QUANTIZATION AND ERASURES IN FRAME REPRESENTATIONS

by

PETROS T. BOUFOUNOS

B.Sc. Economics, Massachusetts Institute of Technology (2000),
B.Sc. EECS, Massachusetts Institute of Technology (2002),
M.Eng. EECS, Massachusetts Institute of Technology (2002).

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2006

Author _____
Department of Electrical Engineering and Computer Science
January 20, 2006

Certified by _____
Alan V. Oppenheim
Ford Professor of Engineering
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Quantization and Erasures in Frame Representations

by

Petros T. Boufounos

Submitted to the Department of Electrical Engineering and Computer Science
on January 20, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Electrical Engineering and Computer Science

Abstract

Frame representations, which correspond to overcomplete generalizations to basis expansions, are often used in signal processing to provide robustness to errors. In this thesis robustness is provided through the use of projections to compensate for errors in the representation coefficients, with specific focus on quantization and erasure errors. The projections are implemented by modifying the unaffected coefficients using an additive term, which is linear in the error. This low-complexity implementation only assumes linear reconstruction using a pre-determined synthesis frame, and makes no assumption on how the representation coefficients are generated.

In the context of quantization, the limits of scalar quantization of frame representations are first examined, assuming the analysis is using inner products with the frame vectors. Bounds on the error and the bit-efficiency are derived, demonstrating that scalar quantization of the coefficients is suboptimal. As an alternative to scalar quantization, a generalization of Sigma-Delta noise shaping to arbitrary frame representations is developed by reformulating noise shaping as a sequence of compensations for the quantization error using projections. The total error is quantified using both the additive noise model of quantization, and a deterministic upper bound based on the triangle inequality. It is thus shown that the average and the worst-case error is reduced compared to scalar quantization of the coefficients.

The projection principle is also used to provide robustness to erasures. Specifically, the case of a transmitter that is aware of the erasure occurrence is considered, which compensates for the erasure error by projecting it to the subsequent frame vectors. It is further demonstrated that the transmitter can be split to a transmitter/receiver combination that performs the same compensation, but in which only the receiver is aware of the erasure occurrence. Furthermore, an algorithm to puncture dense representations in order to produce sparse approximate ones is introduced. In this algorithm the error due to the puncturing is also projected to the span of the remaining coefficients. The algorithm can be combined with quantization to produce quantized sparse representations approximating the original dense representation.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

Acknowledgments

I owe a great debt, monetary, emotional, and intellectual to my parents, Theodosis and Maria. They made sure I brushed my teeth every night, ate my food, and had fun. They tried to make sure I did my homework while in school but I managed to cut some corners with that... sorry mom! I guess it does not matter now; the workload at M.I.T. more than made up for my prior slackness. My parents didn't teach me everything I know, but made sure I get the best education they could provide. They are the best credit card, moral guidance, and emotional support I could ever have. Their advice is always valuable and I would never be here without it. I hope I will be able to repay them for all the great things they have done for me. I dedicate this thesis to them.

My intellectual mentor these past few years in MIT is my research advisor, Al Oppenheim. I would like to thank him for taking me when I was "green" and letting me mature seemingly on my own. Al is kind of mentor who lets you discover the academic world, but he is always there to protect you from falling. He is always involved in research but never dictates a problem to be solved, and I thank him for that. I look forward to future collaboration, and I hope I will be as good a mentor and as good a teacher as Al is.

Of course, this thesis would not have been possible without the involvement of the other two committee members, George Verghese and Vivek Goyal. Their comments and their questions throughout the meetings and during the writing process made this thesis significantly better. George and Vivek, thank you for all the time and effort you devoted into guiding me.

Many thanks to all the past and present members of the groups at the 6th floor of building 36, especially the DSPG group, for making sure there are enough intellectually stimulating (or not) conversations to keep me busy. Wade, Yonina, Andrew, Charles and Maya, made me feel welcome in the

group when I first joined. After they were gone, Sourav, little Al, Zahi, Joonsung, Tom, Melanie, Joe, Ross, Matt, Denis, and, of course, Charlie were always there to chat, answer my questions, or just keep me company in the office. Sourav receives an honorable mention, being my officemate, and being the one who tolerated all my random rambling when I was web surfing, reading e-mail, or (occasionally) doing research. Special thanks also goes to Alecia and Eric, the two DSPG assistants while I was here. Thanks to them all the administrative details were covered and we had nothing to worry about.

I am grateful to my academic advisor, Greg Wornell, for making sure I do not fall behind academically and for guiding me through the course selection problems. He always had great advice on picking courses and making difficult academic and scheduling choices. Of course some of that would not have been possible without his assistant, Tricia, who also managed to keep the 6th floor well fed every Wednesday with sandwiches from the Area I faculty meeting.

The people at the Mitsubishi Electric Research Lab, especially Paris Smaragdis together with Bhiksha Raj, provided two valuable summer experiences. My summer internships there were very stimulating, productive, and fun. They were distracting enough to return to MIT with a fresh view on my research and a clear mind, but not too distracting to delay my research. My experience in MERL was invaluable.

Grad school is not all about work, and my friends reminded me of that quite often. Lia Kolokouri provided me with indispensable emotional support and made sure I was well-nourished—especially with carbohydrates—while I was writing my thesis. Several waves of Boston-based Greek (or honorary Greek) friends helped me maintain my Greek language skills, especially in parties or clubs. Special thanks in no particular order to Paris, Karrie, Elias, Carl, Nicholas, Olga, Andy,Sav, George Z., Natasa, Maria K., Elina, Theodore K. (also for his proofreading skills), Anna S., Costas P., George C., Angelina, George K., Peggy, Mari, Rozita, Panos, Michalis, Marianna, Alexandra, Polina, Yannis K., Thetis, Hip, Apostolos, Dafni, Anastasia, my cousins (especially Costas, Vicky, Eleni, and Dimitris), and everyone else I am forgetting, including the extended kb group (you know who you are...). On the international side, I thank Arin, Joanna, Kavita, Mike, Daniele, Ozge, and the remaining happybunch. Life in Boston and vacations in Greece would not have been the same without my friends.

I would also like to thank everyone who participated in my small polls on the thesis template. I probably managed to disappoint all of you with the final thesis appearance, but I did take every opinion into consideration. You may like the result or blame it to “design by committee.” I hope it is the former.

Writing the acknowledgments is one of the most enjoyable but also one of the most difficult parts of writing the thesis. I want to list all my professors, my teachers, the administrators, and everyone who contributed to me being here. This includes people I don’t know, such as the MIT admissions officers who gave me a chance to come here. Unfortunately, listing everybody is not possible. I am also bound to forget people I know, and I apologize for that.

Contents

1 Introduction **17**

2 Background **21**

2.1 Linear Representations of Vectors 21

2.1.1 Bases and Basis Representations 21

2.1.2 Frames and Frame Representation 22

2.1.3 Frames as a Transformation of Orthonormal Bases 23

2.1.4 Decoupling the Analysis from the Synthesis 25

2.1.5 Frames Implied by Matrix Operations 26

2.1.6 Frames Implied by Discrete-time Filters 26

2.1.7 Frames Implied by Filterbanks 29

2.1.8 Useful Families of Frames 29

2.2 Orthogonal Projection of Vectors 32

2.2.1 Projections and Frame Expansions 33

2.3 Quantization 34

2.3.1 Scalar Quantization . . . 34
2.3.2 Vector Quantization . . . 35
2.3.3 Additive Noise Models . . . 37

3 Compensation Using Projections **39**
3.1 Error Compensation Using Representation Coefficients . . . 39
3.1.1 Computation of the Projection Coefficients . . . 41
3.1.2 Projections and Re-expansion of the Error . . . 43
3.2 Pre-compensation Followed by Post-correction . . . 44

4 Quantization of Frame Representations **47**
4.1 Quantization of Orthonormal Basis Expansions . . . 48
4.2 Quantization Grids and Frame Representations . . . 49
4.2.1 Linear Reconstruction from the Quantized Coefficients . . . 49
4.2.2 Analysis Followed by Scalar Quantization . . . 50
4.3 Limits of Scalar Quantization of the Analysis Coefficients . . . 51
4.3.1 Representation Bits Use . . . 52
4.3.2 Lower Bound on the Quantization Error . . . 52
4.3.3 Discussion . . . 54
4.4 Intersection of a Hyperplane with a Hypercube Lattice . . . 55
4.4.1 Definitions . . . 56
4.4.2 Intersection of a Single Cell with a Hyperplane . . . 57
4.4.3 Intersection of Cells in the Hypercube Lattice . . . 58
4.5 Efficiency of Frame Representations . . . 59

5 Quantization Noise Shaping on Finite Frame Representations **61**
5.1 Introduction . . . 61
5.2 Concepts and Background . . . 63
5.2.1 Frame Representation and Quantization . . . 63
5.2.2 Sigma-Delta Noise Shaping . . . 63
5.3 Noise shaping on Frames . . . 65
5.3.1 Single Coefficient Quantization . . . 65

5.3.2 Sequential Noise Shaping Quantizer 66
5.3.3 Tree Noise Shaping Quantizer 67
5.4 Error Models and Analysis 68
5.4.1 Additive Noise Model 68
5.4.2 Error Magnitude Upper Bound 69
5.4.3 Analysis of the Error Models 70
5.5 First Order Quantizer Design 71
5.5.1 Simple Design Strategies 72
5.5.2 Quantization Graphs and Optimal Quantizers 72
5.6 Further Generalizations 74
5.6.1 Projection Restrictions 74
5.6.2 Higher Order Quantization 75
5.7 Experimental Results 77
5.8 Noise Shaping with Complete Compensation 80
5.8.1 Error Upper Bound 80
5.8.2 Determination of the Residual Vectors 81
5.8.3 Noise Shaping on Finite Shift Invariant Frames 82

6 Noise Shaping for Infinite Frame Representations 85
6.1 Extensions to Infinite Frames 85
6.1.1 Infinite Shift Invariant Frames 86
6.1.2 First Order Noise Shaping 87
6.1.3 Higher Order Noise Shaping 87
6.2 Multistage D/A Converters 89
6.2.1 Elimination of the Discrete-time Filter 90
6.2.2 Multistage Implementation 91
6.2.3 Conversion Performance 92
6.3 Tunable Sigma-Delta Conversion 94
6.3.1 Tunable Digital to Analog Conversion 95
6.3.2 Tunable Analog to Digital Conversion 96
6.3.3 Optimal Tuning and Quantization Precision 97

7 Compensation for Erasures **99**

7.1 Erasure Compensation Using Projections . . . 101

7.1.1 Problem Statement . . . 102

7.1.2 Compensation of a Single Erasure . . . 102

7.1.3 Compensation of Multiple Coefficients . . . 103

7.2 Causal Compensation . . . 105

7.2.1 Transmitter-aware Compensation . . . 105

7.2.2 Pre-compensation with Correction . . . 106

7.2.3 Compensation Stability . . . 109

7.2.4 Simulation Results . . . 110

7.3 Puncturing of Dense Representations . . . 111

7.3.1 Puncturing Algorithm . . . 112

7.3.2 Error Evaluation . . . 113

7.3.3 Sparsification Schedule . . . 115

7.3.4 Quantization Combined with Sparsity . . . 116

8 Conclusions and Future Work **119**

8.1 Error Compensation Using Projections . . . 119

8.2 Quantization Limits . . . 120

8.3 Generalization of Sigma Delta Noise Shaping . . . 120

8.4 Compensation for Erasures . . . 121

8.5 Suggested Research Directions . . . 121

Bibliography **123**

List of Figures

2-1 General Reconstruction filterbank 29

2-2 General Analysis filterbank 30

2-3 Signal processing systems computing the upsampling (top) and the oversampling (bottom) frame expansion coefficients. 31

2-4 Examples of scalar quantizers. 35

2-5 Examples of vector quantizers in two dimensions. 36

2-6 Scalar and vector quantizer operating on a sequence of inputs 37

3-1 Architecture of the system implementing pre-compensation followed by post-correction of a single coefficient error. 45

4-1 Example of a scalar quantizer, and the square lattice generated by the scalar quantizer operating on a two dimensional basis expansion. 49

4-2 A scalar quantization example of a two-vector frame operating on a one-dimensional space. 51

4-3 An example of the hypercube (square) lattice, lattice boundaries, cells, and arbitrary N-blade (line) formed in the $M = 2$, $N = 1$ case of the intersection problem. 56

5-1 Traditional first order noise shaping quantizer 64

5-2 Examples of graph representations of first order noise shaping quantizers on a frame with five frame vectors. Note that the weights shown represent the upper bound of the quantization error. To represent the average error power the weights should be squared. 73

5-3 Histogram of the reconstruction error under (a) direct coefficient quantization, (b) natural ordering and error propagation without projections, (c) skip-two-vectors ordering and error propagation without projections. In the second row, natural ordering using projections, with (d) first, (e) second, and (f) third order error propagation. In the third row, skip-two-vectors ordering using projections, with (g) first and (h) second order error propagation (the third order results are similar to the second order ones but are not displayed for clarity of the legend). 78

6-1 Noise shaping quantizer, followed by filtering 88

6-2 Classical Sigma-Delta DAC architecture 89

6-3 Simplified Sigma-Delta DAC architecture with the low-pass filter $H_0(z)$ replaced by a gain r. This architecture has the same performance as the one in figure 6-2. 90

6-4 Two-stage simplified Sigma-Delta DAC architecture with the same performance as the one in figure 6-2. 91

6-5 Two-stage simplified Sigma-Delta DAC architecture with the gain r_2 placed after the D/C converter. Moving the gain effectively modifies the quantization interval Δ of the quantizer, thus affecting the quantization performance. 92

6-6 Performance of two-stage Sigma-Delta quantizers, with an interpolation filter used only after the first stage. In (a) the filter of the second stage is replaced by a gain factor r_2. In (b) the gain factor is placed in the system output. Note that the y-axis scale is different in the two plots. 93

6-7 Tunable digital to analog converter. 94

6-8 Tunable analog to digital converter. The gains c_i are tunable components. Their value is determined by inserting the autocorrelation of the tunable filter $h[n]$ in equation (3.21). 95

6-9 Tradeoff between error due to quantization and (a) filter bandwidth f_w or (b) filter redundancy r, assuming an ideal lowpass synthesis filter and optimal compensation of order p. 96

7-1 Erasure-aware transmitter projecting erasure errors. 107

7-2 Transmitter and receiver structure projecting erasure errors. Only the receiver is aware of the erasure. 107

7-3 Performance of erasure compensation using projections for the uniform oversampling frame, with oversampling ratios $r = 4$ (left) and $r = 8$ (right). The top plots demonstrate the optimal (unstable) systems. In the bottom plots optimality is traded for stability. In the legend, p denotes the compensation order, and q the probability of erasure. 111

List of Tables

6.1 Gain in dB in in-band noise power comparing p^{th} order classical noise shaping with p^{th} order noise shaping using projections, for different oversampling ratios r. 88

CHAPTER 1 *Introduction*

The use of redundancy as a robustness mechanism is very common in signal processing and communications applications. For example, channel codes provide robustness to communication errors and oversampling is often used to reduce distortion due to quantization. This thesis uses the redundancy in frame representations in order to provide robustness to quantization and erasures.

The use of frames to generate representations that are robust to errors has been considered in several contexts. For example, [27, 31, 7, 8, 14, 34] demonstrate the robustness of general frame expansions to erasures, while [28, 4, 5, 9, 17, 27] discuss the case of quantization. These methods mostly assume that the frame is used to analyze a signal using inner products with the frame vectors. Depending on the error type, the synthesis method is appropriately modified to accommodate for the error. In some cases ([34, 14, 43, 27], for example), the frame design problem is also considered. In these cases, an analysis method, a synthesis method, and an error type are imposed by the problem. The issue is the selection of vectors in the frame most appropriate for the specific problem.

This thesis approaches the problem assuming the synthesis method is predetermined. In most of this work, a linear synthesis equation with a pre-specified frame is considered. To accommodate for errors, the representation coefficients are modified instead of the synthesis method. Specifically, an error on any representation coefficient is compensated for by removing its projection from the remaining unaffected coefficients. The details of this projection principle are examined in chapter 3. The

frame design problem is not considered; the frame is assumed already designed or pre-determined by the application.

The use of projections has several advantages, most due to the linearity of the projection operator. For example, in most of the applications considered, the system implementing the projection, and its parameters are only determined once, at the design stage. To project the error, it is only necessary to scale the parameters according to the error magnitude. Furthermore, linearity often allows the superposition of projections to compensate for errors on different coefficients by compensating for the error on each coefficient separately. Using these properties most of the algorithms described can be implemented efficiently using linear systems.

Chapter 2 provides an overview of frame representations, projections and quantization. Its purpose is to establish the notation used through the thesis. It also serves as a quick reference for the definitions and the properties used in the remainder of the thesis.

Chapter 3 introduces the main tool used repeatedly in this thesis: error compensation using projections. Specifically, this chapter examines how the error introduced in one coefficient can be compensated for using the unaffected coefficients. This compensation performs, essentially, a frame analysis, the computation of which is discussed. Using this method to compensate for errors makes an implicit choice of computational simplicity over other properties. This choice is also discussed. The tools developed in chapter 3 are used in chapters 5 through 7.

Chapter 4 discusses the quantization of frame representations. Simple analysis using inner products followed by scalar quantization of the coefficients is shown to be suboptimal in terms of the bit use and the error decay as a function of the frame redundancy. The result is independent of the frame, or the reconstruction method used. The results in this chapter motivate the use of complex analysis and quantization methods, followed by linear reconstruction, instead of linear analysis using inner products and scalar quantization followed by complex synthesis methods.

One method to improve the performance of scalar quantization is Sigma-Delta noise shaping, discussed in chapters 5 and 6. Specifically, chapter 5 develops first-order Sigma-Delta noise shaping for arbitrary finite frames. Two methods to measure the performance are discussed, and two noise shaping algorithm designs are presented. The results are also generalized to higher order noise shaping. Chapter 6 extends these results to frame expansions in infinite dimensional spaces. The chapter also includes a discussion on simplifications and extensions of classical Sigma-Delta converters used in A/D and D/A conversion.

In chapter 7 the use of projections to compensate for erasures is examined. It is shown that projection of the erasure error is equivalent to projection of the data. Thus, several properties of the compensation are derived. The projection algorithm is used to causally compensate for erasure errors. The advantage of this method is the simplicity and the causality of the resulting system. Furthermore, this approach does not assume a signal analysis using inner products, only the synthesis using a

linear frame synthesis operator. Two equivalent systems are presented, one assuming that the transmitter is aware of the erasure occurrence, and one assuming that only the receiver is. The use of the same principle to intentionally introduce erasures and sparsify dense representations is also considered.

CHAPTER 2 *Background*

This chapter provides a brief overview of the concepts and the definitions used through the thesis, namely basis and frame expansions, projections, and quantization. The primary emphasis is on the definitions and the properties that are used in the remainder of the thesis, the aim being to establish the notation and serve as a quick reference.

2.1 Linear Representations of Vectors

The signals we consider in this thesis are vectors, elements of Hilbert spaces. Vectors such as $\mathbf{x}$ are denoted using boldface, and the Hilbert spaces using $\mathcal{W}$ or $\mathcal{H}$. In most of this thesis the convention $\mathcal{W} \subseteq \mathcal{H}$ is followed, unless otherwise noted. Subscripts are used to denote multiple subspaces wherever necessary.

2.1.1 Bases and Basis Representations

A set of vectors $\{\mathbf{b}_k \in \mathcal{H}\}$ form a basis for $\mathcal{H}$ if they are linearly independent and span $\mathcal{H}$. A Riesz basis further satisfies the following condition:

$$A||\mathbf{x}|| \leq \sum_k |\langle \mathbf{x}, \mathbf{b}_k \rangle| \leq B||\mathbf{x}||, \tag{2.1}$$

for some bounds $A > 0$, and $B < \infty$, and for all $\mathbf{x}$. The upper bound ensures that the basis expansion converges, and the lower bound that the vectors span the space.

Any vector $\mathbf{x} \in \mathcal{H}$ is uniquely expressed as a linear combination of the basis vectors using the synthesis, or reconstruction, sum:

$$\mathbf{x} = \sum_k a_k \mathbf{b}_k. \tag{2.2}$$

The analysis of $\mathbf{x}$ to the representation coefficients a_k is performed using inner products of $\mathbf{x}$ with the dual basis:

$$a_k = \langle \mathbf{x}, \underline{\mathbf{b}}_k \rangle, \tag{2.3}$$

in which the dual basis $\{\underline{\mathbf{b}}_k\}$ is the unique set of biorthogonal vectors satisfying:

$$\langle \mathbf{b}_k, \underline{\mathbf{b}}_l \rangle = \delta_{k,l}. \tag{2.4}$$

A basis is orthonormal if and only if it is self-dual. In this case, all the basis vectors have unit magnitude and each vector is orthogonal to the others. An orthonormal basis has Riesz bounds $A = B = 1$.

