Tandem Filterbank / DFT Code for Bursty Erasure Correction

Joaquin Alejandro Zepeda Salvatierra

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

August 2006

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Electrical Engineering.

2006/08/25

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-28637-1
Our file Notre référence
ISBN: 978-0-494-28637-1

Abstract

Discrete Fourier Transform (DFT) encoding over the real (or complex) field has been proposed as a means to reconstruct samples lost in multimedia transmissions over packet-based networks. A collection of simple sample reconstruction (and error detection) algorithms makes DFT codes an interesting candidate. A common problem with DFT code sample reconstruction algorithms is that the quantization associated with practical implementations results in reconstruction errors that are particularly large when lost samples occur in bursts (bursty erasures).

Following a survey of DFT decoding algorithms, we present herein the Tandem Filterbank / DFT Code (TFBD). The TFBD code consists of a tandem arrangement of a filterbank and DFT encoder that effectively creates DFT codes along the rows (temporal codevectors) and columns (subband codevectors) of the frame under analysis. The tandem arrangement ensures that subband codevectors (the frame columns) will be DFT codes, and we show how the temporal codevectors (frame rows) can also be interpreted as DFT codes. All the subband and temporal codevectors can be used to reconstruct samples entirely independently of each other. An erasure burst along a particular codevector can then be broken up by reconstructing some lost samples along the remaining orientation; these samples can then be used as received samples in reconstructing the original codevector, a technique that we refer to as pivoting. Expressions related to the performance of the Tandem Filterbank / DFT (TFBD) code, including an expression for the temporal code reconstruction error and for temporal-to-subband pivoting operations, are derived and verified through simulations. The expressions also prove useful in the selection of the many parameters specifying a TFBD encoder. The design process is illustrated for two sample TFBD codes that are then compared to a benchmark DFT code at the same rate. The results show that the TFBD encoder is capable of reconstruction error improvements that are more than four orders of magnitude better than that of the benchmark DFT code.

Sommaire

L'encodage de Transforme de Fourier Discrte (TFD) sur le champ rél (ou complexe) a été proposé comme un outil permettant la reconstruction d'échantillons perdus lors d'une transmission multimédia à travers des réseaux à base de paquets. Un ensemble d'algorithmes de reconstruction d'échantillons (et de détection d'erreurs) rend les codes TDF un candidat particulièrement intéressant. Un probléme de base auquel se heurtent les algorithmes de reconstruction d'échantillons TDF est que la quantification associée avec une implémentation pratique implique des erreurs de reconstructions particulièrement élevées lorsque les échantillons perdus apparaissent en rafales.

Suite à une étude des algorithmes de décodage TDF, nous présentons le code Batterie de Filtres et Encodeur TDF en Tandem (*Tandem Filterbank / DFT, TFBD*). Le code utilise une batterie de filtres et un encodeur TDF en tandem, créant ainsi des codes TDF le long des rangées (vecteurs de code temporel) et colonnes (vecteur de code de sousbande) de la trame sous analyse. L'arrangement tandem assure que les vecteurs de sousbande seront bel et bien des codes TDF et nous montrons aussi comment les vecteurs de code de temporels peuvent être interprétés comme des codes TDF. En effet, tout les vecteurs de code temporel et de sous bande peuvent être utilisé pour la reconstruction d'échantillons d'une manière entièrement indépendante. Une suppression en rafale le long d'un vecteur de code en particulier peut tre interrompue par la reconstruction de certains échantillons perdus le long des orientations restantes; ces échantillons peuvent alors être utilisés pour la reconstruction du vecteur de code original, une technique que nous appelons le pivot.

Nous dérivons et validons à travers des simulations, des expressions reliées à la performance du code TFBD. Entre autres, nous dérivons une expression l'erreur de reconstruction du code temporel ainsi que une expression pour un les opérations de pivot temporel-à-sousbande. Ces expressions se montrent aussi particulièrement utiles lors de la sélection des paramètres spécifiant un encodeur TFBD. Notre processus de conception est illustré pour deux exemples de codes TFBD qui sont par la suite comparé à un code TDF de repère du même taux. Selon les résultats, l'encodeur TFBD démontre une amélioration par rapport aux erreurs de reconstructions de plus de quatre ordre de grandeur lorsque comparé avec l'encodeur TDF de base.

Acknowledgments

I would like to thank my parents for having, without hesitation and with all their love, tackled the monumental task of funding my education, and for inspiring in me an inextinguishable sense of responsibility.

I would like to thank my wife for motivating me to complete this work, and for her understanding during times of separation, and for her truly unconditional love.

I would like to thank my older brother for his constant support, and for his friendship, sincere concern and wisdom; my younger brother I would like to thank for being a reason to live a life well lived; and my family I thank for their help in many ways throughout my studies.

I would like to thank Prof. Fabrice Labeau for his academic guidance, his having offered a friendly hand in countless occasions, and for having opened a window into an interesting topic.

Thanks to Prof. Peter Radziszewski of the Comminution Dynamics Laboratory for having provided me with various opportunities throughout my studies, and for having shared his inspiring enthusiasm. Thanks to all the TSP students of MC 735 for their friendship and insights into this and their research topics. In particular, I would like to thank Karim Ali for doing the French translation of the abstract and Rui Ma and Mireille Sendashonga for help with the submission.

Contents

List of Figures

Chapter 1

Introduction

Traditional data transmission schemes consist of a tandem source/channel coder arrangement. In designing the source coder, however, it is assumed that the channel code is flawless in its correction of all channel errors [1], and this is not the case in practice. An immediate consequence of this is that, in the presence of channel errors, there will be a threshold after which further increasing the source code rate will result in increased distortion [2].

Common source and channel coding schemes take a finite input alphabet and map it to a finite output alphabet. For the case of space-time channel codes used in wireless communications, the alphabets are determined by the symbol constellation scheme (*e.g.,* M-PSK, QAM); the code performance is often measured by the probability of error in the symbol decision at the receiver [3]. Two well known examples of such codes are Alamouti's block code [4] and Tarokh's full-diversity convolutional codes [5].

A different set of codes known as *complex codes* carries out the encoding in the field of complex valued numbers, where the input and output alphabets are continuous and correspond to the entire complex plane (*real codes* can be seen as a special case of complex codes where the encoding is done over the real line). In the following section, we present the Impulse-Noise / Erasure Channel (INEC) model used in designing complex codes, and show how codes thus designed can be interpreted as joint source-channel codes.

1.1 Joint Source-Channel Code Design Using the Impulse Noise / Erasure Channel Model

The Impulse-Noise / Erasure Channel (INEC) model commonly used in the design of complex codes [6–9] is illustrated in Fig. 1.1. A complex valued input sequence $x[m]$ is encoded onto a complex valued vector sequence $\underline{y}[n]$. The encoded sequence is then quantized in preparation for transmission using traditional schemes (*e.g.,* a wireless network using space-time codes). The Impulse Noise / Erasure Channel (INEC) indicated in Fig. 1.1 provides an abstraction for this transmission and the distortion that might be introduced by it. At the receiving end, the decoder first attempts to detect and correct channel distortions along the received codevectors $\underline{\hat{y}}$, followed by regeneration of the original input sequence, thus producing the estimate $\hat{x}[m]$.

The INEC (Impulse Noise / Erasure Channel, *cf.* Fig. 1.1) accounts for two types of channel distortions: *impulse noise* and *erasures.* Impulse noise (denoted by $\underline{\nu}$ in Fig. 1.1) is additive noise restricted to a subset of samples of $\underline{\ddot{y}}$. It is meant to model errors in symbol decision at the receiver, which are impulsive in nature. Erasures represent samples that become lost (*e.g.,* due to excessive network traffic) along the transmission. Note that the decoder is aware of the position of erased samples along the transmitted vector, and thus erasures only need to be reconstructed. The position of impulse errors, on the other hand, is unknown. Hence erroneous samples first need to be detected and then reconstructed, where erasure reconstruction algorithms may be used [7, 10].

One can consider the combined effect of the mapping $x[m] \mapsto \underline{y}[n]$ and quantizer operation to be a joint source-channel code. The INEC model provides an abstraction representing all remaining elements in the path to the decoder, accounting particularly for errors and erasures that might be introduced along this path. One can then use the INEC to design a joint source-channel code that will provide for the correction of erroneous and lost samples, thus avoiding the traditional source code design assumption of flawless channel decoding.

1.1.1 Applications of Erasure Reconstruction Methods

This thesis addresses the erasure reconstruction aspect of a special type of complex codes known as DFT codes, particularly when the erasures are *bursty*, meaning that they occur

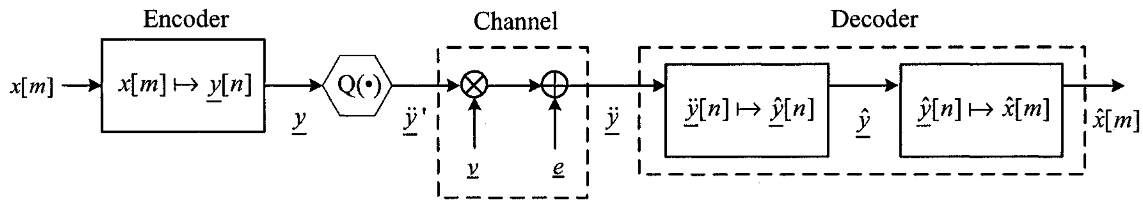

Fig. 1.1 *Code design for the impulse-noise / erasure channel model:* The error and erasure vectors are $\underline{\nu}$ and $\underline{e}$, respectively. The first stage in the decoder compensates for channel artifices, producing the estimate $\underline{\hat{y}}$ of $\underline{y}$, the second stage regenerates the message sequence. The $Q(\cdot)$ operator represents quantization. Note all the specified vectors (*e.g.*, $\underline{\nu}$) are functions of time n (*e.g.*, $\underline{\nu}[n]$).

at circularly contiguous positions along the transmitted vector.[1] In the context of packet-based network transmissions, packet loss due to high network traffic can result in bursty erasures. Current network protocols deal with these events by either ignoring the lost packets, or triggering retransmission mechanisms. The first method greatly affects the quality of the transmission, while the later increases the delay. Real-time multimedia data is particularly sensitive to both artifacts, and would benefit from mechanisms designed to provide improved distortion performance (than that when lost packets are ignored), at reduced delays [8, 9].

Ferreira mentions other applications for lost sample resilient complex codes including algorithm based fault-tolerant computing, where failure of a processing unit will result in lost output samples [11]. Adding redundancy to the computation in a well planned manner can enable the remaining processing units to reconstruct the lost output data. Yet another application occurs in the context of data encoding for optical disks, where erasure bursts can occur as a result of scratches or other physical damage [12]. As mentioned previously, sample reconstruction also has applications in the correction of impulse noise, where erroneous samples, once detected, can be treated as lost samples. Several examples of this approach exist in the recent literature [6,7,9]. Labeau *et al.* [6] use the analysis stage of a filterbank to arrive at a maximum-likelihood (ML) optimal impulse noise reconstruction method. Using statistics corresponding to (bursty) erasures in place of the Gaussian impulse noise model used therein would also result in an ML optimal solution to the (bursty) erasure

[1] For the case of DFT codes, circularly contiguous erasures suffer the same reconstruction problems (in the presence of quantization) as bursty erasures.

reconstruction problem.

1.2 DFT Codes

Throughout this work, particular emphasis is given to a class of complex codes known as *DFT codes* [13]. An (N, K)-DFT code maps an input message vector $\underline{x}[n]$ (K-blocked version of input sequence $x[m]$) of size K onto an output code vector $\underline{y}[n]$ of size N.[2] The mapping is done through a *DFT generator matrix* $\boldsymbol{G}$:

$$\underline{y} = \boldsymbol{G}\underline{x}.$$

As discussed in the literature [7–10], decoding algorithms for DFT codevectors exist that enable detection of as many as $\lfloor (d - E)/2 \rfloor$ errors (here E is the number of erasures) and correction of as many as $E \leq d$ erasures, where $d = N - K$.

A DFT generator matrix takes an input K-dimensional message vector and generates an N-point codevector. This is done by padding the K-point DFT of the message vector with $N - K$ zeros, and then taking the N-point IDFT. In Section 1.2.1 we present the form and qualities of DFT generator matrices in more detail, and for this purpose we now introduce some notation. We let the operator $\boldsymbol{W}_N$ represent the DFT matrix; the element at its k-th row and l-th column is (we use subscript $\langle k, l \rangle$, to denote this element, $k, l \in 0, \ldots, N-1$):

$$\boldsymbol{W}_{N\langle k,l \rangle} = \frac{1}{\sqrt{N}} \mathrm{e}^{-j2\pi kl/N}.$$

The DFT matrix $\boldsymbol{W}_N$ carries out the DFT operation on its input vector. Likewise, its hermitian transpose $\boldsymbol{W}_N^{\mathcal{H}}$ carries out the related inverse DFT (IDFT) operation; we will call $\boldsymbol{W}_N^{\mathcal{H}}$ the IDFT matrix. Note that the columns of the DFT (and IDFT) matrix are orthonormal, and its hermitian transpose equals its conjugate ($\boldsymbol{W}_N^{\mathcal{H}} = \boldsymbol{W}_N^*$).

1.2.1 Interpretation as Block Codes Over the Complex Field

The general form for a DFT generator matrix is obtained from the matrix $\boldsymbol{W}_{N \times K}^{\mathcal{H}}$ formed from a subset of K columns of the IDFT matrix $\boldsymbol{W}_N^{\mathcal{H}}$, multiplied by an arbitrary full rank

[2] DFT codes are memoryless, but also for codes with memory we drop the time indices, for simplicity, and use $\underline{x}$ and $\underline{y}$.

matrix of the form $\boldsymbol{A}\boldsymbol{W}_K$, where $\boldsymbol{A}$ is size $K \times K$ and $\boldsymbol{W}_K$ is the DFT matrix of size $K \times K$:

$$\boldsymbol{G} = \boldsymbol{W}^{\mathcal{H}}_{N\times K}\boldsymbol{A}\boldsymbol{W}_K \tag{1.1}$$

If we (i) consider the vector $\underline{\gamma}$ formed by linearly combining some columns of the (orthonormal) IDFT matrix $\boldsymbol{W}^{\mathcal{H}}_N$ and (ii) note that its DFT will result from pre-multiplication by the DFT matrix $\boldsymbol{W}_N$, it follows that the DFT of $\underline{\gamma}$ will be non-zero only at those positions corresponding to the indices of the selected columns. Referring to (1.1), the codevector

$$\begin{aligned} \underline{y} &= \boldsymbol{G}\underline{x} \\ &= \boldsymbol{W}^{\mathcal{H}}_{N\times K}\boldsymbol{A}\boldsymbol{W}_K\underline{x} \\ &= \boldsymbol{W}^{\mathcal{H}}_{N\times K}\underline{x}' \end{aligned}$$

can be seen to be a linear combination of the K columns of $\boldsymbol{W}^{\mathcal{H}}_N$ found in $\boldsymbol{W}^{\mathcal{H}}_{N\times K}$. Hence, the DFT $\underline{Y}$ of $\underline{y}$ will have a set of non-zero frequencies that we will call the *generator frequencies* or *passband frequencies*, and a set of null frequencies that we will call the *parity* or *stopband* frequencies.[3]

To summarize, the defining factor of a DFT codevector $\underline{y} = \boldsymbol{G}\underline{x}$ is that its DFT will have nulls at all parity frequencies, corresponding to the redundancy introduced by the code; the message information will be restricted to the generator frequencies. With this in mind, the simplest form of a DFT generator matrix will be:

$$\boldsymbol{G} = \boldsymbol{W}^{\mathcal{H}}_{N\times K}. \tag{1.2}$$

We will refer to this as *simple generator matrices.* We will use the term *simple, upper-parity generator matrix* for the special case when the columns comprising $\boldsymbol{G}$ are the first K columns of $\boldsymbol{W}_N$ (we use subscript $\langle *, \{0, \ldots K-1\}\rangle$ to denote all these columns):

$$\boldsymbol{G} = \boldsymbol{W}^{\mathcal{H}}_{N\langle *,\{0,\ldots K-1\}\rangle}. \tag{1.3}$$

Simple generator matrices and simple upper-parity generator matrices are used extensively throughout this work.

Another special case of $\boldsymbol{G}$ resulting in lowpass, real codevectors can be constructed for

[3] The terms *generator* and *parity* frequencies are borrowed from Rath and Guillemot [8, 9, 14]

odd K by letting $\boldsymbol{A} = \boldsymbol{I}$ and selecting the first $1 + (K-1)/2$ and last $(K-1)/2$ columns of $\boldsymbol{W}_N^{\mathcal{H}}$ to form $\boldsymbol{W}_{N\times K}^{\mathcal{H}}$ [13]:

$$\boldsymbol{G} = \boldsymbol{W}_{N\times K}^{\mathcal{H}} \boldsymbol{W}_K. \tag{1.4}$$

The resulting form for the codevector $\underline{y}$ will correspond to an oversampled version of the original message vector $\underline{x}$ [7].

To summarize, DFT generator matrices map an input K-dimensional vector $\underline{x}$ onto a K-dimensional subspace of $\mathbb{C}^N$ (spanned by K columns of $\boldsymbol{W}_N^{\mathcal{H}}$) referred to as the *generator space*. The complement of the generator space (the *parity space*) is spanned by the remaining $d = N - K$ columns of $\boldsymbol{W}_N^{\mathcal{H}}$, which we group to form the *parity matrix* of the code, $\boldsymbol{H}$. The analogy to M-ary block codes [3] can be extended in defining the *syndrome vector* $\underline{s}$, which is null for all codevectors $\underline{y}$:

$$\underline{s} = \boldsymbol{H}^{\mathcal{H}}\underline{y} \;=\; \underline{0} \quad \forall \underline{y} \text{ in code.} \tag{1.5}$$

In the presence of quantization and channel artifacts, the syndrome vector will be non-zero. As we will see later, it will be useful in the detection and correction of errors and erasures.

1.3 Contributions: Combatting Bursty Erasures with the TFBD Code

The problem with sample reconstruction in DFT codevectors is that once quantization is carried out, the error detection and sample reconstruction capabilities are diminished. Regardless of the decoding method used, this problem is greatly accentuated when the erasures occur in bursts [7, 9, 15]. One possible approach to avoid bursty erasures is to use an interleaver at the transmitter, but for the case of delay sensitive applications like multimedia transmissions, this approach might not be practical. In this work we focus on the "coding-theoretic" erasure reconstruction method presented by Rath [9] (and covered in Chapter 2), where it was also shown that the reconstruction error of DFT codevectors increases exponentially with the length of the erasure burst.

Addressing the case of bursty erasures is the intent of the proposed *Tandem Filter-Bank DFT code* (TFBD code) developed in this work. The TFBD code is a 2-D code. As shown in Fig. 1.2, we arrange a sequence of TFBD output codevectors $\underline{y}[n]$ into a two dimensional

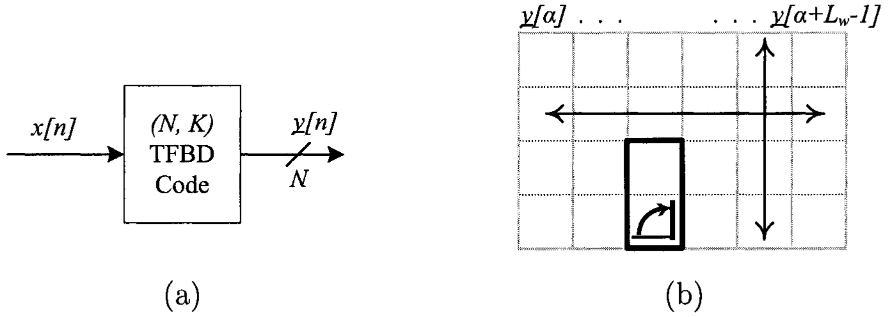

Fig. 1.2 *The TFBD frame and the subband / temporal codevectors:* Input/output relationship (a) of a TFBD encoder with code rate K/N. Output codevectors (b) are stacked to form a frame. The samples constituting a single temporal codevector and a single subband codevector are indicated by the horizontal and vertical arrows, respectively. The heavy-lined region denotes lost samples and the sample marked ◿ denotes a pivot sample.

frame (matrix). The columns of this matrix we will refer to as *subband codevectors*, the rows we will refer to as *temporal codevectors.* Given the form of the TFBD setup, the subband codevectors are DFT codevectors; we will also see that the TFBD setup allows us to treat the temporal codevectors as DFT-like codevectors. Given the dimensions $N \times L_w$ of the frame, the TFBD code provides $N + L_w$ DFT codevectors per frame to choose from in reconstructing samples. The codevectors will be *disjoint*, meaning that they can be used to reconstruct samples independently of each other. We can thus choose the reconstruction orientation of each sample (subband or temporal) in order to break up erasure bursts. We illustrate this in Fig. 1.2, where the heavy lined-block denotes an example erasure burst. Decoding both lost samples along the temporal orientation would reduce the erasure burst length to one, whereas subband reconstruction alone would imply reconstructing a burst of length two.

The previous discussion suggests the following: since subband and temporal codevectors can be used to reconstruct samples independently of each other, we can treat reconstructed samples resulting from previous reconstructions along one orientation as received samples in further reconstructions along the remaining orientation, a technique we refer to as *pivoting.* We use the term pivoting because the reconstruction orientation is rotated about a given (*pivot*) sample, and also because this pivot sample is used as a means of support in reconstructing a different erasure burst. In Fig. 1.2 we denote a pivot sample with the circular arrow symbol ↻; it corresponds to a lost sample reconstructed along the temporal orientation. Following this temporal reconstruction, the pivot is used to reconstruct the remaining lost sample along the subband orientation, where the pivot is used as a received sample.

And so, the TFBD code (i) provides a set of disjoint codevectors that one can choose from to reconstruct lost samples and (ii) allows for previous reconstructed samples (pivots) to be used to the benefit of subsequent reconstructions. The problem at hand is, in the first case, how to select the reconstruction orientation for a given sample and in the second case, how to choose pivot samples and order of reconstruction along a given frame. We provide answers to these questions by deriving performance analysis expressions for the various reconstruction operations available with the TFBD code. We first derive exact expressions (and a simplified upper bound) for the reconstruction performance of temporal codevectors. In doing so, we also derive the correlation matrix of a (windowed) subband from a uniform filterbank (further taken over an arbitrary DFT frequency band). After formally introducing the concept of pivoting, we derive exact expressions (as well as a simplified upper-bound) for the reconstruction error when using previously reconstructed samples along the subband orientation to break up bursts along the temporal orientation. These expressions can be used in designing a decoder that reduces the large reconstruction error resulting from bursty erasures by selecting both the reconstruction order and pivot samples for a given erasure burst. We illustrate this with the design of two sample codes that, depending on the dimensions of the 2-D (subband / temporal) erasure burst, can improve the reconstruction error by more than four orders of magnitude relative to the DFT code at the same rate.

1.4 Organization

The organization of the remainder of this work is as follows: We begin by presenting a survey of INEC decoding methods for DFT codes in Chapter 2. We emphasize the coding-theoretic erasure reconstruction method (*cf.* Section 2.3.1) applicable to DFT codes, as it will be the method used in TFBD erasure reconstructions. We go on to present and justify the TFBD encoder setup in Chapter 3. In Chapter 4 we derive performance measures for the temporal code and for pivoted reconstructions, along with other related expressions including the uniform filterbank subband correlation matrix mentioned above. Simulation results for all the expressions derived in Chapter 3 and Chapter 4 are presented in Chapter 5, where we also demonstrate how the derived expressions can be used as design tools in the selection of the many parameters that specify a TFBD encoder.

1.5 Notation

(For a full list of notational conventions used in this work, please refer to ***Tbl. A.1****, pg. 89.)* We use bold characters such as $\boldsymbol{G}$ to denote matrices, and underlined characters to denote column vectors (*e.g.,* $\underline{x}$). Generally, vectors will be time sequences denoted by $\underline{x}[n]$, but we will often drop the time index for convenience. The k-th entry of vector $\underline{x}$ will be referred to by $\underline{x}_{\langle k \rangle}$; the entry in the k-th row and l-th column of matrix $\boldsymbol{G}$ will be given by $\boldsymbol{G}_{\langle k,l \rangle}$. For the case of matrices $\boldsymbol{G}_{\langle k,* \rangle}$ and $\boldsymbol{G}_{\langle *,l \rangle}$ will denote the k-th row and l-th column, respectively. We will form sub-vectors and sub-rows by taking the entries from a vector/row-vector over a set of indices; this we will denote by $\underline{x}_{\langle \mathbb{I} \rangle}$ and $\boldsymbol{G}_{\langle \mathbb{I},l \rangle}$, for vectors, and $\boldsymbol{G}_{\langle k,\mathbb{I} \rangle}$ for row-vectors, where $\mathbb{I}$ denotes a set of indices. We let $W_N = \mathrm{e}^{-j2\pi/N}$, and use $\boldsymbol{W}_N$ to denote the $N \times N$ DFT matrix with (k,l) entry $\frac{1}{\sqrt{N}} W_N^{kl}$. We will also use $\mathbf{0}$ and $\underline{0}$ to denote the zero matrix and the zero vector, respectively. Superscripts $\mathcal{T}$, $\mathcal{H}$, $+$ and $*$ denote transpose ($\boldsymbol{G}^{\mathcal{T}}$), hermitian transpose ($\boldsymbol{G}^{\mathcal{H}}$), pseudoinverse ($\boldsymbol{G}^{+}$) and conjugation ($\boldsymbol{G}^{*}$) operations, respectively. Finally, we will use (N,K)*-code* to specify a code with codeword size N and input message vector size K; the rate of this code will be K/N.

Chapter 2

Survey of Complex Codes for the Impulse Noise / Erasure Channel

As mentioned in the previous chapter (*cf.* Section 1.2.1), an (N, K) DFT code is defined as having codevectors $\underline{y}$ of size N with nulls at the $d = N - K$ parity frequencies; the information is stored in the remaining K generator frequencies. We group the parity frequency components to form the syndrome vector $\underline{s}$, which will be non-zero in the presence of channel distortion. Given a message vector $\underline{x}$, the codevector will be $\underline{y} = \boldsymbol{G}\underline{x}$ and its syndrome will be $\underline{s} = \boldsymbol{H}^{\mathcal{H}}\underline{y}$, where $\boldsymbol{G}$ and $\boldsymbol{H}$ are the generator and parity matrices, respectively.

Following quantization and transmission over an INEC (Impulse Noise / Erasure Channel, *cf.* Fig. 1.1), some samples will be lost, and these we will replace by zeroes. Of those that are available, some will be erroneous (*i.e.*, errors and erasures occur at mutually exclusive positions). We can write this as follows:

$$\underline{\ddot{y}} = \underline{y} + \underline{\nu} + \underline{e} + \underline{q}, \tag{2.1}$$

where $\underline{\nu}$, $\underline{e}$, and $\underline{q}$ denote the error, erasure and quantization vectors, respectively. The quantization vector $\underline{q}$ is null at the erasure positions; at all other positions it is assumed to have uniformly distributed, i.i.d. entries with variance σ_q^2. The error vector is an impulse noise vector; only ν of its entries are non-zero, and these are assumed to occur at positions $e_0, \ldots, e_{\nu-1}$, where $e_i \in 0, \ldots N-1$. The E erasure positions will be $e_\nu, \ldots, e_{\nu+E-1}$. The erasure vector $\underline{e}$ is null at all positions except these E erasure positions, where $\underline{e}_{\langle k\rangle} = -\underline{y}_{\langle k\rangle}$.

As stated before, the N-th roots of unity will be given by $W_N^k = \mathrm{e}^{-j2\pi k/N}$, $k = 0, \ldots, N$, and for convenience, we will let

$$X_k = W_N^{e_k}, \qquad k = 0, \ldots, E + \nu - 1$$

be the roots corresponding to the error and erasure positions e_k.

In this chapter we present a brief survey of methods designed to correct the distortions of the INEC (Impulse Noise / Erasure Channel). We first present a complex field version of the Reed-Solomon polynomial-based algorithm for error and erasure correction. Then we present a related error detection method based on the ESPRIT parameter estimation method of array signal processing theory. As we will see, the error in sample reconstruction is particularly large when erasures are bursty, and this we discuss in the context of two erasure reconstruction algorithms. As motivation for the development of a novel TFBD algorithm, we present, in the last section of this chapter, the complex-field-convolutional-code interpretation of oversampled filterbanks; as an example, we discuss an error correction method that we present in the context of erasure reconstruction.

2.1 Reed-Solomon Decoding in the Complex Field

Blahut [10] states that complex field operations (*e.g.,* DFT, matrix determinants, singular value decomposition) that can be defined in terms of additions and multiplications can also be defined over finite fields, where addition and multiplication (and their inverses) also exist. He then applies this argument to present a generalized Reed-Solomon decoding algorithm applicable to the complex field, where DFT codes (with circularly contiguous parity frequencies) are seen to be the complex field equivalent of Reed-Solomon codes. In this section we summarize the decoding method presented therein, showing how to both detect error positions and reconstruct (lost or erroneous) samples in DFT codevectors.

As mentioned in Chapter 1, an intuitive divide-and-conquer attempt at tackling the problem consists in breaking it up into two steps: (i) detecting the positions of erroneous samples (for erasure reconstruction, we skip this step) and (ii) reconstructing the erroneous samples. Rather than reconstructing the erroneous samples directly, complex field Reed-Solomon codes obtain the values of the errors, and then subtract these from the received codevector.

