] J. K. Wolf. Redundancy, the discrete Fourier transform, and impulse noise cancellation. *IEEE Transactions on Communications*, 31(3):458–461, March 1983.

[125] A. D. Wyner. Recent results in the Shannon theory. *IEEE Transactions on Information Theory*, IT-20(1):2–10, Jan. 1974.

[126] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, Jan. 1976.

[127] L. Xiaoqian, L. Rongke, and W. Runxin. A novel rate-adaptive distributed source coding scheme using polar codes. *IEEE Communications Letters*, 17(1):143–146, Jan. 2013.

[128] Z. Xiong, A. D. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21(5):80–94, September 2004.

[129] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao. Wyner-Ziv coding based on TCQ and LDPC codes. *IEEE Transactions on Communications*, 57(2):376–387, 2009.

[130] S. Yoon, I. Song, and S. Y. Kim. Code acquisition for DS/SS communications in non-Gaussian impulsive channels. *IEEE Transactions on Communications*, 52(2):187–190, 2004.

[131] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Transactions on Information Theory*, 42(6):2073–2084, 1996.

[132] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory*, 48(6):1250–1276, June 2002.

[133] G. C. Zhu and F. Alajaji. Turbo codes for nonuniform memoryless sources over noisy channels. *IEEE Communications Letters*, 6(2):64–66, 2002.