If the basis is orthonormal, Parseval's theorem holds, stating that:

$$||\mathbf{x}||^2 = \sum_k |\langle \mathbf{x}, \mathbf{b}_k \rangle|^2 = \sum_k |a_k|^2. \tag{2.5}$$

More discussion on these properties and the applications of basis expansions can be found in several linear algebra and signal processing texts [2, 36, 20].

2.1.2 Frames and Frame Representation

Frames are a generalization of bases, first introduced in [24]. A set of vectors $\{\mathbf{f}_k \in \mathcal{W}\}$ forms a frame if there exist constant frame bounds $0 < A \leq B < +\infty$, such that for all $\mathbf{x} \in \mathcal{W}$:

$$A||\mathbf{x}|| \leq \sum_k |\langle \mathbf{x}, \mathbf{f}_k \rangle| \leq B||\mathbf{x}||. \tag{2.6}$$

As with Riesz bases, the left side of the inequality guarantees the vectors span the space, while the right side ensures the convergence of infinite frame expansions. A Riesz basis is a frame, although a frame is not necessarily a Riesz basis—linear independence is not required in the definition of frames.

A vector $\mathbf{x}$ in the space $\mathcal{W}$ can be represented with the synthesis equation:

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k \tag{2.7}$$

In contrast to basis representations, the frame expansion coefficients a_k are not necessarily unique. Similar to basis expansions, however, they can be determined using an analysis equation:

$$a_k = \langle \mathbf{x}, \phi_k \rangle, \tag{2.8}$$

in which the $\{\phi_k\}$ is an analysis frame corresponding to the synthesis frame $\{\mathbf{f}_k\}$.

The analysis frame is not unique given a synthesis frame. Still, the dual frame, $\{\underline{\mathbf{f}}_k\}$, is the unique frame that minimizes the energy of the representation, $\sum_k |a_k|^2$, compared to all the possible analysis frames $\{\phi_k\}$. The lower and upper frame bounds of the dual frame are $1/B$ and $1/A$, respectively.

A frame is tight if its dual frame is the frame itself scaled by a constant. A frame is normalized tight if it is self-dual, i.e. this constant is 1. A normalized tight frame behaves similarly to an orthonormal basis. Tight frames have equal frame bounds A and B, which are equal to unity if the frame is normalized tight. Details on the relationships of the analysis and synthesis vectors can be found in a variety of texts such as [20, 36].

Frame expansions are usually overcomplete. The redundancy of the frame is denoted by r. A finite frame has redundancy $r = M/N$ where M is the number of frame vectors and N is the dimension of the space. If $r = 1$, the frame forms a basis. A normalized tight frame with redundancy $r = 1$ is an orthonormal Riesz basis. In this thesis we exploit the redundancy of frame expansions to compensate for degradation of the expansion coefficients from quantization and erasures.

2.1.3 Frames as a Transformation of Orthonormal Bases

It is often convenient to view frames as a linear transformation of orthonormal bases. Specifically, a space $\mathcal{H}$, an invertible linear operator $F : \mathcal{W} \to \mathcal{H}$, and an orthonormal basis $\{\mathbf{b}_k\}$ on $\mathcal{H}$ can be directly mapped onto an analysis frame $\{\underline{\mathbf{f}}_k\}$ in $\mathcal{W}$ by requiring that the expansion coefficients are the same:

$$\begin{aligned} a_k &= \langle F\mathbf{x}, \mathbf{b}_k \rangle && (2.9)\\ &= \langle \mathbf{x}, F^*\mathbf{b}_k \rangle && (2.10)\\ \Rightarrow \underline{\mathbf{f}}_k &= F^*\mathbf{b}_k, && (2.11) \end{aligned}$$

in which F^* denotes the adjoint operator of F. Alternatively, starting from $\mathcal{H}$ and $\{\mathbf{b}_k \in \mathcal{H}\}$, we define F using $\{\underline{\mathbf{f}}_k\}$:

$$F\mathbf{x} = \sum_k \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle \mathbf{b}_k. \tag{2.12}$$

Using the uniqueness of the adjoint operator it can be verified that the two definitions are equivalent.

Although the choice of $\mathcal{H}$ and $\{\mathbf{b}_k\}$ can be arbitrary, the requirement that F is invertible has an implication about the dimensionality of $\mathcal{H}$, and the cardinality of $\{\mathbf{b}_k\}$: they are both at least as large as the cardinality of the frame expansion (which is equal to M if the frame is finite). We call F the frame analysis operator.[1] of the frame $\{\underline{\mathbf{f}}_k\}$ The basis and the target space $\mathcal{H}$ is sometimes not of particular significance, and they are omitted. In this case, the implied space for $\mathcal{H}$ is $\mathbb{R}^M$ or l^2, depending on the cardinality of the frame, and the implied basis is the set of the unit

[1] Sometimes it is also referred to just as the frame operator, a terminology not followed in this thesis.

vectors along each coordinate direction, δ_k. Thus, the analysis frame operator maps $\mathbf{x}$ to the vector corresponding to the coefficients of the frame expansion:

$$(Fx)_k = \langle x, \underline{\mathbf{f}}_k \rangle = a_k, \tag{2.13}$$

$$\Rightarrow Fx = \begin{bmatrix} a_1 & \dots & a_k & \dots \end{bmatrix}^T. \tag{2.14}$$

This convention is often followed in this thesis, unless otherwise specified.

The singular values of F have particular significance in characterizing the frame. For example, it can be shown [27] that the frame is tight if and only if all the non-zero singular values are equal. Furthermore, the smallest and the largest singular values are associated with the lower and upper bounds A and B of the frame definition in equation (2.6). When all the singular values are non-zero, the frame is non-redundant and becomes an oblique basis. If they are all non-zero and equal, the frame is an orthonormal basis. Because of the significance of the singular values, the operators FF^* or $\sqrt{FF^*}$ are important for the frame. They are sometimes referred to as frame operators in the literature ([14], for example). However, this is not a convention we follow[2].

Given a frame $\{\underline{\mathbf{f}}_k\}$, it is straightforward to reconstruct $\mathbf{x}$ from the coefficients of the expansion using that frame. Since F is invertible, it is possible to determine T, an inverse of F, such that $TF = I$, where I is the identity operator. Using (2.12):

$$F\mathbf{x} = \sum_k a_k \mathbf{b}_k \tag{2.15}$$

$$\Rightarrow \mathbf{x} = TF\mathbf{x} = \sum_k a_k T\mathbf{b}_k, \tag{2.16}$$

making the set of vectors $\{T\mathbf{b}_k\}$ a synthesis set for $\{\underline{\mathbf{f}}_k\}$. In general, T and the corresponding synthesis set is not unique. However when the inversion uses the unique left pseudoinverse of the frame operator $T = F^\dagger$, then the corresponding set is the unique dual frame $\{\mathbf{f}_k = F^\dagger \mathbf{b}_k\}$, of $\{\underline{\mathbf{f}}_k\}$. Given a frame $\{\underline{\mathbf{f}}_k\}$, the dual of its dual is the frame itself. Dual frames have particularly nice properties (for some examples, see [20, 27] and references within), and are often used in pairs for the analysis and the synthesis set.

Given a pair of dual frame sets, either can be used in the analysis equation with the other used in the synthesis equation. Therefore, naming one of the two as the analysis frame and the other as the synthesis frame implies a design choice has been made, and we denote the sets using $\{\underline{\mathbf{f}}_k\}$ and $\{\mathbf{f}_k\}$ respectively. Although it is often the case in practice, explicit mention of an analysis and a synthesis set, and the corresponding equations, does not necessarily imply that the sets are dual of each other, or, even, that both sets form a frame. Unless otherwise noted, the remainder of this thesis does not assume duality of the analysis and synthesis sets. Of course, if the frame is not

[2] Referring to FF^* as the frame operator creates potential confusion with the frame analysis or synthesis operators. Furthermore, F itself is often referred to as the frame operator (omitting the word *analysis*), the potential for confusion is greater.

redundant, it forms a basis, and the expansion is unique. In this case the analysis frame is the dual of the synthesis frame.

2.1.4 Decoupling the Analysis from the Synthesis

In the same way that a frame analysis defines a linear operator, so does the synthesis equation (2.7). We refer to this operator as the frame synthesis operator, denoted by S, although this name is not common in the literature[3].

$$S : \mathcal{H} \rightarrow \mathcal{W}, \text{ s.t. } S\mathbf{y} = \sum_k \langle \mathbf{y}, \mathbf{b}_k \rangle \mathbf{f}_k. \tag{2.17}$$

As with the frame operator, the space $\mathcal{H}$ is sometimes assumed to be $\mathbb{R}^M$ or l^2, with the corresponding bases $\mathbf{b}_k = \delta_k$. The analysis and synthesis operators using the same frame set are adjoints of each other: $S = F^*$. Furthermore, if the frame is normalized tight, the synthesis operator is the pseudoinverse of the analysis one: $S = F^\dagger$.

If the synthesis frame forms a complete basis, the synthesis operation is full rank. When the analysis followed by synthesis is required to be the identity, this completely determines the corresponding analysis operator, and, therefore, the analysis frame. However, in the case of the synthesis with a redundant frame, the domain $\mathcal{H}$ of S is in general larger than the range $\mathcal{W}$, which implies that the synthesis operator has a non-zero nullspace null(S).

The existence of this nullspace in redundant frames decouples the analysis process from the synthesis one. Given a specific synthesis equation, a set of frame expansion coefficients can be modified in many ways. The synthesized vector $\mathbf{x}$ remains the same, as long as the modifications only affect the nullspace of the synthesis operation. The analysis method itself can be modified to produce coefficients with components in the nullspace of the synthesis. This essentially decouples the usual analysis-synthesis linear operation pair associated with basis expansions.

The flexibility of modifying the expansion coefficients allows for the pursuit of other desirable properties in the expansion coefficients. For example, for processing and transmission the coefficients need to be quantized. Sparsity is usually desired in compression applications. Additive noise immunity and tolerance to erasures is also desired in transmission applications. The rest of this thesis presents some examples and applications that this flexibility enables.

Most of this work only assumes that a frame is used for the synthesis, using the synthesis equation, making no assumptions about the analysis method. We should note that this is not the only approach. Often the analysis is given by the application and

[3] This name does occur, however [14], as it should. There is no particular reason why only the analysis should be associated with an operator. Since in this work the focus is on the synthesis operation, this term is very useful. The terminology conventions over the frame operator (as well as other aspects of frame representations) have not yet been stabilized and are often contradictory. Some discussion exists in [14].

the synthesis is designed to accommodate modifications on the coefficients during processing (see [38, 28, 27, 7, 8] for some examples). We explore an important aspect of this choice when discussing quantization in chapter 4. However, depending on the application, either the synthesis or the analysis might be imposed from the setup.

2.1.5 Frames Implied by Matrix Operations

Any matrix $\mathbf{F} \in \mathbb{R}^{M \times N}$ with $\text{rank}\{\mathbf{F}\} = N$ defines an analysis frame operator, with the corresponding frame vectors being the rows of $\mathbf{F}$ transposed. When the matrix operates on a vector $\mathbf{x} \in \mathbb{R}^N$, it computes its frame expansion:

$$\mathbf{F}\mathbf{x} = \begin{bmatrix} -\underline{\mathbf{f}}_1^T- \\ \vdots \\ -\underline{\mathbf{f}}_M^T- \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \langle \underline{\mathbf{f}}_1, \mathbf{x} \rangle \\ \vdots \\ \langle \underline{\mathbf{f}}_M, \mathbf{x} \rangle \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} = \mathbf{a}. \tag{2.18}$$

Any left inverse $\mathbf{T}$ of $\mathbf{F}$ can be used to recover $\mathbf{x}$ from the expansion coefficients, since

$$\mathbf{T}\mathbf{a} = \mathbf{T}\mathbf{F}\mathbf{x} = \mathbf{x}. \tag{2.19}$$

The columns of any such left inverse form a synthesis frame corresponding to $\{\underline{\mathbf{f}}_k\}$. The unique dual frame $\{\mathbf{f}_k\}$ is the one derived from the pseudoinverse $\mathbf{T} = \mathbf{F}^\dagger$:

$$\mathbf{T}\mathbf{a} = \begin{bmatrix} | & & | \\ \mathbf{f}_1 & \cdots & \mathbf{f}_M \\ | & & | \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} = \sum_{k=1}^{M} a_k \mathbf{f}_k = \mathbf{x}. \tag{2.20}$$

The singular values of $\mathbf{F}$ determine the frame bounds A and B for equation (2.6). Specifically, $A = \sigma_{\min}$ and $B = \sigma_{\max}$ are the largest lower frame bound and the smallest upper frame bound respectively. If all the singular values of $\mathbf{F}$ are equal—i.e. $\mathbf{F}^T\mathbf{F} = \sigma^2\mathbf{I}$—then the two bounds, A and B are also equal, and the frame is tight. In this case the dual frame vectors are the frame vectors scaled by $1/\sigma^2$:

$$\mathbf{F}^T\mathbf{F} = \sigma^2\mathbf{I} \Leftrightarrow \left(\frac{1}{\sigma^2}\mathbf{F}^T\right)\mathbf{F} = \mathbf{I} \Leftrightarrow \mathbf{f}_k = \frac{1}{\sigma^2}\underline{\mathbf{f}}_k. \tag{2.21}$$

Therefore, a tight frame in $\mathbb{R}^N$ corresponds to a $M \times N$ matrix whose transpose is its pseudoinverse within a scalar.

2.1.6 Frames Implied by Discrete-time Filters

Any LTI discrete-time filter can be viewed as the synthesis operator for a frame representation. In particular, filtering of a signal $a[n] \in l_2$ with an impulse response $h[n]$ produces the convolution sum:

$$x[n] = \sum_k a[k]h[n-k] = \sum_k a[k]\mathbf{f}_k, \tag{2.22}$$

in which $a[k]$ is the input to the filter and $x[n]$ is the output, also in l_2. This equation has the same form as the synthesis equation (2.7) with the coefficients $a[k]$ taking the place of the frame representation coefficients a_k. In general, the representation coefficients are not produced using inner products with an analysis frame. Still, they represent the signal at the output of the filter.

The convolution sum can also be viewed as an analysis operator. Using $x[k]$ to denote the input, and $a[k]$ to denote the output of an LTI filter $g[n]$:

$$a[k] = \sum_n x[n]g[k-n] = \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle. \tag{2.23}$$

Thus, the vector $\mathbf{x}$ is analyzed using the analysis frame $\underline{\mathbf{f}}_k = g[k-n]$ and the output $a[k]$ corresponds to the analysis coefficients a_k. In this case, the analysis frame is translations of the impulse response time-reversed. For most of this thesis, we consider only synthesis from frame representations, and, therefore, this view is not emphasized. However, it is an important aspect of the duality of frame expansions, and an important consequence of the time invariance of LTI filters.

It can be shown that the adjoint operation of filtering using the impulse response $h[n]$ is filtering using a time-reversed impulse response, $h[-n]$. It follows that the singular values of the filtering operator are determined by the magnitude of the Fourier transform, $|H(e^{j\omega})|$. As expected, since the space is infinite dimensional, the singular values are infinite in number, indexed by discrete-time or continuous-time frequency.[4]

The range of a filter is any signal in the space of linear combinations of complex exponentials, chosen in the frequencies in which the Fourier transform of the filter is not zero. This is also the span of the vector set formed by translations $h[n-k]$ of the filter impulse response. The nullspace of the filter is the linear combination of all complex exponentials for which its frequency response is zero. It should be noted that if the frequency response has zero crossings in frequency (as opposed to frequency intervals in which the frequency response is zero), then the signals that produce 0 output are infinite length complex exponentials. These are not in l_2, the assumed input space of the filter.

To form a Riesz basis from an impulse response $h[n]$ and its shifts, the filter with the impulse response should have frequency response with finite magnitude, lower bounded away zero form in all frequencies:

$$0 < A \leq H(e^{j\omega}) \leq B < +\infty, \tag{2.24}$$

in which (assuming the Fourier transform is properly normalized) the constants A

[4] There is a subtlety involved in the cardinality of the integers, which index the filter taps in discrete-time filters, versus the cardinality of $(-\pi, \pi]$, which index the singular values in the discrete-time frequency domain. This should be resolved using finite length signals and circular convolution, for which the magnitude of the DFT provides the singular values. As we let the length of the signals grow to infinity, the cardinality of the filter indices in time, and the DFT coefficient indices stays the same, eliminating the issue. There is a further subtlety in defining the Fourier transform, so that the singular vectors are unit-norm, but this is not important for the purposes of this discussion.

and B correspond to the lower and upper bounds of the Riesz basis definition (2.1). This simple rule, unfortunately does not extend to frames derived from filters.

When the input to a filter is restricted in l_2, equation (2.22) defines a linear synthesis operator $S : l_2 \rightarrow \mathcal{W}$, in which $\mathcal{W}$ is the range of the filter, as described in the previous section. If the magnitude of the frequency response of the filter is finite and positive for all frequencies, then the filter defines a Riesz basis, and, therefore, a non-redundant frame. If the frequency response contains intervals in which $H(e^{j\omega}) = 0$ then $\mathcal{W}$ is the space of signals with frequency content limited to the span of the frequency response. In order for the filter to form a frame, however, it should also satisfy the bounds A and B of the definition (2.6). Thus, the filter frequency response in the range of the filter should be greater than the lower bound and finite. Therefore, the filter frequency response magnitude should satisfy:

$$0 < A \leq |H(e^{j\omega})| \leq B < +\infty \quad \text{for} \quad \omega \in I \tag{2.25}$$

$$\text{and } |H(e^{j\omega})| = 0 \quad \text{for} \quad \omega \in I^c, \tag{2.26}$$

in which $I \cup I^c = [-\pi, \pi)$, and I is the set of frequency intervals for which the response is positive. Thus, a filter forms a frame if it either has non-zero frequency response in all frequencies, or discontinuities around its nullspace. Otherwise, if the filter frequency response is continuous and positive in the neighborhood of a null, then the lower frame bound A becomes 0.

Given a synthesis filter $H(e^{j\omega})$, a corresponding analysis filter $G(e^{j\omega})$ should satisfy

$$G(e^{j\omega}) = 1/H(e^{j\omega}) \text{ if } H(e^{j\omega}) \neq 0. \tag{2.27}$$

The dual frame, determined by the psudoinverse $H^\dagger(e^{j\omega})$ further satisfies:

$$H^\dagger(e^{j\omega}) = \begin{cases} 1/H(e^{j\omega}) & \text{if } H(e^{j\omega}) \neq 0 \\ 0 & \text{if } H(e^{j\omega}) = 0. \end{cases} \tag{2.28}$$

If the lower bound A is zero, any $G(e^{j\omega})$ is infinite around the null of $H(e^{j\omega})$, which makes the analysis equation unstable.

In most of this thesis we only use filters as synthesis operators, thus this instability is not an issue. The frames for which we need to determine the duals are usually finite. In the cases in which the dual of a filter should be used, we assume the filter forms a frame. In practice, for numerical stability of the computations the nullspace of the filter is taken as the frequency interval in which the frequency response magnitude is small. The filter corresponding to the dual frame should have frequency response magnitude near zero in the same interval. This practical approach is similar to allowing passband and stopband ripple when designing filters to approximate ideal designs.

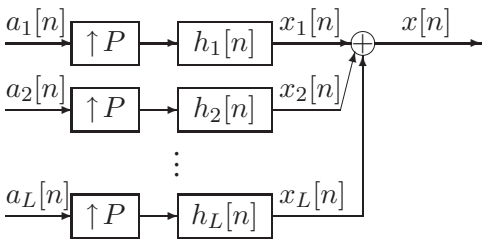


FIGURE 2-1: General Reconstruction filterbank

2.1.7 Frames Implied by Filterbanks

A similar synthesis equation exists for reconstruction filterbanks such as the one shown in figure 2-1:

$$x[n] = \sum_{k,l} a_l[k] h_l[n - kP] = \sum_{k,l} a_l[k] \mathbf{f}_{l,k} \tag{2.29}$$

In this case the indexing is two dimensional, denoting time and filterbank channel. In other words, the frame vectors are all translations of vectors in a generating set $\{\mathbf{f}_{0,l}\}$. Furthermore, depending on the filterbank the vectors in the generating set might be related. For example, all the filters of modulated filterbanks are modulated versions of a fundamental filter.