Error Detection in the Absence of Erasures: We initially consider the case when no erasures or quantization are present:

$$\ddot{\underline{y}} = \underline{y} + \underline{\nu}. \tag{2.2}$$

We formulate the divide-and-conquer approach mathematically with the help of the *error locator polynomial*, defined as follows:

$$\begin{aligned} \Lambda(x) &= \prod_{i=0}^{\nu-1}(1 - X_i x^{-1}) && \text{(2.3a)} \\ &= \sum_{i=0}^{\nu} \Lambda_i x^{-i}, \text{ where } \Lambda_0 = 1. && \text{(2.3b)} \end{aligned}$$

We can express the error locator polynomial in N-dimensional vector form by padding the coefficients Λ_k with zeros as follows:[1]

$$\underline{\Lambda} = [\Lambda_0, \ldots, \Lambda_\nu, \mathbf{0}_{1\times N-\nu-1}]^T. \tag{2.4}$$

The use of $\underline{\Lambda}$ lies in that the k-th entry $\underline{\lambda}_{\langle k\rangle}$ of its N-point IDFT $\underline{\lambda}$ will be $\Lambda(W_N^k)$:

$$\underline{\lambda}_{\langle k\rangle} = \Lambda(W_N^k),\ k = 0, \ldots N-1. \tag{2.5}$$

Since $\Lambda(x)$ has zeros at the roots of unity X_k corresponding to the error positions (we are neglecting erasures for now), $\underline{\lambda}$ will have nulls only at those entries with indices corresponding to error positions (hence the name *error locator polynomial* for $\Lambda(x)$):

$$\underline{\lambda}_{\langle k\rangle}\underline{\nu}_{\langle k\rangle} = 0,\ k = 0, \ldots, N-1. \tag{2.6}$$

We take the DFT of this expression to write it as a circular convolution:

$$\sum_{k=0}^{N} \underline{\Lambda}_{\langle k\rangle}\underline{V}_{\langle m-k\rangle_N} = 0, \tag{2.7}$$

[1] Note that $\Lambda(x)$ we introduce specifically to define the vector $\underline{\Lambda}$, and it is the latter that is of interest in our discussion. We pad $\underline{\Lambda}$ with extra zeroes and make it an N-dimensional vector for consistency, since we are interested in its N-point IDFT $\underline{\lambda}$.

where we use subscript $\langle l \rangle_N$ to denote the entry at position l modulo N. We carry out the first step towards our solution (that of detecting the error positions) by evaluating (2.7) for several m in order to obtain a system of equations where only the $\Lambda_l, l = 1, \ldots, \nu$ are unknowns (the error positions follow from the roots of $\Lambda(x)$, *cf.* (2.3)). This is possible because, in the absence of erasures and quantization, the syndrome vector $\underline{s} = \boldsymbol{H}^{\mathcal{H}}\underline{\ddot{y}}$ is the DFT $\underline{V}$ of the error vector $\underline{\nu}$, evaluated over the set of (circularly contiguous) parity frequencies. If we let q_0 denote the first of the circularly contiguous parity frequencies, we can write this as

$$\underline{s}_{\langle k \rangle} = \underline{V}_{\langle q_0 + k \rangle}. \tag{2.8}$$

And hence, using $m = q_0 + \nu, \ldots, q_0 + d - 1$ in (2.7) will yield the desired system. The system will consist of $d - \nu$ equations, and these need to be at least as many as the ν errors, limiting the number of detectable errors to:

$$\nu \leq \lfloor d/2 \rfloor. \tag{2.9}$$

The second step towards our solution consists of obtaining the error vector $\underline{\nu}$, or equivalently, its DFT $\underline{V}$. Since we already have the error vector components over the parity frequencies (found in the syndrome vector), we need only obtain those corresponding to the generator frequencies, and these can be obtained recursively from (2.7) by expressing $\underline{V}_{\langle m \rangle}$ in terms of its ν previous components:

$$\underline{V}_{\langle m \rangle} = -\sum_{k=1}^{N} \underline{\Lambda}_{\langle k \rangle} \underline{V}_{\langle m-k \rangle_N}.$$

Error Detection with Erasures: The case when erasures are present can be addressed by converting the (partial) received codevector $\underline{\ddot{y}}$ with impulse errors and erasures into a different, complete DFT codevector $\underline{\ddot{y}}_z$ with impulse errors at the original positions and without erasures (and this we already know how to fix). The form for the received codevector will be

$$\underline{\ddot{y}} = \underline{y} + \underline{\nu} + \underline{e}. \tag{2.10}$$

Following the transformation onto a complete DFT codevector, the resulting form will be

$$\underline{\ddot{y}}_z = \underline{y}_z + \underline{\nu}_z, \tag{2.11}$$

where $\underline{y}_z$ will be a DFT codevector with reduced number of parity frequencies, and $\underline{\nu}_z$ will be an impulse noise vector with non-zero entries at the same positions as the original noise vector $\underline{\nu}$. Equation (2.11) is of the same form as (2.2), and can hence be solved accordingly.

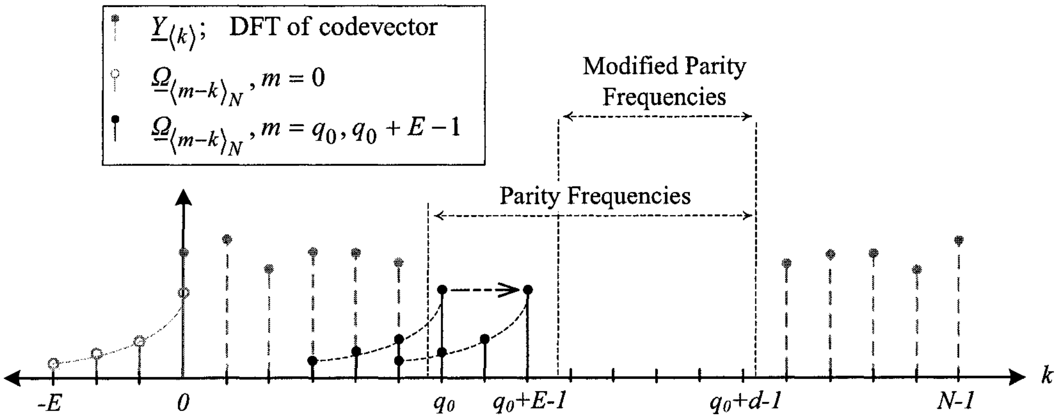
$\underline{Y}_{\langle k\rangle}$; DFT of codevector
$\underline{\Omega}_{\langle m-k\rangle_N}$, $m = 0$
$\underline{\Omega}_{\langle m-k\rangle_N}$, $m = q_0, q_0 + E - 1$
Modified Parity Frequencies
Parity Frequencies
-E
0
q_0
q_0+E-1
q_0+d-1
N-1
k

Fig. 2.1 *Error detection with erasures using the modified codevector:* Circularly convolving the erasure locator coefficients $\underline{\Omega}_{\langle m-k\rangle_N}$ with the entries $\underline{y}_{\langle k\rangle}$ of the original codevector eliminates the E parity frequencies at $k = q_0, \ldots, q_0 + E - 1$. Note the signal shapes where chosen strictly for illustrative purposes. *(Adapted from Rath and Guillemot [8].)*

We first define the *erasure locator polynomial* $\Omega(x)$ as follows:

$$\Omega(x) = \prod_{i=0}^{E-1}(1 - X_{\nu+i}x^{-1}) \tag{2.12a}$$

$$= \sum_{i=0}^{E} \Omega_i x^{-i}, \; \Omega_0 = 1 \tag{2.12b}$$

and the related vector $\underline{\Omega}$ as follows:

$$\underline{\Omega} = [\Omega_0, \ldots, \Omega_{E-1}, \mathbf{0}_{1\times N-E-1}]. \tag{2.12c}$$

We use $\underline{\omega}$ to denote the N-point IDFT vector of $\underline{\Omega}$. As before (*cf.* (2.3), (2.5)), $\underline{\omega}$ vanishes only at the erasure positions (corresponding to the only zeros of $\Omega(x)$): $\underline{e}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} = 0 \;\; \forall k$.

We multiply $\underline{\omega}$ entry-wise with $\underline{\ddot{y}}$ to obtain: [2]

$$\begin{aligned}
\underline{\ddot{y}}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} &= (\underline{y}_{\langle k\rangle} + \underline{e}_{\langle k\rangle})\underline{\omega}_{\langle k\rangle} + \underline{\nu}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} \\
\underline{\ddot{y}}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} &= \underline{y}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} + \underline{\nu}_{\langle k\rangle}\underline{\omega}_{\langle k\rangle} \qquad (2.13a) \\
\underline{\ddot{y}}_{z\langle k\rangle} &= \underline{y}_{z\langle k\rangle} + \underline{\nu}_{z\langle k\rangle} \qquad (2.13b)
\end{aligned}$$

Since (i) $\underline{\omega}$ is null only at erasure positions and (ii) error and erasure positions are mutually exclusive, one can see that $\underline{\nu}_z$ is an impulse noise vector, with errors at the same positions as $\underline{\nu}$. To see that $\underline{y}_z$ is a DFT codevector, we express its DFT $\underline{Y}_z$ as the circular convolution between $\underline{Y}$ and $\underline{\Omega}$ (*cf.* (2.7)):

$$\underline{Y}_{z\langle m\rangle} = \sum_{k=0}^{N-1} \underline{\Omega}_{\langle m-k\rangle_N}\underline{Y}_{\langle k\rangle}, \quad m = 0, \ldots, N-1. \qquad (2.14)$$

In Fig. 2.1 we have plotted the factors $\underline{\Omega}_{\langle m-k\rangle_N}$ and $\underline{Y}_{\langle k\rangle}$ in the summation of (2.14) versus k to illustrate the following point: since there are $E+1$ erasure locator coefficients Ω_l ($E+1$ contiguous non-zero elements in $\underline{\Omega}$, *cf.* (2.12b)), the resulting vector $\underline{y}_z$ will loose E DFT frequency nulls (relative to $\underline{y}$), one each for the summation indices $m = q_0, \ldots, q_0+E-1$ (the parity frequencies are assumed circularly contiguous starting at q_0). In the figure (Fig. 2.1), we plot the first and last corresponding shifted versions of $\underline{\Omega}_{\langle m-k\rangle_N}$ (those at $m = q_0$ and $m = q_0+E-1$), where the horizontal arrow denotes the remaining intermediate shift values ($m = q_0+1, \ldots, q_0+E-2$). We refer to the resulting vector $\underline{y}_z$ as the modified codevector; $\underline{\nu}_z$ will be the modified error vector. Note (*cf.* (2.13)) that $\underline{\nu}_z$ is still an impulse noise error vector with errors at the same positions as $\underline{\nu}$, and that we can extract $\underline{\nu}$ from $\underline{\nu}_z$ (using (2.13)). Since $\underline{y}_z$ will have nulls at the modified parity frequencies, $\underline{y}_z$ exists in the span of a DFT generator matrix (*cf.* Section 1.2.1), effectively creating a new received DFT codevector while removing the effect of erasures. One can build the error locator polynomial in (2.3b) and carry out the related detection and reconstruction of errors on $\underline{y}_z$ as in the case where there were no erasures, keeping in mind that, given the

[2] Here we provide a different analysis than that of [8] and [10] by presenting the decomposition of (2.13), and showing that the modified codevector $\underline{y}_z$ is a DFT codevector, while $\underline{\nu}_z$ remains an impulse noise vector.

number of modified parity frequencies $(d-E)$, at most

$$l = \lfloor (d-E)/2 \rfloor \tag{2.15}$$

errors can be corrected ($\nu \leq l$, *cf.* (2.9)).

Error Detection with Quantization: Both in the case where erasures were present and otherwise, the described algorithms ignored quantization noise. In the presence of quantization, the algorithms would be carried out in the same manner, and the resulting detected error positions and error/erasure reconstructions will only be estimates of the actual values.

2.2 Subspace Methods for Error Detection

Rath and Guillemot [8] improved the detection performance of quantized DFT codes by presenting a set of new methods for detection of erroneous samples in the presence of erasures. We group them here under the term *syndrome-based subspace methods.* The methods are based on the ESPRIT algorithm [14] used for parameter estimation in array processing theory [16]. Here we present a brief overview of syndrome-based subspace methods as introduced by Rath and Guillemot [8,14].

Assume again that ν errors are located at codevector positions $e_0, \ldots, e_{\nu-1}$, and that E erasures are located at positions $e_\nu, \ldots, e_{\nu+E-1}$. Again we let $X_k = W_N^{e_k}$ $(k = 0, \ldots, E+\nu-1)$ be the N-th roots of unity corresponding to the error and erasure positions. We define the *error locator matrix* $\boldsymbol{V}_\nu$ as follows:

$$\boldsymbol{V}_\nu = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ X_0 & X_1 & \ldots & X_{\nu-1} \\ \vdots & \vdots & \vdots & \vdots \\ X_0^{l+E} & X_1^{l+E} & \ldots & X_{\nu-1}^{l+E} \end{bmatrix} = \begin{bmatrix} \underline{v}_{e_0} & \underline{v}_{e_1} & \cdots & \underline{v}_{e_{\nu-1}} \end{bmatrix} \tag{2.16}$$

where the constant l is the maximum number of detectable errors, given also by (2.15) [8]. We note in particular that the vectors $\underline{v}_{e_k} = [1, X_k^1, \ldots, X_k^{l+E}]^T$ comprising the columns of $\boldsymbol{V}_\nu$ consist of powers of distinct roots of unity. For arbitrary roots of unity, we will use $\underline{v}_e = [1, W_N^e, \ldots, W_N^{e(l+E)}]^T$, $e = 0, \ldots, N-1$.

We define the *erasure locator matrix* $\boldsymbol{V}_E$ likewise (*cf.* (2.16)), with columns $\underline{v}_{e_k}, k = \nu, \ldots, \nu + E - 1$:

$$\boldsymbol{V}_E = \left[\begin{array}{cccc} \underline{v}_{e_\nu} & \underline{v}_{e_{\nu+1}} & \cdots & \underline{v}_{e_{\nu+E-1}} \end{array} \right] \tag{2.17}$$

The error locator matrix and the erasure locator matrix span the *error locator subspace* and the *erasure locator subspace*, respectively. Together, they span the *error-erasure subspace* $\text{span}\{[\boldsymbol{V}_\nu | \boldsymbol{V}_E]\} \subset \mathbb{C}^{l+E+1}$. Since the X_k are all distinct roots of unity, and $[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]$ is size $(l + E + 1) \times (\nu + E)$ (*i.e.*, a rectangular, tall matrix, since $\nu \leq l$), its columns correspond to rows of Vandermonde matrices [4], and are hence linearly independent. For the same reason, if we take any other vector $\underline{v}_{e_x}$ that is not a column of $[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]$, the matrix $[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E \,|\, \underline{v}_{e_x}]$ of size $(l + E + 1) \times (\nu + E + 1)$ will also have linearly independent columns. Hence any $\underline{v}_{e_x}$ will always have a non-zero component in the orthogonal complement space of the error-erasure subspace. Only the $\underline{v}_{e_k}$ (corresponding to error/erasure positions e_k) will be entirely contained in the error-erasure subspace:

$$\underline{v}_e \perp \text{span}\{[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]\}^\perp = \underline{0} \quad \text{if and only if} \quad e \in \{e_k | k = 1, \ldots, \nu + E - 1\} \tag{2.18}$$

where we let $\perp$ denote the orthogonal projection operator and use $\text{span}\{\cdot\}^\perp$ for the orthogonal complement space.

The basic idea behind all the subspace methods [8] is to find an orthonormal basis of the complement space $\text{span}\{[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]\}^\perp$, which we refer to as the *noise subspace.* We gather a corresponding orthonormal basis ($l + 1 - \nu$ vectors) to form the matrix $\boldsymbol{U}$. Using (2.18), we can find the set of error and erasure positions (and from this the error positions, since the erasure positions are assumed known) from those $e \in \{0, \ldots, N - 1\}$ satisfying:

$$\underline{v}_e^{\mathcal{H}} \boldsymbol{U} = \mathbf{0}. \tag{2.19}$$

The error-erasure subspace $\text{span}\{[\boldsymbol{V}_\nu^{(l+E+1)} \,|\, \boldsymbol{V}_E^{(l+E+1)}]\}$ is obtained from the syndrome correlation matrix $\boldsymbol{R}_s = \boldsymbol{S}\boldsymbol{S}^h$, where $\boldsymbol{S}$ is formed from the elements of the syndrome vector $\underline{s} = \boldsymbol{H}^{\mathcal{H}} \underline{y}$:

$$\boldsymbol{S} = \left[\begin{array}{cccc} \underline{s}\langle 0\rangle & \underline{s}\langle 1\rangle & \cdots & \underline{s}\langle d-E-l-1\rangle \\ \underline{s}\langle 1\rangle & \underline{s}\langle 2\rangle & \cdots & \underline{s}\langle d-E-l\rangle \\ \vdots & \vdots & \vdots & \vdots \\ \underline{s}\langle l+E\rangle & \underline{s}\langle 1+l+E\rangle & \cdots & \underline{s}\langle d-1\rangle \end{array} \right] \tag{2.20}$$

The matrix $\boldsymbol{S}$ is full rank and spans the error-erasure subspace exactly (ignoring quantization, see Appendix B for a proof). And so, a singular value decomposition of $\boldsymbol{R}_s = \boldsymbol{S}\boldsymbol{S}^{\mathcal{H}}$ will yield $\nu + E$ non-zero eigenvalues (and hence the number of errors); the corresponding eigenvectors will span the error-erasure subspace (*cf.* Appendix B). The zero eigenvalue will have order $1 + l - \nu \geq 1$, and the associated eigenvectors will span the noise subspace. Note that this justifies the dimensions selected for $\boldsymbol{S}$, as the noise subspace is guaranteed to have at least dimension one (as long as $\nu \leq l$) and can thus be used to find the ν roots of unity with corresponding $\underline{v}_e$ satisfying (2.19).

In the presence of quantization noise, it is only possible to obtain an approximation of $\boldsymbol{U}$ in (2.19), and the resulting estimate for the set of error positions can be taken as the ν indices e ($e \in 1, \ldots, N$) that minimize $|\underline{v}_e^{\mathcal{H}} \boldsymbol{U}|$. In this situation, the eigenvalues corresponding to the noise subspace will no longer be zero. Rather, a threshold needs to be defined that distinguishes between low-valued singular values corresponding to the noise subspace and high-valued singular values corresponding to the error-erasure subspace. The detection performance will no longer be exact. It will be a function of the disparities between the noise subspace eigenvalues and those of the error-erasure subspace, and better detection performance will occur at higher error-to-quantization-noise power ratios (*i.e.*, larger error magnitudes).

2.3 Erasure Reconstruction and the Bursty Erasure Phenomenon

In this section we present two methods used to reconstruct lost samples in DFT codevectors. The methods are particularly interesting because they lend themselves intuitively to an explanation of the bursty erasure phenomenon. As mentioned previously, quantization noise affects the erasure reconstruction capabilities of DFT codes. Bursty erasures in particular will result in reconstruction errors with magnitudes that grow exponentially with the length of the burst [9]. We refer to this explosion in the reconstruction error as the *bursty erasure phenomenon.*

We first present the coding-theoretic approach to sample reconstruction [9]: the parity matrix $\boldsymbol{H}$ is used in deriving a matrix $\boldsymbol{T}$ that will reproduce the erased samples given those received. A geometrical interpretation of both the encoding process ($\underline{y} = \boldsymbol{G}\underline{x}$) and the reconstruction process will provide one explanation for the bursty erasure phenomenon. The coding-theoretic reconstruction method will be the one used in the Tandem Filter-Bank

DFT (TFBD) code that is the focus of this thesis, and we will see that the geometrical interpretation provides justification for the TFBD setup. We next present a systems theory based reconstruction approach [7] where the lost samples are obtained as the zero-input solution to a difference equation. The syndrome vector will serve as the set of initial conditions of the system. Using this systems theory approach, the bursty erasure phenomenon will be explained in terms of the clustering of zeros in the transfer function of the resulting solution.

We will let E denote the number of erasures in a received codevector; R will be the number of received samples. Since we ignore error detection in our discussion, the positions of the erasures along the received codevector will be denoted by $e_0, \ldots, e_{E-1}$. The resulting form for the received vector will be:

$$\ddot{\underline{y}} = \underline{y} + \underline{e} + \underline{q} \tag{2.21}$$

2.3.1 Coding-Theoretic Erasure Reconstruction

We can demonstrate the *coding theoretic* erasure recovery process when using DFT codes by considering the null syndrome constraint of any DFT code vector: $\boldsymbol{H}^{\mathcal{H}}\underline{y} = \underline{0}$. For notational simplicity, we will use $\boldsymbol{C} = \boldsymbol{H}^{\mathcal{H}}$ to rewrite the null syndrome constraint as:

$$\boldsymbol{C}\underline{y} = \underline{0}. \tag{2.22}$$

We group all the erased samples from $\underline{y}$ to form vector $\underline{y}_E$ ($\underline{y}_{E\langle k\rangle} = \underline{y}_{\langle e_k\rangle}$); we group the received samples to form $\underline{y}_R$ (using r_k to denote the k-th received position, $\underline{y}_{R\langle k\rangle} = \underline{y}_{\langle r_k\rangle}$, *cf.* **Tbl. A.1**, pg. 89). We break up $\boldsymbol{C}$ respectively so that $\boldsymbol{C}_E$ consists of those columns of $\boldsymbol{C}$ with indices corresponding to erasure positions ($\boldsymbol{C}_{E\langle *,k\rangle} = \boldsymbol{C}_{\langle *,e_k\rangle}$), and $\boldsymbol{C}_R$ consists of those with indices corresponding to received positions ($\boldsymbol{C}_{R\langle *,k\rangle} = \boldsymbol{C}_{\langle *,r_k\rangle}$). Using this, we can rewrite (2.22) as follows:

$$\begin{bmatrix} \boldsymbol{C}_E & \boldsymbol{C}_R \end{bmatrix} \begin{bmatrix} \underline{y}_E \\ \underline{y}_R \end{bmatrix} = \underline{0}. \tag{2.23}$$

Whenever $E < d = N - K$, $\underline{y}_E$ can be recovered from $\underline{y}_R$ by rearranging terms as follows:

$$\begin{aligned} \underline{y}_E &= -\boldsymbol{C}_E^{+}\boldsymbol{C}_R\underline{y}_R = \boldsymbol{T}\underline{y}_R, && \text{(2.24a)} \\ \text{where } \boldsymbol{T} &= -\boldsymbol{C}_E^{+}\boldsymbol{C}_R. && \text{(2.24b)} \end{aligned}$$

Here we have used the plus superscript $(\cdot)^+$ to denote the pseudo-inverse. Referring to (2.21), the actual vector $\ddot{\underline{y}}_R$ available to the decoder is a quantized version of $\underline{y}_R$. In this case, $\boldsymbol{T}\ddot{\underline{y}}_R$ will provide an estimate $\hat{\underline{y}}_E$ of the lost samples. We model the quantization using $\underline{q}$; samples of $\underline{q}$ at received positions will yield vector $\underline{q}_R$ (i.i.d. entries, uniform distribution with mean zero and variance σ_q^2). We can write the samples $\hat{\underline{y}}_{E\langle k\rangle}$ of $\hat{\underline{y}}_E$ and their related mean square error (*code sample mean square error* at the k-th erasure position e_k, or $\Psi(e_k)$) as follows [9], where $\boldsymbol{t}_k$ denotes the k-th row of $\boldsymbol{T}$:

$$\begin{aligned} \hat{\underline{y}}_{E\langle k\rangle} &= \boldsymbol{t}_k\ddot{\underline{y}}_R && \text{(2.25)} \\ &= \boldsymbol{t}_k(\underline{y}_R + \underline{q}_R) \\ &= \underline{y}_{E\langle k\rangle} + \boldsymbol{t}_k\underline{q}_R \\ & && \text{(2.26)} \end{aligned}$$

The resulting code sample mean square error will be:

$$\begin{aligned} \Psi(e_k) &= \mathrm{E}\big[|\hat{\underline{y}}_{E\langle k\rangle} - \underline{y}_{E\langle k\rangle}|^2\big] && \text{(2.27)} \\ &= \mathrm{E}\big[\boldsymbol{t}_k\underline{q}_R\underline{q}_R^{\mathcal{H}}\boldsymbol{t}_k^{\mathcal{H}}\big] \\ &= \sigma_q^2|\boldsymbol{t}_k|^2 && \text{(2.28)} \end{aligned}$$

Rath and Guillemot [9] prove that $|\boldsymbol{t}_k^{\mathcal{H}}|^2$ (and hence also $\Psi(e_k)$, *cf.* (2.28)) grows exponentially with the length of the erasure burst. This we refer to as the *bursty erasure phenomenon.* By interpreting the encoding operation $\underline{y} = \boldsymbol{G}\underline{x}$ as the projection of $\underline{x}$ onto the rows $\boldsymbol{g}_k$ of $\boldsymbol{G}$, we can obtain an intuitive explanation for this phenomenon, as we now show.

2.3.2 Geometrical Interpretation of the Bursty Erasure Phenomenon

In this section we present a geometrical explanation for the bursty erasure phenomenon [9]. For the purpose of illustration, we consider a (5, 2) real-valued DFT code. The available samples are those at positions 0 and 1, samples 2 through 4 need to be reconstructed.

Referring to Fig. 2.2, the encoding operation $\underline{y} = \boldsymbol{G}\underline{x}$ can be interpreted as the projections g_k in $\mathbb{C}^K$ of the message vector $\underline{x}$ onto the lines defined by the rows $\boldsymbol{g}_k$ of $\boldsymbol{G}$:

$$g_k = \boldsymbol{g}_k\underline{x}/|\boldsymbol{g}_k^{\mathcal{H}}|$$

In the absence of quantization, any K or more projections (*i.e.,* the received samples $\underline{y}_R$) will specify $\underline{x}$ exactly, and this will yield the remaining lost samples $\underline{y}_E$. Note that this reconstruction is carried out directly by (2.24a). In the presence of quantization, the projections g_k are known to exist in the interval $\ddot{g}_k \pm \Delta/2$, where the $\ddot{g}_k$ are the available projections, and Δ is the quantization step-size. Rather than specifying $\underline{x}$ exactly, the intervals $\ddot{g}_k \pm \Delta/2$ intersect at a parallelogram $\mathbb{P}$ containing $\underline{x}$. The resulting projection of $\mathbb{P}$ onto rows corresponding to lost samples will be larger (*e.g.,* a diagonal of $\mathbb{P}$) than Δ. Given the constraints on $\boldsymbol{G}$ imposed by DFT codes, the $\boldsymbol{g}_k$ will be positioned in space such that bursty erasures in particular will both increase the lengths of diagonals and align them to rows corresponding to lost samples [9].

The previous discussion motivates the search for a method to break up erasure bursts by obtaining a better estimate for some or all lost samples. We illustrate the idea behind such a method geometrically in Fig. 2.2: The original received samples correspond to the projections $\ddot{g}_0$ and $\ddot{g}_1$, where Δ is the quantization bin size; the original message vector $\underline{x}$ is thus known to exist in $\mathbb{P}$. If we somehow were to obtain a better estimate for one (or more) of the lost samples (in this case $\hat{g}_{4\dagger}$) than that associated with the projection of $\mathbb{P}$ onto the corresponding row, we could use the new estimate to (statistically) crop $\mathbb{P}$, thus reducing the uncertainty in the remaining reconstructions ($\hat{g}_3$ in the figure). In Chapter 4 we will see how the TFBD code can produce such estimates using a technique that we refer to as pivoting.

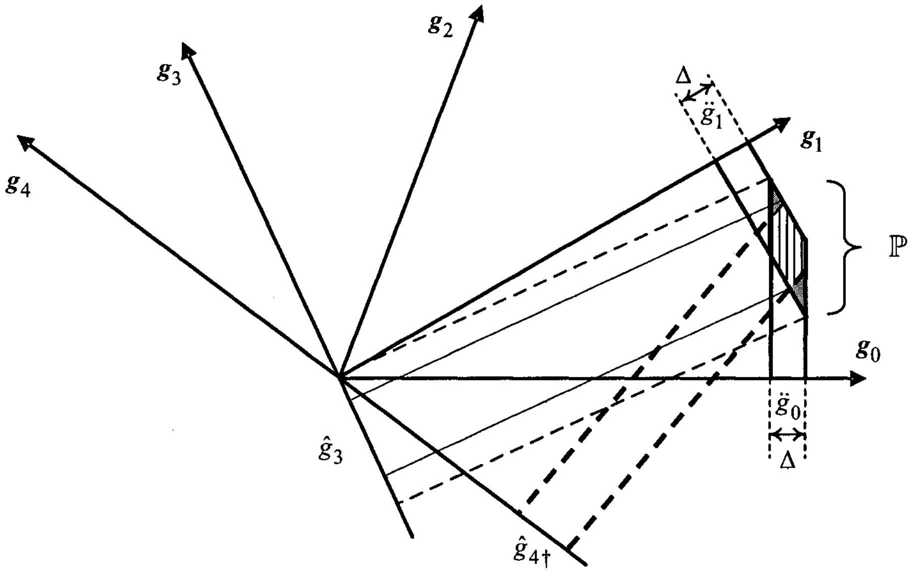

Fig. 2.2 *Geometrical interpretation of bursty erasure phenomenon:* Obtaining a better estimate $\hat{g}_{4\dagger}$ for projection g_4 reduces the dimensions of $\mathbb{P}$, thus improving all remaining projections. *(Adapted from Rath and Guillemot [9].)*

2.3.3 Systems Theory Interpretation of the Bursty Erasure Phenomenon

Marvasti [7] presents another facet of the erasure correction capabilities of DFT codes in the form of the BERT (*Burst Error Recovery Technique*) algorithm. The method is based on the error locator polynomial [8,10] given by (2.12). After showing that lost samples can be obtained from the zero-input response of a system described by a linear constant coefficient difference equation (LCCDE), the explosive magnitude of the reconstruction error in the presence of bursty erasures is explained in the context of systems theory.

In Section 2.1 we presented a complex field Reed-Solomon decoder. The decoding process was carried out in two steps, the first yielding the positions of the errors and the second their value. This same method can be adapted to develop a new erasure reconstruction method, with the simplification that the positions of the erasures are known. The information regarding their positions can be expressed using the erasure locator polynomial and its equivalent vector representation $\underline{\Omega}$ (given by (2.12b)), and we wish to obtain the DFT $\underline{E}$ of the erasure vector $\underline{e}$. We can substitute $\underline{\Omega}$ and the DFT $\underline{E}$ of the erasure vector in place of $\underline{\Lambda}$ and $\underline{V}$ (respectively) in (2.7) to write

$$\sum_{k=0}^{N} \underline{\Omega}_{\langle m-k\rangle_N} \underline{V}_{\langle k\rangle} = 0, \quad \Omega_0 = 1$$

Using the commutative property of the circular convolution, we rewrite this as

$$\begin{aligned} \sum_{k=0}^{E} \underline{\Omega}_{\langle k\rangle} \underline{V}_{\langle m-k\rangle_N} &= 0, \quad \Omega_0 = 1, \quad \text{and rearranging,} && (2.29) \\ \underline{E}_{\langle m\rangle} &= -\sum_{k=1}^{E} \underline{\Omega}_{\langle k\rangle} \underline{E}_{\langle m-k\rangle_N}, \quad m = 0, \ldots, N-1, && (2.30) \end{aligned}$$

where we have used the fact that $\underline{\Omega} = [\Omega_0, \ldots, \Omega_E, \mathbf{0}_{1\times N-E-1}$, *cf.* (2.12). We note that (2.30) corresponds to a linear constant coefficient difference equation (LCCDE) defined by the non-zero taps Ω_k (first E entries of $\underline{\Omega}$). Given any E continuous values of $\underline{E}_{\langle l\rangle}$, we can obtain the remaining values from the zero-input response of this LCCDE. Since $\underline{y}$ is a DFT codevector, we have available at most $d = N - K$ initial conditions in the form of the syndrome $\underline{s}$. Hence, for the system in (2.30) to be solvable, it must be of order at most d (at most $d + 1$ taps $\underline{\Omega}_k$).[3] Since there are $E + 1$ taps Ω_k, we write $E \leq d$ and hence d is the maximum number of reconstructions possible with this method. Note that with the coding theoretic method discussed in Section 2.3.1, the maximum number of reconstructions is $K = N - d$. We can solve $d > N - d$ to see that Marvasti's BERT method [7] (the one discussed presently) will correct more (less) errors whenever $N > 2K$ ($N < 2K$). At higher code rates, the coding theoretic method discussed in Section 2.3.1 will be capable of reconstructing more lost samples.

To get new insight into the bursty erasure phenomenon (*cf.* Section 2.3), we group the taps Ω_k in the (IIR) system of (2.30) to form the (FIR) filter $h[k] = \Omega_k$, $k = 1, \ldots, E$ [7]. The z-transform of $h[k]$ will be given by:

$$H(z) = \sum_{k=0}^{E} h[k] z^{-k} = \sum_{k=0}^{N-1} \Omega_k z^{-k}$$

[3] This discussion on the maximum number of correctable erasures and comparison to the complex field Reed-Solomon method of Section 2.1 is not given by Marvasti [7].