The signals $a_l[n]$ on every channel are often produced using an analysis filterbank such as the one in figure 2-2. The corresponding analysis equation is:

$$a_l[n] = \sum_{k,l} x[k] \bar{h}_l[nP - k] = \langle \mathbf{x}, \underline{\mathbf{f}}_{l,k} \rangle \tag{2.30}$$

in which $\bar{h}_l[n]$ is the impulse response of the l^{th} analysis filter.

The issues in treating filterbanks as frames are similar to the ones explored in section 2.1.6, and we do not examine them in more detail here. In this thesis we use filterbanks mostly to synthesize vectors, not for analysis. Although we assume that whenever a synthesis is performed through a filter bank, the filterbank forms a frame, in practice the assumption can relaxed in ways similar to the ones described for filters. More details on frame expansions derived from analysis and synthesis filterbanks can be found in a variety of references, such as [31, 19, 10].

2.1.8 Useful Families of Frames

While any full-rank linear operator implicitly defines a frame, as described above, there are two frame families that are particularly interesting in signal processing ap-

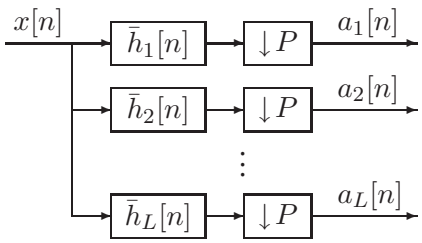


FIGURE 2-2: General Analysis filterbank

plications, especially in the context of this thesis. We present a brief overview in this section.

The Sampling Frame

Oversampling in time of bandlimited signals is a well studied class of frame expansions, although not often described in the terminology of frames. Historically it evolved outside the context of frame theory, and it has proved very useful in signal processing systems, such as Sigma-Delta converters [13], and sampling rate converters.

A discrete time signal $x[n]$ or a continuous time signal $x(t)$ bandlimited to π/T is upsampled or oversampled to produce a sequence a_k. In the terminology of frames, the upsampling operation is a frame expansion in which $\underline{\mathbf{f}}_k = r\mathbf{f}_k = \mathrm{sinc}((n-k)/r)$, with $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$. The sequence a_k is the corresponding ordered sequence of frame coefficients:

$$a_k = \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle \quad = \quad \sum_n x[n]\mathrm{sinc}((n-k)/r), \tag{2.31}$$

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k \quad \Rightarrow \quad x[n] = \sum_k a_k \frac{1}{r}\mathrm{sinc}((n-k)/r). \tag{2.32}$$

Similarly for oversampled continuous time signals:

$$a_k = \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle \quad = \quad \int_{-\infty}^{+\infty} x(t)\mathrm{sinc}((t-krT)/r), \tag{2.33}$$

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k \quad \Rightarrow \quad x(t) = \sum_k a_k \frac{1}{rT}\mathrm{sinc}((t-krT)/r), \tag{2.34}$$

in which $r \geq 1$ and $2\pi/T$ is the Nyquist sampling rate for $x(t)$. The case of $r = 1$ corresponds to sampling at the Nyquist rate and the resulting frame expansion forms a non-redundant orthogonal basis.

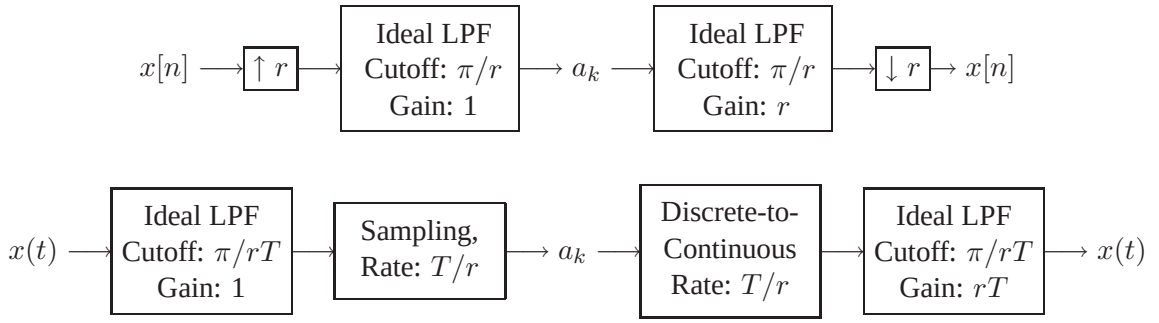


FIGURE 2-3: Signal processing systems computing the upsampling (top) and the oversampling (bottom) frame expansion coefficients.

A subtlety in representing oversampling as a frame expansion, especially in the discrete-time case, involves the spaces $\mathcal{W}$ and $\mathcal{H}$ in which the vector $\mathbf{x}$ lies. Specifically, if $\mathbf{x}$ is taken as a member of l_2, with $x[n]$ its corresponding basis expansion on the basis $\delta[n-k]$, then the upsampling operator is a linear map $F : l_2 \rightarrow l_2$. This means that the target space $\mathcal{H}$ has the same cardinality as the original space and the redundancy of the frame can only be inferred by the measure of the nullspace of the synthesis frame operator, not by directly comparing the space measures (which are equal).

To avoid that complication we consider $\mathbf{x}$ as a member of a subspace in l_2 of functions bandlimited to π/r. The coefficients $x[k]$ are the basis expansion coefficients of $\mathbf{x}$ using a basis for that subspace: $b[n-k] = \mathrm{sinc}(n/r-k) \Rightarrow \mathbf{x} = \sum_k x[k]b[n-k] \in l_2$. Thus, the analysis frame operator F is a linear map from the space $\mathcal{W}$ of series in l_2 bandlimited to π/r to the space $\mathcal{H} = l_2$. The map is the identity over $\mathcal{W}$. Combined with the requirement that the frame vectors should lie in $\mathcal{W}$, the domain is extended to l_2 by making $\mathcal{W}^{\perp}$ its nullspace. Therefore, F is a low-pass filter with cutoff frequency π/r. The pseudoinverse of the analysis frame operator $F^{\dagger}$ is also a low-pass filter with the same cutoff frequency, which implies that the frame is tight. Tightness can also be shown by the magnitude of the Fourier transform. The redundancy r is computed by taking the ratio of measures of $\mathcal{H}$ and $\mathcal{W}$.

In practice, the frame expansion coefficients can be computed using simple signal processing systems, such as the ones in figure 2-3.

The harmonic frames [28] is a class of unit-norm tight frames in $\mathbb{R}^N$, for which the frame vectors are:

$$\text{If } N \text{ is even: } \mathbf{f}_k = \sqrt{\frac{2}{N}} \left[\cos \frac{2\pi k}{M}, \sin \frac{2\pi k}{M}, \cos \frac{2\pi 2k}{M}, \sin \frac{2\pi 2k}{M}, \ldots, \cos \frac{2\pi \frac{Nk}{2}}{M}, \sin \frac{2\pi \frac{Nk}{2}}{M} \right]^T. \tag{2.35}$$

$$\text{If } N \text{ is odd: } \mathbf{f}_k = \sqrt{\frac{2}{N}} \left[\frac{1}{\sqrt{2}}, \cos \frac{2\pi k}{M}, \sin \frac{2\pi k}{M}, \cos \frac{2\pi 2k}{M}, \sin \frac{2\pi 2k}{M}, \ldots, \cos \frac{2\pi \frac{(N-1)k}{2}}{M}, \sin \frac{2\pi \frac{(N-1)k}{2}}{M} \right]^T. \tag{2.36}$$

This class of frames are a proof that a unit-norm tight frame exists for any combination of N, and M, $M \geq N$. One of their most useful property is that any subset of N vectors from a harmonic frame still spans the space, as shown in [27], which also proves that frames with this property exist for any N, and M. Thus, we often refer to the harmonic frames in this thesis as an example of a frame with this property. In [34] this property is defined as maximal robustness to erasures, and a general construction for frames with this property is described.

2.2 Orthogonal Projection of Vectors

Given an inner product space $\mathcal{H}$, two vectors $\mathbf{v}$ and $\mathbf{u}$ are orthogonal if their inner product $\langle \mathbf{v}, \mathbf{u} \rangle$ is equal to 0. For any subspace $\mathcal{W} \subset \mathcal{H}$, the orthogonal complement $\mathcal{W}^\perp$ of $\mathcal{W}$ in $\mathcal{H}$ is the set of all vectors $\mathbf{u}$ that are orthogonal to all the vectors $\mathbf{v} \in \mathcal{W}$:

$$\mathcal{W}^\perp = \{\mathbf{u} \in \mathcal{H} | \, \forall \mathbf{v} \in \mathcal{W} : \langle \mathbf{u}, \mathbf{v} \rangle = 0\}, \tag{2.37}$$

$$\Rightarrow \quad (\mathcal{W}^\perp)^\perp = \mathcal{W}. \tag{2.38}$$

Any vector $\mathbf{x} \in \mathcal{H}$ is uniquely expressed as the sum:

$$\mathbf{x} = \mathbf{u} + \mathbf{v}, \ \mathbf{u} \in \mathcal{W} \text{ and } \mathbf{v} \in \mathcal{W}^\perp. \tag{2.39}$$

to form the direct sum decomposition of $\mathcal{H}$ into $\mathcal{W}$ and its orthogonal complement $\mathcal{W}^\perp$, denoted using $\mathcal{H} = \mathcal{W} \oplus \mathcal{W}^\perp$.

The orthogonal projection[5] of $\mathbf{x}$ onto $\mathcal{W}$ is the operator $\mathcal{P}_\mathcal{W}(\cdot)$ that maps $\mathbf{x}$ to the corresponding $\mathbf{u} \in \mathcal{W}$, as uniquely defined in (2.39). Combined with (2.38):

$$\mathbf{x} = \mathcal{P}_\mathcal{W}(\mathbf{x}) + \mathcal{P}_{\mathcal{W}^\perp}(\mathbf{x}), \ \mathcal{P}_\mathcal{W}(\mathbf{x}) \in \mathcal{W} \text{ and } \mathcal{P}_{\mathcal{W}^\perp}(\mathbf{x}) \in \mathcal{W}^\perp. \tag{2.40}$$

[5] Unless otherwise noted, for the rest of this thesis, the term projection is used interchangeably with the term orthogonal projection (as opposed to an oblique one).

The projection is a linear operator, with several important properties, some of which we state here. The proofs are discussed in a variety of linear algebra texts such as [2].

It can be shown that $\mathcal{P}_{\mathcal{W}}$ computes the vector $\mathbf{u} \in \mathcal{W}$ that minimizes the distance $||\mathbf{x} - \mathbf{u}||$:

$$||\mathbf{x} - \mathcal{P}_{\mathcal{W}}(\mathbf{x})|| \leq ||\mathbf{x} - \mathbf{u}||, \text{ for all } \mathbf{u} \in \mathcal{W}, \tag{2.41}$$

with equality if and only if $\mathbf{u} = \mathcal{P}_{\mathcal{W}}(\mathbf{x})$.

If the vector $\mathbf{z}$ projected onto $\mathcal{W}$ already belongs to $\mathcal{W}$, then the direct sum decomposition is $\mathbf{z} = 0 \oplus \mathbf{z}$. Thus the projection is the identity operator. From that it follows that $\mathcal{P}_{\mathcal{W}}(\mathcal{P}_{\mathcal{W}}(\mathbf{x})) = \mathcal{P}_{\mathcal{W}}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{H}$. An implication is that any vector $\mathbf{z} \in \mathcal{W}$ is an eigenvector of $\mathcal{P}_{\mathcal{W}}(\cdot)$ with corresponding eigenvalue $\lambda = 1$. Similarly, any vector $\mathbf{y} \in \mathcal{W}^{\perp}$ is an eigenvector of $\mathcal{P}_{\mathcal{W}}(\cdot)$ with corresponding eigenvalue $\lambda = 0$. The multiplicity of $\lambda = 1$ and $\lambda = 0$ in the eigenvalue decomposition of $\mathcal{P}_{\mathcal{W}}(\cdot)$ is equal to the dimensionality of the corresponding spaces—$\dim(\mathcal{W})$, and $\dim(\mathcal{W}^{\perp})$ respectively. A projection operator has no other eigenvalues.

Using the eigenvalues it can be shown that the projection cannot increase the magnitude of a vector. Indeed, $||\mathcal{P}_{\mathcal{W}}(\mathbf{x})|| \leq ||\mathbf{x}||$ for all $\mathbf{x} \in \mathcal{H}$, with equality if and only if $\mathbf{x} \in \mathcal{W}$. It can be similarly shown that a projection is a positive semidefinite operator, i.e. $\langle \mathcal{P}_{\mathcal{W}}(\mathbf{x}), \mathbf{x} \rangle \geq 0$ for all $\mathbf{x} \in \mathcal{H}$, with equality if and only if $\mathbf{x} \in \mathcal{W}^{\perp}$.

Projections are extensively studied operators, both because of their important properties and their usefulness in several fields. This section by no means exhausts the known properties of projections. Extensive discussion can be found in linear algebra textbooks such as [2].

2.2.1 Projections and Frame Expansions

The analysis of a vector using an analysis frame followed by synthesis using a corresponding synthesis frame projects the vector to the space $\mathcal{W}$ spanned by the two frames. Using (2.39), the analysis of the vector $\mathbf{x} \in \mathcal{H}$ using the analysis frame is:

$$\begin{aligned} a_k &= \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle && (2.42)\\ &= \langle \mathcal{P}_{\mathcal{W}}(\mathbf{x}), \underline{\mathbf{f}}_k \rangle + \langle \mathcal{P}_{\mathcal{W}^{\perp}}(\mathbf{x}), \underline{\mathbf{f}}_k \rangle && (2.43)\\ &= \langle \mathcal{P}_{\mathcal{W}}(\mathbf{x}), \underline{\mathbf{f}}_k \rangle + 0. && (2.44) \end{aligned}$$

Therefore, the analysis of $\mathbf{x}$ is equal to the analysis of $\mathcal{P}_{\mathcal{W}}(\mathbf{x})$. Since analysis followed by synthesis is the identity for all vectors in $\mathcal{W}$, it follows that analysis of any vector $\mathbf{x} \in \mathcal{H}$, followed by synthesis is the projection of that vector onto $\mathcal{W}$:

$$\begin{aligned} &\sum_k \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle \mathbf{f}_k = \mathcal{P}_{\mathcal{W}}(\mathbf{x}) && (2.45)\\ &\Leftrightarrow S \cdot F(\mathbf{x}) = \mathcal{P}_{\mathcal{W}}(\mathbf{x}), && (2.46) \end{aligned}$$

in which the domain of the analysis frame operator F is extended to $\mathcal{H}$ by setting $F(\mathbf{u}) = 0$ for all $\mathbf{u} \in \mathcal{W}^{\perp}$.

Often it is also convenient to view the frame synthesis operator $S : \mathcal{H} \rightarrow \mathcal{W}$ as the combination of two operators: a projection $\mathcal{P}_{\mathcal{W}} : \mathcal{H} \rightarrow \mathcal{W}$, followed by a full rank operator $S_f : \mathcal{W} \rightarrow \mathcal{W}$. This implies that $\mathcal{W} \subseteq \mathcal{H}$. The frame is tight if and only if the full rank operator S_f is unitary. The projection operator $\mathcal{P}_{\mathcal{W}}$ rejects the nullspace of the frame operator, and, thus, is responsible for the redundancy of the frame.

2.3 Quantization

Quantization is a non-invertible process that maps a continuous space to a set of discrete points in that space. Quantization introduces distortion in the signal. Optimal quantizers should minimize distortion for a pre-determined number of output points or use as few output point as possible to achieve a pre-determined average distortion. Often, optimality is traded off for other features such as implementation simplicity.

2.3.1 Scalar Quantization

A scalar quantizer is a non-linear, non-invertible map $Q : \mathbb{R} \rightarrow P$, in which $P = \{p_1, p_2, \ldots, p_L \in \mathbb{R}\}$ is a discrete set of L levels, with L usually finite. The quantization function assigns each real number to one of these levels. It is completely defined by a set of disjoint intervals I_i covering the reals, and the corresponding levels p_i, such that each scalar in an interval is assigned to the corresponding level.

$$\left\{(I_i, p_i) \,\middle|\, \forall\, i \neq j :\ I_i \cap I_j = \emptyset, \bigcup_{i=1}^{L} I_i = \mathbb{R}\right\} \tag{2.47}$$

$$\Rightarrow \qquad \hat{a} = Q(a) = p_i \text{ if } a \in I_i \tag{2.48}$$

Although any function with a discrete set of output levels can be used as a quantizer, it is reasonable to impose that the function is monotonic.

In order to minimize the distortion given the set of levels, the quantizer should assign each scalar to the closest level. In this case, the set of points P completely defines the set of intervals $\{I_i\}$, and thus, the quantizer[6]:

$$a \in I_i \Leftrightarrow i = \text{argmin}||a - p_i||, \tag{2.49}$$

in which $||\cdot||$ is the distortion measure. In most practical applications the distortion is monotonic in the magnitude $|\cdot|$ of the difference, and therefore can be replaced by that magnitude.

A quantizer might also trade-off distortion for implementation simplicity. For example a round-towards-zero quantizer quantizes a to the closest point p_i that has

[6] We ignore the boundaries of the intervals I_i, which can be open or closed, as long as the union covers the space. The boundaries are a set of measure 0, and the assignment of each boundary to either of the adjacent intervals has no effect in the average distortion.

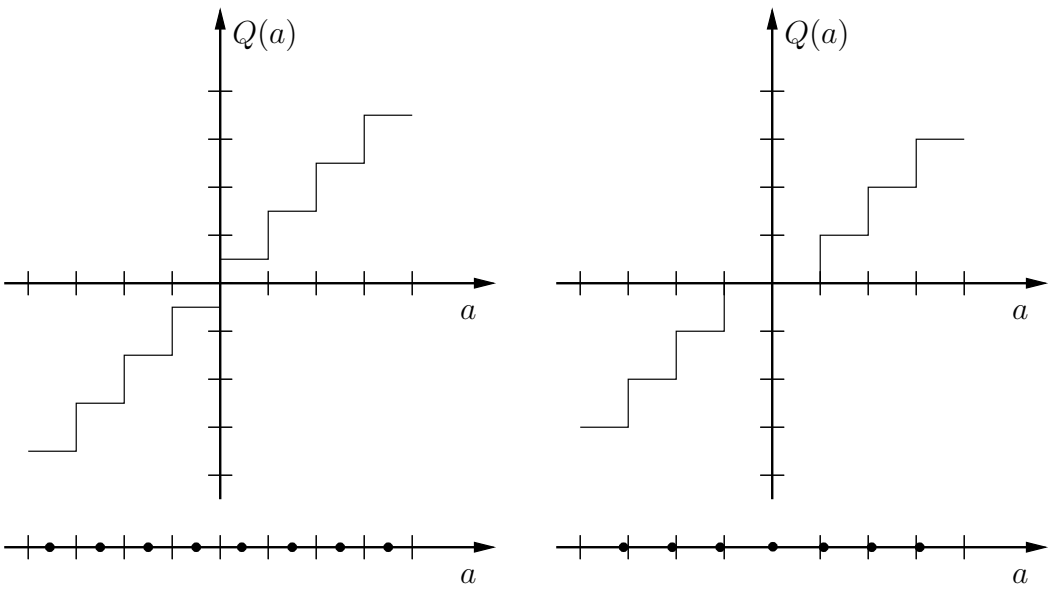


FIGURE 2-4: Examples of scalar quantizers.

magnitude smaller than a:

$$a \in I_i \Leftrightarrow i = \text{argmin}_{|p_i|<|a|} ||a - p_i||. \tag{2.50}$$

Such a quantizer can be implemented by removing, for example, the bits below a certain accuracy level of the binary representation of coefficient.

For the remainder of this thesis, unless otherwise noted, we assume that scalar quantization is performed using a uniform quantizer. A uniform quantizer has uniformly spaced quantization levels $p_i = \mu + i\Delta$, in which i is an integer. Depending on the application, the number of levels L is finite or infinite. The corresponding intervals $I_i = (p_i - \Delta/2, p_i + \Delta/2]$ are chosen using (2.49) to minimize the distortion due to quantization. Thus, the maximum error of a uniform quantizer is $|e_{max}| = \Delta/2$.

A scalar quantizer can be represented graphically by marking the points and the corresponding intervals on a real line or by drawing the function $\hat{a} = Q(a)$, as shown in figure 2-4. The figure shows a uniform quantizer and a round-towards-zero one. These commonly used quantizers combine implementation efficiency with low distortion, depending on the distortion measure.

2.3.2 Vector Quantization

Vector quantizers generalize scalar quantizers to a multidimensional vector space $\mathcal{W}$. They are defined through a set of points $P = \{\mathbf{p}_i \in \mathcal{W}\}$ and a corresponding set of

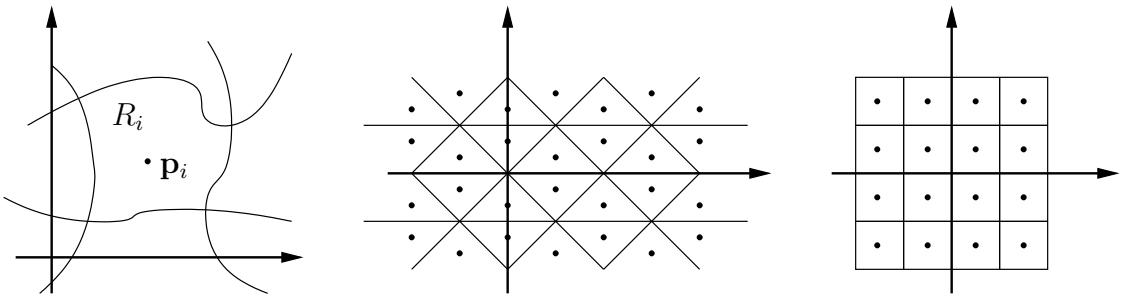


FIGURE 2-5: Examples of vector quantizers in two dimensions.

disjoint regions $\{R_i\}$ that partition $\mathcal{W}$:

$$\{(R_i, \mathbf{p}_i) \,|\forall\, i \neq j :\; R_i \cap R_j = \emptyset, \bigcup_i R_i = \mathcal{W}\} \tag{2.51}$$

$$\Rightarrow \qquad \hat{\mathbf{x}} = Q(\mathbf{x}) = \mathbf{p}_i \text{ if } \mathbf{x} \in R_i, \tag{2.52}$$

in which $Q : \mathcal{W} \rightarrow P$ is the non-linear, non-invertible quantization function. As with a scalar quantizer, to achieve minimum distortion given the quantization points, the regions should be such that any vector in the space is quantized to the nearest quantization point[7]:

$$\mathbf{x} \in R_i \Leftrightarrow i = \text{argmin}||\mathbf{x} - \mathbf{p}_i||, \tag{2.53}$$

in which $||\cdot||$ is the distance measure in the vector space.

Graphical representations of vector quantizers are possible in two dimensions, by extending the real line of figure 2-4 to the two dimensional plane and drawing the corresponding region-point pairs as shown in figure 2-5. The figure shows an arbitrary quantizer, a triangle lattice quantizer and a square lattice quantizer. For certain quantizers with regular structures visualization is also possible in three dimensional spaces, although the figures can be confusing.

The vector generalization of a uniform quantizer is a lattice quantizer. The quantization points of a lattice quantizer are defined as the sum of integer multiples of a linear independent vector set and a vector offset:

$$\mathbf{p}_{i_1,\ldots,i_M} = \mu + \sum_k i_k \mathbf{b}_k, \tag{2.54}$$

in which each i_k is an integer taking values in a finite or infinite range, depending on the quantizer model. In a lattice quantizer all the regions $R_{i_1,\ldots,i_M}$ corresponding to each point have the same shape and are all translations of a single fundamental cell

[7] As with the scalar quantizer, the region boundaries are a set of measure 0, and they can be assigned to any of the neighboring regions R_i.

(a) Scalar Quantizer (b) Vector Quantizer

FIGURE 2-6: Scalar and vector quantizer operating on a sequence of inputs

R, which is usually assumed to contain the origin. For this work we define

$$R \equiv R_{i_1,\ldots,i_M} - \mathbf{p}_{i_1,\ldots,i_M}, \tag{2.55}$$

which is the same for any choice of $i_1, \ldots, i_M$.

In general, vector quantizers are designed to partition the multidimensional space with less average distortion than scalar quantizers. Assuming a uniform source distribution, the minimum distortion is achieved by making the quantization cell shape as close to a hypersphere as possible (given the cell's volume). A hypersphere achieves the best worst-case and average error performance. Still, hyperspheres do not cover the space without overlap, and, therefore, cannot be used as quantization cells. Lattice vector quantizers attempt to create efficient, easy to use structures that have cells close to hyperspheres. Scalar quantizers on basis and frame expansions can also be viewed in terms of lattice vector quantizers, a view explored in chapter 4.

2.3.3 Additive Noise Models

The non-linearities of quantizers make them quite difficult to analyze, especially in the context of linear systems [29, 13, 30]. To facilitate the analysis it is often convenient to model the quantizer using an additive stochastic model, although the quantizer itself is a deterministic function. The models, first introduced in [6] are especially useful if the quantizer is used in a sequence of coefficients or vectors. We present them here in such a setup, as shown in figure 2-6.

In the case of a scalar quantizer we assume a linear quantizer with quantization interval Δ, properly scaled not to overflow. The quantizer quantizes each input a_k to $\hat{a}_k = Q(a_k) = a_k + e_k$, in which e_k is the additive error due to the quantization. The e_k is modeled as a white process, uncorrelated with the input a_k, with variance σ_e^2. Often it is further modeled as uniform in the interval $[-\Delta/2, \Delta/2]$, which implies $\sigma_e^2 = \Delta^2/12$ [29, 13].

Similarly for a vector quantizer, as shown in figure 2-6(b), the additive error $\mathbf{e}_k$ can be modeled as an uncorrelated sequence of vectors, independent of the data $\mathbf{x}_k$, and uniform in the quantization cell R, as defined in (2.55) [26, 42].

These stochastic models aim to describe average behavior of the quantizer over many signals. They provide no guarantees in individual realizations and their assumptions can often be quite inaccurate. For fine quantization levels these models are well motivated and provide a good description of the quantizer output. Their use in coarser quantization grids is not as well justified, but they are commonly used in practice, even in extreme quantization situations [29, 13, 28, 9]. In this work we complement these models with deterministic bounds to guarantee the performance of the quantizer in the worst case conditions.

CHAPTER 3 *Compensation Using Projections*

The redundancy of a frame expansion can be exploited in a variety of ways. For example, large frame dictionaries can lead to sparse representations useful in data compression [28]. They are also useful in the cases of signal degradation due to additive noise, quantization, or erasures [27, 20]. In this thesis we exploit the redundancy of frame expansions to linearly compensate for errors. Although the basic principle is straightforward, it occurs in several different contexts. Specifically we use this principle to analyze coefficient quantization and erasures. A recurring and important theme in the remainder of this thesis is that of compensation using projections. This chapter briefly introduces projections in the context of frame expansions and this thesis.

3.1 Error Compensation Using Representation Coefficients

Most of this thesis examines errors that corrupt one or more frame expansion coefficients. We compensate for these errors by modifying other coefficients in the frame expansion. Specifically, we assume that some coefficient a_i is corrupted and replaced by $\hat{a}_i = a_i + e_i$. The coefficients $\{a_k | k \in S_i\}$ are to be modified to $\{a'_k | k \in S_i\}$ to compensate for the known error e_i, in which $S_i = \{k_1, \ldots, k_p\}$ is the set of indices of the coefficients to be used. Of course, i cannot be a member of S_i, otherwise the error can be perfectly compensated for by modifying $\hat{a}_i$ back to a_i.

The reconstruction is performed using the synthesis equation (2.7) with the updated

coefficients, to produce $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \hat{a}_i \mathbf{f}_i + \sum_{k \in S_i} a'_k \mathbf{f}_k + \sum_{k \notin (S_i \cup \{i\})} a_k \mathbf{f}_k \tag{3.1}$$

The coefficients $\{a'_k | k \in S_i\}$ should be chosen to minimize the magnitude of the synthesis error:

$$\begin{aligned} \mathcal{E} &= \mathbf{x} - \hat{\mathbf{x}} && (3.2) \\ &= -e_i \mathbf{f}_i + \sum_{k \in S_i} (a_k - a'_k) \mathbf{f}_k && (3.3) \end{aligned}$$

To minimize the norm of the error we let $\mathcal{W}_i$ denote the space spanned by the set of vectors $\{\mathbf{f}_k, k \in S_i\}$ and recognize that $\sum_{k \in S_i} (a'_k - a_k)\mathbf{f}_k$ spans that space by appropriate choice of the a'_k. Thus, as discussed in the previous chapter, the magnitude of the error $||\mathcal{E}|| = || - e_i \mathbf{f}_i + \sum_{k \in S_i} (a_k - a'_k)\mathbf{f}_k||$ is minimized if and only if the vector formed by the sum is the projection of $e_i \mathbf{f}_i$ onto $\mathcal{W}_i$:

$$\begin{aligned} \sum_{k \in S_i} (a_k - a'_k)\mathbf{f}_k &= \mathcal{P}_{\mathcal{W}_i}(e_i \mathbf{f}_i) && (3.4) \\ \Leftrightarrow \sum_{k \in S_i} a'_k \mathbf{f}_k &= \sum_{k \in S_i} a_k \mathbf{f}_k - e_i \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i) && (3.5) \\ \Leftrightarrow \sum_{k \in S_i} a'_k \mathbf{f}_k &= \sum_{k \in S_i} (a_k - e_i c_{i,k,S_i}) \mathbf{f}_k, && (3.6) \end{aligned}$$

in which $\sum_{k \in S_i} e_i c_{i,k,S_i} \mathbf{f}_k$ is the frame expansion of $\mathcal{P}_{\mathcal{W}_i}(e_i \mathbf{f}_i)$ onto $\mathcal{W}_i$. By the linearity of the projection operator, the coefficients $\{c_{i,k,S_i}, k \in S_i\}$ are not a function of the error e_i or the data a_k, and, therefore, can be determined using only the frame vectors. Specifically we define these coefficients such that:

$$\begin{aligned} \mathcal{P}_{\mathcal{W}_i}(e_i \mathbf{f}_i) &= \sum_{k \in S_i} e_i c_{i,k,S_i} \mathbf{f}_k && (3.7) \\ \Rightarrow \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i) &= \sum_{i \in S_i} c_{i,k,S_i} \mathbf{f}_k, && (3.8) \end{aligned}$$

In other words, c_{i,k,S_i} are the frame expansion coefficients of the projection of $\mathbf{f}_i$ onto the space $\mathcal{W}_i$, defined by the span of the frame vectors $\{\mathbf{f}_k | k \in S_i\}$. We refer to the c_{i,k,S_i} as the compensation coefficients. Using these, we update each of the a_k to:

$$a'_k = a_k - e_i c_{i,k,S_i}. \tag{3.9}$$

This ensures that equation (3.4) is satisfied.

Although the projection $\mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$ is unique, the corresponding compensation coefficients and the assignment (3.9) are not necessarily unique. Specifically, the coefficients are unique if and only if the frame vectors $\{\mathbf{f}_k | k \in S_i\}$ are linearly independent. Otherwise the frame formed for $\mathcal{W}_i$ is redundant, and the expansion, as

discussed in chapter 2, is not unique.

If the vector $\mathbf{f}_i$ on which the error occurs is linearly dependent with the vectors $\{\mathbf{f}_k | k \in S_i\}$, then the error can be perfectly compensated for. Otherwise, the resulting error is $e_i \left(\mathbf{f}_i - \mathcal{P}_{\mathcal{W}}(\mathbf{f}_i)\right)$. We separate the magnitude from the direction of the error by defining the *error coefficient* $\tilde{c}_{i,S_i}$, and the *residual direction* vector $\mathbf{r}_{i,S_i}$:

$$\tilde{c}_{i,S_i} = ||\mathbf{f}_i - \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)|| \tag{3.10}$$

$$\mathbf{r}_{i,S_i} = \frac{\mathbf{f}_i - \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)}{||\mathbf{f}_i - \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)||}, \tag{3.11}$$

such that the error is $e_i \tilde{c}_{i,S_i} \mathbf{r}_{i,S_i}$. These two quantities are particularly useful in the performance analysis of the algorithms presented in chapters 5 through 7.

The error coefficient is always positive since it is the magnitude of a vector. Also, the residual direction is a vector that has, by definition, unit magnitude. Furthermore, it should be emphasized that the error vector is orthogonal to the space $\mathcal{W}_i$, which includes all the frame vectors $\{\mathbf{f}_k | k \in S_i\}$ used in the compensation.

In denoting the coefficients the set S_i of indices used to compensate a_i is explicitly mentioned. This choice was made to emphasize that modifying the set of indices used also modifies the corresponding compensation and error coefficients, and the residual error. The notation used in this chapter disambiguates potential confusion on which coefficients should be used. However, this notation can become cumbersome in simple situations in which the set of indices is clear. Therefore, in the remainder of this thesis the set S_i might or might not be included, depending on the context. For the remainder of this chapter the notation $c_{i,k}$, $\mathbf{r}_i$, and $\tilde{c}_i$ is used, and the set S_i is implied.

In most of the applications considered in this thesis, the frame is known in advance. Thus the compensation coefficients can be precomputed off-line, at the design stage of the system, together with the error coefficients and the residual directions. Furthermore, the error can often be detected when it occurs, and, therefore the compensation can be computed at run-time by appropriately scaling the compensation coefficients. Even if the error cannot be explicitly obtained at run-time, it is sometimes possible to pre-compensate for the error such that after the error occurs, the data can be restored while the error is compensated for.

3.1.1 Computation of the Projection Coefficients

As described in equation (3.8) in the previous section, the $c_{i,k}$ should be such that $\sum_{i \in S_i} c_{i,k} \mathbf{f}_k = \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$. Thus, the compensation coefficients are the frame expansion coefficients of $\mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$ using the frame $\{\mathbf{f}_k | k \in S_i\}$. From the discussion in chapter 2, it follows that an analysis frame for $\{\mathbf{f}_k | k \in S_i\}$ exists and could be used to determine the compensation coefficients. We use $\{\phi_k^{S_i} | k \in S_i\}$ to denote this set, which is different than the subset of the analysis frame vectors from the original set corresponding to the coefficients to be used for compensation.

The projection coefficients can, in principle, be computed by first calculating the $\{\phi_k^{S_i}\}$, and using these to compute the inner products:

$$c_{i,k} = \langle \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i), \phi_k^{S_i} \rangle. \tag{3.12}$$

The vectors $\phi_k^{S_i}$ lie in $\mathcal{W}_i$ by construction. Therefore the projection operator $\mathcal{P}_{\mathcal{W}_i}(\cdot)$ can be removed using:

$$\begin{aligned} \langle \mathbf{f}_i, \phi_k^{S_i} \rangle &= \langle \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i) + \mathcal{P}_{\mathcal{W}_i^\perp}(\mathbf{f}_i), \phi_k^{S_i} \rangle & (3.13) \\ &= \langle \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i), \phi_k^{S_i} \rangle & (3.14) \\ \Rightarrow c_{i,k} &= \langle \mathbf{f}_i, \phi_k^{S_i} \rangle. & (3.15) \end{aligned}$$

Calculating the dual set $\{\phi_k^{S_i}\}$, however, can be computationally expensive, and it is not necessary in order to calculate the projection coefficients. Instead, using the inner product of (3.8) with $\mathbf{f}_l$, for all $l \in \mathcal{W}_i$, it follows that:

$$\begin{aligned} \langle \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i), \mathbf{f}_l \rangle &= \sum_{k \in S_i} \langle c_{i,k} \mathbf{f}_k, \mathbf{f}_l \rangle & (3.16) \\ \Leftrightarrow \begin{bmatrix} R_{i,k_1} \\ \vdots \\ R_{i,k_p} \end{bmatrix} &= \begin{bmatrix} R_{k_1,k_1} & \cdots & R_{k_1,k_p} \\ \vdots & \ddots & \vdots \\ R_{k_p,k_1} & \cdots & R_{k_p,k_p} \end{bmatrix} \begin{bmatrix} c_{i,k_1} \\ \vdots \\ c_{i,k_p} \end{bmatrix} & (3.17) \\ \Leftrightarrow \rho &= \mathbf{R}\mathbf{c}, & (3.18) \end{aligned}$$

in which $R_{k,l} = \langle \mathbf{f}_k, \mathbf{f}_l \rangle$ is the frame autocorrelation. Satisfying this equation is equivalent to computing the projection coefficients with an analysis frame, as described above. If the frame $\{\mathbf{f}_k | k \in \mathcal{W}_i\}$ is redundant, the matrix $\mathbf{R}$ of autocorrelations is not full rank. Any left inverse can be used to compute the projection coefficients. The use of the left pseudoinverse in the solution of (3.18) is equivalent the use of the dual frame of $\{\mathbf{f}_k | k \in S_i\}$ in $\mathcal{W}_i$ as the $\{\phi_k^{S_i} | k \in S_i\}$ in (3.15).

Computation for Shift-invariant Frames

If the frame is shift invariant, the frame autocorrelation is a function only of the difference of the indices of the frame vectors:

$$R_{i,i+k} = R_{0,k} \equiv R_k. \tag{3.19}$$

If, furthermore, the set S_i consists of the p coefficients subsequent to the corrupted one (i.e. $S_i = \{i+1, \ldots, i+p\}$), then the projection coefficients are also shift-invariant:

$$c_{i,i+k} = c_{0,k} \equiv c_k. \tag{3.20}$$

In this case, equation (3.18) takes the special form of the autocorrelation normal equations, or the Yule-Walker equations [41, 40]:

$$\rho = \begin{bmatrix} R_1 \\ \vdots \\ R_p \end{bmatrix} = \begin{bmatrix} R_0 & \cdots & R_{p-1} \\ \vdots & \ddots & \vdots \\ R_{p-1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_p \end{bmatrix} = \mathbf{R}\mathbf{c}. \tag{3.21}$$

Although the solution to these equations can be determined using a general matrix inversion algorithm, it is more efficient to use the Levinson-Durbin recursion [32, 25].

In addition to the computational efficiency, the Levinson recursion algorithm, being recursive in the equation order, provides the solution to the equations for all intermediate orders $1, \ldots, p$. In certain applications the intermediate solutions are useful for the compensation of subsequent errors. Section 5.8 provides an example in which the intermediate solutions are used in the compensation system. In that case, the use of the Levinson recursion reduces the computation from $O(M^4)$ to $O(M^2)$.

If the frame is redundant, the system (3.21) might be underdetermined, and the solution is not unique. The Levinson recursion determines one possible solution to the problem. This solution is not the one corresponding to the left pseudoinverse of the problem, which is not necessarily an issue. However, it is a property of the recursion to be aware of during system design.

3.1.2 Projections and Re-expansion of the Error

As noted in the previous section, depending on the frame and the set S_i used for the projection, the compensation coefficients might or might not be unique. If the former is the case, then the solution to equation (3.18) is uniquely determined by inverting the matrix $\mathbf{R}$. However, if the solution is not unique, it can be determined using a variety of algorithms. Although all solutions are optimal in terms of the compensation error, each algorithm might have significant advantages over others depending on the application.

It is important to further recognize that equation (3.18) is derived assuming equations (3.8) and (3.9). These are sufficient but not necessary to minimize the error magnitude. The correction should only satisfy (3.4). The necessary and sufficient condition for (3.4) is that the sum $\sum_{k \in S_i} (a_k - a'_k)\mathbf{f}_k$ is the frame expansion of $\mathcal{P}_{\mathcal{W}_i}(e_i \mathbf{f}_i)$ using $\{\mathbf{f}_k | k \in \mathcal{W}_i\}$ as the synthesis frame. Equations (3.8) and (3.9) are derived from (3.4) only with computational simplicity and linearity in mind.

As discussed in chapter 2, this expansion can be computed using a variety of methods, adaptive or not, such as inner products with an analysis frame or the use of the matching pursuit [33, 28]. The choice implied by (3.9) is equivalent to using inner products with an analysis frame. Furthermore, inverting $\mathbf{R}$ using the pseudoinverse implies that the analysis frame is the dual frame. This, however might not be the best choice for the application considered. For example, a sparse solution reduces the computation in updating the coefficients using (3.9). Alternatively, minimizing the

maximum magnitude of the projection coefficients (i.e. the l_∞ norm of $\mathbf{c}$) reduces the effect of the coefficient updates to each of the updated coefficients.