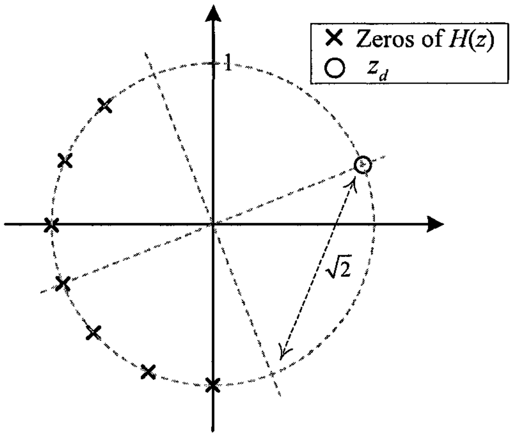

Fig. 2.3 *Zero clustering interpretation of bursty erasure phenomenon:* Zero plot of $H(z)$ for the case of bursty erasures. The point z_d is diametrically opposite to the cluster of zeros.

Referring to (2.12b) we can see that $H(z) = \Omega(z)$ and hence $H(z)$ has zeros at the E roots of unity corresponding to the erasure positions, and these will be contiguous when erasures are bursty. A heuristic explanation [7] for the bursty erasure phenomenon can be obtained by considering the zero plot of $H(z)$ for bursty erasures (see the sample in Fig. 2.3). Assuming an (N, K) DFT code with $N < 2K$, the zeros of $H(z)$ will be clustered on the same half of the complex plane. If we were to evaluate $H(z)$ on the point z_d on the unit circle diametrically opposite to the cluster of zeros, a lower bound on its magnitude would be $(\sqrt{2})^E$, which grows exponentially with the length of the burst (*cf.* Fig. 2.3). To see how this will affect the solution to (2.30), we write the z-transform of the solution to (2.30) as follows [7]:

$$E(z) = \frac{\sum_{k=1}^{E}\sum_{r=1}^{k} h[k]\underline{E}_{\langle r\rangle} z^{k-r+1}}{1+\sum_{k=1}^{E} h[k] z^k} \tag{2.31}$$

In the presence of quantization, the zeros and poles of the filter will be given by $z_i + \Delta z_i$ and $z_p + \Delta z_p$, respectively. The Δz_i and Δz_p represent the error in the position of the zeros and poles resulting from the error $\Delta\underline{E}_{\langle k\rangle}$ in the $\underline{E}_{\langle k\rangle}$ used for the initial state of the filter (corresponding to the syndrome coefficients). Both Δz_i and Δz_p can be expressed in terms of the $\Delta\underline{E}_{\langle k\rangle}$ using differential calculus. The results [7] are:

$$\Delta z_i = \sum_{r=1}^{E}\left(\frac{-\sum_{k=1}^{E} h[k] z_i^{k-r+E}}{\prod_{\substack{l=1\\ l\neq i}}^{E}(z_i - z_l)}\Delta\underline{E}_{\langle r\rangle}\right) \tag{2.32a}$$

$$\Delta z_p = \sum_{r=1}^{E}\left(\frac{-z_p^{E-r}}{\prod_{\substack{l=1\\ l\neq i}}^{E}(z_p - z_l)}\Delta h[r]\right) \tag{2.32b}$$

Based on our previous discussion, the values for $h[n]$ can be very large in the presence of bursty erasures ($|H(z)|$ grew faster than exponentially on the point z_d, *cf.* Fig. 2.3), and we see that these parameters appear in the numerators of (2.32). As a result, we can expect large errors in the LCCDE solution $E(z)$ and the related erasure estimates.

2.4 Filterbanks as Convolutional Codes Over the Complex Field

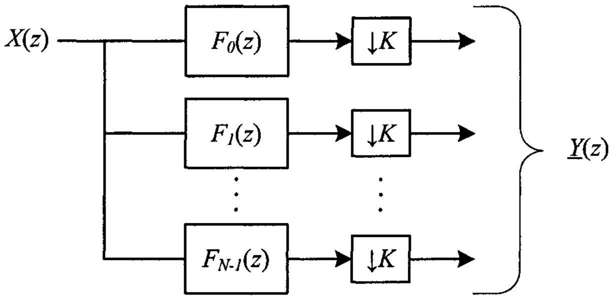

Fig. 2.4 *Filterbank analysis stage:* Standard analysis stage of an oversampled filterbank ($N > K$).

Consider the analysis stage of the oversampled filterbank of Fig. 2.4. We let $x[m]$, $f_i[n]$, and $\underline{y}[n]$ represent the time domain versions of $X(z)$, $F_i(z)$ and $\underline{Y}(z)$ respectively. The l-th polyphase component $f_i^l[n]$ [17, 18] of the filter $f_i[n]$ is obtained by left-shifting the filter by l samples and keeping only every K-th sample:

$$f_i^l[n] = f_i[nK + l]$$

We can use the K resulting polyphase components $f_i^l[n]$ ($l = 0, \ldots, K-1$) thus defined to reconstruct the original filter, where we use $\left[f_i^l[n]\right]_{(\uparrow K)}$ to denote $f_i^l[n]$ upsampled by K:

$$f_i[n] = \sum_{l=0}^{K} \left[f_i^l[n-l]\right]_{(\uparrow K)}$$

Equivalently in the z-domain:

$$F_i(z) = \sum_{l=0}^{K-1} F_i^l(z^K) z^{-l} \tag{2.33}$$

By grouping all the polyphase components $F_i^l(z)$ into the *polyphase matrix* $\boldsymbol{G}(z)$, we can express (2.33) in matrix form as follows:

$$\boldsymbol{G}(z) = \begin{bmatrix} F_0^0(z) & \dots & F_0^{K-1}(z) \\ \vdots & \ddots & \vdots \\ F_{N-1}^0(z) & \dots & F_{N-1}^{K-1}(z) \end{bmatrix} \Rightarrow \underbrace{\begin{bmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_{N-1}(z) \end{bmatrix}}_{\underline{F}(z)} = \boldsymbol{G}(z^k) \underbrace{\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-K+1} \end{bmatrix}}_{\underline{D}_K(z)} \tag{2.34}$$

Here $\underline{F}(z)$ is the vector composed of all the filters $F_i(z)$, and $\underline{D}_K(z)$ is the vector composed of delays. Using this, we redraw the filterbank block diagram in Fig. 2.4 in matrix form as illustrated in Fig. 2.5. The first noble identity [17, 18] establishes a useful equivalent representation for multi-rate systems involving decimators (see its matrix form in Fig. 2.6). Applying it to Fig. 2.5 yields the form shown in Fig. 2.7 for our oversampled filterbank. This form is the polyphase representation of the filterbank, and it allows us to represent the filterbank transfer function using the following matrix equation:

$$\underline{Y}(z) = \boldsymbol{G}(z)\underline{X}(z) \tag{2.35}$$

The time domain vector sequence $\underline{x}[n]$ corresponding to $\underline{X}(z)$ will be the K-blocked version of the message sequence $x[m]$:

$$\underline{x}_{(k)}[n] = x[nK + k]$$

The mapping from $x[m]$ to its K-blocked version $\underline{x}[n]$ is done by means of the operator $\underline{D}_K(z)$ (*cf.* (2.33)).

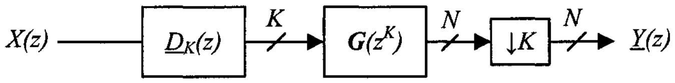

Fig. 2.5 *Matrix form for oversampled filterbank.*

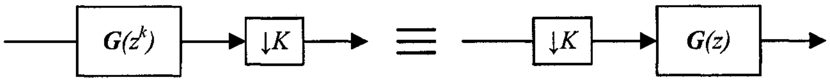

Fig. 2.6 *Matrix form of first noble identity.*

The expression of (2.35) is a generalization of the complex field block-code $\underline{y} = \boldsymbol{G}\underline{x}$ seen previously. While block-codes encode a K-dimensional vector $\underline{x}$ to an N-dimensional vector $\underline{y}$, (2.35) encodes the corresponding vector sequence $\underline{x}[n]$ onto the vector sequence $\underline{y}[n]$. If we assume the filters $F_i(z)$ to be FIR, the polyphase matrix $\boldsymbol{G}(z)$ will also be FIR; we can write $\boldsymbol{G}(z)$ as follows:

$$\boldsymbol{G}(z) = \sum_{l=0}^{L_g-1} \boldsymbol{G}_l z^{-l}$$

Using this, we express the encoding operation (2.35) in the time domain as follows:

$$\underline{y}[n] = \sum_{l=0}^{Lg-1} \boldsymbol{G}_l \underline{x}[n-l] \tag{2.36}$$

The expression is reminiscent of those representing convolutional codes in finite fields [3], and we can say that (2.36) is the complex field extension of the finite field convolutional codes used extensively in digital communications.

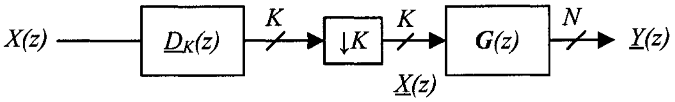

Fig. 2.7 *Polyphase form for oversampled filterbank.*

2.4.1 Erasure Correction Using Filterbanks

We saw in the previous section how filterbanks can be interpreted as convolutional codes over the complex field. Labeau *et al.* [6] derive a decoder for such a code that addresses the distortions of the INEC, and we cover it in the present section. The method was presented in the context of impulse noise detection and correction, but we discuss it here in the context of sample reconstruction. The resulting algorithm will be optimal in the mean square sense. However, derivations where provided [6] for the PDF of the impulse error magnitude given the available information, and it seems promising to adapt this method using statistics corresponding to (bursty) erasures to obtain a maximum-likelihood (bursty) erasure reconstruction method.

Since the generator matrix $\boldsymbol{G}(z)$ in (2.36) is rectangular (assumed full column rank almost everywhere), we can define the related *parity matrix* $\boldsymbol{H}(z)$ similarly to our previous definition as spanning the null-space of $\boldsymbol{G}(z)$ [6]:

$$\boldsymbol{H}(z)^{\mathcal{H}}\boldsymbol{G}(z) = \boldsymbol{0}.$$

The syndrome definition follows likewise as $\underline{S}(z) = \boldsymbol{H}^{\mathcal{H}}(z)\underline{Y}(z)$, which in the absence of errors or quantization, will be the zero vector $\underline{0}$. We express the z-transform of the received vector sequence $\ddot{\underline{Y}}(z)$ and its syndrome (where we use $\boldsymbol{C}(z) = \boldsymbol{H}^{\mathcal{H}}(z)$) as follows:

$$\begin{aligned} \ddot{\underline{Y}}(z) &= \underline{y}(z) + \underline{e}(z) + \underline{q}(z), && (2.37)\\ \underline{s}(z) &= \boldsymbol{C}(z)\underline{e}(z) + \boldsymbol{C}(z)\underline{q}(z) && (2.38)\\ &= \boldsymbol{C}(z)\underline{e}(z) + \underline{q}'(z). && (2.39) \end{aligned}$$

We assume $\boldsymbol{C}(z)$ to have entries that are FIR of length at most L_c, and express it as

$$\boldsymbol{C}(z) = \sum_{k=0}^{L_c-1} \boldsymbol{C}_k z^{-k},$$

where the $\boldsymbol{C}_k$ consist of constant values. We use $\boldsymbol{C}_{k\langle *,l\rangle}$ to denote the l-th column of $\boldsymbol{C}_k$. Following the work of Labeau [6], we only consider a single lost sample $e = -\underline{y}_{\langle m\rangle}[l]$ affecting the m-th subband at time l. In the time domain, the resulting form for (2.39) can be expressed as:

$$\underline{s}[l+k] = -\boldsymbol{C}_{k\langle *,m\rangle} e + \underline{q}'[l+k], \qquad k = 0, \ldots, L_c - 1$$

where we note that, given the FIR nature of $\boldsymbol{C}(z)$ and the position of the erasure, only the syndromes at times $l, \ldots, l + L_c - 1$ will be affected. We can stack these to form $\underline{S}[l]$:

$$\underline{S}[l] = \begin{bmatrix} \underline{s}[l] \\ \underline{s}[l+1] \\ \cdots \\ \underline{s}[l+L_c-1]^{\mathcal{T}} \end{bmatrix} = -e \begin{bmatrix} \boldsymbol{C}_{0\langle *,m\rangle} \\ \boldsymbol{C}_{1\langle *,m\rangle} \\ \vdots \\ \boldsymbol{C}_{L_c-1\langle *,m\rangle} \end{bmatrix} + \begin{bmatrix} \underline{q}'[l] \\ \underline{q}'[l+1] \\ \vdots \\ \underline{q}'[l+L_c-1] \end{bmatrix} \tag{2.40}$$

The mean square estimate for the erased sample e follows directly by left multiplying $\underline{S}[l]$ by the pseudoinverse of $[\boldsymbol{C}_{0\langle *,m\rangle}|\boldsymbol{C}_{1\langle *,m\rangle}|\ldots|\boldsymbol{C}_{L_c\langle *,m\rangle}]^{\mathcal{T}}$.

2.5 Summary

We began this chapter by showing how DFT codes could be interpreted as the complex field extension of discrete Reed-Solomon codes. We then presented several ways in which one could use DFT codes to detect and correct errors, as well as reconstruct erasures. For completeness, we went on to present subspace-based methods of error detection. We then presented several erasure reconstruction methods and showed how for the case when erasures where contiguous (bursty erasures), the reconstruction error was very large. Finally, we presented an erasure reconstruction method based on filterbanks and showed how such a transmitter could be interpreted as a complex-field convolutional code.

In the next chapter we introduce the proposed *Tandem Filterbank / DFT code* (TFBD code) that is the focus of this thesis. The code aims at preserving the decoding simplicity and resourcefulness of DFT codes, while improving their reconstruction abilities in the presence of bursty erasures. As we will see, the TFBD code will be a 2-D code: a TFBD frame can be interpreted as being composed of DFT codevectors along its rows and columns. By judiciously choosing the DFT codevector used for a particular reconstruction, one can avoid bursty erasures and thus achieve the desired improvement in reconstruction error.

Chapter 3

Proposed Tandem Filterbank / DFT Generator Matrix Setup

We now begin our discussion of the Tandem Filterbank / DFT code (TFBD code), designed and analyzed as the main focus of this thesis. The TFBD code is meant to improve the bursty erasure reconstruction performance of quantized DFT codes, and sample reconstruction (as opposed to error detection) is thus the main focus throughout this and the following chapters. However, as we will see, all the decoding (error and erasure correction) methods applicable to DFT codevectors (*cf.* Chapter 2) also apply to TFBD codevectors.

We begin this chapter by presenting the TFBD encoder setup and deriving the related transfer function. We then justify the setup by explaining the many tools it exposes in dealing with bursty erasures. In the next chapter, we will carry out a mathematical analysis of the capabilities of these tools.

3.1 Tandem Filterbank / DFT Encoder Setup

The TFBD encoding process is illustrated in Fig. 3.1 and Fig. 3.2 (we show their equivalence in Section 3.2). From Fig. 3.1, it is straightforward to see that the spectral shape of subbands comprising $\underline{y}'[n]$ will be the same. Linear combination in the form of $\boldsymbol{G}$ will not alter this spectral shape in the resulting subbands of $\underline{y}[n]$. From Fig. 3.2, one can see the relationship between the TFBD encoder and the standard filterbank analysis stage of Fig. 2.4.

The encoding process can be broken up into two stages: the first stage maps the input sequence $x[n]$ to the vector sequence $\underline{y}'[n]$. Since there are K_b subbands comprising vector $\underline{y}'[n]$, and each subband is downsampled by a factor of K, the expansion factor for this stage is K_b/K, where $K_b > K$. The second stage linearly combines the subbands in $\underline{y}'[n]$ via a DFT generator matrix $\boldsymbol{G}$ of size $N \times K_b$, thus forming the output vector sequence $\underline{y}[n]$. Given the dimensions of $\boldsymbol{G}$, the expansion factor for this stage is N/K_b, for a net system code rate of K/N. We will refer to this as an (N, K_b, K) TFBD code, where we emphasize the following relationship:

$$\boxed{N > K_b > K} \tag{3.1}$$

In the following section of this chapter we will derive the TFBD (Tandem Filterbank / DFT) transfer function. We will use it in the following chapter (Chapter 4) to derive expressions for the performance of the TFBD code.

3.2 TFBD Transfer Function

The transfer function from $x[n]$ to the l-th entry $\underline{y}'_{\langle l\rangle}[n]$ of $\underline{y}'[n]$ can be obtained by inspection from Fig. 3.1, where L_f is the length of the prototype filter $f[n]$, and subscript $(\downarrow K)$ denotes downsampling by K:

$$\begin{aligned} \underline{y}'_{\langle l\rangle}[n] &= \left[(Kx[n]W_{K_b}^{-ln}) * f[n]\right]_{(\downarrow K)} && (3.2)\\ &= K\sum_{a=0}^{L_f-1} f[a]\,(x[nK-a]W_{K_b}^{-l(nK-a)}) \qquad \forall l \in [0,\ldots,K_b-1] && (3.3)\end{aligned}$$

This can be rearranged to obtain:

$$\underline{y}'_{\langle l\rangle}[n] = KW_{K_b}^{-lKn}\sum_{a=0}^{L_f-1}(f[a]W_{K_b}^{al})\,x[nK-a]. \tag{3.4}$$

Here the term in parentheses corresponds to the l-th analysis filter $f_l[a]$ in Fig. 3.2. We substitute to get:

$$\underline{y}'_{\langle l\rangle}[n] = KW_{K_b}^{-lKn}\sum_{a=0}^{L_f-1} f_l[a]\,x[nK-a] \tag{3.5}$$

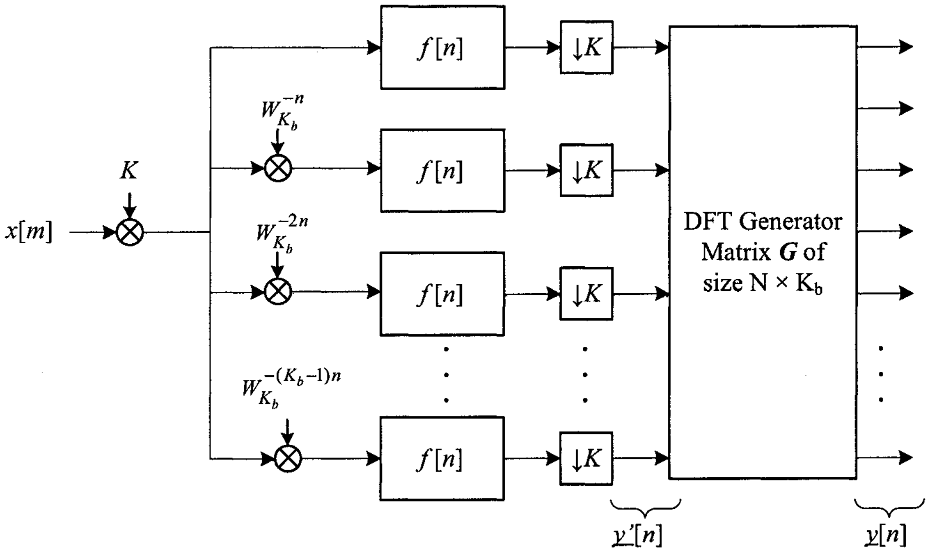

Fig. 3.1 *Proposed TFBD encoder:* The input sequence is modulated so that subsequent frequency bands are brought down to the baseband, followed by lowpass filtering with $f[n]$, downsampling, and DFT encoding.

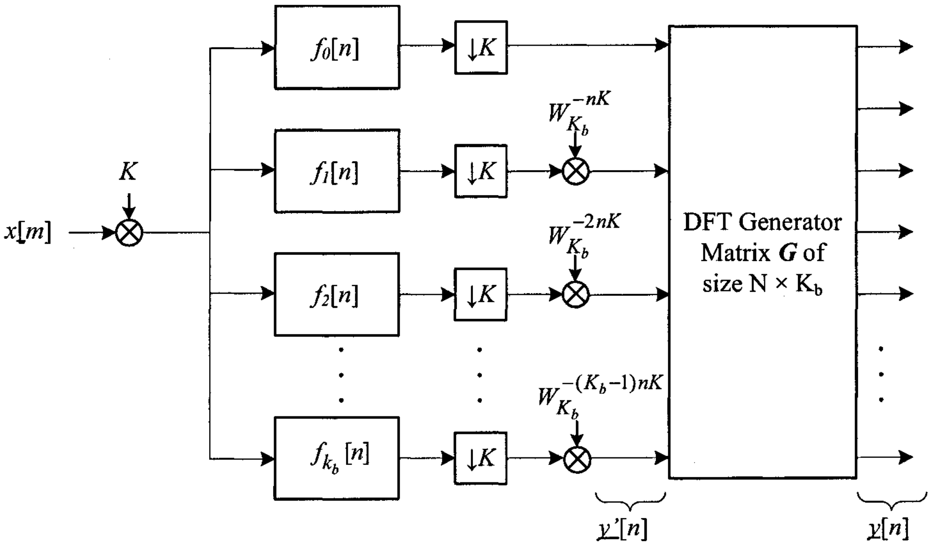

Fig. 3.2 *Equivalent form for TFBD encoder:* The analysis filters $f_l[n] = f[n]W_{K_b}^{ln}$ are modulated versions of the prototype filter $f[n]$ in Fig. 3.1.

where the summation and scaled time index 'nK' represent the subband filtering and downsampling operations in Fig. 3.2, respectively, and the complex exponential represents the subband modulation. Thus we can see that the two configurations given in Fig. 3.1 and Fig. 3.2 are equivalent. We can use (3.5) to write an expression for the entries $\underline{y}_{\langle k \rangle}[n]$ of $\underline{y}[n]$:

$$\underline{y}_{\langle k \rangle}[n] = \sum_{l=0}^{K_b-1} \boldsymbol{G}_{\langle k,l \rangle} \underline{y}'_{\langle l \rangle}[n] \tag{3.6a}$$

$$\boxed{\underline{y}_{\langle k \rangle}[n] = K \sum_{l=0}^{K_b-1} \sum_{a=0}^{L_f-1} \boldsymbol{G}_{\langle k,l \rangle} f_l[a] x[nK-a] W_{K_b}^{-lKn}} \tag{3.6b}$$

In Chapter 4 we will use equation (3.6b) in deriving performance measures for the reconstruction capabilities of the TFBD code.

3.3 Justification for TFBD Encoder Setup

For our prototype filter $f[n]$ we will use a lowpass filter with a nominal cutoff frequency $1/2K_b$, where we have normalized the $[0, 2\pi]$ angular frequency axis to the interval [0, 1]; we will refer to this normalized axis as the f-axis. This cutoff frequency is chosen because it divides the spectrum of $x[n]$ into K_b portions of equal length, each corresponding to one component of $\underline{y}'[n]$. Following the downsampling operation, the subband components $\underline{y}'_{\langle k \rangle}[n]$ in Fig. 3.1 will have a maximum frequency of $K/2K_b$. We are particularly interested in the $\underline{y}_{\langle k \rangle}[n]$, and since these are linear combinations of the $\underline{y}'_{\langle k \rangle}[n]$, they will retain the lowpass shape with maximum frequency $K/2K_b$. We use $\mathbb{I}_p$ to denote the passband (support) of $\underline{y}_{\langle k \rangle}[n]$ and $\mathbb{I}_s$ to denote its stopband, as indicated in Fig. 3.3. The length of this passband (we use $|\mathbb{I}|$ to denote the length of interval $\mathbb{I}$) will be given by:

$$|\mathbb{I}_p| = K/K_b < 1, \tag{3.7}$$

where the inequality follows from the TFBD requirement $K < K_b$. The complement of $\mathbb{I}_p$ along the f-axis corresponds to the stopband $\mathbb{I}_s$ of the $\underline{y}_{\langle k \rangle}[n]$, and its length will be:

$$|\mathbb{I}_s| = 1 - |\mathbb{I}_p| = 1 - K/K_b < 1 \tag{3.8}$$

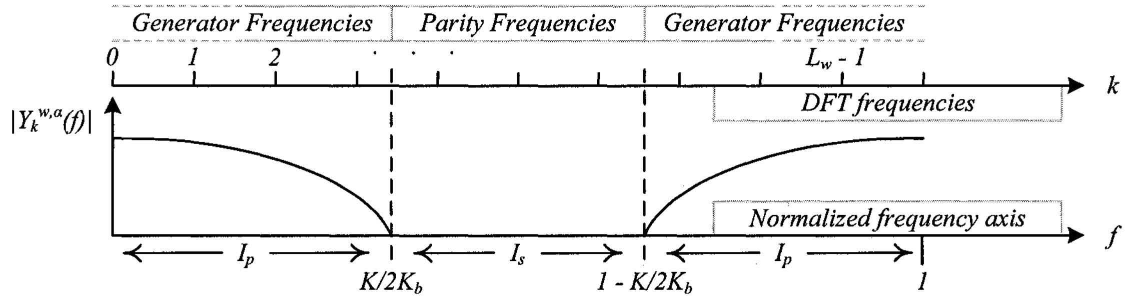

Fig. 3.3 *Parity and generator frequencies of temporal code:* The spectral shape of TFBD temporal codevectors $\underline{y}_k^{w,\alpha}$ is plotted on the f-axis. The corresponding position of temporal codevector parity and generator frequencies is indicated on the k-axis.

The lowpass shape of the time sequences $\underline{y}_{\langle l \rangle}[n]$ could potentially be used to block quantization power falling outside the passband $\mathbb{I}_p$, reducing it by $|\mathbb{I}_p|^2 = (K/K_b)^2$. Since the vectors $\underline{y}[n]$ exist in the span of $\boldsymbol{G}$ (*cf.* Fig. 3.1), we could then reconstruct samples using the corresponding reconstruction matrix $\boldsymbol{T}$ given in (2.24). However, as discussed in Section 1.2, the reconstruction error of DFT codes (given by (2.28) as $\sigma_q^2|\boldsymbol{t}_k|^2$) in the presence of bursty erasures is dominated by the very large magnitude of the rows $\boldsymbol{t}_k$ of $\boldsymbol{T}$. When erasures are bursty, the rows $\boldsymbol{t}_k$ grow exponentially in magnitude with a decreased number of available samples [9]. As explained next, the TFBD code allows for lost samples to be reconstructed along both the temporal and subband orientations, thus making provisions for breaking up erasure bursts and taking advantage of the available exponential gain in $|\boldsymbol{t}_k|^2$.

3.3.1 The Subband and Temporal Codevectors

The crux of the matter regarding the TFBD code is that, by bandlimitting the scalar time sequences $\underline{y}_{\langle l \rangle}[n]$ (*cf.* Fig. 3.1), we are in fact implementing a separate DFT-like code along time. To see this we first consider a TFBD frame, defined with the help of a window $w[n]$ centered at zero, and of length $L_w = 2D_w + 1$. We take L_w contiguous TFBD codevectors $\underline{y}[n]$, weighed by the corresponding window entry, and stack them to form the matrix that we refer to as the TFBD frame:

$$\left[w[-D_w]\, \underline{y}[\alpha - D_w] \;\middle|\; \ldots \;\middle|\; w[D_w]\, \underline{y}[\alpha + D_w] \right]. \tag{3.9}$$

The window $w[n]$ adds the flexibility of spectral shaping (along the rows of the frame) while allowing us to process the sequence $\underline{y}[n]$ by blocks of L_w weighted vectors. We will refer to the rows of the frame as the *temporal codevectors*; the columns we will call the *subband codevectors.* From Fig. 3.1 it is easy to see that the subband codevectors exist in the span of a DFT generator matrix $\boldsymbol{G}$ (the subband generator matrix), and are hence DFT codevectors: they can be used to reconstruct samples using a reconstruction matrix $\boldsymbol{T}$ built according to (2.24b). We will also see that the temporal codevectors also behave as DFT codevectors, and have a related temporal generator matrix $\boldsymbol{\mathcal{G}}$. As with any DFT code, the temporal code can be used to reconstruct samples using a reconstruction matrix $\boldsymbol{\mathcal{T}}$, also built according to (2.24b). However, the expression for the CSMSE will no longer be given by (2.28); we derive error expressions for the temporal code and for the related *pivoting* application in Chapter 4.

To see how the bandlimited sequences $\underline{y}_{\langle l \rangle}[n]$ behave as DFT-like codevectors, we first consider the characteristics of the subband DFT code (and DFT codes in general): the subband codevectors $\underline{y}[n]$ exist in the span of some DFT generator matrix $\boldsymbol{G}$ with general form given by (1.1) and reproduced here as:

$$\boldsymbol{G} = \boldsymbol{W}^{\mathcal{H}}_{N\times K_b}\boldsymbol{A}\boldsymbol{W}_{K_b}.$$

As described in Section 1.2, the generator matrix structures the subband codevectors by forcing nulls at all parity frequencies; the encoded message information will be contained in the set of generator or passband frequencies. As seen from (2.24), reconstruction matrices are uniquely determined by these parity frequencies and the position of the erasures, regardless of the particular form of the DFT generator matrix.

Consider now the finite sequence formed from subbands $\underline{y}_{\langle l \rangle}[n]$ in Fig. 3.1:

$$y_l^{w,\alpha}[n] = w[n-\alpha]\underline{y}_{\langle l \rangle}[n], \tag{3.10a}$$

with n the sequence index and $l = 0, \ldots, K_b - 1$ the subband index; $w[n]$ is length $L_w = 2D_w + 1$ and centered at zero. Using this, we can express the temporal codevectors in vector form (*i.e.*, the transpose of the rows of (3.9)) as follows:

$$\boxed{\underline{y}_l^{w,\alpha} = \begin{bmatrix} y_l^{w,\alpha}[\alpha - D_w] \\ \vdots \\ y_l^{w,\alpha}[\alpha + D_w] \end{bmatrix} \in \mathbb{C}^{L_w}} \tag{3.10b}$$

If we assume for a moment that $f[n]$ is an ideal lowpass filter, and that $w[n]$ is an ideal window (one that preserves perfectly the ideal lowpass shape), then $\underline{y}_l^{w,\alpha}$ will always have nulls at all DFT frequencies (for an L_w-point DFT) corresponding to the stopband. The message information will be stored in the $\lfloor L_w|\mathbb{I}_p|\rfloor$ passband (or generator, *cf.* Section 1.2) frequencies. The remaining $L_w - \lfloor L_w|\mathbb{I}_p|\rfloor$ (null) frequencies will be the parity frequencies of the temporal code. Hence, $\underline{y}_l^{w,\alpha}$ exists in the span of the lowpass DFT generator matrix $\boldsymbol{G}$ of size $L_w \times L_w|\mathbb{I}_p|$, and with knowledge of the erasure positions, we can construct the temporal reconstruction matrix $\mathcal{T}$and recover lost samples as with any DFT code. In combination with the subband codevectors, the temporal codevectors enable the reconstruction process to be carried out along the temporal or subband orientation in order to avoid erasure bursts along a particular direction (we present the related pivoting application in Chapter 4).

In this sense we say that a TFBD frame consists of several DFT codevectors, those falling along the horizontal orientation, referred to as temporal codevectors, and those falling along the vertical orientation, referred to as subband codevectors. We say that all the codevectors in a frame are disjoint in the sense that they can be decoded entirely independently of each other. In the next chapter we will see that, under practical assumptions for the prototype filter $f[n]$ and window $w[n]$, the temporal codevector $\underline{y}_l^{w,\alpha}[n]$ can be decomposed as

$$\underline{y}_l^{w,\alpha}[n] = \underline{y}_p[n] + \underline{y}_s[n]$$

The component $\underline{y}_p[n]$ will be shown to be a DFT codevector, while $\underline{y}_s[n]$ will be responsible for reconstruction errors akin to those resulting from quantization noise present in any practical DFT code setup.

3.3.2 Variable Code Length and Position

We now discuss other ways in which the temporal code can be used to improve reconstruction performance. As mentioned previously, the temporal code has $\lfloor L_w\mathbb{I}_p\rfloor$ generator frequencies and $L_w - \lfloor L_w \mathbb{I}_p \rfloor$ parity frequencies, meaning that $\underline{y}_l^{w,\alpha}$ is an $(L_w, \lfloor L_w \mathbb{I}_p \rfloor)$ DFT code. For illustration purposes, we let $L_w = AK_b$; applying (3.7), the temporal code will be (AK_b, AK). Hence the rate of the temporal code is fixed at K/K_b but its length can be varied through parameter A. The length of the subband code, on the other hand, is fixed at N. Fig. 3.4 illustrates the potential application of the variable length of $\underline{y}_l^{w,\alpha}$. The figure plots the magnitude $|\boldsymbol{\tau}_k|^2$ of the rows of the temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$ corresponding to an (AK_b, AK) code (fixed rate, variable length) for a fixed number of erasures $E = 8$ and A=1, 2 and 50. As can be seen, most of the substantial reduction in $|\boldsymbol{\tau}_k|^2$ can be obtained for $A = 2$, which implies doubling the size of the DFT generator matrix used in defining the temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$, and with this a related increase in complexity in constructing and using $\boldsymbol{\mathcal{T}}$. Referring to (2.28), we saw that for DFT codes, the CSMSE (code sample mean square error) was directly proportional to the magnitude of rows of the reconstruction matrix. The temporal code is a DFT-like code and hence we can expect similar benefits, and we can thus say that the variable length of the temporal code allows us to trade processing complexity for reduced CSMSE. In Chapter 4 we derive exact performance expressions for the temporal code, and in Chapter 5 we test it through simulations.

We now note that the position α of the temporal window used in constructing the temporal codevector (*cf.* (3.10a)) can be varied as part of the decoding process. One can use this to split temporal erasure bursts amongst two consecutive temporal codevectors $\underline{y}_l^{w,\alpha_0}$ and $\underline{y}_l^{w,\alpha_1}$ lying along the same subband. Their positions α_0 and $\alpha_1 = \alpha_0 + L_w$ (they are consecutive) will be such that they meet in the middle of the temporal erasure burst under consideration. We illustrate this graphically in Fig. 3.5, where the two temporal erasure codevectors are denoted by $\underline{y}_l^{w,\alpha_0}$ and $\underline{y}_l^{w,\alpha_1}$. The lost samples forming the temporal erasure burst are delineated by the heavy line, and the positions α_0 and α_1 of the temporal codevectors are chosen so each codevector is responsible for correcting half the samples in the burst, effectively reducing the length of the maximum reconstructed burst by half (note that we are neglecting subband decoding for illustrative purposes, as decoding along the subbands would only require reconstructions of a single lost sample).

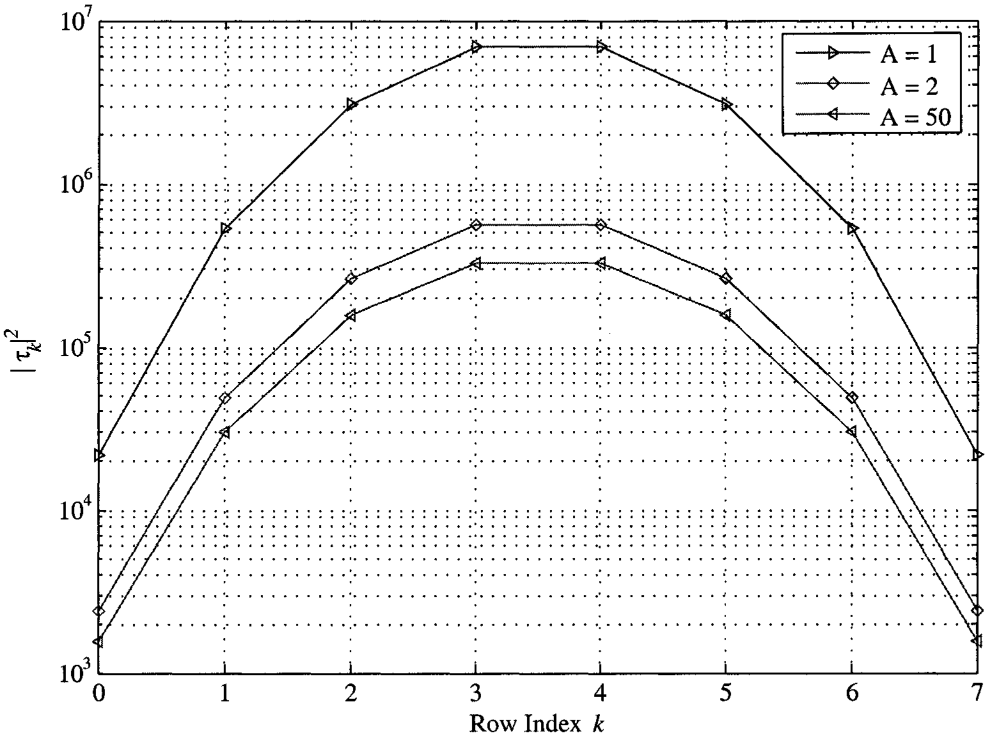

Fig. 3.4 *Row magnitudes of temporal reconstruction matrix vs. code length:* The horizontal axis indicates the row index of the temporal reconstruction matrix. The reconstruction matrix used corresponds to a (K_b, K) DFT code for $K_b = 19$ and $K = 11$.

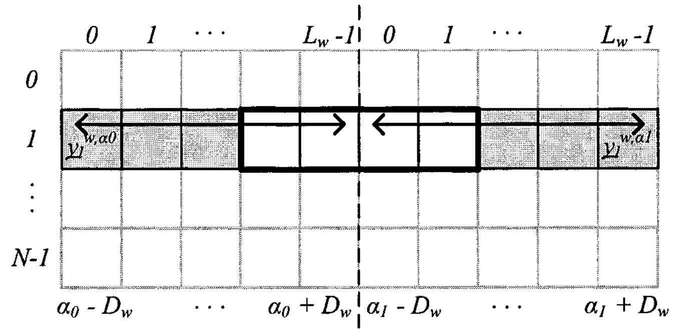

Fig. 3.5 *Splitting temporal erasure bursts:* The temporal erasure burst is split along the middle so that $\underline{y}_l^{w,\alpha_0}$ and $\underline{y}_l^{w,\alpha_1}$ are each used to reconstruct half the samples. The heavy frame delineates the temporal erasure burst, and the dashed line separates the two frames under analysis. Double-sided arrows specify the two temporal codevectors under analysis, and we highlight the available samples in both codevectors. Upper column labels specify the frame-relative time indices; bottom column labels specify the time index n; row labels specify subband indices.

3.4 Summary

We began the present chapter by introducing the Tandem Filterbank / DFT (TFBD) encoder: it consists of the tandem arrangement of a uniform filterbank with baseband-modulated subbands, followed by a DFT generator matrix. The resulting TFBD frame consists of temporal codevectors (along the rows) and subband codevectors (along the columns). We then derived the TFBD transfer function; it will prove useful in the next chapter, where we derive expressions for the TFBD reconstruction error. We finished our introduction by discussing ways in which the resulting TFBD frame could be exploited to improve bursty erasure reconstructions.

In the next chapter we discuss better ways of exploiting the TFBD frame structure. We begin by presenting a mathematical analysis of the TFBD temporal code and show how it can be interpreted as a DFT code. We then provide expressions for the code sample reconstruction error (CSMSE) of the temporal code. In the last sections of the chapter, we introduce *pivoting*, a method by which samples reconstructed along one orientation can be used as received samples in carrying out reconstructions along the remaining orientation. As a means of empirical support for the correctness of our expression (we test expressions experimentally in Chapter 5), we conclude by revisiting the geometrical interpretation of the bursty erasure phenomenon(*cf.* Section 2.3) and show how our pivoting expression corresponds to the geometrical model.

Chapter 4

Mathematical Analysis of the TFBD Code

In the previous chapter we introduced the basic form of the TFBD encoder and saw how it effectively creates disjoint DFT codevectors along the rows and columns of the frame under analysis. By disjoint we mean that these codevectors can be reconstructed independently of each other. We thus have at least two possible ways of reconstructing any given sample: along the temporal or horizontal orientations. To select the reconstruction orientation we choose the one that minimizes the code sample mean square error (CSMSE) of the related sample. One problem with this approach is that the temporal code does not have exact nulls at the parity frequencies, meaning that the CSMSE expression of DFT codevectors given in Section 1.2 does not apply. We are hence lacking the tools to make decisions regarding the reconstruction orientation.

We begin the present chapter by introducing the temporal reconstruction matrix $\mathcal{T}$that we will use to carry out reconstructions along the temporal orientation. We then present the stopband / passband decomposition of the temporal codevector. We will see that the passband component is a DFT codevector responsible for the DFT-code-like qualities of the temporal code, while the stopband component will be responsible for reconstruction errors akin to those resulting from quantization noise. We next derive an exact expression for the CSMSE of the temporal code in the presence of quantization noise, which we then use to derive a simplified upper bound. The expressions will be in terms of the correlation matrix of the stopband component of temporal codevectors, and this we derive next. We go on to show how samples reconstructed along one orientation can be used as received samples in carrying out reconstructions along the remaining orientation, a technique we refer to as *pivoting*. We then derive an exact expression for temporal-to-subband pivoting operations, as well as the corresponding simplified upper bound.

4.1 Reconstructing Samples With the Temporal Code

A CSMSE performance analysis of the temporal code can be carried out by expanding on the method [9] presented in Section 2.3.1. There we saw that one could partition the null syndrome requirement of DFT codevectors ($\boldsymbol{H}^{\mathcal{H}}\underline{y} = \boldsymbol{C}\underline{y} = \underline{0}$, $\boldsymbol{C} = \boldsymbol{H}^{\mathcal{H}}$) to build a reconstruction matrix $\boldsymbol{T}$: We group the received samples to form the received vector $\underline{y}_R$, and do likewise with the erased samples to form the erasure vector $\underline{y}_E$. Reordering the columns of $\boldsymbol{C}$ correspondingly, we rewrite the null syndrome requirement as:

$$[\boldsymbol{C}_R|\boldsymbol{C}_E]\begin{bmatrix}\underline{y}_R\\ \underline{y}_E\end{bmatrix} = \boldsymbol{C}_R\underline{y}_R + \boldsymbol{C}_E\underline{y}_E = 0. \tag{4.1}$$

As long as $R \geq E$, we can solve for $\underline{y}_E$ to arrive at the general form of a DFT reconstruction matrix:

$$\boldsymbol{T} = -\boldsymbol{C}_E^{+}\boldsymbol{C}_R \tag{4.2}$$

In the absence of quantization, the reconstruction $\underline{y}_E = \boldsymbol{T}\underline{y}_R$ will be exact. In practice, the decoder only has access to the quantized samples

$$\underline{\ddot{y}}_R = \underline{y}_R + \underline{q}_R.$$

The resulting reconstruction will only yield the estimate $\underline{\hat{y}} = \boldsymbol{T}\underline{\ddot{y}}_R$. As we saw in (2.28), by letting the rows of $\boldsymbol{T}$ be denoted by $\boldsymbol{t}_k$, we can write the related CSMSE (code sample mean square error) as

$$\Psi(e_k) = |\boldsymbol{t}_k|^2\sigma_q^2, \tag{4.3}$$

where e_k is the position of the k-th erasure and σ_q^2 is the quantization noise power.

We note that in order to build a general DFT reconstruction matrix we need two pieces of information: Since the parity matrix $\boldsymbol{H}$ (and $\boldsymbol{C}$ follows from $\boldsymbol{C} = \boldsymbol{H}^{\mathcal{H}}$) is formed from the columns of the IDFT matrix $\boldsymbol{W}_N^{\mathcal{H}}$ with indices corresponding to those of the parity frequencies, we first need to know the set of parity frequencies. Secondly, in order to carry out the partitioning $\boldsymbol{C} = [\boldsymbol{C}_R|\boldsymbol{C}_E]$, we need to know the position of the erasures. And so a general DFT reconstruction matrix $\boldsymbol{T}$ is entirely specified from the parity and erasure indices.

As discussed in the last chapter, we have seen that the temporal code can be said to have parity frequencies at the stopband of the $\underline{y}_{\langle k \rangle}[n]$ (this is illustrated in Fig. 3.3). And hence, with knowledge of the erasure positions, we can build a *temporal reconstruction matrix* $\boldsymbol{\mathcal{T}}$as with any DFT codevector and use it in reconstructing lost samples. However, since the temporal code does not have exact nulls at the parity frequencies, the code sample mean square error (CSMSE) given in (2.28) (and reproduced in (4.3)) will no longer apply.

4.1.1 The Stopband and Passband Components of the Temporal Codevector

In order to carry out a CSMSE analysis of the temporal code, we first decompose the temporal codevector into a stopband and passband component. The passband component will have exact nulls at DFT frequencies in the stopband, and so we can use these as parity frequencies in carrying out reconstructions, while treating the stopband component as noise. For notational convenience we will drop the subband and position indices l and α in $\underline{y}_l^{w,\alpha}$ and let $\underline{y}^w$ denote a general temporal codevector.

Let $\underline{y}_s^w$ and $\underline{y}_p^w$ denote the stopband and passband components of the temporal code vector, respectively (we neglect quantization for now):

$$\underline{y}^w = \underline{y}_p^w + \underline{y}_s^w. \tag{4.4}$$

The stopband (respectively passband) component is the IDFT of the DFT of $\underline{y}^w$ after zeroing out all DFT components corresponding to the passband (respectively stopband). For instance, the passband component $\underline{y}_p^w$ can be written as:

$$\begin{aligned} \underline{y}_p^w &= \boldsymbol{W}_{L_w}^{\mathcal{H}} \boldsymbol{D}_p \boldsymbol{W}_{L_w} \underline{y}^w & (4.5) \\ &= \boldsymbol{W}_{L_w}^{\mathcal{H}} \boldsymbol{D}_p \underline{Y}_w & (4.6) \end{aligned}$$

Where $\boldsymbol{D}_p$ corresponds to the identity matrix with zeros along diagonal entries indexed over the stopband frequencies. Likewise, using $\boldsymbol{D}_s = \boldsymbol{I} - \boldsymbol{D}_p$ the stopband component can be expressed as

$$\underline{y}_s^w = \underbrace{\boldsymbol{W}_{L_w}^{\mathcal{H}} \boldsymbol{D}_s}_{\boldsymbol{F}_s} \underline{Y}_w, \tag{4.7}$$

where we have also defined the stopband operator $\boldsymbol{F}_s$, used later in the chapter.

For clarity, we now summarize our notation as follows (see **Tbl. A.1**, pg. 89 for a full list of notational conventions): subscripts p and s (*e.g.*, $\underline{y}_p^w$, $\underline{y}_s^w$) denote passband and stopband components, respectively. Subscripts E and R (*e.g.*, $\underline{y}_E^w$, $\underline{y}_R^w$) will denote vectors composed of samples with erased/received indices. We will combine this notation to denote received/erased samples taken from the stopband/passband components of $\underline{y}^w$. For example, $\underline{y}_{Rs}^w$ will contain samples with received indices from vector $\underline{y}_s^w$.

We now show how the temporal codevector can be interpreted as a DFT code. Referring to (4.6), we rearrange the columns of $\boldsymbol{D}_p$ and partition it as $\boldsymbol{D}_p = [\boldsymbol{E}_g \,|\, \boldsymbol{0}]$, where $\boldsymbol{E}_g$ will contain all the non-zero columns. We reorder $\underline{Y}_w$ accordingly to obtain $[\underline{Y}_p^{w'\mathcal{T}} \,|\, \underline{Y}_s^{w'\mathcal{T}}]^{\mathcal{T}}$,[1] so that we can rewrite (4.6) as follows:

$$\underline{y}_p^w = \boldsymbol{W}_{L_w}^{\mathcal{H}} [\boldsymbol{E}_g \,|\, \boldsymbol{0}] \begin{bmatrix} \underline{Y}_p^{w'} \\ \underline{Y}_s^{w'} \end{bmatrix} \tag{4.8a}$$

$$= \boldsymbol{W}_{L_w}^{\mathcal{H}} \boldsymbol{E}_g \underline{Y}_p^{w'} \tag{4.8b}$$

$$= \boldsymbol{G}_p \underline{Y}_p^{w'} \tag{4.8c}$$

where $\boldsymbol{G}_p$ is a simple DFT generator matrix (*cf.* (1.2))consisting of the columns of $\boldsymbol{W}_{L_w}$ with indices corresponding to passband frequencies. This shows that $\underline{y}_p^w$ is a DFT codevector and that there must exist a related *temporal reconstruction matrix* $\boldsymbol{\mathcal{T}}$ of size $E \times R$ allowing us to write:

$$\underline{y}_{Ep}^w = \boldsymbol{\mathcal{T}} \underline{y}_{Rp}^w, \tag{4.9}$$

where $\underline{y}_{Ep}^w$ (respectively $\underline{y}_{Rp}^w$) is composed of samples from $\underline{y}_p^w$ with erased (received) indices.

To summarize, we began by decomposing the temporal codevector into its passband and stopband components. The passband component had nulls at the stopband frequencies and was hence a DFT codevector. We showed this by deriving an expression for the corresponding DFT generator matrix. We can thus build a temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$and use it in estimating lost samples along the temporal orientation. We note that even

[1] Note $\underline{Y}_p^{w'}$ (respectively $\underline{Y}_s^{w'}$) is composed only of the non-zero elements of the DFT of $\underline{y}_p^w$ ($\underline{y}_s^w$).

in the absence of quantization, the resulting estimates $\underline{y}^w_{Ep\langle k\rangle}$ produced by (4.9) will only be an approximation to the actual lost samples $\underline{y}^w_{E\langle e_k\rangle}$, their usefulness being a function of the stopband attenuation of the prototype filters $f[n]$ (*cf.* Fig. 3.1). Furthermore, rather than having access to the passband samples $\underline{y}^w_{Rp\langle k\rangle}$ (*cf.* (4.9)), we only have at our disposal those corresponding to the temporal codevector $\underline{y}^w_R$, adding additional error to our reconstruction. We take these points into account in the following discussion, where we derive exact expressions for the CSMSE of temporal reconstruction operations.

4.1.2 CSMSE Performance Analysis of the Temporal Code

In the presence of quantization, the vector $\ddot{\underline{y}}^w_R$ available to the decoder has the form

$$\begin{aligned} \ddot{\underline{y}}^w_R &= \underline{y}^w_R + \boldsymbol{D}_{Rw}\underline{q}_R \\ \ddot{\underline{y}}^w_R &= \underline{y}^w_{Rs} + \underline{y}^w_{Rp} + \boldsymbol{D}_{Rw}\underline{q}_R \end{aligned}$$

where $\underline{q}_R$ represents the quantization noise, modeled as i.i.d. uniform samples with variance σ_q^2; $\boldsymbol{D}_{Rw}$ is a diagonal matrix with entries of $w[n]$ corresponding to received samples along its main diagonal. Using $\ddot{\underline{y}}^w_R$ in (4.9), we can build an estimate $\hat{\underline{y}}^w_{Ep}$ (and use it in place of $\underline{y}^w_E$ in the decoder output) as follows:

$$\begin{aligned} \hat{\underline{y}}^w_{Ep} &= \boldsymbol{\mathcal{T}}\ddot{\underline{y}}^w_R \\ &= \boldsymbol{\mathcal{T}}(\underline{y}^w_{Rp} + \underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R) \\ &= \underline{y}^w_{Ep} + \boldsymbol{\mathcal{T}}(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R) \end{aligned} \tag{4.10}$$

We will use (4.10) to find the code sample mean square error for the passband erasure vector estimate, $\Psi_p(e_k)$, where e_k denotes the position of the k-th erased sample along the original codevector; r_k will be used likewise for the k-th received sample. We note that, being the CSMSE of the passband component, $\Psi_p(e_k)$ will be an approximation to the CSMSE of the temporal codeword, $\Psi_w(e_k)$. Using (4.10), we express the temporal reconstruction error as:

$$\underline{y}^w_{E\langle k\rangle} - \hat{\underline{y}}^w_{Ep\langle k\rangle} = \underline{y}^w_{E\langle k\rangle} - [\underline{y}^w_{Ep\langle k\rangle} + \boldsymbol{\tau}_k(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R)] \tag{4.11a}$$

$$= \underline{y}^w_{Es\langle k\rangle} - \boldsymbol{\tau}_k(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R) \tag{4.11b}$$

The resulting form for $\Psi_w(e_k)$ follows, where we let $\boldsymbol{\tau}_k$ denote the k-th row of the temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$:

$$\begin{aligned}
\Psi_w(e_k) &= \mathrm{E}\big[\,|\underline{y}^w_{E\langle k\rangle} - \hat{\underline{y}}^w_{Ep\langle k\rangle}|^2\,\big] \\
&= \mathrm{E}\big[\,|\underline{y}^w_{Es\langle k\rangle} - \boldsymbol{\tau}_k(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R)|^2\,\big] \\
&= \sigma_s^2(e_k) + \mathrm{E}\big[\,|\boldsymbol{\tau}_k(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R)|^2\,\big] - \mathrm{E}\big[\,2\mathrm{Re}\big[\,\underline{y}^w_{Es\langle k\rangle}\underline{y}^{w\mathcal{H}}_{Rs}\boldsymbol{\tau}_k^{\mathcal{H}}\,\big]\,\big] \\
&= \sigma_s^2(e_k) + \Psi_p(e_k) - 2\mathrm{Re}\big[\,\boldsymbol{R}_{s\langle e_k,\{r_i|i=1,\ldots,R\}\rangle}\boldsymbol{\tau}_k^{\mathcal{H}}\,\big] && (4.12)
\end{aligned}$$

Here $\sigma_s^2(e_k)$ is the power of the stopband component at the corresponding sample, and $\boldsymbol{R}_s$ is the correlation matrix for vector $\underline{y}^w_s$. We obtain our expression for $\Psi_p(e_k)$ from (4.10):

$$\begin{aligned}
\Psi_p(e_k) &= \mathrm{E}\big[\,|\underline{y}^w_{Ep\langle k\rangle} - \hat{\underline{y}}^w_{Ep\langle k\rangle}|^2\,\big] \\
&= \mathrm{E}\big[\,|\boldsymbol{\tau}_k(\underline{y}^w_{Rs} + \boldsymbol{D}_{Rw}\underline{q}_R)|^2\,\big] \\
&= \mathrm{E}\big[\,\boldsymbol{\tau}_k\underline{y}^w_{Rs}\underline{y}^{w\mathcal{H}}_{Rs}\boldsymbol{\tau}_k^{\mathcal{H}} + \boldsymbol{\tau}_k\boldsymbol{D}_{Rw}\underline{q}_R\underline{q}_R^{\mathcal{H}}\boldsymbol{D}_{Rw}^{\mathcal{H}}\boldsymbol{\tau}_k^{\mathcal{H}}\,\big] \\
&= \boldsymbol{\tau}_k\boldsymbol{R}_{Rs}\boldsymbol{\tau}_k^{\mathcal{H}} + \sigma_q^2\boldsymbol{\tau}_k\boldsymbol{D}_{Rw}^2\boldsymbol{\tau}_k^{\mathcal{H}} && (4.13a) \\
&= e_s(k) + e_q(k) && (4.13b)
\end{aligned}$$

Here $\boldsymbol{R}_{Rs}$ is the correlation matrix $\mathrm{E}\big[\,\underline{y}^w_{Rs}\underline{y}^{w\mathcal{H}}_{Rs}\,\big]$ (a submatrix of $\boldsymbol{R}_s$) with singular value decomposition $\boldsymbol{R}_{Rs} = \boldsymbol{U}\boldsymbol{D}_s\boldsymbol{U}^{\mathcal{H}}$. We can use this to rewrite $e_s(k)$ in (4.13) as follows:

$$\begin{aligned}
e_s(k) &= \boldsymbol{\tau}_k\boldsymbol{U}\boldsymbol{D}_s\boldsymbol{U}^{\mathcal{H}}\boldsymbol{\tau}_k^{\mathcal{H}} \\
&= \boldsymbol{\tau}_{ks}\boldsymbol{D}_s\boldsymbol{\tau}_{ks}^{\mathcal{H}} \\
&= \sum_{l=0}^{R-1} |\boldsymbol{\tau}_{ks\langle l\rangle}|^2\lambda_{s,l} \\
&\leq \lambda_{s,max}\sum_{l=0}^{R-1} |\boldsymbol{\tau}_{ks\langle l\rangle}|^2 && (4.14) \\
&= \lambda_{s,max}|\boldsymbol{\tau}_k^{\mathcal{H}}|^2 && (4.15)
\end{aligned}$$

Similarly, using $\lambda^2_{w,max}$ for the largest entry of $\boldsymbol{D}^2_{Rw}$, we can write $e_q(k) \leq \sigma_q^2\lambda^2_{w,max}|\boldsymbol{\tau}_k^{\mathcal{H}}|^2$. Using this and (4.15) in (4.13) yields:

$$\Psi_p(e_k) \leq (\lambda_{s,max} + \sigma_q^2\lambda^2_{w,max})|\boldsymbol{\tau}_k^{\mathcal{H}}|^2. \tag{4.16}$$

In Chapter 5, where we discuss our results, we will see that (for the TFBD codes considered) the passband CSMSE $\Psi_p(e_k)$ is a very good approximation to the temporal CSMSE $\Psi_e(e_k)$, and hence (4.16) can also be used for the latter. Note that (4.16) gives a loose indication of the length required for the prototype filter: as long as $\lambda_{s,max}$ is sufficiently smaller than $\sigma_q^2\lambda_w^2$, filters with deeper stopbands will not improve the *upper bound* on the passband CSMSE Ψ_p.

4.2 Expression for the Stopband Correlation Matrix of the Temporal Codevector

The temporal reconstruction error $\Psi_w(e_k)$ is expressed in (4.12) and (4.13) in terms of the stopband correlation matrix $\boldsymbol{R}_s$, for which we now derive an expression. Assuming that a model is available for the input message sequence $x[n]$, we wish to express the samples of the stopband component $\underline{y}_s^w$ in terms of the input message samples. This will in turn allow us to express the stopband correlation matrix $\boldsymbol{R}_s$ in terms of the statistics of the input message sequence. This approach will enable us to obtain an exact form for $\boldsymbol{R}_s$. However, the resulting form involves a matrix expressed in element-wise form, where each matrix element will consist of a quadruple sum. Assuming that the input sequence $x[n]$ is i.i.d. will enable simplification to a double summation over a convolution.

We consider first the vector $\underline{y}_l^\alpha$ ($\underline{y}^\alpha$ for brevity) corresponding to the temporal codevector $\underline{y}_l^{w,\alpha}$ ($\underline{y}^w$) before windowing, where the window is assumed to have a length $L_w = 2D_w + 1$ (*cf.* Fig. 3.1):

$$\underline{y}_l^\alpha = \left[\ \underline{y}_{\langle l\rangle}[\alpha - D_w]\ \mid\ \cdots\ \mid\ \underline{y}_{\langle l\rangle}[\alpha + D_w]\ \right]^{\mathcal{T}} \quad (= \underline{y}^\alpha). \tag{4.17}$$

The vector $\underline{y}^\alpha$ can be mapped to $\underline{y}^w$ through $\boldsymbol{D}_w$, the diagonal matrix containing the window along its main diagonal:

$$\underline{y}^w = \boldsymbol{D}_w \underline{y}^\alpha \tag{4.18}$$

Using (4.18) and the stopband operator $\boldsymbol{F}_s$ given in (4.7), we can write $\underline{y}_s^w$ (we omit the subband index l) as $\underline{y}_s^w = \boldsymbol{F}_s\boldsymbol{D}_w\underline{y}^\alpha$ and express $\boldsymbol{R}_s$ as:

$$\begin{aligned}\boldsymbol{R}_s &= \mathrm{E}\big[\,\underline{y}_s^w\underline{y}_s^{w\mathcal{H}}\,\big]\\ &= \boldsymbol{F}_s\boldsymbol{D}_w\mathrm{E}\big[\,\underline{y}^\alpha\underline{y}^{\alpha\mathcal{H}}\,\big]\boldsymbol{D}_w^{\mathcal{H}}\boldsymbol{F}_s^{\mathcal{H}}\\ &= \boldsymbol{F}_s\boldsymbol{D}_w\boldsymbol{R}_\alpha\boldsymbol{D}_w^{\mathcal{H}}\boldsymbol{F}_s^{\mathcal{H}} \qquad (4.19)\end{aligned}$$

We now obtain an element-wise expression for the general form of $\boldsymbol{R}_\alpha$. The elements of $\boldsymbol{R}_\alpha$ are given by (*cf.* (4.17)):

$$\begin{aligned}\boldsymbol{R}_{\alpha\langle c,d\rangle} &= \mathrm{E}\big[\,\underline{y}_{\langle c\rangle}^{\alpha}\underline{y}_{\langle d\rangle}^{\alpha *}\,\big] \qquad (4.20\text{a})\\ &= \mathrm{E}\big[\,\underline{y}_{\langle l\rangle}[c_\alpha]\;\underline{y}_{\langle l\rangle}^{*}[d_\alpha]\,\big] \qquad (4.20\text{b})\end{aligned}$$

where we let $c_\alpha = \alpha - D_w + c$ and $d_\alpha = \alpha - D_w + d$. Using (3.6b) we get the following form (where L_f is the length of the prototype filter $f[n]$):

$$\begin{aligned} &= K^2\mathrm{E}\big[\sum_{m=0}^{K_b-1}\sum_{n=0}^{L_f-1}\boldsymbol{G}_{\langle l,m\rangle}f_m[n]x[c_\alpha K-n]W_{K_b}^{-mKc_\alpha}\\ &\qquad \sum_{p=0}^{K_b-1}\sum_{q=0}^{L_f-1}\boldsymbol{G}_{\langle l,p\rangle}^{*}f_p^{*}[q]x^{*}[d_\alpha K-q]W_{K_b}^{pKd_\alpha}\,\big] \qquad (4.20\text{c})\\ &= K^2\sum_{m,n,p,q}\boldsymbol{G}_{\langle l,m\rangle}\boldsymbol{G}_{\langle l,p\rangle}^{*}f_m[n]f_p^{*}[q]\\ &\qquad R_x(c_\alpha K-n, d_\alpha K-q)W_{K_b}^{-(mc_\alpha-pd_\alpha)K} \qquad (4.20\text{d})\end{aligned}$$

Note that the summation indices in (4.20d) take the same values as in (4.20c) and $R_x(k,l) = \mathrm{E}\big[\,x[k]x^*[l]\,\big]$. Assuming that the message sequence is i.i.d. allows us to substitute the summation over q with the single non-zero term at $q = (d_\alpha - c_\alpha)K + n$, the resulting form being:

$$\begin{aligned}\boldsymbol{R}_{\alpha\langle c,d\rangle} &= K^2\sigma_x^2\sum_{m,n,p}\Big[\boldsymbol{G}_{\langle l,m\rangle}\boldsymbol{G}_{\langle l,p\rangle}^{*}f_m[n]f_p^{*}[K(d_\alpha-c_\alpha)+n]W_{K_b}^{-(mc_\alpha-pd_\alpha)K}\Big] \quad (4.21\text{a})\\ &= K^2\sigma_x^2\sum_{m,p}\Big[\boldsymbol{G}_{\langle l,m\rangle}\boldsymbol{G}_{\langle l,p\rangle}^{*}\Big[f_m[n]*f_p^{*}[-n]\Big]_{(c_\alpha-d_\alpha)K}W_{K_b}^{-(mc_\alpha-pd_\alpha)K}\Big] \quad (4.21\text{b})\end{aligned}$$

Using (4.21) in (4.19), one can then compute the correlation matrix $\boldsymbol{R}_s$ of the stopband component of temporal codevectors that is needed for the CSMSE computation in (4.12) and (4.13).