The optimal method might also be data dependent. If we recognize that:

$$\sum_{k \in S_i} a'_k \mathbf{f}_k = \sum_{k \in S_i} a_k \mathbf{f}_k - e_i \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i) \tag{3.22}$$

satisfies (3.4), then the $\{a'_k | k \in S_i\}$ are the frame expansion of $\sum_{k \in S_i} a_k \mathbf{f}_k - e_i \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$ using the $\{\mathbf{f}_k | k \in \mathcal{W}_i\}$ as the synthesis frame. The $\{a'_k | k \in S_i\}$ can be determined using any analysis algorithm, and used to replace the corresponding $\{a_k\}$. This is not necessarily the same as expanding only the error vector $e_i \mathbf{f}_i$ and adding the expansion to the existing $\{a_k\}$. For example, to maintain coefficient quantization, taking into account coefficients a_k before computing the update can be beneficial. Still, in this thesis, we only consider the linear, data independent case. It should be emphasized however, that all the solutions are optimal if the error magnitude is the metric.

Even in the case for which the solution in (3.18) is unique, it might be more important for an application to tolerate more error in order to improve other aspects of the design. For example, in chapters 5 and 6 it is demonstrated that in the case of Sigma-Delta noise shaping the projection coefficients are modified from the optimal choice in order to eliminate products and reduce the computational complexity in the feedback loop. Section 7.2.3 shows that a suboptimal solution can improve the stability of the resulting systems.

Projections can be extended to compensate for errors that affect sets of coefficients at once, such as block erasures or vector quantization, using a set of coefficients that is not affected by the error. In this case the whole error vector should be projected to the space used for the compensation, using the same principle. We do not explicitly develop the formulation of this problem in the applications presented here, but the setup and solution is straightforward as long as care is taken in the bookkeeping of the indices. For example, the compensation coefficient vector $\mathbf{c}$ should become a matrix, and so should ρ in equation (3.18).

3.2 Pre-compensation Followed by Post-correction

Implementation of equation (3.9) is straightforward if the error is known during compensation. A more surprising result, however, is that in some cases compensation using projections can be performed even if the error is not known at the point of compensation. In certain cases it is possible to pre-compensate for the error before it occurs, and undo the compensation after the error occurrence. For the remainder of this section we assume that the application allows two systems to be inserted one before and one after the error occurs, before the signal synthesis. The systems are not allowed to share information directly, but are allowed to modify the coefficients on which the error occurs. We also assume the error occurs only on one coefficient,

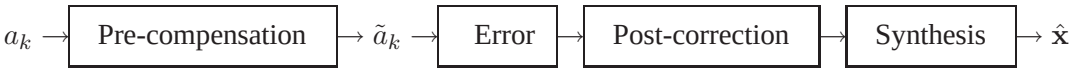


FIGURE 3-1: Architecture of the system implementing pre-compensation followed by post-correction of a single coefficient error.

a_i, and is not dependent[1] on the coefficients used for the compensation. Sequential application of the principle to multiple coefficients is also possible. The details of sequential application vary depending on the application. Thus, they are discussed in the corresponding chapters.

The pre-compensation algorithm can be implemented using the system in figure 3-1. In the figure the system implementing the pre-projection modifies coefficients $\{a_k | k \in S_i\}$ to:

$$\tilde{a}_k = \begin{cases} a_k + a_i c_{i,k}, & \text{if } k \in \mathcal{S}_i \\ a_k, & \text{otherwise.} \end{cases} \tag{3.23}$$

The modified coefficients are used instead of the original ones, thus representing the sum of $\mathbf{x}$ and the projection of $a_i \mathbf{f}_i$ onto $\mathcal{W}_i$: $\tilde{\mathbf{x}} = \mathbf{x} + a_i \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$. The error modifies coefficient $\tilde{a}_i = a_i$ to:

$$\hat{a}_i = \tilde{a}_i + e_i = a_i + e_i, \tag{3.24}$$

which are subsequently input to the post-correction system to modify $\{\tilde{a}_k | k \in S_i\}$ to a'_k:

$$\begin{aligned} a'_k &= \tilde{a}_k - \hat{a}_i c_{i,k} && (3.25) \\ &= a_k - e_i c_{i,k}. && (3.26) \end{aligned}$$

The a'_k are used for the reconstruction $\hat{\mathbf{x}}$ of $\mathbf{x}$. The error is equal to:

$$\begin{aligned} \mathcal{E} &= e_i \mathbf{f}_i - e_i \sum_{k \in S_i} c_{i,k} \mathbf{f}_k && (3.27) \\ &= e_i (\mathbf{f}_i - \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)) && (3.28) \\ &= e_i \tilde{c}_i \mathbf{r}_i, && (3.29) \end{aligned}$$

which is the same as if the error was known before the compensation.

[1] In this statement, "not dependent" can be interpreted in two different ways: a) if the error is random, then it should be statistically independent to the coefficients $\{a_k, k \in S_i\}$ used for compensation, or b) if the error is a deterministic but unknown function of the data, then it should not be a function of the coefficients $\{a_k, k \in S_i\}$.

CHAPTER 4

Quantization of Frame Representations

This chapter examines scalar quantization on frame representations, and derives bounds on the performance of the quantizer. The performance is evaluated by considering how many of the available quantization points are used by the quantizer, not by considering the shape of the corresponding quantization regions. Thus a lower bound on the error and the bit waste as a function of the redundancy and the number of quantization levels is derived.

A similar bound has been previously demonstrated in [39] for the oversampling frame of periodic functions. In that work the quantization is shown to partition the signal space in a particular structure called hyperplane wave structure. An upper bound on the cell density is derived, which is subsequently used to derive a lower bound on the error decay of the quantization as a function of the redundancy. The analysis assumes an infinite level uniform quantizer, although it is shown that a finite level quantizer can only perform worse.

Although the analysis in [39] can be applied to any finite frame, this chapter takes a different view of the problem. Specifically, we consider the map of the frame analysis operator from the signal space $\mathcal{W}$ to a higher dimensional space $\mathcal{H}$. Scalar quantization of the analysis coefficients is equivalent to scalar quantization of a basis expansion in $\mathcal{H}$. By considering the hyper-cube quantization lattice generated by the scalar quantizer on the basis expansion, it is shown that the image of $\mathcal{W}$ under the

frame operator does not reach all the quantization cells, and therefore cannot use all the quantization points.

The analysis in this chapter explicitly assumes a fixed number of quantization levels and provides a slightly different bound than [39]. Specifically, the bound we provide is a function of the quantization levels and does not depend on the quantizer being uniform. Asymptotically, both bounds demonstrate the same growth rates. However, the analysis in this chapter allows us to quantify the error decay and the bit waste of any scalar quantizer both as a function of the redundancy and the number of quantization levels of the quantizer. In principle, the same result can be derived by extending the approach in [39], but the proof in this chapter provides a more straightforward generalization.

4.1 Quantization of Orthonormal Basis Expansions

In this section we assume an orthonormal basis $\{\mathbf{b}_k\}$ in a space $\mathcal{H}$. The basis expansion coefficients can each take any of L quantization levels, uniformly spaced with interval Δ:

$$p_i = \mu + i\Delta,\ i = 1, \ldots, L. \tag{4.1}$$

Thus, the L^M points in $\mathcal{H}$ that can be represented using the quantized expansion are:

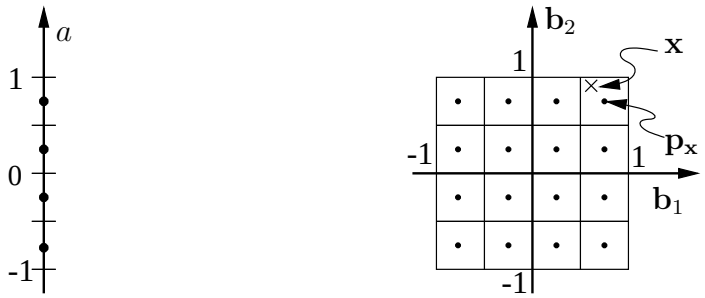
$$\mathbf{p}_{i_1,\ldots,i_M} = \sum_{k=1}^{M} p_{i_k}\mathbf{b}_k \tag{4.2}$$

$$= \mu \sum_{k} \mathbf{b}_k + \Delta \sum_{k=1}^{M} i_k \mathbf{b}_k,\ \ i_k \in \{1, \ldots, L\},\ k = 1, \ldots, M, \tag{4.3}$$

in which k denotes the dimension, and i_k the corresponding quantization level in each dimension.

The quantization cells in this lattice, denoted by R_i, are hypercubes of size Δ^M centered at each of the L^M quantization points. A scalar quantizer quantizes any vector $\mathbf{x}$ in a cell to the corresponding quantization point at the center of the cell. For a basis expansion, this is an optimal coefficient quantization strategy. Figure 4-1 demonstrates (a) a scalar quantizer and (b) the two-dimensional square lattice generated if the scalar quantizer operates on a two dimensional basis expansion. In the figure, vector $\mathbf{x}$ is quantized to the nearest point $\mathbf{p}_\mathbf{x}$.

The assumption of a uniform quantizer is not necessary for the development in this chapter, except where explicitly noted. Any finite level scalar quantizer with L levels can be used instead. The effect is that the lattice ceases to be uniform and the cells are not translations of a fundamental hypercube. Instead the cells become arbitrary hypercuboids—generalizations of rectangles in higher dimensions. For the purposes of clarity, the assumption of a uniform quantizer is maintained, although it is not necessary, unless the quantizer parameter Δ is explicitly used.



(a) Scalar Quantizer (b) Corresponding Square Lattice

FIGURE 4-1: Example of a scalar quantizer, and the square lattice generated by the scalar quantizer operating on a two dimensional basis expansion.

4.2 Quantization Grids and Frame Representations

Quantization of frame representations can also be analyzed using lattices and lattice points. Both frame analysis and synthesis can be related to quantization points on orthogonal lattices using the analysis and synthesis operators, respectively, as described in chapter 2. However, the decoupling of the analysis from the synthesis operation implies that quantizing the analysis coefficients should be approached separately from the synthesis from quantized coefficients. The next section explores the synthesis from quantized coefficients, while section 4.2.2 examines the quantization of analysis coefficients.

4.2.1 Linear Reconstruction from the Quantized Coefficients

To understand the synthesis from quantized samples, we consider the synthesis operator S, as defined in equation (2.17)

$$S : \mathcal{H} \rightarrow \mathcal{W}, \text{ s.t. } S\mathbf{y} = \sum_k \langle \mathbf{y}, \mathbf{b}_k \rangle \mathbf{f}_k, \tag{2.17}$$

in which $\mathbf{y}$ is any vector in $\mathcal{H}$, and $\mathbf{b}_k$ is the basis set assumed by the synthesis operator. In synthesizing quantized frame representations, $\mathbf{y}$ is one of the quantization points, $\mathbf{p}_i$, and the inner products $\langle \mathbf{p}_i, \mathbf{b}_k \rangle$ take one of the discrete values of the scalar quantizer. All the quantization points lie on a square lattice in $\mathcal{H}$ defined by the interval spacing Δ of the quantizer and the basis $\{\mathbf{b}_k\}$.

The frame operator reconstructs these points $\mathbf{p}_i$ to $S\mathbf{p}_i$ in the low dimensional space $\mathcal{W}$, in which the frame lies. An expansion method that assumes a predetermined linear synthesis should quantize a vector $\mathbf{x} \in \mathcal{W}$ to the point i that minimizes the distance $||\mathbf{x} - S\mathbf{p}_i||$ in $\mathcal{W}$. This is not necessarily the same point that minimizes the distance $||\underline{F}\mathbf{x} - p_i||$, in which $\underline{F}$ is the analysis operator of the dual frame or any

other analysis frame.

In principle the desired point can be determined using exhaustive search, although in practice this is usually not possible, especially in the case of infinite frames. The quantized matching pursuit [28] or Sigma-Delta noise shaping on frames, described in the next section, are examples of efficient expansion methods that assume a fixed synthesis frame and aim to find the quantized representation closest to the original vector. Neither method claims the optimality of exhaustive search, but they are far simpler and practical, even for infinite frame expansions.

4.2.2 Analysis Followed by Scalar Quantization

If, instead, a linear analysis using inner products followed by scalar quantization is assumed, the signal space $\mathcal{W}$ is mapped to the higher dimensional space $\mathcal{H}$ through the analysis operator F:

$$F\mathbf{x} = \sum_k \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle \mathbf{b}_k. \tag{2.12}$$

Scalar quantization of the coefficients corresponds to scalar quantization of the basis expansion of $F\mathbf{x}$. Thus, the synthesis operation should reconstruct each quantization point $\mathbf{p}_i$ to the vector $\hat{\mathbf{x}}$ that minimizes the average distance from all the vectors that were quantized to $\mathbf{p}_i$. Linear reconstruction from the quantized samples, which corresponds to setting $\hat{\mathbf{x}} = S\mathbf{p}_i$ using the dual frame for S, is known to be suboptimal. Instead consistent reconstruction methods [38, 28] have been shown to improve the quantization performance.

To design the reconstruction and analyze the quantizer performance, we need to examine the map of $\mathcal{W}$ onto $\mathcal{H}$ through the analysis operator F. Since F is invertible, it has rank N, same as the dimensionality of $\mathcal{W}$. Thus, the image of $\mathcal{W}$ under F is an N dimensional subspace in $\mathcal{H}$, which we denote using $F(\mathcal{W})$. Scalar coefficient quantization of the frame expansion is equivalent to vector quantization of that subspace $F(\mathcal{W})$ using the lattice defined by the scalar quantizer and the basis $\{\mathbf{b}_k \in \mathcal{H}\}$ implied by the frame operator F.

Figure 4-2 illustrates an example for an arbitrary two-vector frame operating on a one-dimensional signal space $\mathcal{W}$. Any vector $\mathbf{x} \in \mathcal{W}$ is quantized to the point $\mathbf{p}_i$ closest to its map $F\mathbf{x}$. This implies that all the vectors in the intersection of $F(\mathcal{W})$ with the square cell R_i are quantized to the point $\mathbf{p}_i$. Consistent reconstruction produces the vector $\hat{\mathbf{x}}$ that minimizes the error from all the points in the inverse image of the intersection $F^\dagger(F(\mathcal{W}) \cap R_i)$ [38, 28].

In the figure it is also obvious that $F(\mathcal{W})$ only intersects a small number of the available quantization cells. The next section uses an upper bound on the number of cells intersected to derive a bounds on the quantization error and the bit use of the quantizer.

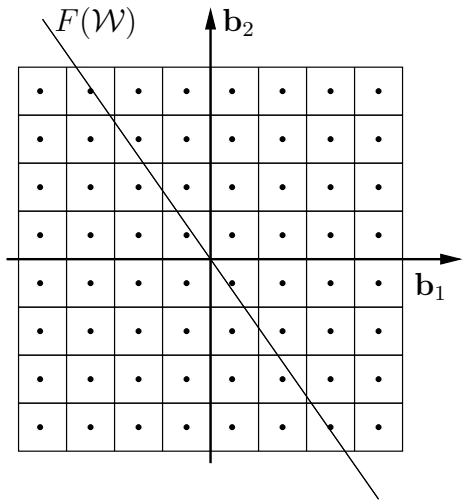


FIGURE 4-2: A scalar quantization example of a two-vector frame operating on a one-dimensional space.

4.3 Limits of Scalar Quantization of the Analysis Coefficients

Analysis using inner products followed by scalar quantization quantizes a vector $\mathbf{x}$ to the quantization point $\mathbf{p}_i \in \mathcal{H}$ if and only if the image of the point $F\mathbf{x} \in \mathcal{H}$ lies in the quantization cell R_i. Therefore, scalar quantization of the coefficients can only produce one of the quantization points in the cells intersected by $F(\mathcal{W})$. This is only a fraction of the L^M possible quantization points that can be represented by L possible quantization levels for each of the M coefficients. Assuming no subsequent entropy coding, this representation uses at least $\log_2(L)$ bits per coefficient, i.e. $M \log_2(L)$ bits in total to represent the coefficients.

An L-level uniform scalar quantizer operating on an M-dimensional basis expansion forms an M-dimensional hypercube lattice. The hypercube has width ΔL on each dimension for a total volume of $(\Delta L)^M$, in which Δ is the interval spacing of the quantizer. The lattice consists L^M cells, each of which is a smaller M-dimensional hypercube of size Δ^M. It is also not necessarily centered at zero; its position in the space depends on the constant μ of the scalar quantizer in equation (4.1). The lattice is intersected by $F(\mathcal{W})$, which is an N-dimensional subspace in $\mathcal{H}$.

We use $I(M, N, L)$ to denote the maximum number of cells that any hyperplane of dimension N intersects in a hypercube lattice of dimension M, with L cells per dimension. In section 4.4 it is shown that:

$$I(M, N, L) \leq (2L)^N \binom{M}{N}. \tag{4.4}$$

Thus, independent of the frame used for the expansion, at most $I(M,N,L)$ out of the possible L^M points are used. The binomial coefficient is upper bounded by:

$$\binom{M}{N} \leq \left(\frac{Me}{N}\right)^N \tag{4.5}$$

$$\Rightarrow I(M,N,L) \leq \left(\frac{2LMe}{N}\right)^N \tag{4.6}$$

$$= (2Lre)^N, \tag{4.7}$$

in which $r = M/N$ is the frame redundancy rate.

4.3.1 Representation Bits Use

The representation of the coefficients, assuming no subsequent entropy coding, uses at least $\log_2\left(L^M\right) = M\log_2(L) = rN\log_2(L)$ bits. However, only $I(M,N,L)$ cells are ever reached, i.e $I(M,N,L)$ points are ever used. To uniquely represent each of these points approximately $\log_2\left(I(M,N,L)\right)$ are necessary. Thus the ratio of necessary bits to the number of used bits is:

$$\frac{\log_2\left(I(M,N,L)\right)}{\log_2(L^M)} = \frac{N\log_2(2Lre)}{rN\log_2(L)} \tag{4.8}$$

$$= \frac{\log_2(2Lre)}{r\log_2(L)}. \tag{4.9}$$

As the quantization becomes finer (i.e. $L \to \infty$) the redundancy is not as helpful in reducing the quantization error in the signal. Therefore, as expected, the ratio of necessary to used bits tends to $1/r$, which implies that all the redundant coefficients can be removed without loss in the reconstruction. Similarly, for a constant L, as the redundancy r increases, the fraction grows as $O\left(\log_2(r)/r\right)$, and the ratio of necessary to used bits decreases.

The analysis above also provides a target bit rate for subsequent entropy coder. Specifically, independent of the distribution of the source $\mathbf{x}$ in $\mathcal{W}$, subsequent entropy coding of the representation should use at most $\log_2(I(M,N,L))$ bits, i.e. at most a fraction of $\frac{\log_2(2Lre)}{r\log_2(L)}$ of the input bits.

4.3.2 Lower Bound on the Quantization Error

Equation (4.4) can also be used to bound the error of a uniform quantizer. Without loss of generality we assume that the analysis frame vectors are normalized such that they have magnitude $||\underline{\mathbf{f}}_k|| \leq 1$, and examine two different cases. In the first case, the vectors represented by the frame have bounded length $||\mathbf{x}|| \leq R$. To ensure the quantizer does not overflow, the quantization interval is set to $\Delta = 2R/L$. In the second case, the quantization interval Δ is constant. For the quantizer not to overflow, the vectors to be represented should have magnitude bounded by $||\mathbf{x}|| \leq L\Delta/2$.

In the first case, the bound on the vector magnitude,

$$||\mathbf{x}|| \leq R, \tag{4.10}$$

implies that the vectors occupy a volume:

$$V = \frac{2\pi^{N/2}R^N}{N\Gamma(N/2)}, \tag{4.11}$$

in which $\Gamma(\cdot)$ is the Gamma function, and N is the dimensionality of the low dimensional signal space $\mathcal{W}$. This volume is divided among all the attainable quantization points $I(M,N,L)$. Thus, there is some quantization point $\mathbf{p}$ that has corresponding volume at least $V/I(M,N,L)$. The maximum distance $\epsilon = ||\mathbf{x}-\mathbf{p}||$ of a vector $\mathbf{x}$ in that volume from the quantization point $\mathbf{p}$ follows:

$$\begin{aligned}
\frac{2\pi^{N/2}\epsilon^N}{N\Gamma(N/2)} &\geq \frac{V}{I(M,N,L)} = \frac{2\pi^{N/2}R^N}{N\Gamma(N/2)I(M,N,L)} && (4.12)\\
\Rightarrow \epsilon &\geq R\cdot I(M,N,L)^{-N} && (4.13)\\
\Rightarrow \epsilon &\geq R(2Lre)^{-1}, && (4.14)
\end{aligned}$$

which implies that the worst case error magnitude decreases as $\Omega(1/(Lr))$ as the redundancy r or the number quantization levels L increases.