4.2.1 Power of TFBD Code Samples

Being the correlation matrix of the temporal codevector before windowing, $\boldsymbol{R}_\alpha$ can be used to obtain an approximation to the power σ_y^2 of TFBD code samples (*i.e.,* the power for all entries of $\underline{y}[n]$ in Fig. 3.1, at all times n). We do this by assuming that the prototype filter $f[n]$ is an ideal lowpass filter (with cutoff frequency $f_c = 1/2K_b$, as explained in Section 3.3). Referring to the convolution expression in (4.21b), the assumption of ideal $f[n]$ zeros the convolution for all values other than $m = p$. Furthermore, we are only interested in the diagonals of $\boldsymbol{R}_\alpha$, and hence we can use $d_\alpha = c_\alpha$, thus simplifying the convolution expression to the sum of magnitude-squared entries of $f[n]$; note that these same simplifications ($m = p$ and $d_\alpha = c_\alpha$) reduce the complex exponential in (4.21b) to one, and thus we can write the power of TFBD code samples as follows:

$$\sigma_y^2 = \sigma_x^2 K^2 \left[\sum_{m=0}^{K_b-1} |\boldsymbol{G}_{\langle l,m\rangle}|^2\right] \left[\sum_{k=0}^{L_f-1} |f[k]|^2\right]$$

For the case when $\boldsymbol{G}$ is a simple, upper-parity matrix (as given by (1.3)), $\boldsymbol{G}_{\langle l,m\rangle} = \frac{1}{\sqrt{N}}\mathrm{e}^{-j2\pi kl/N}$. The resulting form for σ_y^2 will be

$$\boxed{\sigma_y^2 = \sigma_x^2 \left(\frac{K^2 K_b}{N}\right) \sum_{k=0}^{L_f-1} |f[k]|^2} \tag{4.22}$$

Under practical considerations for $f[n]$, the convolution operation in (4.21b) will not be exactly zero for values other than $m = p$. However, as we will see in Chapter 5 (for the cases considered), the approximation given by (4.22) will display a relative error better than 10^{-5}, meaning that, for practical purposes, it is as good as the actual value. Note that the exact value for the power of a given TFBD sample can be obtained from the first K_b entries of the diagonal of $\boldsymbol{R}_\alpha$ at the corresponding subband (the diagonal is K_b periodic).

Power Spectrum of Temporal Codevector: The matrix $\boldsymbol{R}_\alpha$ will also be useful in computing the power spectrum $\underline{S}_w$ of the temporal codevector. We define $\underline{S}_w$ as having entries given by

$$\underline{S}_{w\langle k\rangle} = |\underline{Y}^{w}_{\langle k\rangle}|^2,$$

where $\underline{Y}^w$ is the DFT of the temporal codevector. We can express $\underline{Y}^w$ as follows:

$$\underline{Y}^w = \boldsymbol{W}_{L_w} \underbrace{\underline{y}^w}_{\boldsymbol{D}_w \underline{y}^\alpha \ \textit{cf.}\ (4.18)} \quad ,$$

where $\boldsymbol{W}_{L_w}$ is the DFT matrix of size L_w. The entries of $\underline{S}_w$ will correspond to the diagonals of the correlation matrix of $\underline{Y}^w$, as shown below, where we use $\mathrm{diag}_k(\cdot)$ to denote the k-th entry along the diagonal of a matrix:

$$\begin{aligned} \underline{S}_{w\langle k\rangle} &= \mathrm{diag}_k(\mathrm{E}[\underline{Y}^w \underline{Y}^{w\mathcal{H}}]) \\ &= \mathrm{diag}_k(\boldsymbol{W}_{L_w} \boldsymbol{D}_w \boldsymbol{R}_\alpha \boldsymbol{D}_w^{\mathcal{H}} \boldsymbol{W}_{L_w}^{\mathcal{H}}) \end{aligned} \tag{4.23}$$

In Chapter 5, expression (4.23) will be useful as a tool for the selection of the parity frequencies of the temporal code.

4.3 Disjoint DFT Codes and Pivoting

In Chapter 3 we saw how an immediate benefit of TFBD came from the fact that either the temporal or subband code can be selected in order to avoid bursts when reconstructing erasures. This is possible because the reconstruction capabilities of the codevectors are disjoint: erasure recovery along the temporal orientation can be done entirely independently of erasure recovery along the subband orientation. With this, the TFBD code allows us to entirely avoid bursts occurring along a single orientation (1-D bursts). The related increase in performance can be traded against the delay introduced by interleavers meant to reduce the occurrence of bursts. This would in turn increase the probability that bursts occur simultaneously along both orientations (2-D erasure bursts) and, as we will see, TFBD provides a tool for this situation in the form of *pivots*.

We will use the term *pivot* to refer to a lost sample within a burst; as all samples, it is common to a subband codevector and at least one temporal codevector, and a better estimate can be obtained along one orientation than the other (note this does not necessarily mean that the burst exists only along one direction). The reconstruction can be carried out first along the orientation yielding the best estimate, and then the reconstruction process is rotated (hence the analogy to mechanical pivots) and carried out along the remaining direction, where this new estimate is used as a 'received' sample. In this way, pivots can be used to break up a 2-D burst into smaller bursts, resulting in potentially several orders of magnitude of CSMSE improvement.

A good indication of the benefit available through pivoting operations is given by Fig. 4.1. The figure plots the squared row magnitudes of a sample temporal reconstruction matrix (*i.e.*, $|\boldsymbol{\tau}_k|^2$) as a function of the index k of the corresponding erasure vector (also the row-index of the reconstruction matrix) for a burst of length 13, versus the case when the third sample in the erasure burst is assumed to be received. The row magnitudes are seen to decrease between one and three orders of magnitude and, with a pivot sample displaying a quality reconstruction, we can expect similar benefits given the similarities between CSMSE expressions of the temporal and subband codes (*cf.* (4.16) and (2.28)).

We illustrate pivoting operations by considering the 2-D bursty erasure shown in Fig. 4.2, where the heavy lined region denotes the 2-D erasure burst being reconstructed, and we have highlighted the received samples of temporal code vector $\underline{y}_1^{w,12}$, where $L_w = 25$. Our approach here (in selecting the reconstruction orientation) is merely illustrative, and in Section 4.3.1 we provide decision rules for certain pivoting operations. We first apply the subband codevector at frame column 12; this enables us to rotate the reconstruction process by using the estimate $\hat{\underline{y}}_{\langle 1 \rangle}[12]$ as a new 'received' sample in carrying out reconstructions with the temporal codevector $\underline{y}_1^{w,12}$. We will call $\hat{\underline{y}}_{\langle 1 \rangle}[12]$ a *subband-to-temporal* pivot, denoted by the symbol ⊿. Reconstruction along the temporal orientation (frame row 1) will now yield estimates of the lost samples in the indicated temporal codevector, and these will become *temporal-to-subband* pivots (denoted by ◿) for the remaining reconstructions along the subband orientation.

Probability of 2-D Erasure Bursts: Transmission of the TFBD code frame is done serially, its rows concatenated to form a single transmission vector. The reason why serialization is done along the temporal orientation is that temporal codevectors will generally be longer than subband codevectors, and hence erasure bursts are more likely to fall along a single temporal codevector, resulting in 1-D (rather than 2-D) erasure bursts. For 1-D temporal erasure bursts, all sample reconstructions can be done along the subband orientation, thus avoiding burst reconstructions entirely. In this sense, the TFBD code displays

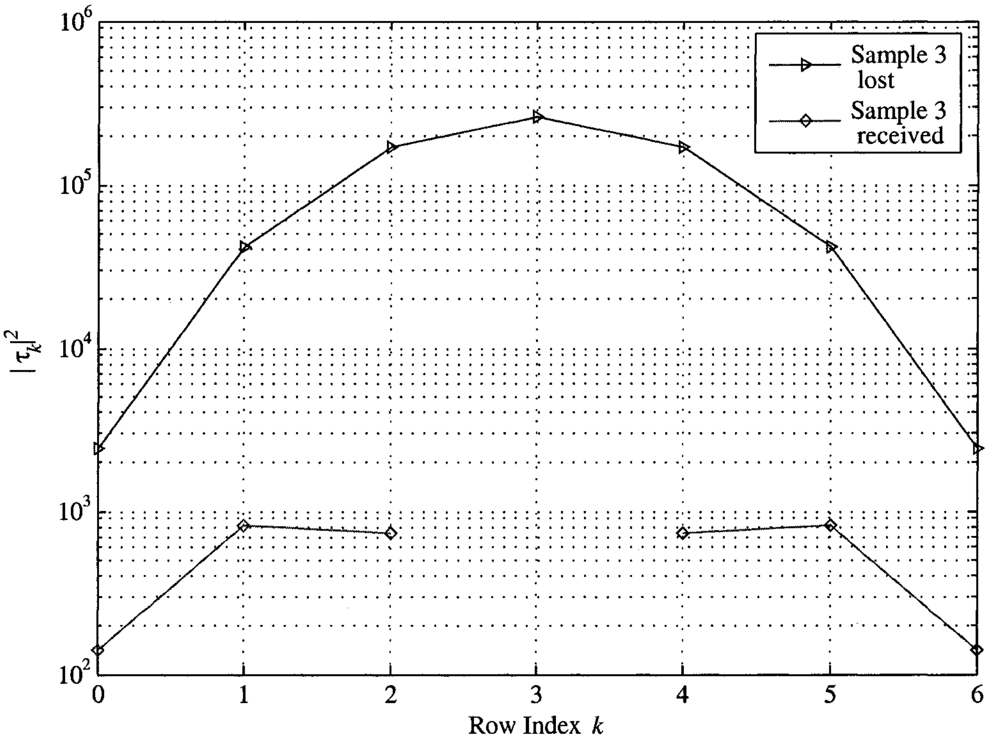

Fig. 4.1 *Pivot effect on row-magnitudes of reconstruction matrix:* Row magnitueds $|t_k^{\mathcal{H}}|$ of reconstruction matrix $\boldsymbol{T}$ for a $(25, 11)$ DFT code. In the first case, $\boldsymbol{T}$ corresponds to a burst of size $E = 7$ erasures; the second case ($E = 6$) is that when the sample at position $k = 3$ (along erasure vector, position e_3 of temporal codevector) is available. Quantization power is normalized to unity. Horizontal axis labels correspond to the erasure vector index (also the row-indices of $\boldsymbol{T}$).

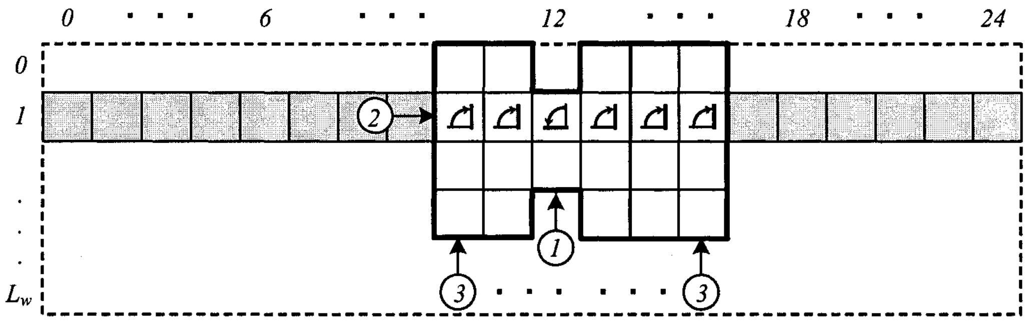

Fig. 4.2 *2-D bursty erasure correction via pivoting:* Row and column indices correspond to temporal and subband codevector indices, respectively. The heavy frame delineates lost samples. Circled indices represent the decoding order of codevectors, the order being irrelevant among codevectors with the same index. The symbol ⤶ denotes temporal-to-subband pivoting; ⤴ denotes subband-to-temporal pivoting.

a greater burst length tolerance than other reconstruction schemes, and hence a smaller required interleaver length and associated delay. In turn, reducing the interleaver length increases the probability that erasure bursts will occur along two or more temporal codevectors, possible resulting in 2-D erasure bursts. In this case, the selection of pivot samples is less obvious, and thus we will now derive expressions that can be used in pivot selection.

4.3.1 Selecting Temporal-To-Subband Pivots

In this section we derive CSMSE expressions for the following simplified case: in using the subband orientation to reconstruct a subband erasure burst within a 2-D burst, which erased samples should we first reconstruct along the temporal orientation? Chosen samples will become temporal-to-subband pivots in decoding the subband vector under consideration. The remaining samples will become subband-to-temporal pivots, and this can be justified in light of the geometrical model of Section 2.3.2, as discussed in Section 4.4 later on in this chapter.

To arrive at our CSMSE expressions, we first create an expanded received vector $\underline{\ddot{y}}_{R\dagger}$

by modifying the received vector $\ddot{\underline{y}}_R \in \mathbb{C}^R$ to include the $R_\dagger$ available temporal pivots ($\ddot{\underline{y}}_\dagger$):

$$\begin{aligned} \ddot{\underline{y}}_{R\dagger} &= \begin{bmatrix} \ddot{\underline{y}}_R \\ \ddot{\underline{y}}_\dagger \end{bmatrix} && (4.24) \\ &= \begin{bmatrix} \underline{y}_R \\ \underline{y}_\dagger \end{bmatrix} + \begin{bmatrix} \underline{q}_R \\ \underline{q}_\dagger \end{bmatrix} && (4.25) \end{aligned}$$

where $\underline{q}_\dagger$ represents the pivot error. Note that since the samples in $\ddot{\underline{y}}_\dagger$ are reconstructions themselves, they will have an implicit error beyond quantization (*e.g.*, the error for temporal reconstructions is given by (4.11)). We now only need to obtain the reconstructed vector $\hat{\underline{y}}_{E\dagger}$, and this can be done by forming the reconstruction matrix $\boldsymbol{T}_\dagger$ corresponding to the case when all the samples in $\ddot{\underline{y}}_{R\dagger}$ were available to begin with:

$$\hat{\underline{y}}_{E\dagger} = \boldsymbol{T}_\dagger \ddot{\underline{y}}_{R\dagger} \qquad (4.26)$$

Using $\boldsymbol{t}_{k\dagger}$ for the k-th row of $\boldsymbol{T}_\dagger$, we obtain the pivoted subband CSMSE $\Psi_{SB\dagger}(e_k)$ as follows:

$$\begin{aligned} \Psi_{SB\dagger}(e_k) &= \mathrm{E}\big[\, |\hat{\underline{y}}_{E\dagger\langle k\rangle} - \underline{y}_{E\dagger\langle k\rangle}|^2 \,\big] && (4.27) \\ &= \mathrm{E}\Big[\, |\boldsymbol{t}_{k\dagger} \begin{bmatrix} \underline{q}_R \\ \underline{q}_\dagger \end{bmatrix} |^2 \Big] && (4.28) \\ &= \mathrm{E}\Big[\, \boldsymbol{t}_{k\dagger} \begin{bmatrix} \underline{q}_R \\ \underline{q}_\dagger \end{bmatrix} \begin{bmatrix} \underline{q}_R^{\mathcal{H}} & \underline{q}_\dagger^{\mathcal{H}} \end{bmatrix} \boldsymbol{t}_{k\dagger}^{\mathcal{H}} \Big] && (4.29) \\ &= \boldsymbol{t}_{k\dagger} \mathrm{E} \begin{bmatrix} \underline{q}_R \underline{q}_R^{\mathcal{H}} & \underline{q}_R \underline{q}_\dagger^{\mathcal{H}} \\ \underline{q}_\dagger \underline{q}_R^{\mathcal{H}} & \underline{q}_\dagger \underline{q}_\dagger^{\mathcal{H}} \end{bmatrix} \boldsymbol{t}_{k\dagger}^{\mathcal{H}} && (4.30) \\ &= \boldsymbol{t}_{k\dagger} \underbrace{\begin{bmatrix} \sigma_q^2 \boldsymbol{I} & 0 \\ 0 & \boldsymbol{R}_\dagger \end{bmatrix}}_{\boldsymbol{R}_{R\dagger}} \boldsymbol{t}_{k\dagger}^{\mathcal{H}} && (4.31) \end{aligned}$$

where in (4.31) we have assumed that the set of quantization noise samples are i.i.d. and uncorrelated with the samples of the original codevector; as a result, they are also uncorrelated with the samples of $\underline{q}_\dagger$.

Using (4.31) and (2.28), we can now state an exact condition that can be used in selecting temporal-to-subband pivots (⊿ and ⊿ represent subband-to-temporal and temporal-to-subband pivoting, respectively):

$$\boxed{\boldsymbol{t}_{k\dagger}\begin{bmatrix}\sigma_q^2\boldsymbol{I} & 0\\ 0 & \boldsymbol{R}_\dagger\end{bmatrix}\boldsymbol{t}_{k\dagger}^{\mathcal{H}} \overset{\text{⊿}}{\underset{\text{⊿}}{\gtrless}} \sigma_q^2|t_k|^2} \tag{4.32}$$

The case of a single pivot sample can be easily written in exact form using (4.12): $\boldsymbol{R}_\dagger$ in (4.31) will be a scalar given by $\Psi_w(e_p)$, where we use the index p to denote the pivot position along the temporal erasure burst. We substitute into (4.31) to write the exact condition as follows:

$$\underbrace{\boldsymbol{t}_{k\dagger}\begin{bmatrix}\sigma_q^2\boldsymbol{I} & 0\\ 0 & \Psi_w(e_p)\end{bmatrix}\boldsymbol{t}_{k\dagger}^{\mathcal{H}}}_{\Psi_{SB\dagger}(e_k)} \overset{\text{⊿}}{\underset{\text{⊿}}{\gtrless}} \underbrace{\sigma_q^2|\boldsymbol{t}_k^{\mathcal{H}}|^2}_{\Psi_{SB}(e_k)} \tag{4.33}$$

Note that the row vectors $\boldsymbol{t}_k$, $\boldsymbol{t}_{k\dagger}$ and $\boldsymbol{\tau}_k$ correspond to the rows of the subband, pivoted, and temporal reconstruction matrices, respectively.

4.3.2 Suboptimal Condition for Temporal-to-Subband Pivoting

Given the form of $\boldsymbol{R}_{R\dagger}$ in (4.31) we know that the first R of its singular values are given by σ_q^2; the remaining $R_\dagger$ will be those of $\boldsymbol{R}_\dagger$, which we group to form the diagonal matrix $\boldsymbol{D}_\dagger$. Using $\boldsymbol{U}_{R\dagger}$ for the matrix of eigenvectors of $\boldsymbol{R}_{R\dagger}$, we can rewrite $\Psi_{SB\dagger}(e_k)$ in (4.31) as follows:

$$\begin{aligned}\Psi_{SB\dagger}(e_k) &= \boldsymbol{t}_{k\dagger}\boldsymbol{U}_{R\dagger}\begin{bmatrix}\sigma_q^2\boldsymbol{I} & 0\\ 0 & \boldsymbol{D}_\dagger\end{bmatrix}\boldsymbol{U}_{R\dagger}^{h}\boldsymbol{t}_{k\dagger}^{\mathcal{H}}\\ &= \boldsymbol{t}_{k\ddagger}\begin{bmatrix}\sigma_q^2\boldsymbol{I} & 0\\ 0 & \boldsymbol{D}_\dagger\end{bmatrix}\boldsymbol{t}_{k\ddagger}^{\mathcal{H}}\end{aligned} \tag{4.34}$$

where we have let $\boldsymbol{t}_{k\ddagger} = \boldsymbol{t}_{k\dagger}\boldsymbol{U}_{R\dagger}$. We can again (*cf.* (4.16)) obtain an upper bound for $\Psi_{SB\dagger}(e_k)$ by selecting only the maximum of the singular values:

$$\Psi_{SB\dagger}(e_k) \le \max\{\sigma_q^2; \boldsymbol{D}_{\dagger\langle l,l\rangle} \mid l = 1,\ldots,R_\dagger\}\ \ |\boldsymbol{t}_{k\dagger}^{\mathcal{H}}|^2 \tag{4.35}$$

When one does not use pivoted samples in the reconstruction, the subband CSMSE $\Psi_{SB}(e_k)$ is given by (2.28) as $\sigma_q^2|\boldsymbol{t}_k^{\mathcal{H}}|^2$. We combine this with (4.35) to state a (suboptimal) condition for pivot selection:

$$\max\{\sigma_q^2; \boldsymbol{D}_{\dagger\langle l,l\rangle} \mid l = 1, \ldots, R_\dagger\} \;\; |\boldsymbol{t}_{k\dagger}^{\mathcal{H}}|^2 \overset{\text{⊿}}{\underset{\text{⊿}}{\gtrless}} \sigma_q^2|\boldsymbol{t}_k^{\mathcal{H}}|^2 \tag{4.36}$$

4.4 The Geometrical Interpretation Revisited

In Section 2.3.2 we presented a geometrical interpretation of the encoding operation of a general DFT codevector $\underline{y} = \boldsymbol{G}\underline{x}$ and the related reconstruction operation $\underline{y}_E = \boldsymbol{T}\underline{y}_R$. We found that, in the presence of quantization, the encoding operation corresponded to a parallelogram $\mathbb{P}$ defined by the quantization interval Δ (*cf.* Fig. 2.2). $\mathbb{P}$ was known to contain the original message vector $\underline{x}$. The error resulting from the reconstruction operation was given by the projections of this parallelogram onto the lines defined by the rows of $\boldsymbol{G}$ corresponding to lost samples. The analogy was extended in noting that by cutting down on the length of one or more of these projections, the corresponding dimensions of the parallelogram would be reduced along with its projections (along rows of $\boldsymbol{G}$) related to the remaining reconstructions.

Agreement of geometrical and mathematical model: The pivoting application of the TFBD code presented in the last section provided one possible means of cutting down on the available projections, and we now test (intuitively) how well our geometrical model agrees with the derivations. Consider the case when a single temporal-to-subband pivot sample is available to reconstruct bursts along the subband orientation. The subband erasure burst under consideration is a 1-D burst of length 5 at positions $6, \ldots, 10$, as shown in Fig. 4.3. The single pivot available is located in the 8-th subband. In Fig. 4.3, we plot the reference curve $\Psi_{SB}(e_k)$ corresponding to subband reconstruction with no pivot, given in the right-hand side of (4.33). We also plot $\Psi_{SB\dagger}(e_k)$ (left-hand side of (4.33)) corresponding to the pivoted subband reconstruction for various values of pivot CSMSE (*i.e.,* we sweep the scalar $\Psi_w(e_p)$ in (4.33); this corresponds to varying the position/number of erasures along the temporal codevector providing the temporal-to-subband pivot).

As predicted by our geometrical model, whenever the temporal reconstruction yields a better pivot estimate than the subband reconstruction, all other samples in the burst benefit monotonically from the improved pivot estimate. Also, whenever the temporal reconstruction yields a pivot with the same CSMSE (no change in dimensions of $\mathbb{P}$) as the subband reconstruction, both sides of (4.33) produce the same CSMSE for the remaining burst samples, meaning that the geometrical model of Section 2.3.2 and the mathematical model derived in the present chapter agree. The reconstructed samples, however, can only benefit from the pivot estimate up to a point, as seen in Fig. 4.3, and this prediction of the mathematical model can also be explained in terms of the geometrical model: The single pivot sample can only vary the size of $\mathbb{P}$ along a single dimension. At the limit when there is no estimation error (*i.e.,* estimate converges with the actual value), $\mathbb{P}$ reduces to a line segment, and this segment will still have a non-zero projection onto the remaining rows of $\boldsymbol{G}$.

Temporal-to-subband pivots: In the previous pages we derived expressions that allowed us to select temporal-to-subband pivots for a given subband burst (*e.g.,* (4.33)). The samples that were not selected became subband-to-temporal pivots in carrying out reconstructions along the related temporal codevectors. To show that this yields better reconstructions for the temporal codevectors, we again consider the geometrical model: Selecting the subband-reconstructed samples will decrease the corresponding projected dimension of the parallelogram related to the temporal reconstruction. Hence the remaining projections (and associated reconstruction errors) will also be reduced.

4.5 Summary

We began the present chapter by showing mathematically how one could interpret the temporal code as a DFT code. We then carried out a mathematical analysis that yielded the CSMSE expression for temporal reconstructions. We also derived the CSMSE expression for the related pivoting application, and showed how the expression coincided with the geometrical model of Chapter 2. We will begin the next chapter by verifying our derivations through simulations. Having thus certified the correctness of the expressions, we will analyze them and show how they can be used as design tools in the selection of the many TFBD code parameters.

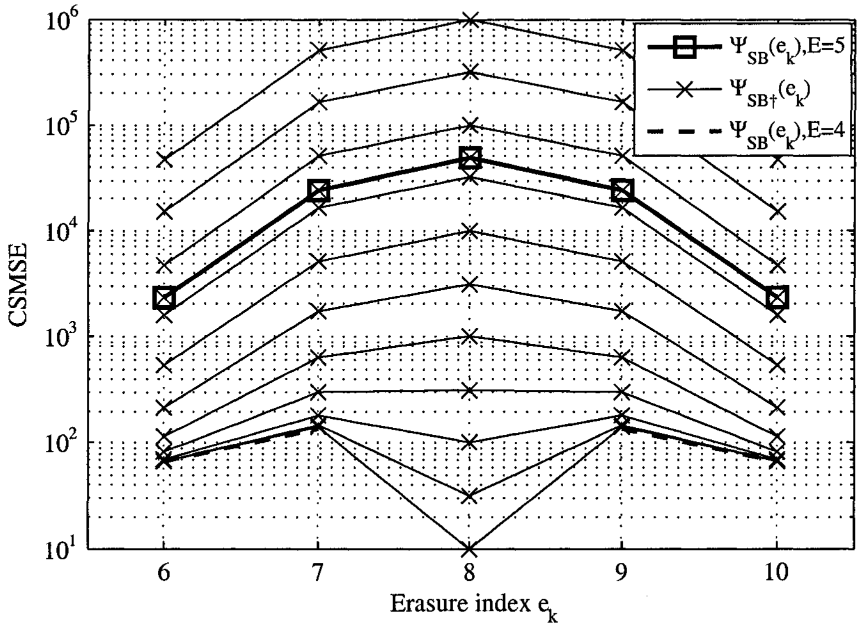

Fig. 4.3 *Pivoting and the geometrical interpretation revisited:* Plot of subband reconstruction CSMSE ($\Psi_{SB}(e_k)$, $E = 5$) vs. pivoted reconstruction CSMSE ($\Psi_{SB\dagger}(e_k)$, as given by left hand side (4.33) with $e_p = 8$) for various values of pivot error; (15, 10, 7) TFBD code. The horizontal axis corresponds to the subband codevector indices e_k. The $\Psi_{SB\dagger}(e_k)$ curves decrease monotonically (over all samples e_k) along with the pivot error up to a limit (heavy dashed line, $\Psi_{SB}(e_k)$, $E = 4$) corresponding to the case when the pivot sample was received.

Chapter 5

Results

We begin the present chapter by verifying the expressions derived in Chapter 4 through simulations. After showing the correctness of the derivations, we use the expressions to construct plots that facilitate the design of TFBD encoders. The analysis will shed light on the selection of TFBD parameters including the temporal parity frequencies, the prototype filter, and the temporal window.

5.1 Experimental Evaluation of Derived Expressions

We begin first by verifying the expression for the stopband correlation matrix given in (4.19) for the case of i.i.d. input message sequence $x[n]$. We then verify the expressions for the temporal CSMSE $\Psi_w(e_k)$ given in (4.12), and that for the pivoted CSMSE $\Psi_{SB\dagger}(e_k)$ given in (4.33). In all cases the simulation results coincide with the derived theoretical values.

5.1.1 Stopband Correlation Matrix

In Fig. 5.1 we provide a sample surface plot of $\boldsymbol{R}_s$ (top of figure) as given in (4.19), along with the corresponding plot of the simulated version $\boldsymbol{R}_{s,sim}$ (bottom of figure). The subband generator matrix $\boldsymbol{G}$ (*cf.* Fig. 3.1 in pg. 32) used was a simple upper-parity DFT generator matrix of size 15×10 (as defined by (1.3)). The prototype filter $f[n]$ used was a Kaiser lowpass filter of length $L_f = 101$, Kaiser parameter $\beta = 2.0019$, and cutoff frequency $f_c = 0.0523$ (where $f = 0.5$ corresponds to angular frequency π). The temporal window used was a rectangular window of unit entries with length $L_w = 11$.

In Fig. 5.2 we plot the relative error between the entries of matrices $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,sim}$ displayed in Fig. 5.1. Fig. 5.2 consists of a surface plot of the relative error matrix $\boldsymbol{R}_{s\Delta}$, defined as having (k, l) element:

$$\boldsymbol{R}_{s\Delta\langle k,l\rangle} = \left| \frac{\boldsymbol{R}_{s\langle k,l\rangle} - \boldsymbol{R}_{s,sim\langle k,l\rangle}}{\boldsymbol{R}_{s,\langle k,l\rangle}} \right| \tag{5.1}$$

The maximum relative error was found to be −12.5 dB, and hence the derived theoretical expression coincides well with the experimental data.

5.1.2 CSMSE of Temporal and Pivoted Reconstructions

To verify the temporal and pivoted CSMSE expressions ((4.12) and left-hand side of (4.33), respectively), we carried out simulations and obtained the experimental CSMSE over 10,000 frames. The input message samples $x[n]$ were taken from a Gaussian distribution with variance 0.9; quantization noise was modeled using additive Gaussian noise with variance $\sigma_q^2 = 7.264 \times 10^{-4}$. For the prototype filter we used a Kaiser filter with Kaiser parameter $\beta = 4$, length $L_f = 201$ and cutoff frequency $f_c = 0.03$ (where $f = 0.5$ corresponds to angular frequency π). The temporal window used was a rectangular window of unit entries with length $L_w = 301$. These parameters correspond to the second of two codes designed later in the chapter. The resulting TFBD code sample power (given by (4.22)) was $\sigma_y^2 = 1.66429$. As explained in Section 4.2.1, this value is an approximation, and the exact values can be obtained from the first K_b entries of matrix $\boldsymbol{R}_\alpha$ for the corresponding subband. For this setup, the largest absolute relative error of the approximation was less than 10^{-5}.

Both Fig. 5.3 and Fig. 5.5 plot the CSMSE versus the codevector sample index for all erased samples. In Fig. 5.3, the results are plotted against the derived theoretical value, given by (4.12), and shown to coincide. A single erasure burst of length $E_t = 5$ starting at codevector index 150 along the temporal orientation (1-D burst) was simulated and reconstructed using the temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$ given by (4.9). Fig. 5.5 plots the simulated CSMSE for a subband erasure burst decoded directly (using $\boldsymbol{T}$ in (2.28)) and with pivoting (using $\boldsymbol{T}_\dagger$ in (4.26)); again the simulated values are shown to coincide with the theoretical, given by (4.33) for both decoding methods. A subband burst of length

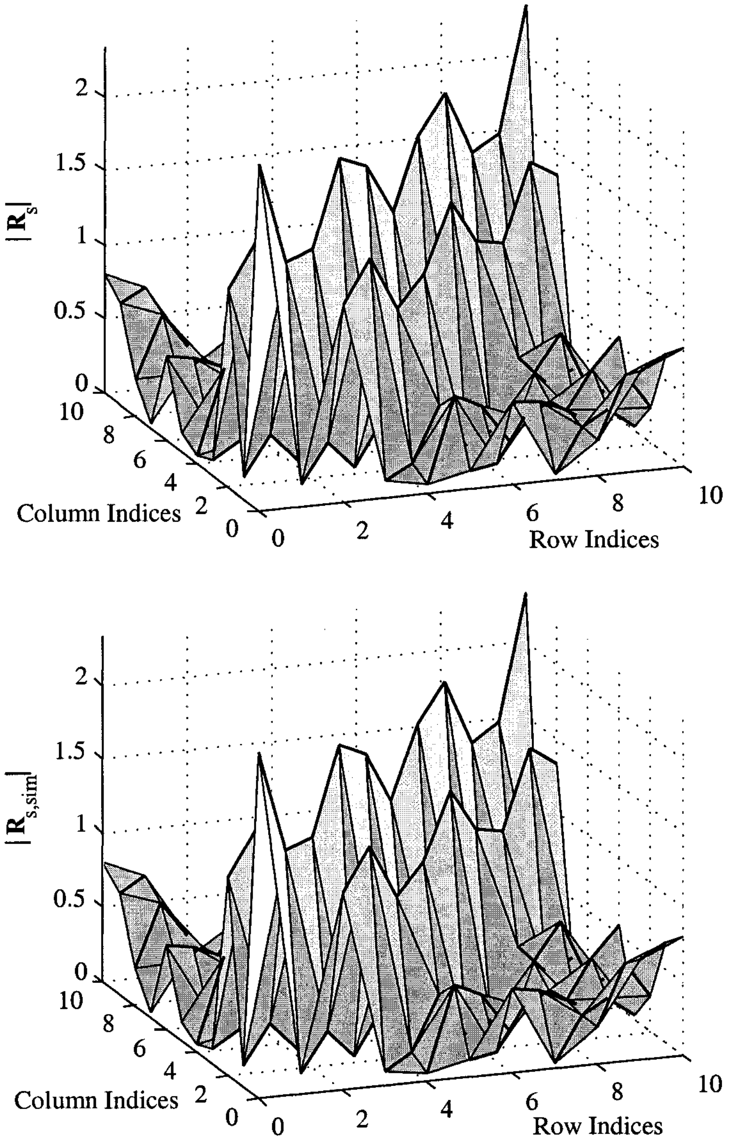

Fig. 5.1 *Stopband Correlation Matrix* $\boldsymbol{R}_s$*:* The view is rotated to look down the main diagonal. The prototype filter $f[n]$ used was a Kaiser lowpass filter of length $L_f = 101$, $\beta = 2.0019$, and $f_c = 0.0523$. The temporal window was rectangular with length $L_w = 11$. Top: Surface plot of magnitude of $\boldsymbol{R}_s$, as given in (4.19). Bottom: corresponding simulated version $\boldsymbol{R}_{s,sim}$ obtained by averaging 100,000 realizations.

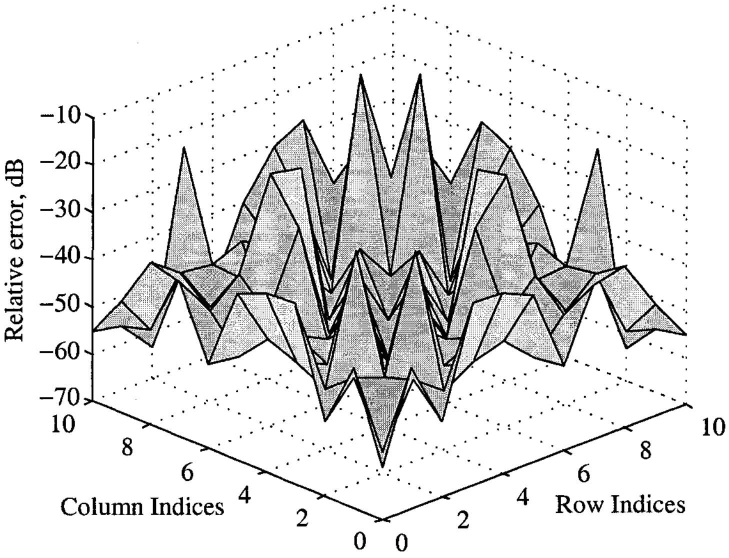

Fig. 5.2 *Simulated vs. derived stopband correlation matrix, relative error:* The relative error is shown as a surface plot, where the element at the k-th row and l-th column is given by $\boldsymbol{R}_{s\Delta\langle k,l\rangle}$ in (5.1). The theoretical correlation matrix $\boldsymbol{R}_s$, and the simulated version $\boldsymbol{R}_{s,sim}$ are the same as in Fig. 5.1. The maximum relative error is -12.5 dB.

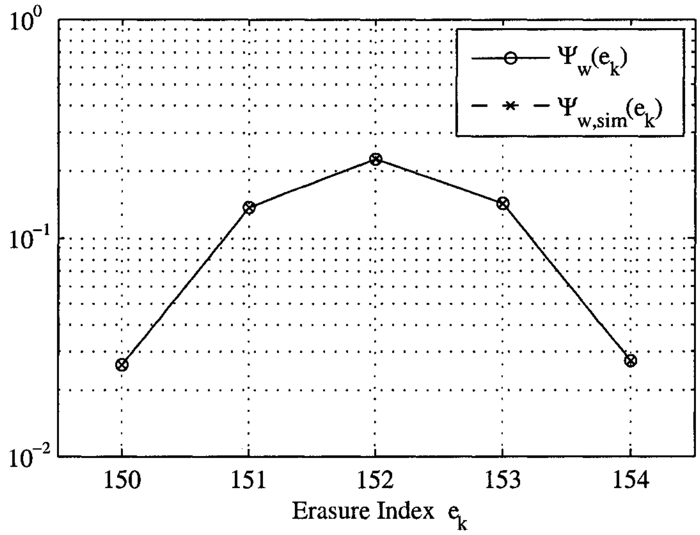

Fig. 5.3 *Experimental evaluation of expression for temporal CSMSE:* Temporal reconstruction error for (15, 10, 7)-TFBD Code, 1-D temporal burst.

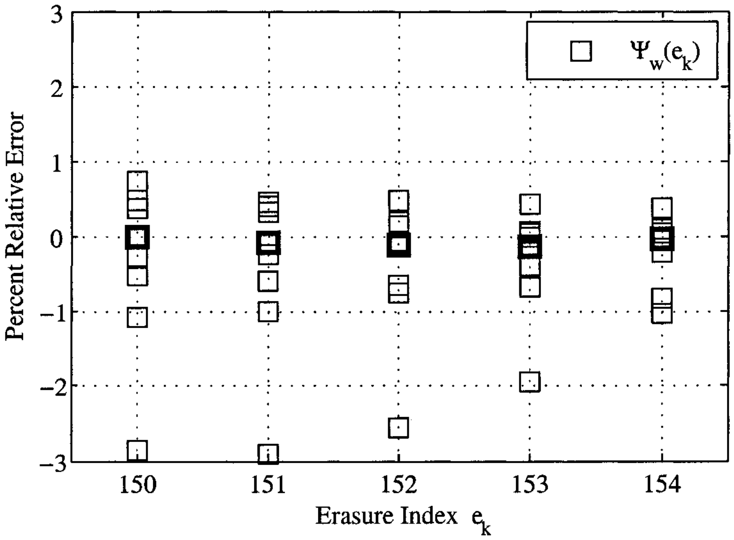

Fig. 5.4 *Relative error of simulated temporal CSMSE:* Relative error of derived CSMSE expression $\Psi_w(e_k)$ vs. simulated values for eight realizations; (15, 10, 7)-TFBD Code, 1-D temporal burst. The bold markers correspond to the values used in Fig. 5.3

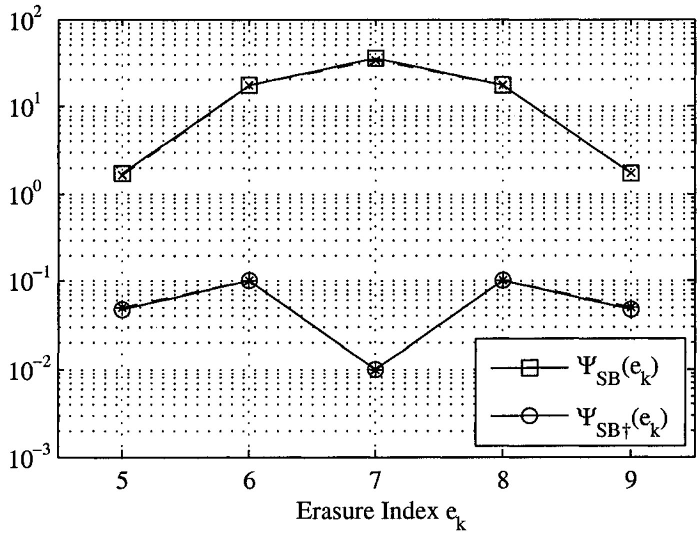

Fig. 5.5 *Experimental evaluation of expression for pivoted CSMSE:* Subband vs. temporal-to-subband pivoted decoding of subband codevector. (15, 10, 7)-TFBD Code, 1-D subband burst. The curves with $\times/*$ markers and dashed lines represent the corresponding simulation values.

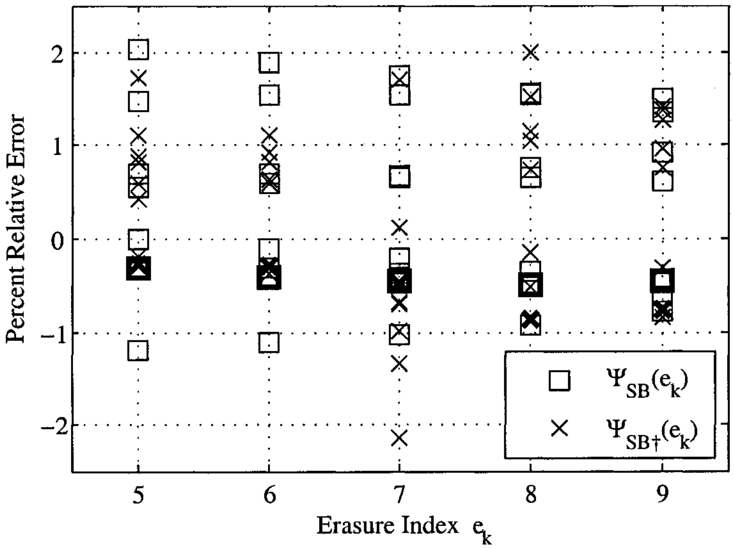

Fig. 5.6 *Relative error of simulated pivoted CSMSE:* Relative error of derived CSMSE expressions $\Psi_w(e_k)$ and $\Psi_{SB\dagger}(e_k)$ vs. simulated values for eight realizations; (15, 10, 7)-TFBD Code, 1-D subband burst. The bold markers correspond to the values used in Fig. 5.5

$E_s = 5$ starting at codevector index 5 was used, and the selected temporal-to-subband pivot was the middle sample from a temporal burst of length $E_t = 3$ located along the seventh subband. The figure illustrates a gain of more than three orders of magnitude for the pivoted sample, and more than one order of magnitude for the samples in the two resulting erasure bursts (relative to the subband TFBD code). Note that the results are only meant to illustrate the correctness of the derivations, as reconstructing all erasures along the temporal orientation would have been optimal.

In Fig. 5.4 and Fig. 5.6, we repeat the experiments of Fig. 5.3 and Fig. 5.5, respectively, seven times each, and plot the relative error for all eight experiments with respect to the derived theoretical values (note that we have used bold markers for the relative errors corresponding to Fig. 5.3 and Fig. 5.5). As can be observed, all simulations are shown to coincide with the theoretical values within a 3% relative error.

5.2 TFBD Code Design

Now that we have experimentally verified the expressions derived in Chapter 4, we show how they can be used to design a TFBD encoder. We will first show how one can find the optimal set of temporal parity frequencies by minimizing the combined stopband component of the temporal CSMSE Ψ_w. As one can expect, the optimal set will be a function of the filterbank's prototype filter $f[n]$. In turn, selecting the prototype filter and temporal window will represent a tradeoff between the reconstruction CSMSE Ψ_w of the temporal code and the distortion introduced by the filterbank component of the TFBD (Tandem Filterbank / DFT) code. We will represent this tradeoff graphically as a family of curves obtained by varying the prototype filter and temporal window.

The Ψ_w/distortion family of curves will also be used to compare the magnitude of Ψ_w relative to that of (i) the subband CSMSE and (ii) the CSMSE of the (N, K) DFT code that we use as a comparison benchmark. For pivoting operations to be applicable, it is necessary that reconstructions along both orientations yield comparable error magnitudes, and thus insight into the CSMSE along both orientations is necessary. Regarding the benchmark (N, K) DFT code: such code will be of the same rate as the (N, K_b, K) TFBD code in question and is thus the appropriate candidate to gauge the performance of the TFBD code.

To finalize the chapter, we will select two TFBD codes and compare their pivoted

CSMSE to that of the corresponding benchmark (N, K) DFT code. The comparison will be made as a function of both the subband and temporal burst lengths. The results will show that for sufficiently long subband burst lengths, the TFBD code will be capable of improvements of more than four orders of magnitude.

5.2.1 Selection of Temporal Parity Frequencies

In Chapter 3 we discussed how the temporal code could be interpreted as a DFT code; the parity frequencies corresponded to the stopband of the temporal codevectors, and using this one could build a temporal generator matrix $\boldsymbol{\mathcal{G}}$ and an associated temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$. The question then is how to select the set of parity frequencies. The solution can be formulated as an optimization problem with the help of the temporal CSMSE expression $\Psi_w(e_k)$. One possible approach is to minimize some function of $\Psi_w(e_k)$ (*e.g.*, component term, maximized over k), over all sets of parity frequencies that are contiguous and centered at $f = 0.5$.[1] Applying this constraint on the possible sets of parity frequencies makes sense since we know that the temporal codevectors have a lowpass spectral shape. The optimization is thus reduced to a search (of linear complexity) over the number of elements P_w in the set of temporal parity frequencies.

In Section 4.1.2 we derived the temporal CSMSE Ψ_w; we reproduce it here for convenience, and further define the cross-term $X_{s\tau}$ and the *combined stopband component* $E_{SB,c}$ of Ψ_w:

$$\begin{aligned}
\Psi_w(e_k) &= \sigma_s^2(e_k) + \underbrace{\boldsymbol{\tau}_k \boldsymbol{R}_{Rs} \boldsymbol{\tau}_k^{\mathcal{H}}}_{e_s(k)} - \underbrace{2\mathrm{Re}\big[\, \boldsymbol{R}_{s\langle e_k,\{r_i|i=1,\ldots,R\}\rangle} \boldsymbol{\tau}_k^{\mathcal{H}} \,\big]}_{X_{s\tau}(k)} \\
&+ \underbrace{\sigma_q^2 \boldsymbol{\tau}_k \boldsymbol{D}_{Rw}^2 \boldsymbol{\tau}_k^{\mathcal{H}}}_{e_q(k)}, \qquad (5.2a) \\
E_{SB,c}(k) &= \sigma_s^2(e_k) + e_s(k) + X_{s\tau}(k). \qquad (5.2b)
\end{aligned}$$

Here $\sigma_s^2(e_k)$ is the stopband power corresponding to the k-th erased sample; $\boldsymbol{R}_s$ is the correlation matrix of the stopband component $\underline{y}_s^w$ of the temporal codevector $\underline{y}^w$; $\boldsymbol{R}_{Rs}$ is the submatrix of $\boldsymbol{R}_s$ corresponding to the samples of $\underline{y}_s^w$ with received indices; and $\boldsymbol{D}_{Rw}^2$

[1] The more general problem would be the minimization over all possible combinations (or sets) of L_w-DFT frequencies, but the number of possible sets grows very quickly. For temporal window lengths of size $L_w = 101$, there are about 2.54×10^{30} possible combinations.

is a diagonal matrix with squared magnitudes of window entries corresponding to received samples along its diagonal. Note that the combined stopband component $E_{SB,c}$ sums up the terms of Ψ_w that originate from the parity frequency component of temporal codevectors, and thus, its magnitude is specified by the set of temporal parity frequencies.

In this work we use the following strategy in selecting the set of temporal parity frequencies: as discussed above, we constrain our search to sets of parity frequencies that are contiguous and centered at $f = 0.5$, and thus we need only find the best set size P_w. The function that we minimize to find this best P_w is the maximum over k of the combined stopband component ($E_{SB,c}$) of $\Psi_w(e_k)$. The justification behind this is as follows: the temporal code is an approximation to a related $(L_w, L_w - P_w)$ DFT code. The combined stopband component $E_{SB,c}$ is missing in the CSMSE expression of the related DFT code (*cf.* (2.28)) and thus we know that the temporal code will always perform worse than the DFT code by at least the magnitude of $E_{SB,c}$.[2]

In Fig. 5.7 we plot the maximum over k of $\Psi_w(e_k)$, its components $e_q(k)$ and $E_{SB,c}(k)$, and the subcomponents of $E_{SB,c}(k)$, $e_s(k)$, $\sigma_s^2(e_k)$, and $X_{s\tau}(k)$. We plot these terms as a function of the number of elements P_w in the set of parity frequencies, using a fixed temporal burst length of $E = 4$ erasures. We will refer to these as *max-vs-parity* curves, as they portray the maximum of the corresponding expression versus the number P_w of parity frequencies. The minimum value possible for P_w is E, the number of erasures, since a DFT code can correct at most P_w erasures; the maximum value is $L_w - 1$, as this corresponds to the minimum code rate of $1/L_w$.[3] The parameters used in the figure are as follows (all parameters were chosen for illustrative purposes): The prototype filter $f[n]$ used was a Kaiser lowpass filter of length $L_f = 201$, cutoff frequency $f_c = 0.0528$ and Kaiser parameter $\beta = 10$. The temporal window was a Hanning window of length 201, the message sequence power σ_x^2 was set to 0.9, and the quantization noise was modeled as zero mean Gaussian noise with variance $\sigma_q^2 = 1.5259 \times 10^{-5}$. As mentioned previously, parity frequencies are

[2] The quantization noise component of the temporal code ($\sigma_q^2\, \boldsymbol{\tau} \boldsymbol{D}_{Rw}^2 \boldsymbol{\tau}^{\mathcal{H}}$, *cf.* (4.12)) is different to the CSMSE of the related $(L_w, L_w - P_w)$ DFT Code ($\sigma_q^2\, |\boldsymbol{\tau}_k^{\mathcal{H}}|^2$, *cf.* (2.28)). However, one can also apply the window $w[n]$ to the corresponding received samples of the DFT code, and then the resulting CSMSE expression would become the same as that for the temporal code. Hence, since the temporal CSMSE $\Psi_w(e_k)$ suffers from an additional combined stopband error component, we can assert that the related DFT code performs better.

[3] An (N, K) DFT code has $P = N - K$ parity frequencies and a rate of K/N. The minimum code rate for a given N is $1/N$ (*i.e.*, $K = 1$). The temporal code can be seen as an $(L_w, L_w - P_w)$ DFT code, with minimum rate occurring at $L_w - P_w = 1$, or $P_w = L_w - 1$.

constrained to be contiguous and centered at $f = 0.5$.

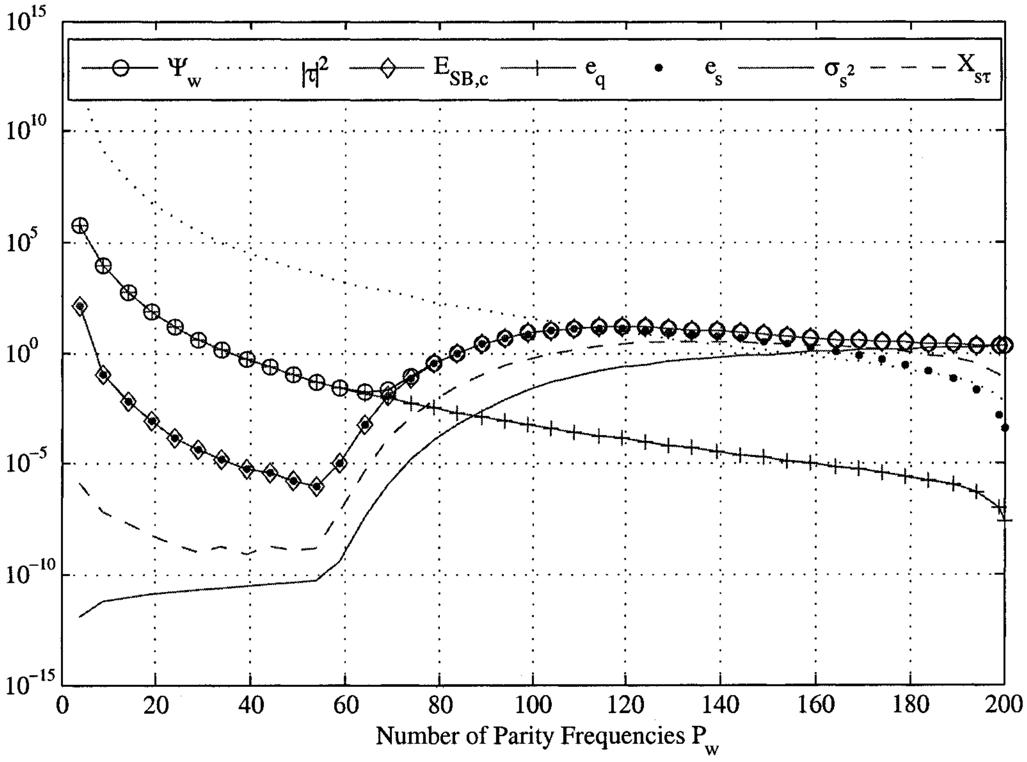

Fig. 5.7 *Max-vs-parity curves:* Maximum over k of $\Psi_w(e_k)$, $E_{SB,c}(k)$, $e_q(k)$, $e_s(k)$, $\sigma_s^2(e_k)$, and $X_{s\tau}(k)$ (*cf.* (5.2)). The horizontal axis corresponds to the number of temporal parity frequencies P_w. See Fig. 5.8 for the corresponding plot of the temporal codevector power spectrum.

We refer first to the plot of e_q (*cf.* Fig. 5.7): as the number of parity frequencies is increased, e_q decreases monotonically, and this is expected as the row magnitudes of $\boldsymbol{\mathcal{T}}$ decrease very rapidly with increased number of parity frequencies [9]. The shape of e_s can be explained in light of the lowpass spectral shape of the temporal codevectors $\underline{y}^w$: As long as parity frequencies are added (*i.e.*, P_w is increased) within the stopband of $\underline{y}^w$, the reduction in row magnitudes of $\boldsymbol{\mathcal{T}}$ sets the trend ($e_s(k) = \boldsymbol{\tau}_k \boldsymbol{R}_s \boldsymbol{\tau}_k^{\mathcal{H}}$, *cf.* (5.2)). But as soon as parity frequencies begin to fall outside the stopband, the increasing magnitude of the stopband correlation matrix $\boldsymbol{R}_s$ begins to set the trend.[4] The transition from stopband to passband

[4] Here we use the *magnitude of a matrix* loosely, but it can be defined, for example, as the sum of squares of its diagonal elements, or as its spectral norm.

drastically changes the magnitude of $\boldsymbol{R}_s$, and this is seen as a rapid increase in e_s following the optimal P_w (found at $P_w = \mathit{56}$). Following this transition, added parity frequencies fall in the passband $\underline{y}^w$, and the magnitude of $\boldsymbol{R}_s$ is affected less drastically as parity frequencies of comparable power content are appended to the set of parity frequencies. Thus we can see a decreased slope for e_s, until eventually the diminishing slope is overcome by the decreasing row magnitudes of $\boldsymbol{\mathcal{T}}$.

Since the cross term $X_{s\mathcal{T}}$ is also composed of (a submatrix of) $\boldsymbol{R}_s$ and rows of $\boldsymbol{\mathcal{T}}$, we can expect that it will also follow a trend similar to that of e_s. This can be verified in the corresponding max-vs-parity curve in Fig. 5.7. The stopband power component σ_s^2 also exhibits the drastic change in slope following the optimal P_w, and the diminishing slope as the parity frequencies venture into the passband. However, as one would expect, increasing the frequency components of the stopband vector $\underline{y}_s^w$ increases the power of its entries monotonically, and this characteristic is also shown in the plot. Note that, in this example, both the cross term $X_{s\mathcal{T}}$ and the stopband power σ_s^2 contribute very little to the net temporal CSMSE Ψ_w. Under this situation, the passband CSMSE Ψ_p (*cf.* (4.12)) will be a good approximation to Ψ_w.

In Fig. 5.8 we have plotted the power spectrum of the temporal codevector and highlighted some selected sets of parity frequencies. Each set is highlighted with a color bar of alternating shade and unique thickness. Color bars are portrayed as layered on top of each other, the lower (thinner) layers highlighting sets of parity frequencies with more elements (higher P_w); the visible areas of each layer highlight the first few and last few parity frequencies for that value of P_w. The circled parity frequencies are the first and last frequencies in the optimal set (that minimizing the max-vs-parity $E_{SB,c}$ curve of Fig. 5.7). Only selected values are shown for P_w ($P_w = 11, 26, 41, \mathit{56}, 61, 66, 71, 76, 81$, optimum value is italicized). Note that sets shown that are smaller (in number of elements) than the optimal set have a difference in P_w of 15 ($P_w = 11, 11+15, \ldots, \mathit{56}$). For larger sets we use a difference in P_w of 5 ($P_w = \mathit{56}, 56+5, \ldots, 81$). This uneven spacing in P_w is done to illustrate how fast the power spectrum grows with P_w after the optimal value, and this again explains the ∪-concave peak in e_s (and $X_{s\mathcal{T}}$) in Fig. 5.7.

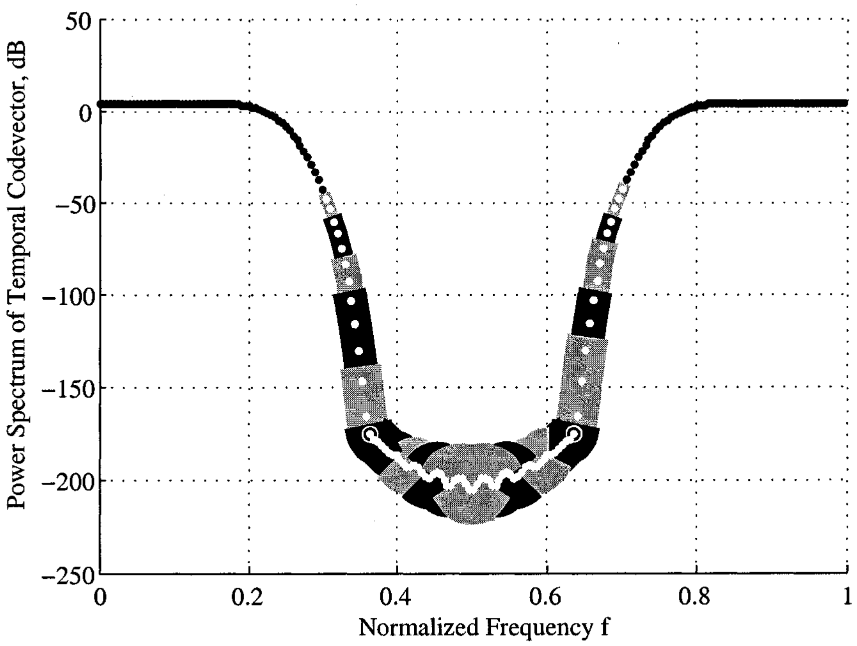

Fig. 5.8 *Selection of temporal parity frequencies:* Plot of power spectrum of temporal codevector ($\underline{S}_{w\langle k\rangle}$ in (4.23) vs. $f = k/L_w$, $L_w = 201$, $k = 0, \ldots, L_w-1$) corresponding to the code in Fig. 5.7. The power spectrum at parity frequency sets corresponding to $P_w = 11, 26, \ldots, \mathit{56}$ (optimum), $61, \ldots, 81$ are highlighted with color bars of different shade/thickness, portrayed as layered on top of each other (lower, thinner layers correspond to higher P_w). The circled points correspond to $\underline{S}w\langle k\rangle$ at k values corresponding to the first and last parity frequencies in the optimal set.

5.2.2 The Prototype Filter and Temporal Window: CSMSE vs. Filterbank Distortion

The selection of a TFBD code's prototype filter $f[n]$ and temporal window $w[n]$ can be seen as a tradeoff between the temporal CSMSE (Ψ_w) capability of the code and the filterbank's distortion. In uniform filterbank design, the optimal prototype filter can be selected from a family of filters (*e.g.,* Kaiser, windowed filters) by minimizing the filterbank's output distortion (see, for example, the work by Yiu and Grbić [19]). The temporal CSMSE Ψ_w, on the other hand, benefits from stopbands with larger widths and greater attenuation. Wider stopbands allow for greater number of parity frequencies, resulting in smaller row-magnitudes for the temporal reconstruction matrix [9]; greater attenuation will reduce the stopband component of Ψ_w.

In the following discussion we begin by defining a filterbank's amplitude distortion A_d and aliasing power A_p. We will then use these, along with the expressions derived in Chapter 4, to construct plots that will prove useful in selecting the Ψ_w/distortion TFBD tradeoff.