Similarly, in the second case, the vectors are bounded by

$$||\mathbf{x}|| \leq L\Delta/2, \tag{4.15}$$

and occupy a volume:

$$V = \frac{2\pi^{N/2}(L\Delta/2)^N}{N\Gamma(N/2)}. \tag{4.16}$$

Using the same analysis as above:

$$\begin{aligned}
\frac{2\pi^{N/2}\epsilon^N}{N\Gamma(N/2)} &\geq \frac{V}{I(M,N,L)} = \frac{2\pi^{N/2}(L\Delta/2)^N}{N\Gamma(N/2)I(M,N,L)} && (4.17)\\
\Rightarrow \epsilon &\geq \left(\frac{L\Delta}{2I(M,N,L)}\right)^N && (4.18)\\
\Rightarrow \epsilon &\geq R\Delta(4re)^{-1}. && (4.19)
\end{aligned}$$

Therefore, the worst case error magnitude is proportional to the quantization interval Δ. The error decreases as $\Omega(\Delta/r)$.

We can also use Zador's formula [30] and an analysis similar to the one in [39] to determine a lower bound in the mean-squared error of quantization that decreases similarly to the worst case squared error, as $\Omega\left((Lr)^{-2}\right)$ in the first case and $\Omega\left((\Delta/r)^2\right)$ in the second. We do not present this here since the solution does not provide further intuition to the problem.

4.3.3 Discussion

The analysis above demonstrates that direct scalar quantization of the frame expansion coefficients is inefficient. Equation (4.9) quantifies the number of bits wasted using such an approach to quantization, or equivalently, what we should expect the minimum gain to be from subsequent entropy coding of the quantizer output. The worst-case or mean squared error decay of $\Omega\left(1/(Lr)\right)$ shows that doubling the redundancy rate r or doubling the number of quantization levels L of the quantizer reduces the quantization error at the same rate. However, doubling the redundancy rate r doubles the bits used by the representation while doubling the number of levels L only uses one more bit per coefficient, i.e. M more bits in total. Similarly for the $\Omega\left((\Delta/r)^2\right)$ case. Therefore, decreasing the error by refining the quantizer is the more rate-efficient approach to decrease the error.

The bound in (4.4) is on the number of cells of the hypercube that any hyperplane can intersect. Therefore, any affine data-independent operation on the frame expansion coefficients has no effect on the result as long as it does not change the redundancy of the coefficients. This is necessary to accommodate arbitrary offsets in the linear quantizers, but it also implies that any data-independent translation of the coefficients before the quantization (such as data-independent deterministic or random dither) does not improve the asymptotic performance of the scalar quantizer. Furthermore, the derivation of the bound does not assume uniform quantization intervals. Therefore, any monotonic scalar transformation of the representation coefficients cannot improve the bit-use efficiency of the quantizer, or the error decay rate—although the constants might be affected.

The synthesis method is not considered in the analysis above. The results provide the lower bound on the error for any synthesis method. However, the use of the synthesis sum in equation (2.7) with the dual frame does not necessarily reach that bound. In fact, it has been shown that the method that achieves the lower bound (at least asymptotically) is consistent reconstruction [28].

A significant conclusion of this chapter is that analysis using inner products followed by individual scalar coefficient quantization is not efficient. If, instead, the expansion method is able to reach all the $L^M = L^{rN}$ quantization points available, then, in principle, the error squared can decay exponentially as $O\left(L^{-r}\right)$. This implies that rate-efficient quantized representations employ non-linear expansion methods such as the quantized matching-pursuit [33, 28] or the generalization to Sigma-Delta noise shaping described in the next chapter. These assume the frame synthesis is predetermined and the determination of the quantized coefficients is a non-linear process taking the synthesis frame vectors into account. Both methods try to determine a quantized representation that has a reconstruction closer to the original signal than simple scalar quantization of the frame expansion coefficients. It has been shown, for example, that using p^{th} order Sigma-Delta noise shaping on the oversampling frame the error decays as $\Omega(r^{-(p-1)})$ [37].

We should also note, that all the results above are best-case results, based only on the number of quantization cells reached by the frame analysis. The advantage is that

this method is independent to the frame used. Frames that reach fewer quantization cells exist and have worse performance. There is also no discussion on how the frame partitions the volume of the vector space to the the corresponding cells. The proof provides only some intuition on how a rate-efficient frame should be designed, but not necessary or sufficient conditions to reach the bound.

4.4 Intersection of a Hyperplane with a Hypercube Lattice

In this section we prove the result we asserted in equation (4.4) using a proof recursive in N and M. The problem, as defined in the previous section, is counting the maximum number of hypercube cells in an M-dimensional space that can be intersected by an N-dimensional hyperspace. We denote the number of these cells using $I(M, N, L)$. Assuming a uniform quantizer, and without loss of generality, we scale, rotate and translate the problem such that the hypercubes are aligned with the integers, and have sides at integer coordinates. Then the subspace becomes an arbitrary N-blade, i.e. an N-dimensional hyperplane in the M-dimensional plane.

The lattice boundaries of the hypercubes become $(M-1)$-blades satisfying $x_k = \langle \mathbf{x}, \mathbf{b}_k \rangle = i$, in which $k \in \{1, \ldots, M\}$ is the coordinate index and $i \in \{0, \ldots, L\}$ is the boundary index along that direction, for a total of $M(L+1)$ lattice boundaries. We call the boundaries at $i_k = 1, \ldots, L-1$ internal, and the boundaries at $i_k = 0, L$ external, for a total of $M(L-1)$ internal lattice boundaries and $2M$ external ones. We assign a direction to the boundaries in each dimension with $i_k = 0$ to be the leftmost and $i_k = L$ the rightmost. As an example, figure 4-3 demonstrates the elements of the problem for $N = 1$ and $M = 2$.

After defining the necessary terms for the proof, in the next section, the proof proceeds as follows:

1. Section 4.4.2 proves that an N-blade intersecting a cell intersects at least N internal cell sides.
2. It is then shown that an N-blade intersecting one side of a lattice boundary intersects at most $I(M-1, N-1, L)$ cell sides.
3. By counting the number of boundary sides $s(M, L)$ in the lattice, an upper bound of $s(M, L)I(M-1, N-1, L)$ to the number of cell sides intersected follows.
4. Dividing this upper bound by the minimum number of cell sides needed to be intersected for a cell to be intersected—shown to be N in section 4.4.2—a recursive expression for the upper bound of $I(M, N, L)$ follows. Expanding that expression provides the upper bound in (4.4).

Steps 2-4 of the proof are presented in section 4.4.3. It should be noted that the counting in steps 1-3 of the proof does not depend on the boundaries of the hypercube lattice having integer coordinates, or even being equally spaced. Therefore, the proof does not rely on the quantizer being uniform—only on having L levels. Still, a uniform quantizer is assumed for clarity of exposition, and to simplify notation.

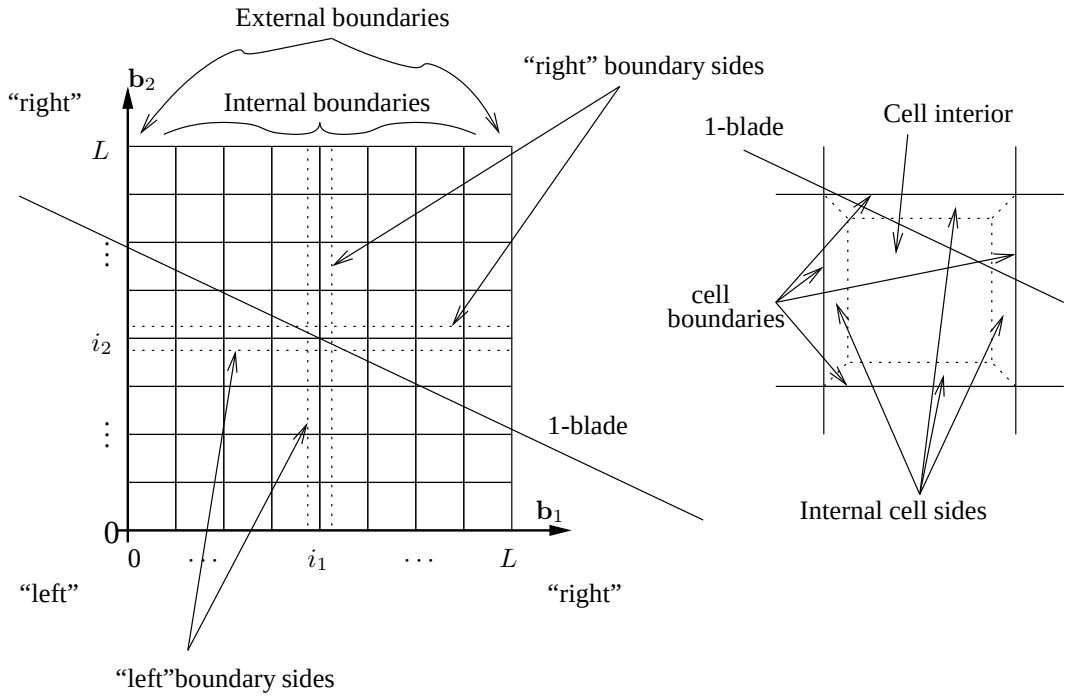


FIGURE 4-3: An example of the hypercube (square) lattice, lattice boundaries, cells, and arbitrary N-blade (line) formed in the $M = 2$, $N = 1$ case of the intersection problem.

4.4.1 Definitions

Each of the lattice boundaries has a left and a right side, defined respectively as the subsets with $i - \epsilon < x_k \leq i$ and $i \leq x_k < i + \epsilon$, for some small ϵ, in which the coordinate index k and the boundary index i determine the boundary in between the two sides. In counting the sides we are not interested in the sides that face outside the hypercube lattice, which is left side of the leftmost lattice boundary, and the right side of the rightmost lattice boundary. Thus, the total number of internal boundary sides s in the hypercube grid are:

$$s(M, L) = 2M + 2M(L - 1) = 2ML, \tag{4.20}$$

which counts one side for each of the $2M$ external lattice boundaries and two sides for each of the $M(L - 1)$ internal ones.

Each cell is identified from the coordinates of its rightmost boundaries in each dimension $(i_1, \ldots, i_M)$, $i_k \in \{1, \ldots, L\}$. The cell boundaries along the l^{th} dimension are defined, as the sets $LB_l(i_1, \ldots, i_M)$ and $RB_l(i_1, \ldots, i_M)$ for the left and the

right boundary, respectively:

$$LB_k(i_1,\ldots,i_M) = \left\{ (x_1,\ldots,x_M) \;\middle|\; \begin{array}{ll} x_k = i_k - 1, & \text{if } k = l \\ i_k - 1 < x_k < i_k, & \text{if } k \neq l \end{array} \right\}, \quad (4.21)$$

$$RB_k(i_1,\ldots,i_M) = \left\{ (x_1,\ldots,x_M) \;\middle|\; \begin{array}{ll} x_k = i_k, & \text{if } k = l \\ i_k - 1 < x_k < i_k, & \text{if } k \neq l \end{array} \right\}. \quad (4.22)$$

The inside facing sides of the cell along the l^{th} dimension are defined as the right facing side of the left cell boundary and the left facing side of the right cell boundary, restricted by the remaining cell boundaries. These are called the left internal cell side and the right internal cell side, and are denoted using $LS_k(\cdot)$ and $RS_k(\cdot)$, respectively:

$$LS_k(i_1,\ldots,i_M) = \left\{ (x_1,\ldots,x_M) \;\middle|\; \begin{array}{ll} i_k \leq x_k + 1 < i_k + \epsilon, & \text{if } k = l \\ i_k - 1 < x_k < i_k, & \text{if } k \neq l \end{array} \right\}, \quad (4.23)$$

$$RS_k(i_1,\ldots,i_M) = \left\{ (x_1,\ldots,x_M) \;\middle|\; \begin{array}{ll} i_k - \epsilon < x_k \leq i_k, & \text{if } k = l \\ i_k - 1 < x_k < i_k, & \text{if } k \neq l \end{array} \right\}, \quad (4.24)$$

for some small $\epsilon > 0$. Any cell in M dimensions has $2M$ internal sides. The interior of the cell $C(i_1,\ldots,i_M)$ is defined as the open set of points inside the cell boundaries, without the boundaries:

$$C(i_1,\ldots,i_M) = \{(x_1,\ldots,x_M) | i_k - 1 < x_k < i_k, \text{ for all } k\}. \quad (4.25)$$

The set of cell boundaries for all cells adjacent to a lattice boundary i_k form a hypercube lattice of dimension $M-1$ on that lattice boundary, which also has dimension $M-1$. Thus, each of the lattice boundaries has an $(M-1)$-dimensional structure, similar to the M-dimensional one defined here. The proof exploits this recursive structure of the hypercube lattice.

4.4.2 Intersection of a Single Cell with a Hyperplane

To prove the desired statement we first prove that a cell intersected by an N-blade has at least N of its internal cell sides and the corresponding boundaries intersected by the blade.[1] An N-blade intersects a cell if its intersection with the interior of the cell is non-zero.

Starting from a point $\mathbf{p}$ on the N-blade, interior to the cell, we determine a vector parallel to the blade and follow it along the blade until it intersects one of the cell boundaries. This implies that the blade intersects the corresponding internal side of the cell. In the same manner, we determine a second vector parallel to the blade and the boundary intersected by the previous step. Following this vector along the blade, starting from the same point $\mathbf{p}$, we intersect another boundary. This boundary is different from the boundary intersected before, since the vector followed is parallel to

[1] With a little more care, it can be demonstrated that at least $N+1$ internal sides and the corresponding boundaries are intersected, but N of them are enough for the purposes of the bound we present.

the first boundary, and the point $\mathbf{p}$ is interior to the cell. We repeat the process making sure that the vector determined in each iteration is parallel both to the blade and all the previously intersected boundaries. This ensures that every vector intersects a boundary and the corresponding internal cell side that has not been intersected before.

This is possible for at least N iterations because the intersection of an N-blade with the cell boundary forms an $(N-1)$-blade that is an affine subspace parallel both to the original N-blade and the boundary. Therefore, after the first iteration there are at least $N-1$ linearly independent vectors that are parallel to the intersected boundary and to the original blade. Using that argument recursively, it follows that after the k^{th} iteration there are $N-k$ linearly independent vectors to pick from, that are all parallel to the original blade and the boundaries that have already been intersected.

4.4.3 Intersection of Cells in the Hypercube Lattice

We denote using $I(M,N,L)$ the upper bound on the number of hypercube cells of an L^M sized hypercube lattice in M dimensions that an arbitrary N-blade intersects. The degenerate case in which a blade is a subset of one of the lattice boundaries is equivalent to the problem $I(M-1,N,L)$, and can be ignored. The intersection of any other arbitrary N-blade with a lattice boundary creates at most an $(N-1)$-blade within that boundary, which is also a sub-blade of the original N-blade. The cell boundaries form an $(M-1)$-dimensional lattice inside each lattice boundary. Therefore, the N-blade intersects at most $I(M-1,N-1,L)$ cell boundaries within each lattice boundary, and the corresponding left and right sides. The only exception is the leftmost and rightmost external lattice boundaries, which only have one side facing the inside of the lattice. In total there are $s(M,L)\cdot I(M-1,N-1,L)$ internal cell sides that are being intersected, in which $s(M,L)$ is the total number of sides, as defined in (4.20).

For each cell being intersected, there should be at least N unique internal cell sides intersected. A recursive upper bound follows:

$$
\begin{aligned}
I(M,N,L) &\leq \frac{s(M,L)\cdot I(M-1,N-1,L)}{N} && (4.26)\\
&= \frac{(2ML)I(M-1,N-1,L)}{N} && (4.27)\\
&\leq \frac{(2L)^N M(M-1)\cdots(M-N+1)I(M-N,0,L)}{N!} && (4.28)\\
&= \frac{(2L)^N M!I(M-N,0,L)}{N!(M-N)!} && (4.29)\\
&= (2L)^N \binom{M}{N} I(M-N,0,L). && (4.30)
\end{aligned}
$$

But $I(M-N,0,L)\leq 1$ since a 0-blade is a single point, which can only be interior

to (i.e. intersect) at most one cell. Thus, (4.4) follows:

$$I(M,N,L) \leq (2L)^N \binom{M}{N}. \tag{4.4}$$

This bound is loose, and is often larger than L^M,the number of total cells in the lattice. However, it is good enough for the purposes of this paper and the asymptotic bounds we proved in this chapter.

The bound can be made tighter using the fact that an N-blade intersects at least $N+1$ internal cell sides of any cell it intersects. Furthermore, it can be shown that:

$$I(M,1,L) \leq M(L-1)+1, \tag{4.31}$$

which can be used to terminate the recursion instead of $I(M,0,L)$—using the recursion on $I(M,1,L)$ results to $I(M,1,L) \leq 2ML$. Thus the upper bound becomes:

$$I(M,N,L) \leq \frac{2^N L^{N-1}(ML-M+1)}{N(N+1)} \binom{M}{N-1}. \tag{4.32}$$

Still, the tighter bound does not change the rates determined in section 4.3.

4.5 Efficiency of Frame Representations

Frame representations are not necessarily inefficient in terms of quantization. Indeed, there are examples that achieve the $O(L^{-r})$ reduction of error magnitude expected if the redundancy in the representation is used efficiently in terms of minimizing the quantization error [18]. Similarly, the next chapter, discusses how Sigma-Delta noise shaping can be generalized to arbitrary frame expansions to reduce the total quantization error.

The proof in the previous sections, demonstrates the limits of direct scalar quantization of frame expansions, not the limits of any other method of computing the quantized frame coefficients. The difference in the methods that achieve further efficiency is the assumption that the synthesis instead of the analysis is predetermined. Under this assumption the analysis is modified to a non-linear method that determines a better set of quantized expansion coefficients, such that the signal reconstruction has an improved error performance.

A side issue in that discussion is the measurement of quantization efficiency. In a classical Sigma-Delta A/D or D/A configuration the signal is 64 or 128 times oversampled using the classical oversampling frame and quantized with a Sigma-Delta converter to a 1-bit per sample representation. Compared to a critically sampled 16-bit signal, for example, the 1-bit, oversample representation uses 4 or 8 times more bits. It is, however, a very efficient representation if the cost of the representation is in the D/A or the A/D converter, not in the storage or transmission cost. Indeed, a 64 times oversampled 1-bit D/A converter is much cheaper than a critically sampled 16-bit D/A because the 1-bit converter can be implemented in hardware using a simple switch, whereas a 16-bit one requires a careful manufacturing process to ensure

linearity and other properties, even though it is running at a slower rate. This demonstrates that implementation cost is an important aspect when comparing quantization strategies. The next two chapters present a low-complexity approach to improve the error performance of direct coefficient quantization by generalizing Sigma-Delta quantization to arbitrary frame expansions.

CHAPTER 5

Quantization Noise Shaping on Finite Frame Representations

This chapter presents how quantization noise shaping can be viewed as a sequence of compensations using projections in the framework of chapter 3. The generalization of noise shaping to arbitrary finite frame expansions follows naturally. Different quantization algorithms are described, together with measures to evaluate them. The generalization to higher order quantization is also considered.

5.1 Introduction

Quantization methods for frame expansions have received considerable attention in the last few years. Simple scalar quantization applied independently on each frame expansion coefficient, followed by linear reconstruction is well known to be suboptimal [20, 17]. Several algorithms have been proposed that improve performance although with significant complexity either at the quantizer [28] or in the reconstruction method [28, 38]. The previous chapter proves that scalar quantization of the frame representation has fundamental performance limits, independent of the reconstruction method. To improve performance an improved quantization method is, therefore, necessary.

One such method, oversampled noise shaping, has been well studied and established for the oversampling frame [29, 13]. In [1] it is shown that noise shaping can be

considered as a causal example of error diffusion, a method often encountered in image halftoning in which error due to quantization of oversampled representations is diffused among multiple coefficients. More recently, frame quantization methods inspired by uniform oversampled noise shaping (referred to generically as Sigma-Delta noise shaping) have been proposed for finite uniform frames [4, 5] and for frames generated by oversampled filterbanks [9]. In [4, 5] the error due to the quantization of each expansion coefficient is subtracted from the next coefficient. The method is algorithmically similar to classical first order noise shaping and uses a quantity called frame variation to determine the optimal ordering of frame vectors such that the quantization error is reduced. In [9] higher order noise shaping is extended to oversampled filterbanks using a predictive approach. That solution performs higher order noise shaping, in which the error is filtered and subtracted from the subsequent frame coefficients.