Quantifying Filterbank Distortion: Referring to Fig. 3.2, the first stage of the TFBD encoder consists of the analysis stage of a filterbank. Since the effect of the modulators and DFT generator matrices are reverted (assuming no erasures) at the decoder, the resulting received message sequence $\hat{x}[m]$ (*cf.* Fig. 1.1) can be seen as the output of an oversampled uniform filterbank with N subbands and a downsampling factor of K_b. One can express $\hat{x}[m]$ in the z-domain as [19]

$$\hat{X}(z) = \sum_{l=0}^{K_b-1} E_l(z) X(zW_{K_b}^l),$$

where $E_l(z)$ is defined as:

$$E_l(z) = \frac{1}{K_b} \sum_{k=0}^{N-1} F(zW_N^k W_{K_b}^l) F^*(zW_K^k).$$

With this, we can define the filterbank's amplitude distortion A_d and aliasing power A_p as follows [19]:

$$A_d = \frac{1}{2\pi K_b}\int_{-\pi}^{\pi}\left(1-\left|E_0(e^{-j\omega})\right|^2\right)^2 d\omega, \tag{5.3a}$$

$$A_p = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{l=1}^{K_b-1}\left|E_l(e^{-j\omega})\right|^2 d\omega. \tag{5.3b}$$

Temporal CSMSE / Filterbank Distortion Tradeoff: A useful tool in selecting the TFBD code tradeoff between temporal CSMSE and filterbank output distortion is a family of plots displaying the attainable tradeoff points as a function of the prototype filter and temporal window used. The plots will account for the temporal CSMSE side of the tradeoff by plotting (i) the *minimax* $E_{SB,c}$ (minimum of $E_{SB,c}$ in (5.2) over its max-vs-parity curve; see Fig. 5.7 for a sample $E_{SB,c}$ max-vs-parity curve) and (ii) the corresponding maximum row-magnitude of the temporal reconstruction matrix. The filterbank distortion side of the tradeoff will be accounted for by plotting A_d and A_p. For all curves (minimax $E_{SB,c}$, row magnitudes of $\mathcal{T}$, A_d, and A_p), the independent variable will correspond to a parameter of the prototype filter (*e.g.*, the cutoff frequency f_c).

One such family of plots is displayed in Figs. 5.9-5.12. Each figure consists of four plots, and the horizontal axis in each plot is the value of the (normalized) prototype filter cutoff frequency f_c. All cases correspond to an $(15, 10, 7)$-TFBD code with simple upper-parity subband generator matrices $\boldsymbol{G}$ (as defined by (1.3)); all prototype filters used are Kaiser lowpass filters. The four plots in each figure correspond to Kaiser β parameter values $\beta = 4, 6, 8$ and 10. The first figure (Fig. 5.9) uses a filter length of $L_f = 101$; the remaining figures use $L_f = 201$. All figures use Hanning windows except the last one (Fig. 5.12), which uses a rectangular window. The window length for the first two plots is $L_w = 101$; the last two plots use $L_w = 301$.

All the minmax $E_{SB,c}$ curves in the family of plots display a monotonically increasing trend. The reason for this is that for the values of f_c that are displayed, the stopband attenuation of $f[n]$ increases along with the cutoff frequency, resulting in temporal codevectors with larger parity frequency components. Another trend can be observed for the $|\boldsymbol{\tau}|^2$ curves. Note that the $|\boldsymbol{\tau}|^2$ values displayed correspond to the max-vs-parity $|\boldsymbol{\tau}|^2$ curve (*cf.* Fig. 5.7) at the optimal P_w (value of P_w yielding the minimax $E_{SB,c}$). In general, the $|\boldsymbol{\tau}|^2$ curves increase until peaking near the center of the f_c-axis, where they drop and then remain constant. The reason for this shape is as follows: Lower cutoff frequencies result in larger stopbands, and thus greater number of temporal parity frequencies P_w. As the cutoff frequency is increased, the number of parity frequencies decreases, and thus the increase in $|\boldsymbol{\tau}|^2$. The cutoff frequency is increased to the point where a stopband is hardly discernible. The corresponding $E_{SB,c}$ max-vs-parity curve for this situation would not exhibit a pronounced dip for values of P_w corresponding to the stopband (*cf.* Fig. 5.7), and hence the minimum $E_{SB,c}$ along the curve would occur at the rightmost end of the P_w-axis (*cf.* Fig. 5.7), where the lower value of $E_{SB,c}$ is dictated by the decreasing row magnitudes of $\boldsymbol{\mathcal{T}}$. This is the reason for (i) the sudden drop in $|\boldsymbol{\tau}|^2$ throughout the family of plots, and (ii) for its constant value following this drop (*i.e.*, P_w becomes fixed at its maximum value of $L_w - 1$).

Thus, when constructing a family of plots such as the one in Figs. 5.9-5.12, one can be sure of having reached the maximum value of interest for f_c when $|\boldsymbol{\tau}|^2$ falls to a constant value (or equivalently, when the optimal P_w equals its maximum of $L_w - 1$). The minimum value of interest for f_c can be set by defining the maximum acceptable amplitude distortion ($A_d = -10$ dB, in our examples).

Quantizer Selection: Both the temporal CSMSE and the subband CSMSE contain a term in σ_q^2 (quantization noise power) and $\boldsymbol{\tau}_k$ or $\boldsymbol{t}_k$ (rows of temporal reconstruction matrix $\boldsymbol{\mathcal{T}}$ and subband reconstruction matrix $\boldsymbol{T}$, respectively). The quantization noise power thus needs to be small enough to overcome the magnitudes of rows of both reconstruction matrices. For the case of Ψ_w, the combined stopband error component $E_{SB,c}$ will constitute a lower bound on the usefulness of reducing σ_q^2 (*cf.* Fig. 5.7). In order for pivoting operations to be applicable, the CSMSE attainable along subband and temporal orientations must be of comparable magnitudes. In this sense, $E_{SB,c}$ can also be seen as a lower bound on subband CSMSE improvement resulting from reduced quantization power σ_q^2.

Hence, the choice of quantization power σ_q^2 can be determined from the magnitudes of rows of subband and temporal reconstruction matrices $\boldsymbol{T}$ and $\boldsymbol{\mathcal{T}}$, and the combined stopband component $E_{SB,c}$. For this purpose, along with the minimax $E_{SB,c}$, we display the corresponding $|\boldsymbol{\tau}|^2$ curves in the family of plots of Figs. 5.9-5.12, as well as the maximum row magnitude of $\boldsymbol{T}$ (dot-dashed line of constant value). We also plot the maximum reconstruction matrix row magnitude for the (N, K) DFT code that we use as our evaluation benchmark (dashed line, also of constant value). In selecting code parameters, we intend to balance the reconstruction performance of the temporal code to that of the subband code. For this reason we use a constant subband/temporal erasure burst length of 4 for all reconstruction matrices in the family of plots (including that for the benchmark DFT code). Note that for a $(15, 10, 7)$ TFBD code, the subband code can reconstruct subband bursts of at most length 5.

Effect of the Filter Length L_f: As one would expect, longer prototype filters can result in better amplitude distortion / Ψ_w tradeoffs. We illustrate this by comparing the codes in Fig. 5.9 and Fig. 5.10. Both figures correspond to the same TFBD code, but in Fig. 5.9 we use a filter length of $L_f = 101$, while in Fig. 5.10 we use a filter length of $L_f = 201$. With the shorter filter, one can achieve an amplitude distortion A_d of about -15 dB at a minimax $E_{SB,c}$ of -10 dB (for the cases $\beta = 4$ and 6). With a length 201 filter, it is possible to achieve A_d as low as -25 dB for the same $E_{SB,c}$ ($\beta = 6$ and 8). The increased length thus allows an improvement in filterbank amplitude distortion of -10 dB at a fixed $E_{SB,c}$ of -10 dB.

Effect of Temporal Window: The temporal window also controls the spectral shape of temporal codevectors: increasing its length L_w will result in temporal codevectors with lower stopband frequency components. Thus increasing L_w should result in a similar effect to that achievable by increasing the prototype filter length L_w. This expected trend is confirmed in Fig. 5.10 and Fig. 5.11; therein we vary the length L_w of the window ($L_w = 101$ and 301, respectively, Hanning window for both cases), while holding all other parameters fixed. As can be observed (Fig. 5.10, $\beta = 8$) at an A_d of about -26 dB, one can achieve an $E_{SB,c}$ slightly above -10 dB. For the longer window (Fig. 5.11, $\beta = 10$), it is possible to achieve an $E_{SB,c}$ that is 10 dB lower, at about the same $A_d = -26$ dB.

It is not difficult to deduce the effect of using a rectangular window in place of the Hanning window: the rectangular window will result in larger temporal codevector stopband components, and with this an increase in $E_{SB,c}$. We provide support for this in Fig. 5.12, where we have taken the setup of Fig. 5.11 and replaced the Hanning window by a rectangular window of the same length. While the Hanning window makes it possible to achieve $E_{SB,c}$ values close to -20 dB at A_d close to -26 dB (Fig. 5.11, $\beta = 10$), at $E_{SB,c}$ values close to -20 dB, the rectangular window yields an A_d at about -11 dB (Fig. 5.12, all values of β), or 15 dB worse.

5.2.3 Comparison to (N, K) DFT Code

We now compare the reconstruction performance of two (N, K_b, K) TFBD codes selected from the family of plots in Figs. 5.9-5.12, to that of the corresponding (N, K) DFT codes. We use (N, K) DFT codes as comparison benchmarks because they operate at the same code rate K/N as the related (N, K_b, K) TFBD codes. The DFT code, however, can only reconstruct samples along the subband orientation, while the TFBD code can also reconstruct samples along the temporal orientation. As can be expected, the reconstruction performance of both codes will depend on the dimensions of the 2-D burst.

The two codes we select for evaluation are those specified by dashed circles in Fig. 5.11 and Fig. 5.12. Both codes are $(15, 10, 7)$ TFBD codes and use window lengths of $L_w = 301$. The first uses a Hanning window and the second a rectangular window. In both cases a Kaiser lowpass filter was used as the filterbank prototype filter $f[n]$, with Kaiser parameter $\beta = 6$ and $f_c = 0.04$ for the first case, and $\beta = 4$ and $f_c = 0.03$ in the second case. The resulting amplitude distortions for both cases where $A_d = -15$ dB and $A_d = -11$ dB, respectively. The resulting values for TFBD code sample power σ_y^2 (as given by (4.22)) where 2.23 and 1.66, respectively. As explained in Section 4.2.1, the expression for σ_y^2 in (4.22) is an approximation, and the exact power for a given TFBD code sample can be obtained from the first K_b entries along the diagonal of $\boldsymbol{R}_\alpha$ for the corresponding subband. For the setups under consideration, the approximation will display an absolute relative error better than 10^{-7} and 10^{-5}, respectively.

To test the performance of the selected TFBD codes, we compare the temporal CSMSE Ψ_w of a given sample against the CSMSE of $(15, 7)$ DFT code reconstructions of the same sample. The 2-D burst used for the comparison had a cross formation, as displayed in Fig. 5.13, where the reconstructed sample used for the comparison is marked with an × symbol (the cross intersection sample). The cross intersection occurred at the middle position along each burst (*i.e.,* odd burst length: middle sample along burst; even burst length: extra lost sample below or to the right of the intersection). The example provided in Fig. 5.13 uses an odd subband burst length and even temporal burst length.

In Fig. 5.14 and Fig. 5.15, we plot the ratio of Ψ_w to the CSMSE of the DFT code (given by (2.28) [9]) as a surface plot, using the subband and temporal burst lengths ($L_{B,T}$

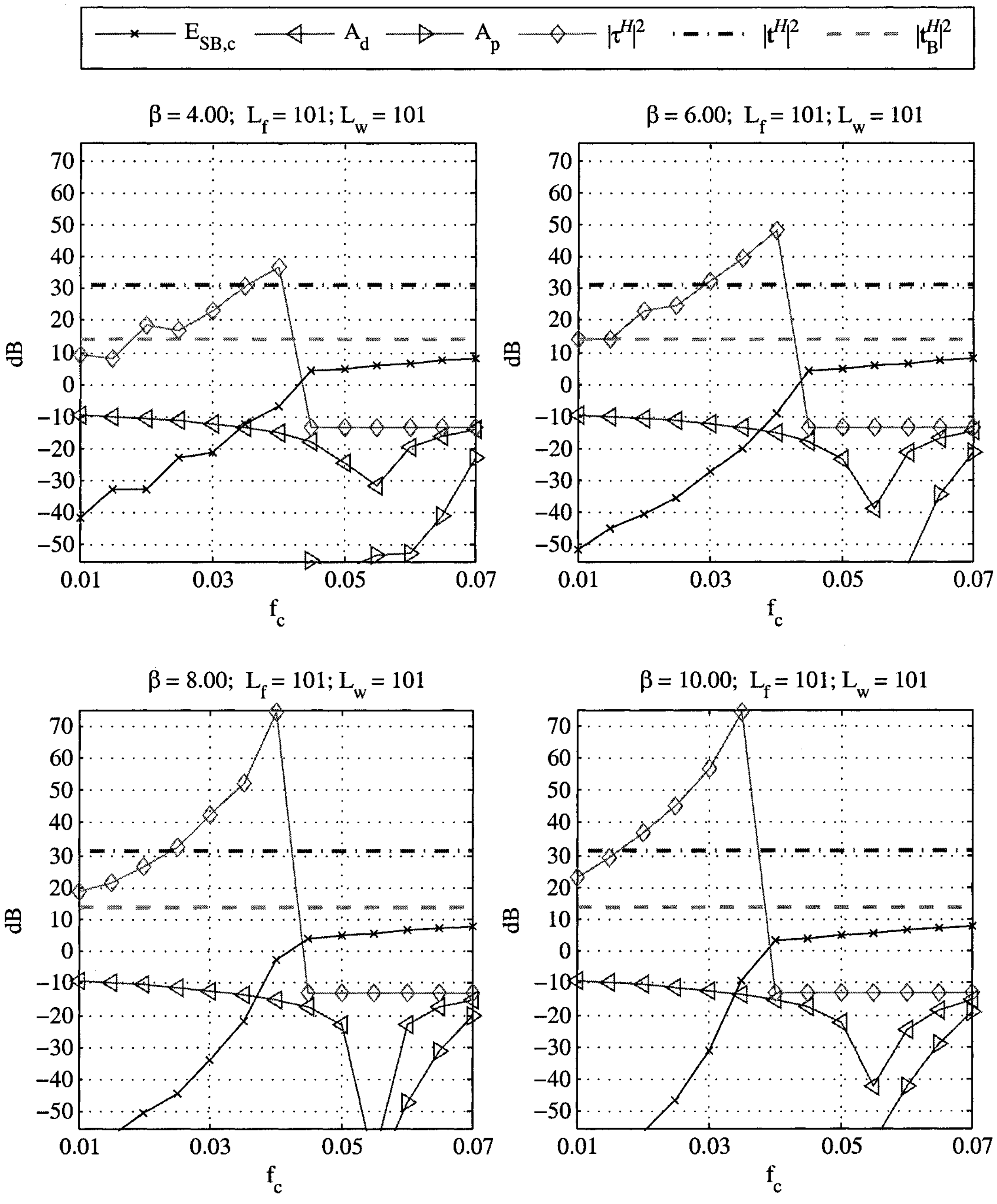

Fig. 5.9 *TFBD code design (Kaiser, $L_f = 101$; Hanning, $L_w = 101$):* Filterbank amplitude distortion (A_d) and aliasing power (A_p), vs. minimax stopband component ($E_{SB,c}$) and maximum row magnitudes of reconstruction matrices for temporal code ($|\tau^{\mathcal{H}}|^2$), subband code ($|t^{\mathcal{H}}|^2$), and benchmark DFT code ($|t_B^{\mathcal{H}}|^2$). A burst length of $E = 4$ was used in all cases.

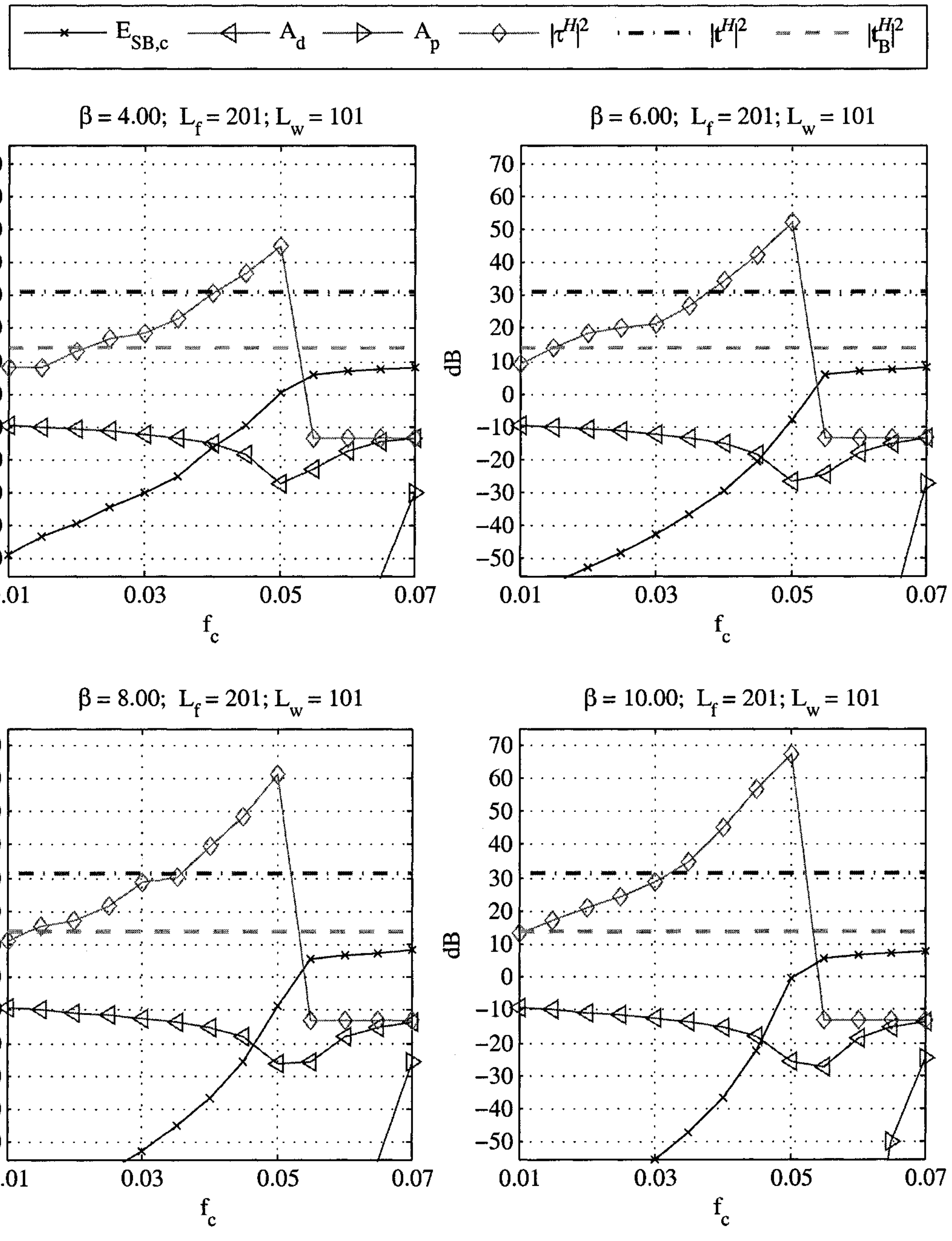

Fig. 5.10 *TFBD code design (Kaiser, $L_f = 201$; Hanning, $L_w = 101$):* Filterbank amplitude distortion (A_d) and aliasing power (A_p), vs. minimax stopband component ($E_{SB,c}$) and maximum row magnitudes of reconstruction matrices for temporal code ($|\tau^{\mathcal{H}}|^2$), subband code ($|t^{\mathcal{H}}|^2$), and benchmark DFT code ($|t_B^{\mathcal{H}}|^2$). A burst length of $E = 4$ was used in all cases.

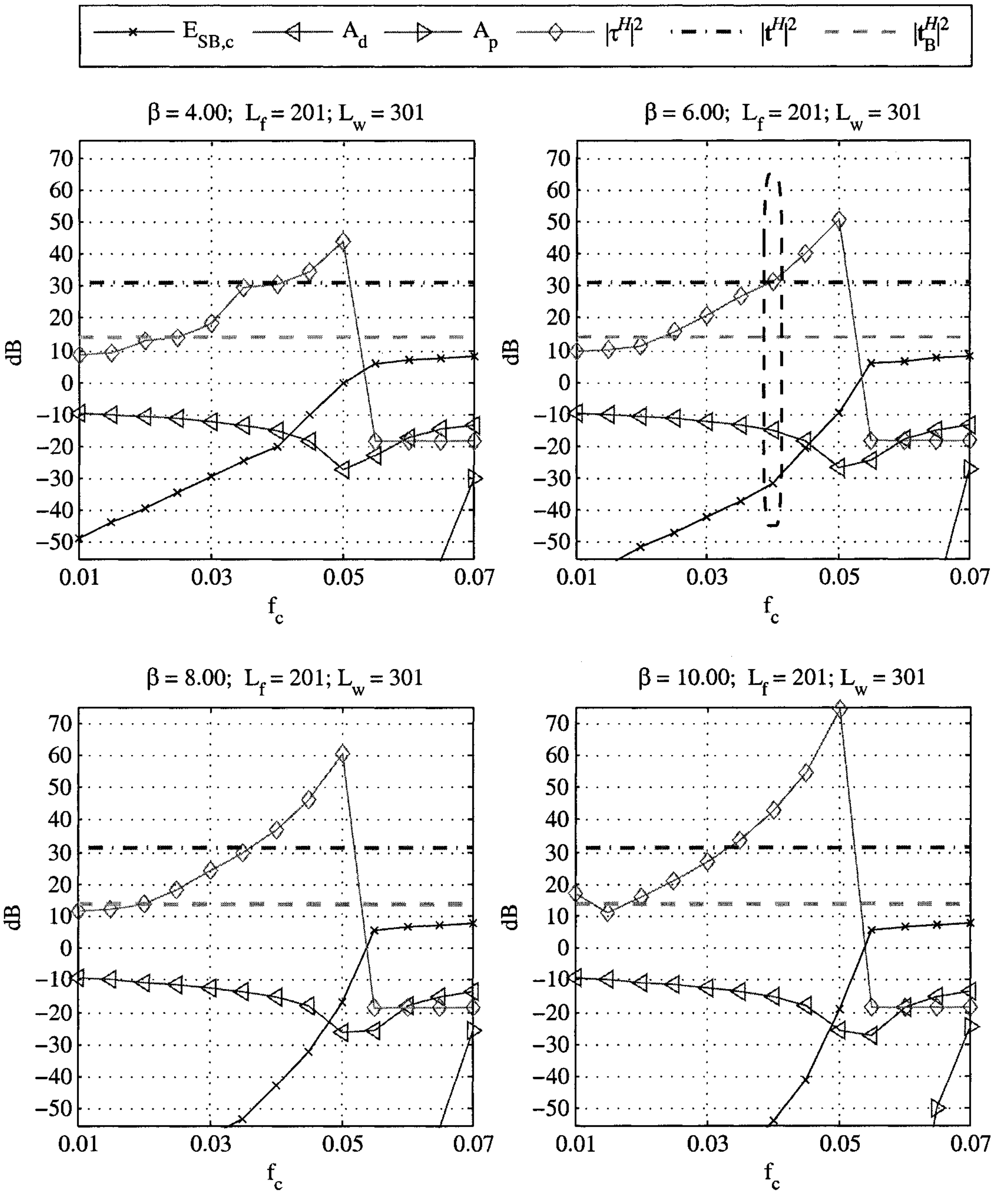

Fig. 5.11 *TFBD code design (Kaiser, L_f = 201; Hanning, L_w = 301):* Filterbank amplitude distortion (A_d) and aliasing power (A_p), vs. minimax stopband component ($E_{SB,c}$) and maximum row magnitudes of reconstruction matrices for temporal code ($|\tau^{\mathcal{H}}|^2$), subband code ($|t^{\mathcal{H}}|^2$), and benchmark DFT code ($|t_B^{\mathcal{H}}|^2$). A burst length of $E = 4$ was used in all cases.

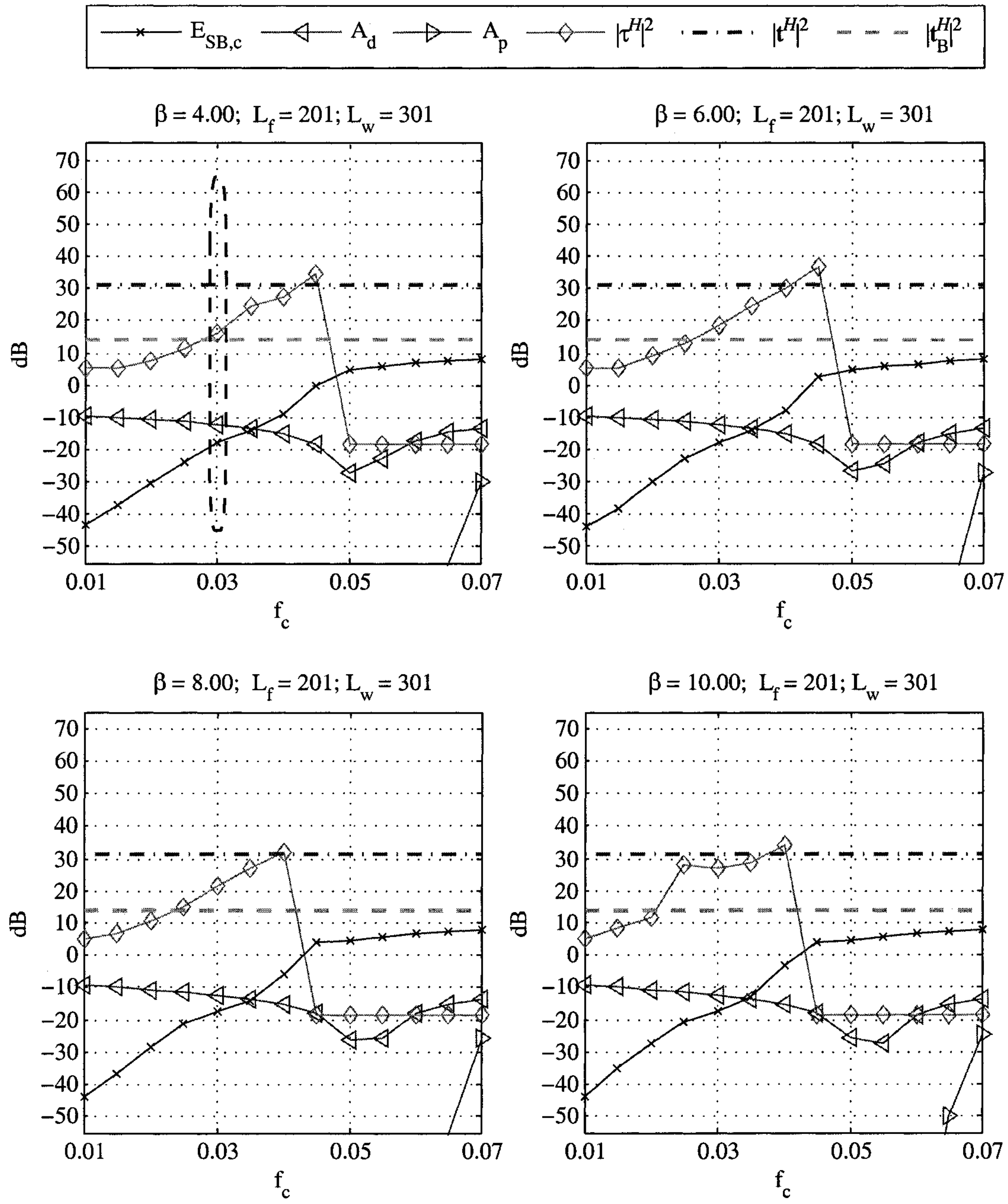

Fig. 5.12 *TFBD code design (Kaiser, $L_f = 201$; rectangular, $L_w = 301$):* Filterbank amplitude distortion (A_d) and aliasing power (A_p), vs. minimax stopband component ($E_{SB,c}$) and maximum row magnitudes of reconstruction matrices for temporal code ($|\tau^{\mathcal{H}}|^2$), subband code ($|t^{\mathcal{H}}|^2$), and benchmark DFT code ($|t_B^{\mathcal{H}}|^2$). A burst length of $E = 4$ was used in all cases.

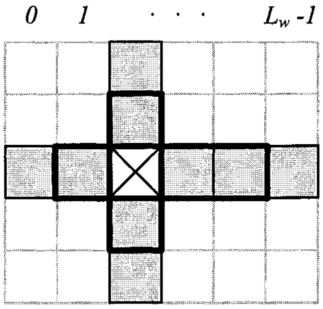

Fig. 5.13 *2-D cross formation erasure burst:* TFBD frame with column indices indicated on top of the figure; the heavy frame delineates lost samples in 2-D cross formation, with odd subband burst length ($L_{B,SB}$) and even temporal burst length ($L_{B,T}$). The intersection sample is marked with an × symbol. To evaluate the TFBD code, we reconstruct the intersection sample using the highlighted (N, K_b, K) TFBD temporal codevector, and compare the CSMSE to that of the (N, K) benchmark DFT codevector corresponding to the highlighted subband codevector.

and $L_{B,SB}$, respectively) for the x and y axes. We have also drawn constant value lines (dotted lines) on the surface plots every decreasing decade starting at unity. At the top of Fig. 5.14 and Fig. 5.15 we use a single cross formation 2-D burst; at the bottom of both figures we use 30 extra lost samples along the temporal codevector (more than 10% samples lost), spaced every ten samples before and after the temporal burst.

Fig. 5.14 and Fig. 5.15 indicate that for some subband burst length ($L_{B,SB}$) / temporal burst length ($L_{B,T}$) combinations, the TFBD code (using temporal reconstruction) can offer CSMSE improvement that can be as high as four orders of magnitude over the benchmark DFT code at the same rate. The temporal code can be seen to outperform the benchmark code for ($L_{B,SB}$, $L_{B,T}$)-pairs corresponding to longer subband erasure burst lengths and shorter temporal erasure burst lengths. This behavior is expected, as row magnitudes of reconstruction matrices (determining factor in both CSMSE expressions) increase rapidly with the length of erasure bursts.