This chapter formulates noise shaping as compensation of the error resulting from quantizing each frame expansion coefficient through a projection onto the space defined by another synthesis frame vector. This requires only knowledge of the synthesis frame set and a pre-specified ordering and pairing for the frame vectors. Instead of attempting a purely algorithmic generalization, we incorporate the use of projections and explore the issue of frame vector ordering. This method improves the average quantization error even if the frame vector ordering is not optimal. However, the benefits from determining the optimal ordering are also demonstrated. The theoretical framework presented provides a design method for noise shaping quantizers under the cost functions presented. This generalization of Sigma-Delta noise shaping improves the error in reconstruction due to quantization even for non-redundant frame expansions (i.e. a basis set) as long as the frame vectors are non-orthogonal. The results in this chapter have also appeared in [11, 12].

Section 5.2 describes classical first-order Sigma-Delta quantizers in the terminology of frames. Section 5.3 offers two generalizations, which we refer to as the sequential quantizer and the tree quantizer, both assuming a known ordering of the frame vectors. Section 5.4 explores two different cost models for evaluating the quantizer structures and determining the frame vector ordering. The first is based on a stochastic representation of the error and the second on deterministic upper bounds. In section 5.5 the optimal ordering of coefficients is considered, assuming the cost measures in section 5.4. It is shown that for finite frames the determination of frame vector ordering can be formulated in terms of known problems in graph theory and that for Sigma-Delta noise shaping the natural (time-sequential) ordering is optimal. Section 5.6 considers cases where the projection is restricted and how these cases relate to the work in [4, 5]. Furthermore, the natural extension to higher order quantization is examined. Section 5.7 presents experimental results on finite frames that verify and validate the theoretical ones. In section 5.8 the special case of quantization followed by complete compensation of the error is further analyzed.

5.2 Concepts and Background

This section establishes the notation and reformulates Sigma-Delta noise shaping using the terminology of frames and projections.

5.2.1 Frame Representation and Quantization

As described in chapter 2, we assume a vector $\mathbf{x}$ in a space $\mathcal{W}$ of finite dimension N represented using a predetermined finite frame:

$$\mathbf{x} = \sum_{k=1}^{M} a_k \mathbf{f}_k, \tag{5.1}$$

in which $\{\mathbf{f}_k, k = 1, \ldots, M\}$ is the synthesis frame. The redundancy of the frame is $r = M/N$. A frame is uniform if all the frame vectors have the same magnitude, i.e. $||\underline{\mathbf{f}}_k|| = ||\underline{\mathbf{f}}_l||$ for all k and l.

The coefficients a_k above are scalar, continuous quantities to be quantized. The simplest quantization strategy, which we call direct scalar quantization, is to quantize each one individually to $\hat{a}_k = Q(a_k) = a_k + e_k$, where $Q(\cdot)$ denotes the quantization function and e_k the quantization error for each coefficient. The total additive error vector from this strategy is equal to

$$\mathcal{E} = \sum_{k=1}^{M} e_k \mathbf{f}_k. \tag{5.2}$$

Section 4.1 demonstrates that if the frame forms an orthonormal basis, then direct scalar quantization is optimal in terms of minimizing the error magnitude. However, as discussed in [4, 5, 9, 13, 17, 20, 28, 38] and shown in section 4.3 this is not the case for all other frame expansions. Noise shaping is one of the possible strategies to reduce the error magnitude. In order to generalize noise shaping to arbitrary frame expansions, we first present traditional oversampling and noise shaping formulated in the context of projections.

5.2.2 Sigma-Delta Noise Shaping

Oversampling in time of bandlimited signals is a well studied class of frame expansions, presented in section 2.1.8. A signal $x[n]$ or $x(t)$ is upsampled or oversampled to produce a sequence a_k. In the terminology of frames, the upsampling operation is a frame expansion in which $\underline{\mathbf{f}}_k = r\mathbf{f}_k = \mathrm{sinc}((n-k)/r)$, with $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$. The sequence a_k is the corresponding ordered sequence of frame coefficients:

$$a_k = \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle = \sum_n x[n]\mathrm{sinc}((n-k)/r) \tag{5.3}$$

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k[n] = \sum_k a_k \frac{1}{r}\mathrm{sinc}((n-k)/r). \tag{5.4}$$

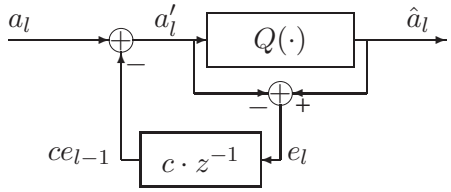


FIGURE 5-1: Traditional first order noise shaping quantizer

Similarly for oversampled continuous time signals:

$$a_k = \langle \mathbf{x}, \underline{\mathbf{f}}_k \rangle = \int_{-\infty}^{+\infty} x(t) \frac{r}{T} \text{sinc}(\frac{rt}{T} - k) \tag{5.5}$$

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k = \sum_k a_k \text{sinc}(\frac{rt}{T} - k), \tag{5.6}$$

where T is the Nyquist sampling period for $x(t)$.

Sigma-Delta quantizers can be represented in a number of equivalent forms [13]. The representation shown in figure 5-1 most directly represents the view that we extend to general frame expansions. Performance of Sigma-Delta quantizers is sometimes analyzed using the additive white noise model for the quantization error presented in section 2.3.3 [13]. Based on this model it can be shown that the quantization noise power at the reconstruction is minimized when the scaling coefficient c is chosen to be $c = \text{sinc}(1/r)$.[1]

The process in figure 5-1 can be viewed as an iterative process of coefficient quantization followed by error projection. The quantizer in the figure quantizes a'_l to $\hat{a}_l = a'_l + e_l$. Consider $x_l[n]$, such that the coefficients up to a_{l-1} have been quantized and e_{l-1} has already been scaled by c and subtracted from a_l to produce a'_l:

$$x_l[n] = \sum_{k=-\infty}^{l-1} \hat{a}_k \mathbf{f}_k[n] + a'_l \mathbf{f}_l[n] + \sum_{k=l+1}^{+\infty} a_k \mathbf{f}_k[n] \tag{5.7}$$

$$= x_{l+1}[n] + e_l(\mathbf{f}_l[n] - c \cdot \mathbf{f}_{l+1}[n]). \tag{5.8}$$

The incremental error $e_l(\mathbf{f}_l[n] - c \cdot \mathbf{f}_{l+1}[n])$ at the l^{th} iteration of (5.8) is minimized if we pick c such that $c \cdot \mathbf{f}_{l+1}[n]$ is the projection of $\mathbf{f}_l[n]$ onto $\mathbf{f}_{l+1}[n]$:

$$c = \langle \mathbf{f}_l[n], \mathbf{f}_{l+1}[n] \rangle / ||\mathbf{f}_{l+1}[n]||^2 = \text{sinc}(1/r). \tag{5.9}$$

This choice of c projects to $\mathbf{f}_{l+1}[n]$ the error due to quantizing a_l and compensates for this error by modifying a_{l+1}. Note that the optimal choice of c in (5.9) is the same

[1] With typical oversampling ratios, this coefficient is close to unity and is often chosen as unity for computational convenience.

as the optimal choice of c under the additive white noise model for quantization. It is also the solution of equation (3.18) in page 42 for the first order case (i.e. $p = 1$).

Minimizing the incremental error is not necessarily optimal in terms of minimizing the overall quantization error. It is, however, optimal in terms of the two cost functions described in section 5.4. Before we examine these cost functions we generalize first order noise shaping to general frame expansions.

5.3 Noise shaping on Frames

This section considers two generalizations of the discussion of section 5.2.2 to arbitrary finite frame representations of length M. Throughout the discussion in this section we assume the ordering of the synthesis frame vectors $(\mathbf{f}_1, \ldots, \mathbf{f}_M)$, and correspondingly the ordering of the synthesis coefficients $(a_1, \ldots, a_M)$ has already been determined.

The ordering of the frame vectors is addressed in section 5.5. However, it should be emphasized that the execution of the algorithm and the ordering of the frame vectors are distinct issues. The optimal ordering can be determined once, off-line, in the design phase. The ordering only depends on the properties of the synthesis frame, not the data or the analysis frame.

5.3.1 Single Coefficient Quantization

To illustrate our approach, we consider quantizing the first coefficient a_1 to $\hat{a}_1 = a_1 + e_1$, with e_1 denoting the additive quantization error. Equation (5.1) then becomes:

$$\mathbf{x} = \hat{a}_1\mathbf{f}_1 + \sum_{k=2}^{M} a_k\mathbf{f}_k - e_1\mathbf{f}_1 \tag{5.10}$$

$$= \hat{a}_1\mathbf{f}_1 + a_2\mathbf{f}_2 + \sum_{k=3}^{M} a_k\mathbf{f}_k - e_1c_{1,2}\mathbf{f}_2 - e_1(\mathbf{f}_1 - c_{1,2}\mathbf{f}_2). \tag{5.11}$$

As in (3.4) and (5.8), the norm of $e_1(\mathbf{f}_1 - c_{1,2}\mathbf{f}_2)$ is minimized if $c_{1,2}\mathbf{f}_2$ is the projection of $\mathbf{f}_1$ onto $\mathbf{f}_2$:

$$c_{1,2}\mathbf{f}_2 = \langle \mathbf{f}_1, \mathbf{u}_2\rangle \mathbf{u}_2 \tag{5.12}$$

$$= \langle \mathbf{f}_1, \frac{\mathbf{f}_2}{||\mathbf{f}_2||}\rangle \frac{\mathbf{f}_2}{||\mathbf{f}_2||} \tag{5.13}$$

$$\Rightarrow c_{1,2} = \frac{\langle \mathbf{f}_1, \mathbf{u}_2\rangle}{||\mathbf{f}_2||} = \frac{\langle \mathbf{f}_1, \mathbf{f}_2\rangle}{||\mathbf{f}_2||^2}, \tag{5.14}$$

where $\mathbf{u}_k = \mathbf{f}_k/||\mathbf{f}_k||$ are unit vectors in the direction of the synthesis vectors. Finally, we incorporate the term $-e_1c_{1,2}\mathbf{f}_2$ in the expansion by updating a_2 as in (3.9):

$$a_2' = a_2 - e_1c_{1,2}. \tag{5.15}$$

This is the same development as presented in chapter 3, assuming first order ($p = 1$) compensation. The only difference is in the notation, in which the set S_i and the space $\mathcal{W}_i$ are ignored here, since they are implied by first order compensation.

After the projection, the residual error is equal to $e_1(\mathbf{f}_1 - c_{1,2}\mathbf{f}_2)$. Consistent with chapter 3 we simplify this expression and define $\mathbf{r}_{1,2}$ to be the direction of the residual error, and $e_1\tilde{c}_{1,2}$ to be the error amplitude:

$$\mathbf{r}_{1,2} = (\mathbf{f}_1 - c_{1,2}\mathbf{f}_2)/||\mathbf{f}_1 - c_{1,2}\mathbf{f}_2|| \tag{5.16}$$

$$\tilde{c}_{1,2} = ||\mathbf{f}_1 - c_{1,2}\mathbf{f}_2|| = \langle \mathbf{f}_1, \mathbf{r}_{1,2} \rangle. \tag{5.17}$$

Thus, the residual error is $e_1\langle \mathbf{f}_1, \mathbf{r}_{1,2} \rangle \mathbf{r}_{1,2} = e_1\tilde{c}_{1,2}\mathbf{r}_{1,2}$, in which $\tilde{c}_{1,2}$ is the error coefficient for this pair of vectors.

Substituting the above, equation (5.11) becomes

$$\mathbf{x} = \hat{a}_1\mathbf{f}_1 + a'_2\mathbf{f}_2 + \sum_{k=3}^{M} a_k\mathbf{f}_k - e_1\tilde{c}_{1,2}\mathbf{r}_{1,2}. \tag{5.18}$$

Equation (5.18) can be viewed as decomposing $e_1\mathbf{f}_1$ into the direct sum $(e_1c_{1,2}\mathbf{f}_2) \oplus (e_1\tilde{c}_{1,2}\mathbf{r}_{1,2})$ and compensating only for the first term of this sum. The component $e_1\tilde{c}_{1,2}\mathbf{r}_{1,2}$ is the final quantization error after one step is completed.

Note that for any pair of frame vectors the corresponding error coefficient $\tilde{c}_{k,l}$ is always positive. Also, if the synthesis frame is uniform, there is a symmetry in the terms we defined: $c_{k,l} = c_{l,k}$ and $\tilde{c}_{k,l} = \tilde{c}_{l,k}$, for any pair $k \neq l$.

5.3.2 Sequential Noise Shaping Quantizer

A sequential first order noise shaping quantizer iterates the process in section 5.3.1 by quantizing the next (updated) coefficient until all the coefficients have been quantized. Specifically, the algorithm continues as follows:

1. Quantize coefficient k by setting $\hat{a}_k = Q(a'_k)$.

2. Compute the error $e_k = \hat{a}_k - a'_k$.

3. Update the next coefficient a_{k+1} to $a'_{k+1} = a_{k+1} - e_k c_{k,k+1}$, where

$$c_{k,l} = \frac{\langle \mathbf{f}_k, \mathbf{f}_l \rangle}{||\mathbf{f}_l||^2}. \tag{5.19}$$

4. Increase k and iterate from step 1 until all the coefficients have been quantized.

Every iteration of the sequential quantization contributes $e_k \tilde{c}_{k,k+1} \mathbf{r}_{k,k+1}$ to the total quantization error, where

$$\mathbf{r}_{k,l} = \frac{\mathbf{f}_k - c_{k,l}\mathbf{f}_l}{||\mathbf{f}_k - c_{k,l}\mathbf{f}_l||}, \text{ and} \tag{5.20}$$

$$\tilde{c}_{k,l} = ||\mathbf{f}_k - c_{k,l}\mathbf{f}_l||. \tag{5.21}$$

Since the frame expansion is finite, the algorithm cannot compensate for the quantization error of the last step $e_M \mathbf{f}_M$. Thus, the total error vector is

$$\mathcal{E} = \sum_{k=1}^{M-1} e_k \tilde{c}_{k,k+1} \mathbf{r}_{k,k+1} + e_M \mathbf{f}_M. \tag{5.22}$$

Note that the definition of $c_{k,l}$ in (5.19) is consistent with the solution of equation (3.18) for the case of $p = 1$. Also, $\tilde{c}_{k,l}\mathbf{r}_{k,l}$ is the residual from the projection of $\mathbf{f}_k$ onto $\mathbf{f}_l$, and has magnitude less than or equal to $\mathbf{f}_k$. Specifically, for all k and l:

$$\tilde{c}_{k,l} \leq ||\mathbf{f}_k||, \tag{5.23}$$

with equality holding if and only if $\mathbf{f}_k$ is orthogonal to $\mathbf{f}_l$. Furthermore note that $\tilde{c}_{k,l}$, being the magnitude of a vector, is always nonnegative.

5.3.3 Tree Noise Shaping Quantizer

The sequential quantizer can be generalized by relaxing the sequence of error assignments: Again, we assume that the coefficients have been pre-ordered and that the ordering defines the sequence in which coefficients are quantized. In this generalization, we associate with each ordered frame vector $\mathbf{f}_k$ another, possibly not adjacent, frame vector $\mathbf{f}_{l_k}$ further in the sequence (and, therefore, for which the corresponding coefficient has not yet been quantized) to which the error is projected using equation (5.15). With this more general approach some frame vectors can be used to compensate for more than one quantized coefficient.

A tree noise shaping quantizer uses the algorithm presented in section 5.3.2, with step 3 modified to:

3. Update a_{l_k} to $a'_{l_k} = a_{l_k} - e_k c_{k,l_k}$, where $c_{k,l} = \frac{\langle \mathbf{f}_k, \mathbf{f}_l \rangle}{||\mathbf{f}_l||^2}$, and $l_k > k$.

The constraint $l_k > k$ ensures that a_{l_k} is further in the sequence than a_k. For finite frames, this defines a tree, in which every node is a frame vector or associated coefficient. If a coefficient a_k uses coefficient a_{l_k} to compensate for the error, then a_k is a direct child of a_{l_k} in that tree. The root of the tree is the last coefficient to be quantized, a_M. The sequential quantizer is a special case of the tree quantizer in which $l_k = k + 1$.

The resulting expression for $\mathbf{x}$ is given by:

$$\mathbf{x} = \sum_{k=1}^{M} \hat{a}_k \mathbf{f}_k - \sum_{k=1}^{M-1} e_k \tilde{c}_{k,l_k} \mathbf{r}_{k,l_k} - e_M \mathbf{f}_M \tag{5.24}$$

$$= \hat{\mathbf{x}} - \sum_{k=1}^{M-1} e_k \tilde{c}_{k,l_k} \mathbf{r}_{k,l_k} - e_M ||\mathbf{f}_M|| \mathbf{u}_M, \tag{5.25}$$

where $\hat{\mathbf{x}}$ is the quantized version of $\mathbf{x}$ after noise shaping, and the e_k are the quantization errors in the coefficients after the corrections from the previous iterations have been applied to a_k. Thus, the total error of the process is:

$$\mathcal{E} = \sum_{k=1}^{M-1} e_k \tilde{c}_{k,l_k} \mathbf{r}_{k,l_k} + e_M \mathbf{f}_M. \tag{5.26}$$

5.4 Error Models and Analysis

In order to compare and design quantizers, we need to be able to compare the magnitude of the error in each. However, the error terms e_k in equations (5.2), (5.22), and (5.26) are data dependent in a very non-linear way. Furthermore, due to the error projection and propagation performed in noise shaping, the coefficients being quantized at every step are different for the different quantization strategies. Therefore, for each k, e_k is different among the equations (5.2), (5.22), and (5.26), making the precise analysis and comparison even harder. In order to compare quantizer designs we need to evaluate them using cost functions that are independent of the data.

To simplify the problem further, we focus on cost measures for which the incremental cost at each step is independent of the whole path and the data. We call these incremental cost functions. In this section we examine two such models, one stochastic and one deterministic. The first cost function is based on the white noise model for quantization, while the second provides a guaranteed upper bound for the error. Note that for the rest of this development we assume uniform quantization, with Δ denoting the interval spacing of the uniform quantizer. We also assume that the quantizer is properly scaled not to overflow.

5.4.1 Additive Noise Model

The first cost function assumes the additive uniform white noise model for quantization error, to determine the expected energy of the error $E\{||\mathcal{E}||^2\}$. All the error coefficients e_k are assumed white and identically distributed, with variance $\Delta^2/12$, where Δ is the interval spacing of the quantizer. They are also assumed to be uncorrelated with the quantized coefficients. Thus, all error components contribute

additively to the error power, resulting in:

$$E\{||\mathcal{E}||^2\} = \frac{\Delta^2}{12}\left(\sum_{k=1}^{M}||\mathbf{f}_k||^2\right), \tag{5.27}$$

$$E\{||\mathcal{E}||^2\} = \frac{\Delta^2}{12}\left(\sum_{k=1}^{M-1}\tilde{c}_{k,k+1}^2 + ||\mathbf{f}_M||^2\right), \text{ and} \tag{5.28}$$

$$E\{||\mathcal{E}||^2\} = \frac{\Delta^2}{12}\left(\sum_{k=1}^{M-1}\tilde{c}_{k,l_k}^2 + ||\mathbf{f}_M||^2\right), \tag{5.29}$$

for the direct, the sequential and the tree quantizer respectively.

This model, further described in section 2.3.3 is a generalization of the additive noise model sometimes used to characterize noise shaping on the oversampling frame. The model has been applied to other frame expansions [9, 28], although its assumptions are often inaccurate. This model only attempts to describe average behavior and provides no guarantees on performance for individual realizations. It is possible that quantizing a particular signal using noise shaping generates more error than using direct coefficient quantization.

5.4.2 Error Magnitude Upper Bound

As an alternative cost function, we can also consider an upper bound for the error magnitude. For any set of vectors $\mathbf{u}_i$, $||\sum_k \mathbf{u}_k|| \leq \sum_k ||\mathbf{u}_k||$, with equality only if all vectors are collinear, in the same direction. This leads to the following upper bound on the error:

$$||\mathcal{E}|| \leq \frac{\Delta}{2}\left(\sum_{k=1}^{M}||\mathbf{f}_k||\right), \tag{5.30}$$

$$||\mathcal{E}|| \leq \frac{\Delta}{2}\left(\sum_{k=1}^{M-1}\tilde{c}_{k,k+1} + ||\mathbf{f}_M||\right), \text{ and} \tag{5.31}$$

$$||\mathcal{E}|| \leq \frac{\Delta}{2}\left(\sum_{k=1}^{M-1}\tilde{c}_{k,l_k} + ||\mathbf{f}_M||\right), \tag{5.32}$$

for direct, sequential and tree quantization, respectively.