The setup of Fig. 5.14 resulted in an amplitude distortion of $A_d = -15$ dB, and for this case, improvements were observed for subband burst lengths as small as $L_{B,SB} = 3$. Of all 64 $(L_{B,SB}, L_{B,T})$-pairs considered, 17 yielded temporal CSMSE benefits over the benchmark DFT code. Increasing the amplitude distortion to −11 dB (Fig. 5.15) enabled CSMSE improvements for subband bursts as small as $L_{B,SB} = 1$; in this case, 29

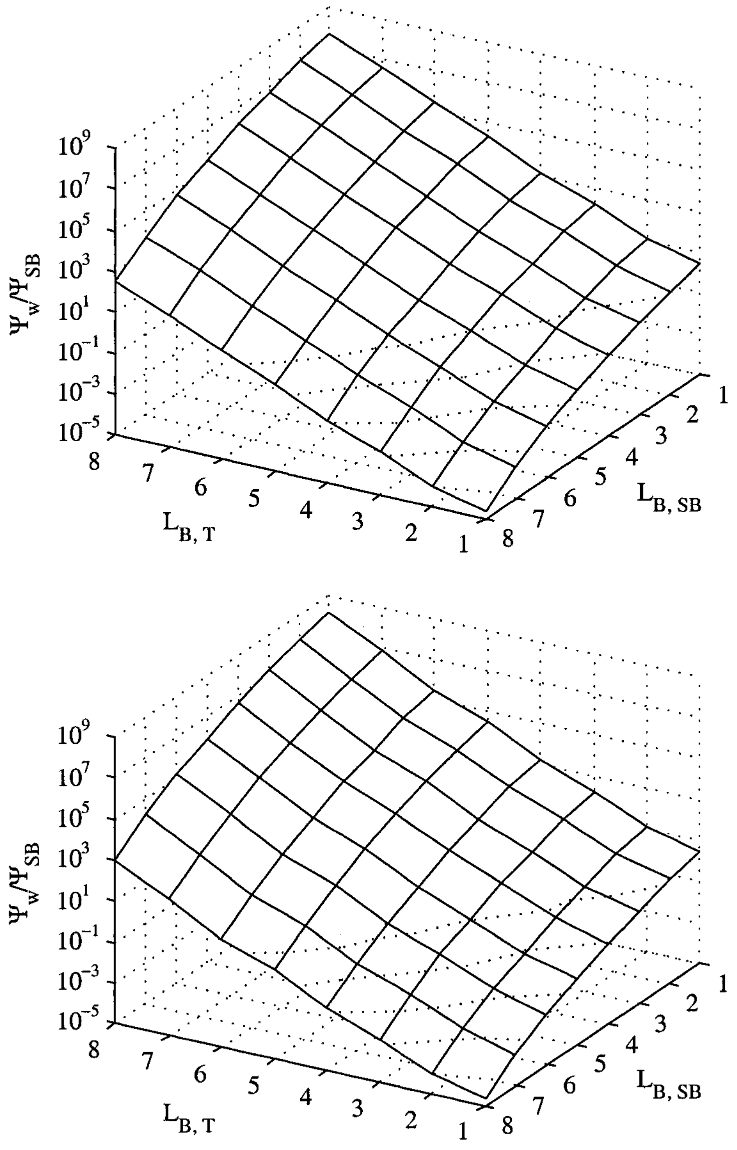

Fig. 5.14 *Temporal code vs. benchmark DFT code, Hanning window:* Ratio of Ψ_w of (15,10,7) TFBD code to CSMSE of (15, 7) DFT code at intersection of 2-D cross formation burst. $L_{B,T}$ and $L_{B,SB}$ are the length of the temporal and subband bursts, respectively. Contour lines drawn every 10^{-1} starting at unity. Setup as circled in Fig. 5.11. Top: single 2-D cross formation burst, with comparison sample at cross intersection (temporal position 150 and subband position 7). Bottom: same cross formation burst with 30 extra temporal erasures at positions 0,10,...,140, and 160,170,..., 300.

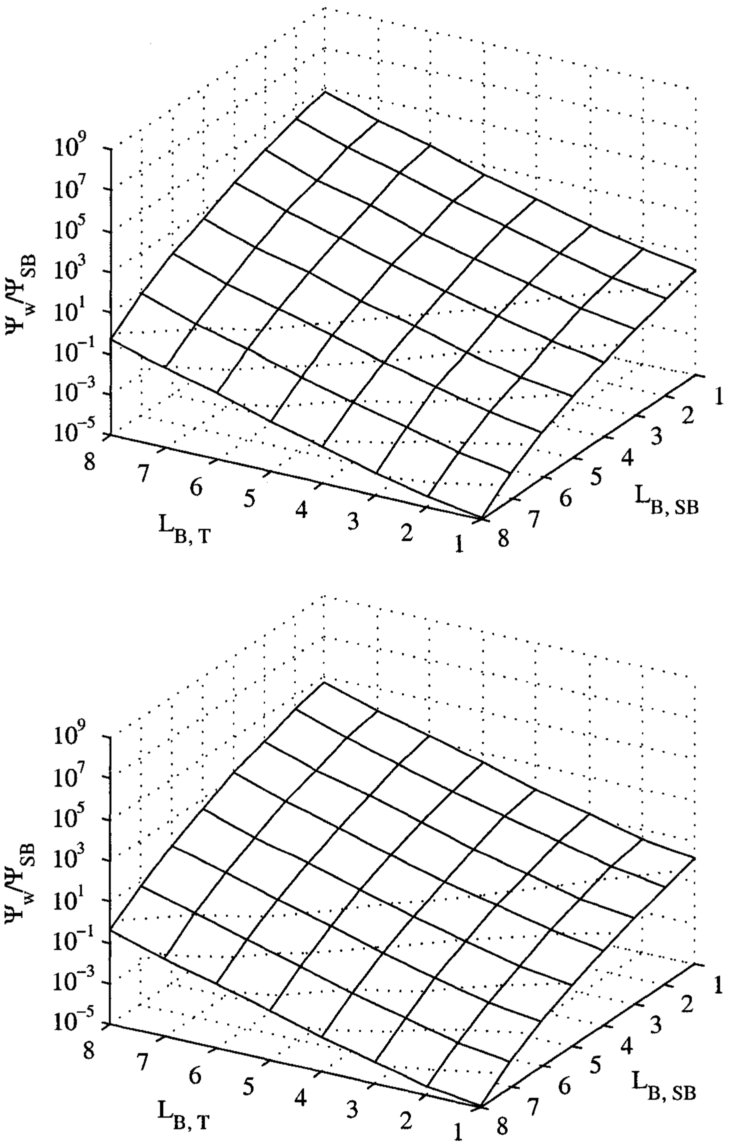

Fig. 5.15 *Temporal code vs. benchmark DFT code, rectangular window:* (Same as Fig. 5.14, but with setup circled in Fig. 5.12.) Ratio of Ψ_w of (15,10,7) TFBD code to CSMSE of (15,7) DFT code at intersection of 2-D cross formation burst. $L_{B,T}$ and $L_{B,SB}$ are the length of the temporal and subband bursts, respectively. Contour lines drawn every 10^{-1} starting at unity. Top: single 2-D cross formation burst, with comparison sample at cross intersection (temporal position 150 and subband position 7); Bottom: same cross formation burst with 30 extra temporal erasures at positions 0,10,...,140, and 160,170,..., 300.

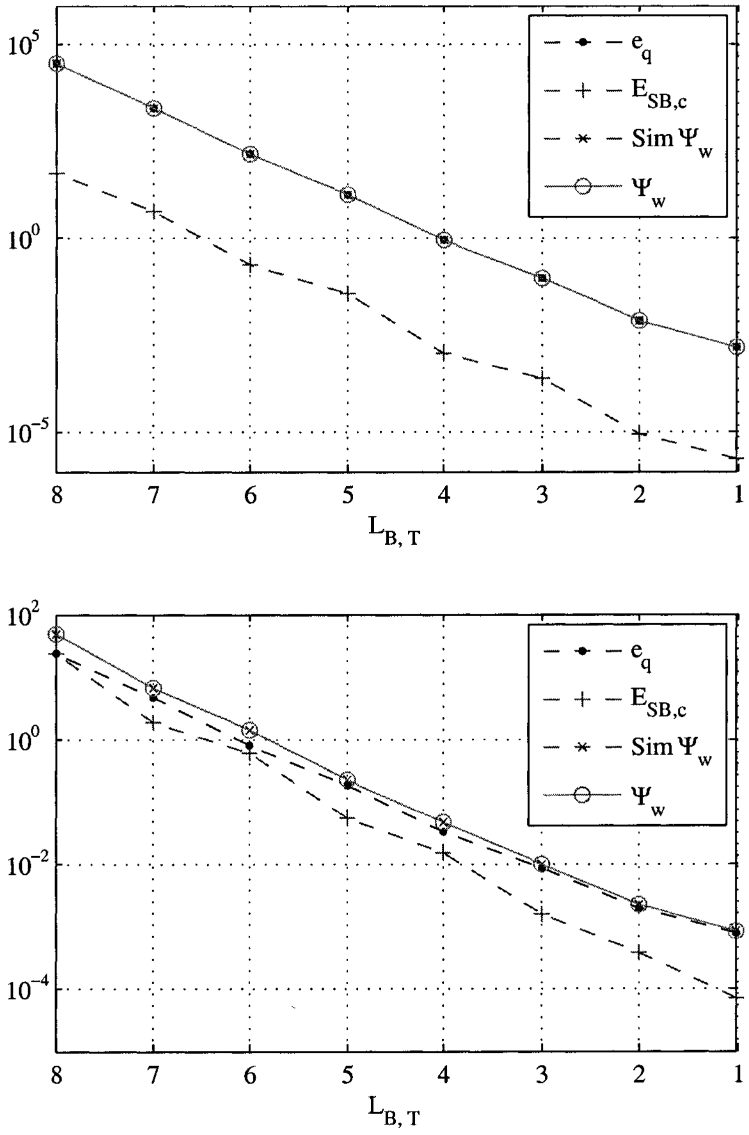

Fig. 5.16 *Simulation results for selected codes:* Theoretical and simulation values for Ψ_w, along with combined stopband component $E_{SB,c}$ and quantization noise component e_q. Simulation results displayed correspond to 10,000 realizations for each temporal burst length $L_{B,T}$. Both plots correspond to the case of a single, cross-formation burst, with no extra temporal erasures. Top: results corresponding to top of Fig. 5.14; Bottom: results corresponding to top of Fig. 5.15.

out of 64 $(L_{B,SB}, L_{B,T})$-pairs resulted in better temporal reconstructions.[5] These benefits remained constant even when extra temporal erasures were placed throughout the temporal codevector.

5.3 Summary

We began the present chapter by verifying the expressions derived in Chapter 4 experimentally. After verifying their correctness, we used them to construct plots that represented the tradeoffs involved in TFBD design. We then selected two sample codes and tested them against a benchmark DFT code of the same rate. The results showed that TFBD was capable of improvements in reconstruction error of more than four orders of magnitude, depending on the dimensions of the burst involved.

[5] The rectangular window is also capable of splitting bursts with frame boundaries, as discussed in Section 3.3.2.

Chapter 6

Conclusion

DFT codes over the real (or complex) field have been proposed [9] as a means for correcting errors and reconstructing lost data in packet-based network transmissions. Different authors have recurred to a wide range of techniques to find various DFT decoding methods, resulting in an array of decoding algorithms applicable to DFT codes. DFT codes have been interpreted as block codes over the complex field, and decoding methods similar to those of finite field block codes are also available in the complex field. Blahut shows that DFT codes are a complex field version of Reed-Solomon codes [10], and error detection and sample reconstruction methods akin to finite field Reed-Solomon decoding also exist in the complex field. Rath and Guillemot [8] presented an algorithm for DFT code error detection that was an extension to the ESPRIT algorithm of array processing theory. Marvasti *et al.* reformulated DFT code sample reconstruction as the zero-input solution to a linear constant coefficient difference equation [7].

With regards to sample reconstruction, a common problem with all the available algorithms is that for the case of bursty erasures (contiguous blocks of lost samples), the resulting MSE of the reconstruction (code sample mean square error, or CSMSE) is very large, growing quickly with the length of the burst [9]. We referred to this rapid growth in the CSMSE of the reconstruction as the bursty erasure phenomenon. For the difference-equation-based reconstruction method of Marvasti *et al.* mentioned above, the large reconstruction magnitudes were shown to be caused by zero clustering (resulting from bursty erasures) in the z-transform of the erasure locator polynomial. Rath and Guillemot [9] presented a geometrical interpretation of the same phenomenon: in the absence of quantization, the DFT reconstruction operations corresponded to projecting the message vector $\underline{x}$ onto rows of $\boldsymbol{G}$ with row-index corresponding to a lost sample index. Under quantization, the point $\underline{x}$ was known to exist within a hyperparallelogram $\mathbb{P}$; the projections of $\mathbb{P}$ onto rows of $\boldsymbol{G}$ with received sample row-index were equal to the quantization interval length. Reconstruction CSMSE, on the other hand, corresponded to the projection of $\mathbb{P}$ onto rows with lost sample row-index, and these were necessarily larger than the quantization interval (*e.g.,* a diagonal of $\mathbb{P}$). Bursty erasures particularly affected the length of diagonals of $\mathbb{P}$, and thus the associated sensitivity of the CSMSE to bursty erasures.

A recent work by Labeau *et al.* presented oversampled filterbanks in the context of coding theory, where filterbanks play the role of convolutional codes. The filterbank's polyphase matrix (z-domain matrix of size $N \times K$, $N > K$) can be seen as a generator matrix for a code, and a corresponding orthonormal matrix would play the role of parity matrix. Using the resulting syndrome sequence (*i.e.,* sequence resulting from running the input to the synthesis stage of the filterbank through the parity matrix), one could obtain an estimate for lost samples that was optimal in the mean-square sense.

Motivated by the simplicity and decoding resourcefulness of DFT codes, and the potential benefit in extending them from block codes to convolutional codes over the complex field, we began our work as a search for a convolutional code version of DFT codes that improved on their shortcomings when it came to bursty erasure. The result was a novel TFBD (Tandem Filterbank / DFT) code that achieves this goal by creating frames with rows and columns that can be treated as independent DFT codes. The codevectors along the columns of the frame are referred to as *subband codevectors*; those along the rows are the *temporal codevectors.* Erasure bursts along a given orientation can thus be avoided by carrying out the reconstructions along the remaining orientation. We further explored the possibility of using samples reconstructed along a given orientation, as received samples in reconstructions along the remaining orientations, a technique we referred to as *pivoting.*

We realized the concept behind the TFBD code by deriving expressions for the reconstruction operations of the TFBD code. As the name suggests, the Tandem Filterbank / DFT code consists of a tandem arrangement of a filterbank and DFT encoder. The result is that the subband codevectors are exact DFT codevectors, and hence the CSMSE expressions presented by Rath and Guillemot [9] apply. Temporal codevectors, on the other hand, are windowed segments of signals with lowpass shape and hence violate the null parity frequency requirement of DFT codes. However, any practical DFT code implementation must suffer quantization noise and non-null parity frequency components along with this. Hence, keeping in mind the near-zero stopband power of temporal codevectors, we derived exact expressions for the CSMSE of temporal reconstructions. The expressions were in terms of the correlation matrix of the stopband component of temporal codevectors, and thus we went on to derive a corresponding expression. Following this, we also derived expressions for the related temporal-to-subband pivoting application.

After verifying that all our expressions agreed with experimental values obtained through simulation, we applied the derived expressions towards the design of TFBD encoders. The selection of the set of parity frequencies, for example, was done by minimizing the combined stopband error component of the temporal CSMSE over the number P_w of parity frequencies in the set, where we constrained our search to those sets with parity frequencies that were contiguous and centered at $f = 0.5$. The choice of the prototype filter and temporal window was done by balancing the amplitude distortion of the TFBD filterbank and the temporal reconstruction CSMSE, keeping in mind that, in order for pivoting applications to be applicable, temporal and subband reconstruction errors had to be of comparable magnitude.

After selecting two sample codes, we tested their performance against a benchmark DFT code at the same rate. The results were very promising: at amplitude distortions of -11 dB and -15 dB, the codes displayed a relative improvement in CSMSE of as much as four orders of magnitude. The relative magnitude of temporal to benchmark CSMSE was a function of the combination of subband and temporal burst lengths. As expected, longer burst lengths resulted in larger row-magnitudes of reconstruction matrices, and hence the trend was that the better CSMSE went to the code with shorter erasure bursts. At -15 dB, 17 out of 64 combinations of subband/temporal burst lengths considered were in favor of the temporal code. At -11 dB, 29 out of 64 combinations went to the temporal code.

The topic of constructing complex field convolutional codes for sample reconstruction can be considered to be a relatively new one. While the trend in communications technology research has moved from the original analog modulation methods to digital transmissions with finite field encoding, complex field encoding might seem somewhat anachronic. However, it seems possible that such encoding methods might find their application in areas such as real-time multimedia transmissions, where the low delay (such as provided by DFT codes) is essential. Furthermore, the fact that codes such as the TFBD code do away with the assumption of flawless channel decoding used in designing tandem source/channel coders might make them suitable for applications requiring high source code rates. Ferreira [11] mentions another interesting application as is the construction of parallel computing systems with algorithm-based fault tolerance. Computing algorithms are usually thought up by the user as being carried out over the real or complex field (as opposed to over a quantized representation of this). Adding algorithm-based fault tolerance to such systems lends itself more intuitively to encoding formulated over the complex field. Labeau *et al.* trespass and then revert the analog/digital anachronism by proposing their oversampled filterbank (an encoder over the complex field) for the detection of erroneous symbols at a digital receiver. After all, signal constellations used in digital transmissions are defined over the complex plane. In all these and other applications, the TFBD codes presented in this work could find their place, and we leave such explorations for a later date.

Appendix A

Notational Conventions

Table A.1: Table of Notational Conventions. (Continued on next page...)

Symbol	Definition
	GENERAL CONVENTIONS
$\boldsymbol{x}$, $\boldsymbol{X}$	Matrix or row vector.
$\underline{x}$, $\underline{X}$	Vector.
$\underline{x}^{\mathcal{T}}$, $\boldsymbol{X}^{\mathcal{T}}$	Transpose of $\underline{x}$ or $\boldsymbol{X}$
$\underline{x}^{\mathcal{H}}$, $\boldsymbol{X}^{\mathcal{H}}$	Hermitian transpose of $\underline{x}$ or $\boldsymbol{X}$
$\underline{x}_{\langle k \rangle}$	Scalar corresponding to the k-th entry of vector $\underline{x}$.
$\underline{x}_{\langle \{i_k \vert k=1,\ldots,N\} \rangle}$	Vector composed of the entries of $\underline{x}$ indexed by the set $\{i_k \vert k = 1, \ldots, N\}$
$\boldsymbol{X}_{\langle k,l \rangle}$	Scalar corresponding to the (k, l)-entry of matrix $\boldsymbol{X}$.
$\boldsymbol{X}_{\langle k,* \rangle}$ $(\boldsymbol{X}_{\langle *,l \rangle})$	Row k (column l) of matrix $\boldsymbol{X}$.
$\boldsymbol{X}_{\langle l,\, \{i_k \vert k=1,\ldots,N\} \rangle}$ $(\boldsymbol{X}_{\langle \{i_k \vert k=1,\ldots,N\},l \rangle})$	Row (column) vector corresponding to the l-th row (column) of $\boldsymbol{X}$ indexed over the set $\{i_k \vert k = 1, \ldots, N\}$.
f-axis	Represents the angular frequency axis $[-\pi, \pi]$ normalized to the interval $[-0.5, 0.5]$.

Table A.1: Table of Notational Conventions. (Continued from previous page.)

Symbol	Definition
	ACRONYMS
(N, K_b, K) TFBD Code	Tandem Filterbank / DFT Code with generator matrix dimensions $K_b \times N$ and filterbank polyphase matrix dimensions $K \times K_b$; the code rate is K/N.
CSMSE	Code Sample Mean Square Error.
	SUBBAND DFT CODE (ALSO GENERAL DFT CODE)
$\underline{x}$	Original message vector, corresponding to the K-blocked version of the input message sequence $x[m]$.
$\underline{y}$	Original codevector.
$\underline{y}_R$	Received codevector before quantization, *i.e.*, $\underline{y}_{\langle\{r_k \mid k=1,\ldots,R\}\rangle}$.
$\underline{y}_E$	Erasure codevector before quantization, *i.e.*, $\underline{y}_{\langle\{e_k \mid k=1,\ldots,E\}\rangle}$.
$\ddot{\underline{y}}$	Quantized version of $\underline{y}$.
$\hat{\underline{y}}$	Estimate of original codevector $\underline{y}$.
$\boldsymbol{G}$	DFT generator matrix; (N, K_b, K) TFBD subband generator matrices have size $N \times K_b$.
$\boldsymbol{T}$	Reconstruction matrix of size $E \times R$, E and R the number of erasures and received samples, respectively.
$\Psi(e_k)$	Code sample mean square error (CSMSE) at position e_k of reconstructed codevector $\hat{\underline{y}}$.
	TEMPORAL DFT CODE
$\underline{x}_l^{w,\alpha}$, $\underline{x}^w$	Theoretical message vector corresponding to the temporal codevector $\underline{y}_l^{w,\alpha}$.
$\underline{y}_l^{w,\alpha}$, $\underline{y}^w$	Temporal codevector (for l-th subband) centered at time α, windowed with $w[n]$.

Table A.1: Table of Notational Conventions. (Continued from previous page.)

Symbol	Definition
$\underline{y}_l^\alpha$, $\underline{y}^\alpha$	Temporal codevector (for l-th subband) centered at time α, before windowing.
$\mathcal{G}$	DFT generator matrix related to the temporal code.
$\mathcal{T}$	Temporal code reconstruction matrix of size $E \times R$, E and R the number of erasures and received samples, respectively.
$\Psi(e_k)$	Code sample mean square error (CSMSE) at position e_k of reconstructed temporal codevector $\hat{\underline{y}}^w$.
	DFT CODE INDEXING CONVENTIONS
r_k	Received sample index along original codevector.
e_k	Erased sample index along original codevector.
	PIVOTING
↶	Denotes subband-to-temporal pivoting; also used to denote a sample used as a subband-to-temporal pivot.
↷	Denotes temporal-to-subband pivoting; also used to denote a sample used as a temporal-to-subband pivot.

Appendix B

Syndrome Correlation Matrix & Error-Erasure Subspace

In Section 2.2 we discussed subspace methods for error detection, as presented by Rath and Guillemot [8]. The methods consisted in finding the complement $(\text{span}\{[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]\}^\perp)$ of the error-erasure subspace; the error (and erasure) positions could be obtained from those $e \in 0, \ldots, N-1$ yielding vectors $\underline{v}_e = [1, W_N^e, \ldots, W_N^{e(l+E)}]^T$ not found in the complement space. We now prove that, as discussed in Section 2.2, the matrix $\boldsymbol{S}$ spans the error-erasure subspace exactly and thus its complement will be the same as the required complement space $\text{span}\{[\boldsymbol{V}_\nu \,|\, \boldsymbol{V}_E]\}^\perp$.

Matrix $\boldsymbol{S}$ was given by (2.20) in Section 2.2 as:

$$\boldsymbol{S} = \begin{bmatrix} \underline{s}_{\langle 0 \rangle} & \underline{s}_{\langle 1 \rangle} & \cdots & \underline{s}_{\langle d-E-l-1 \rangle} \\ \underline{s}_{\langle 1 \rangle} & \underline{s}_{\langle 2 \rangle} & \cdots & \underline{s}_{\langle d-E-l \rangle} \\ \vdots & \vdots & \vdots & \vdots \\ \underline{s}_{\langle l+E \rangle} & \underline{s}_{\langle 1+l+E \rangle} & \cdots & \underline{s}_{\langle d-1 \rangle} \end{bmatrix} \tag{B.1}$$

where the vector $\underline{s}$ is the syndrome of the received codevector $\underline{\ddot{y}}$. We i) neglect quantization ($\underline{\ddot{y}} = \underline{y} + \underline{\nu} + \underline{e} = \underline{y} + \underline{\epsilon}$, where, for simplicity, $\underline{\epsilon} = \underline{e} + \underline{\nu}$) and ii) assume that the $d = N - K$ parity frequencies are contiguous starting at position q_0. The resulting form for the syndrome vector is

$$\underline{s} = \boldsymbol{H}^{\mathcal{H}} \underbrace{(\underline{e} + \underline{\nu})}_{\underline{\epsilon}} = \begin{bmatrix} 1 & W_N^{q_0} & & W_N^{(N-1)q_0} \\ \vdots & \vdots & \dots & \vdots \\ 1 & W_N^{q_0+d-1} & & W_N^{(N-1)(q_0+d-1)} \end{bmatrix} \underline{\epsilon} \tag{B.2}$$

Each column of $\boldsymbol{H}^{\mathcal{H}}$ ($\boldsymbol{H}$ is the parity matrix, *cf.* Section 1.2.1) consists of powers of a single root of unity, and those columns corresponding to the non-zero entries of $\underline{\epsilon} = \underline{e} + \underline{\nu}$ consist of powers of the error and erasure roots $X_k = W_N^{e_k}$ ($e_0, \dots, e_{\nu-1}$ are the error positions; $e_\nu, \dots, e_{E+\nu-1}$ are the erasure positions). We can thus simplify (B.2) as follows, where $\underline{\epsilon}_{\langle\{e_k\}\rangle}$ denotes the vector composed of the entries $\{e_k\} = \{e_0, \dots, e_{E+\nu-1}\}$ of vector $\underline{\epsilon}$ (see **Tbl. A.1**, pg. 89, for a full list of notational conventions):

$$\underline{s} = \begin{bmatrix} X_0^{q_0} & & X_{\nu+E-1}^{q_0} \\ \vdots & \dots & \vdots \\ X_0^{q_0+d-1} & & X_{\nu+E-1}^{q_0+d-1} \end{bmatrix} \underline{\epsilon}_{\langle\{e_k\}\rangle} = \begin{bmatrix} 1 & & 1 \\ X_0 & & X_{\nu+E-1} \\ \vdots & \dots & \vdots \\ X_0^{d-1} & & X_{\nu+E-1}^{d-1} \end{bmatrix} \begin{bmatrix} X_0^{q_0}\, \underline{\epsilon}_{\langle e_0\rangle} \\ \vdots \\ X_{\nu+E-1}^{q_0}\, \underline{\epsilon}_{\langle e_{\nu+E-1}\rangle} \end{bmatrix}, \tag{B.3}$$

Note that in the second equality we have factored out the first entry from each column in the matrix, and moved it to the corresponding entry of vector $\underline{\epsilon}$. Using this same approach, we can express the k-th column $\boldsymbol{S}_{\langle *,k\rangle}$ of $\boldsymbol{S}$ in (2.20) as follows:

$$\begin{aligned} \boldsymbol{S}_{\langle *,k\rangle} &= \underline{s}_{\langle k,\dots,k+l+E\rangle} \\ &= \begin{bmatrix} X_0^{q_0+k} & & X_{\nu+E-1}^{q_0+k} \\ \vdots & \dots & \vdots \\ X_0^{q_0+k+l+E} & & X_{\nu+E-1}^{q_0+k+l+E} \end{bmatrix} \underline{\epsilon}_{\langle\{e_k\}\rangle} = \underbrace{\begin{bmatrix} 1 & & 1 \\ X_0 & & X_{\nu+E-1} \\ \vdots & \dots & \vdots \\ X_0^{l+E} & & X_{\nu+E-1}^{l+E} \end{bmatrix}}_{\begin{bmatrix} \underline{v}_0 & \dots & \underline{v}_{\nu+E-1} \end{bmatrix} = [\boldsymbol{V}_\nu|\boldsymbol{V}_E];\ cf.\ (2.16),(2.17)} \begin{bmatrix} X_0^{q_0+k}\, \underline{\epsilon}_{\langle e_0\rangle} \\ \vdots \\ X_{\nu+E-1}^{q_0+k}\, \underline{\epsilon}_{\langle e_{\nu+E-1}\rangle} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{V}_\nu | \boldsymbol{V}_E \end{bmatrix} \begin{bmatrix} X_0^{q_0}\, \underline{\epsilon}_{\langle e_0\rangle} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & X_{\nu+E-1}^{q_0}\, \underline{\epsilon}_{\langle e_{\nu+E-1}\rangle} \end{bmatrix} \begin{bmatrix} X_0^k \\ \vdots \\ X_{\nu+E-1}^k \end{bmatrix}. \end{aligned}$$

Since the column index k only appears in the last vector factor in the expression, the resulting form for $\boldsymbol{S}$ will be:

$$\boldsymbol{S} = \left[\boldsymbol{V}_{\nu} | \boldsymbol{V}_{E}\right] \begin{bmatrix} X_0^{q_0} \, \underline{\epsilon}_{\langle e_0 \rangle} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & X_{\nu+E-1}^{q_0} \, \underline{\epsilon}_{\langle e_{\nu+E-1} \rangle} \end{bmatrix} \begin{bmatrix} X_0^0 & \dots & X_0^{d-l-E} \\ & \vdots & \\ X_{\nu+E-1}^0 & \dots & X_{\nu+E-1}^{d-l-E} \end{bmatrix} \tag{B.4}$$

This proves that $\boldsymbol{S}$ will span the error-erasure subspace, and hence its complement will be the required complement space $\text{span}\{[\boldsymbol{V}_{\nu} \,|\, \boldsymbol{V}_{E}]\}^{\perp}$.

□

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: Wiley–Interscience, 1991.

[2] F. Labeau, "Joint source-channel coding and related techniques," *HF Journal*, vol. 1, January 2001.

[3] J. G. Proakis, *Digital Communications.* New York: McGraw Hill, 2001.

[4] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1451–1458, October 1998.

[5] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, March 1998.

[6] F. Labeau, J. Chiang, M. Kieffer, P. Duhamel, L. Vandendorpe, and B. Macq, "Oversampled filterbanks as error correcting codes: Theory and impulse noise correction," *IEEE Transactions on Signal Processing*, vol. 53, pp. 4619–4630, December 2005.

[7] 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*, vol. 47, pp. 1065–1075, April 1999.

[8] G. Rath and C. Guillemot, "Subspace-based error and erasure correction with DFT codes for wireless channels," *IEEE Transactions on Signal Processing*, vol. 52, pp. 3241–3252, November 2004.

[9] G. Rath and C. Guillemot, "Performance analysis and recursive syndrome decoding of DFT codes for bursty erasure recovery," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1335–1350, May 2003.

[10] R. E. Blahut, *Algebraic Methods for Signal Processing and Communications Coding.* New York: Springer-Verlag, 1992.

[11] P. J. Ferreira, "Mathematics for multimedia signal processing II – Discrete finite frames and signal reconstruction," in *Signal Processing for Multimedia* (J. S. Byrnes, ed.), pp. 35–54, New York: IOS Press, 1999.

[12] D. Hoffman, D. Leonard, C. Lindner, K. Phelps, C. Rodger, and J. Wall, *Coding Theory.* New York, New York 10016: Marcel Dekker, Inc., 1991.

[13] T. G. Marshall, "Coding of real-number sequences for error correction: A digital signal processing problem," *IEEE Journal on Selected Areas in Communications*, vol. 2, pp. 381–392, March 1984.

[14] G. Rath and C. Guillemot, "ESPRIT-like error localization algorithm for a class of real number codes," in *Proceedings of Globecom 2003*, IEEE.

[15] P. J. Ferreira, "The stability of a procedure for the recovery of lost samples in band-limited signals," *Signal Processing*, vol. 40, pp. 195–205, December 1994.

[16] H. L. Van Trees, *Optimum Array Processing.* New York: Wiley-Interscience, 2002.

[17] P. P. Vaidyanathan, "Multirate digital filters, filterbanks, polyphase networks, and appliactions: A tutorial," *Proceedings of the IEEE*, vol. 78, pp. 56–93, January 1990.

[18] P. P. Vaidyanathan, *Multirate Systems and filterbanks.* Upper Saddle River, NJ 07458: Prentice Hall, 1993.

[19] K. F. C. Yiu, N. Grbić, S. Nordholm, and K. L. Teo, "Multicriteria design of oversampled uniform DFT filterbanks," *IEEE Signal Processing Letters*, vol. 11, pp. 541–544, June 2004.

[20] J. Zepeda and F. Labeau, "Tandem filterbank-DFT code for bursty erasure correction," *64th IEEE Vehicular Technology Conference*, Fall 2006. Accepted for publication.