The vector $\mathbf{r}_{M-1,l_{M-1}}$ is by construction orthogonal to $\mathbf{f}_M$ and the $\mathbf{r}_{k,l_k}$ are never collinear, making the bound very loose. Thus, a noise shaping quantizer can be expected in general to perform better than what the bound suggests. Still, for the purposes of this discussion we treat this upper bound as a cost function and we design the quantizer such that this cost function is minimized.

5.4.3 Analysis of the Error Models

To compare the average performance of direct coefficient quantization to the proposed noise shaping schemes we only need to compare magnitude of the right hand side of equations (5.27) through (5.29), and (5.30) through (5.32) above. The cost of direct coefficient quantization computed using equations (5.27) and (5.30) does not change even if the order of quantization is different. Therefore, we can assume the ordering of the synthesis frame vectors and the associated coefficients is given, and compare the three strategies. In this section we prove that for any frame vector ordering, the proposed noise shaping strategies reduce both the average error power, and the worst case error magnitude, as described using the proposed functions, compared to direct scalar quantization.

When comparing the cost functions, the multiplicative terms $\frac{\Delta^2}{12}$ and $\frac{\Delta}{2}$ are eliminated because they are the same in all equations. Furthermore, the final additive term $||\mathbf{f}_M||^2$ and $||\mathbf{f}_M||$ does not affect the comparison since it exists in all equations. Therefore, it can also be eliminated. To summarize, we need to compare the following quantities:

$$\sum_{k=1}^{M-1} ||\mathbf{f}_k||^2, \quad \sum_{k=1}^{M-1} \tilde{c}_{k,k+1}^2, \text{ and } \sum_{k=1}^{M-1} \tilde{c}_{k,l_k}^2, \tag{5.33}$$

in terms of the average error power, and

$$\sum_{k=1}^{M-1} ||\mathbf{f}_k||, \quad \sum_{k=1}^{M-1} \tilde{c}_{k,k+1}, \text{ and } \sum_{k=1}^{M-1} \tilde{c}_{k,l_k}, \tag{5.34}$$

in terms of the guaranteed worst case performance. These correspond to direct coefficient quantization, sequential noise shaping, and tree noise shaping respectively.

Using (5.23) it follows that both noise shaping methods have lower cost than direct coefficient quantization for any frame vector ordering. Furthermore, we can always pick $l_k = k + 1$, and, therefore, the tree noise shaping quantizer can always achieve the cost of the sequential quantizer. Therefore, we can always find l_k such that the comparison above becomes:

$$\sum_{k=1}^{M-1} ||\mathbf{f}_k||^2 \geq \sum_{k=1}^{M-1} \tilde{c}_{k,k+1}^2 \geq \sum_{k=1}^{M-1} \tilde{c}_{k,l_k}^2, \text{ and} \tag{5.35}$$

$$\sum_{k=1}^{M-1} ||\mathbf{f}_k|| \geq \sum_{k=1}^{M-1} \tilde{c}_{k,k+1} \geq \sum_{k=1}^{M-1} \tilde{c}_{k,l_k}. \tag{5.36}$$

The relationships above hold with equality if and only if all the pairs $(\mathbf{f}_k, \mathbf{f}_{k+1})$ and $(\mathbf{f}_k, \mathbf{f}_{l_k})$ are orthogonal. Otherwise the comparison with direct coefficient quantization results in a strict inequality. In other words, noise shaping improves the quantization cost compared to direct coefficient quantization even if the frame is not re-

dundant, as long as the frame is not an orthogonal basis.[2] Note that the coefficients $c_{k,l}$ are 0 if the frame is an orthogonal basis. Therefore, the feedback terms $e_k c_{k,l_k}$ in step 3 of the algorithms described in section 5.3 are equal to 0. In this case, the strategies in section 5.3 reduce to direct coefficient quantization, which can be shown to be the optimal scalar quantization strategy for orthogonal basis expansions.

We can also determine a lower bound for the cost, independent of the frame vector ordering, by picking $j_k = \text{argmin}_{l_k \neq k} \tilde{c}_{k,l_k}$. This does not necessarily satisfy the constrain $j_k > k$ of section 5.3.3, therefore the lower bound cannot always be met. However, if a quantizer can meet the lower bound, it is the minimum cost first order noise shaping quantizer, independent of the frame vector ordering, for both cost functions.

The inequalities presented in this section are summarized below.

For given frame ordering, $j_k = \text{argmin}_{l_k \neq k} \tilde{c}_{k,l_k}$ and some $\{l_k > k\}$:

$$\sum_{k=1}^{M} \tilde{c}_{k,j_k} \leq \sum_{k=1}^{M-1} \tilde{c}_{k,l_k} + ||\mathbf{f}_M|| \leq \sum_{k=1}^{M-1} \tilde{c}_{k,k+1} + ||\mathbf{f}_M|| \leq \sum_{k=1}^{M} ||\mathbf{f}_k||, \tag{5.37}$$

and

$$\sum_{k=1}^{M} \tilde{c}_{k,j_k}^2 \leq \sum_{k=1}^{M-1} \tilde{c}_{k,l_k}^2 + ||\mathbf{f}_M||^2 \leq \sum_{k=1}^{M-1} \tilde{c}_{k,k+1}^2 + ||\mathbf{f}_M||^2 \leq \sum_{k=1}^{M} ||\mathbf{f}_k||^2, \tag{5.38}$$

where the lower and upper bounds are independent of the frame vector ordering.

In the development above we proved that the proposed noise shaping reduces the average and the upper bound of the quantization error for all frame expansions. The strategies above degenerate to direct coefficient quantization if the frame is an orthogonal basis. These results hold without any assumptions on the frame, or the ordering of the frame vectors and the corresponding coefficients. Finally, we derived a lower bound for the cost of a first order noise shaping quantizer. In the next section we examine how to determine the optimal ordering and pairing of the frame vectors.

5.5 First Order Quantizer Design

As indicated earlier, an essential issue in first order quantizer design based on the strategies outlined in this chapter is determining the ordering of the frame vectors. The optimal ordering depends on the specific set of synthesis frame vectors, but not on the specific signal. Consequently, the quantizer design (i.e. the frame vector

[2] An oblique basis can reduce the quantization error compared to an orthogonal one if noise shaping is used, assuming the quantizer uses the same Δ. However, more quantization levels might be necessary to ensure that the quantizer does not overflow if an oblique basis is used.

ordering) is carried out off-line and the quantizer implementation is a sequence of projections based on the ordering chosen for either the sequential or tree quantizer.

5.5.1 Simple Design Strategies

An obvious design strategy is to determine an ordering and pairing of the coefficients such that the quantization of every coefficient a_k is compensated as much as possible by the coefficient a_{l_k}. This can be achieved by setting $l_k = j_k$, with $j_k = \text{argmin}_{l_k \neq k} \tilde{c}_{k,l_k}$, as defined for the lower bounds of equations (5.37) and (5.38). When this strategy is possible to implement, i.e. $j_k > k$, it results in the optimal ordering and pairing under both cost models we discussed, simply because it meets the lower bound for the quantization cost.

This is exactly how a traditional Sigma-Delta quantizer works. When an expansion coefficient is quantized, the coefficients that can compensate for most of the error are the ones right before or right after it. This implies that the time sequential ordering of the oversampling frame vectors is the optimal ordering for first order noise shaping (another optimal ordering is the time-reversed, i.e. the anticausal version). We examine this further in section 6.1.1.

Unfortunately, for certain frames, this optimal pairing might not be feasible. Still, it suggests a heuristic for a good coefficient pairing: at every step k, the error from quantizing coefficient a_k is compensated using the coefficient a_{l_k} that can compensate for most of the error, picking from all the frame vectors whose corresponding coefficients have not yet been quantized. This is achieved by setting $l_k = \text{argmin}_{l>k} \tilde{c}_{k,l}$. This, in general is not an optimal strategy, but an easily implementable heuristic. Optimal designs are discussed next.

5.5.2 Quantization Graphs and Optimal Quantizers

From section 5.3.3 it is clear that a tree quantizer can be represented as a graph—specifically, a tree—in which all the nodes of the graph are coefficients to be quantized. Similarly for a sequential quantizer, which is a special case of the tree quantizer, the graph is a linear path passing through all the nodes a_k in the correct sequence. In both cases, the graphs have edges (k, l_k), pairing coefficient a_k to coefficient a_{l_k} if and only if the quantization of coefficient a_k assigns the error to the coefficient a_{l_k}.

Figure 5-2 shows four examples of graph representations of first order noise shaping quantizers on a frame with five frame vectors. The top two figures, (a) and (b), demonstrate two sequential quantizers ordering the frame vectors in their natural and their reverse order respectively. In addition, parts (c) and (d) of the figure demonstrate two general tree quantizers for the same frame.

In the figure a weight is assigned to each edge. The cost of each quantizer is proportional to the total weight of the graph with the addition of the cost of the final term. For a uniform frame the magnitude of the final term is the same, independent of which coefficient is quantized last. Therefore it is eliminated when comparing the

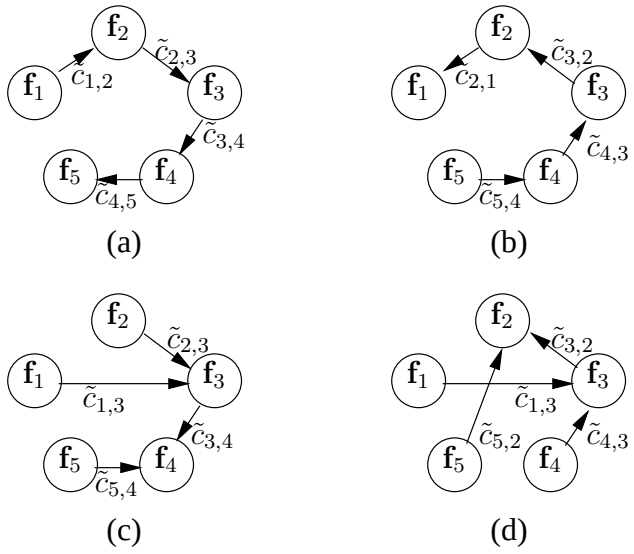


FIGURE 5-2: Examples of graph representations of first order noise shaping quantizers on a frame with five frame vectors. Note that the weights shown represent the upper bound of the quantization error. To represent the average error power the weights should be squared.

cost of quantizer designs on the same frame. Thus, designing the optimal quantizer corresponds to determining the graph with the minimum weight.

We define the quantization error assignment graph which has the frame vectors as nodes $V = \{\mathbf{f}_1, \ldots, \mathbf{f}_M\}$ and edges with weight $w(k,l) = \tilde{c}_{k,l}^2$ or $w(k,l) = \tilde{c}_{k,l}$ if we want to minimize the expected error power or the upper bound of the error magnitude respectively. On this graph, any acyclical path that visits all the nodes—a hamiltonian path—defines a first order sequential quantizer. Similarly, any tree that visits all the nodes—a spanning tree—defines a tree quantizer.

The minimum cost hamiltonian path defines the optimal sequential quantizer. This can be determined by solving the traveling salesman problem (TSP). The TSP is NP-complete in general, but has been extensively studied in the literature [15]. Similarly, the optimal tree quantizer is defined by the solution of the minimum spanning tree problem. This is also a well studied problem, solvable in polynomial time [15]. Since any path is also a tree, if the minimum spanning tree is a hamiltonian path, then it is also the solution to the traveling salesman problem. These results can be extended to non-uniform frames.

We should note that in general the optimal ordering and pairing depend on which of the two cost functions we choose to optimize for. Furthermore, we should reemphasize that this optimization is performed once, off-line, at the design stage of the

quantizer. Therefore, the computational cost of solving these problems does not affect the complexity of the quantizer.

5.6 Further Generalizations

In this section we consider two further generalizations. In section 5.6.1 we examine the case for which the product term is restricted. In section 5.6.2 we consider the case of noise shaping using more than one vector for compensation. Although a combination of the two is possible, we do not consider it here.

5.6.1 Projection Restrictions

The development in the previous sections uses the product $e_k c_{k,l_k}$ to compensate for the error in quantizing coefficient a_k using coefficient a_{l_k}. Implementation restrictions often do not allow for this product to be computed to a satisfactory precision. For example, typical Sigma-Delta converters eliminate this product altogether by setting $c = 1$. In such cases, the error compensation is not using a projection. Still, the intuition and approach remains applicable.

The restriction we consider is one on the product: the coefficients c_{k,l_k} are restricted to be in a discrete set $\mathcal{A} = \{\alpha_1, ..., \alpha_K\}$. Requiring the coefficient to be an integer power of 2 or to be only ± 1 are examples of such constraints. In this case we use again the algorithms of section 5.3, with $c_{k,l}$ now chosen to be the coefficient in $\mathcal{A}$ closest to achieving a projection, i.e. with $c_{k,l}$ specified as:

$$c_{k,l} = \text{argmin}_{c \in \mathcal{A}} ||\mathbf{f}_k - c\mathbf{f}_l|| \tag{5.39}$$

As in the unrestricted case, the residual error is $e_k(\mathbf{f}_k - c_{k,l}\mathbf{f}_l) = e_k \tilde{c}_{k,l} \mathbf{r}_{k,l}$ with $\mathbf{r}_{k,l}$ and $\tilde{c}_{k,l}$ defined as in equations (5.20) and (5.21), respectively.

To apply either of the error models in section 5.4 we use the new $\tilde{c}_{l,l_k}$, as computed above. However, in this case, certain coefficient orderings and pairings might increase the overall error. A pairing of $\mathbf{f}_k$ with $\mathbf{f}_{l_k}$ improves the cost if and only if

$$||\mathbf{f}_k - c_{k,l_k}\mathbf{f}_{l_k}|| \leq ||\mathbf{f}_k|| \Leftrightarrow \tilde{c}_{k,l_k} \leq ||\mathbf{f}_k||, \tag{5.40}$$

which is no longer guaranteed to hold. Thus, the strategies described in section 5.5.1 need one minor modification: we only allow the compensation to take place if the inequality (5.40) holds. Similarly, in terms of the graphical model of section 5.5.2, we only allow an edge in the graph if the inequality (5.40) holds. Still, the optimal sequential quantizer is the solution to the TSP, and the optimal tree quantizer is the solution to the minimum spanning tree problem on that graph—which might now have missing edges.

The main implication of missing edges is that, depending on the frame we operate on, the graph might have disconnected components. In this case we should solve the traveling salesman problem or the minimum spanning tree on every component. Also, it is possible that, although we are operating on an oversampled frame, noise

shaping is not beneficial due to the constraints. The simplest way to correct for this is to always allow the choice $c_{k,l_k} = 0$ in the set $\mathcal{A}$. This ensures that (5.40) is always met, and therefore the graph stays connected. Thus, whenever noise shaping is not beneficial, the algorithms will pick $c_{k,l_k} = 0$ as the compensation coefficient, which is equivalent to no noise shaping. We should note that the choice of the set $\mathcal{A}$ matters. The denser the set is, the better the approximation of the projection. Thus, the resulting cost is smaller.

An interesting special case is to set $\mathcal{A} = \{1\}$, so that no multiplications are required. As mentioned previously, this is a common design choice in traditional Sigma-Delta converters. Furthermore, it is the case examined in [4, 5], where the issue of the optimal permutation is addressed in terms of the frame variation. The frame variation is defined in [4] motivated by the triangle inequality, as is the upper bound model of section 5.4.2. In that work it is also shown that incorrect frame vector ordering might increase the overall error, compared to direct coefficient quantization.

In the case $\mathcal{A} = \{1\}$ the compensation is improving the cost if and only if $||\mathbf{f}_k - \mathbf{f}_{l_k}|| < ||\mathbf{f}_k||$. The rest of the development remains the same: determining the optimal frame vector ordering requires solving the traveling salesman problem or the minimum spanning tree problem on a possibly disconnected graph. In the example we present in section 5.7, the natural frame ordering becomes optimal using our cost models, yielding the same results as the frame variation criterion suggested in [4, 5]. In section 6.1.1 we show that when applied to classical first order noise shaping this restriction does not affect the optimal frame ordering and does not impact significantly the error power.

5.6.2 Higher Order Quantization

Classical Sigma-Delta noise shaping is commonly done in multiple stages to achieve higher-order noise shaping. Similarly noise shaping on arbitrary frame expansions can be generalized to higher order. Unfortunately, in this case determining the optimal ordering is not as straightforward, and we do not attempt this development. However, we develop the quantization strategy and the error modeling for a given ordering of the coefficients.

The goal of higher order noise shaping is to compensate for quantization of each coefficient using more than one coefficients. There are several possible implementations of a traditional higher order Sigma-Delta quantizers. All have a common property; the quantization error is in effect modified by a p^{th} order filter, typically with a transfer function of the form:

$$H_e(z) = (1 - z^{-1})^p \tag{5.41}$$

and equivalently an impulse response:

$$h_e[n] = \delta[n] - \sum_{i=1}^{p} c_i \delta[n - i], \tag{5.42}$$

for some c_i. Thus, every error coefficient e_k additively contributes a term of the form $e_k(\mathbf{f}_k - \sum_{i=1}^{p} c_i \mathbf{f}_{k+i})$ to the output error. In order to minimize the magnitude of this contribution we need to choose the c_i such that $\sum_{i=1}^{p} c_i \mathbf{f}_{k+i}$ is the projection of $\mathbf{f}_k$ to the space spanned by $\{\mathbf{f}_{k+1}, \ldots, \mathbf{f}_{k+p}\}$. Using (5.41) as the system function is often preferred for implementation simplicity but it is not the optimal choice. This design choice is similar to eliminating the product in figure 5-1. As with first order noise shaping, it is straightforward to generalize this to arbitrary frames.

Given a frame vector ordering, we consider the quantization of coefficient a_k to $\hat{a}_k = a_k + e_k$. This error is to be compensated using coefficients a_{l_1} to a_{l_p}, with all the $l_i > k$. Thus, we desire to project the vector $-e_k \mathbf{f}_k$ to the space $\mathcal{W}_k$, defined by the vectors $\mathbf{f}_{l_1}, \ldots, \mathbf{f}_{l_p}$, as described in chapter 3. We use the analysis in that chapter to determine a set of coefficients that multiply the error e_k in order to project it to the appropriate space.

To perform the projection we view the set $\{\mathbf{f}_l | l \in S_k\}$ as the reconstruction frame for $\mathcal{W}_k$, where $S_k = \{l_1, \ldots, l_p\}$ is the set of the indices of all the vectors that we use for compensation of coefficient a_k. Ensuring that for all $j \geq k,\ k \notin S_j$ guarantees that once a coefficient is quantized, it is not modified again.

We use $c_{k,l}$ to denote the coefficients that perform the projection—the corresponding set S_k and the space $\mathcal{W}_k$ are implied and not included in the notation. These coefficients perform a projection if they satisfy equation (3.18), which becomes:

$$\begin{bmatrix} \langle \mathbf{f}_{l_1}, \mathbf{f}_{l_1} \rangle & \langle \mathbf{f}_{l_1}, \mathbf{f}_{l_2} \rangle & \cdots & \langle \mathbf{f}_{l_1}, \mathbf{f}_{l_p} \rangle \\ \langle \mathbf{f}_{l_2}, \mathbf{f}_{l_1} \rangle & \langle \mathbf{f}_{l_2}, \mathbf{f}_{l_p} \rangle & \cdots & \langle \mathbf{f}_{l_1}, \mathbf{f}_{l_p} \rangle \\ \vdots & & \ddots & \vdots \\ \langle \mathbf{f}_{l_p}, \mathbf{f}_{l_1} \rangle & \langle \mathbf{f}_{l_p}, \mathbf{f}_{l_2} \rangle & \cdots & \langle \mathbf{f}_{l_p}, \mathbf{f}_{l_p} \rangle \end{bmatrix} \begin{bmatrix} c_{k,l_1} \\ c_{k,l_2} \\ \vdots \\ c_{k,l_p} \end{bmatrix} = \begin{bmatrix} \langle \mathbf{f}_{l_1}, \mathbf{f}_k \rangle \\ \langle \mathbf{f}_{l_2}, \mathbf{f}_k \rangle \\ \vdots \\ \langle \mathbf{f}_{l_p}, \mathbf{f}_k \rangle \end{bmatrix}. \tag{5.43}$$

If the frame $\{\mathbf{f}_l | l \in S_k\}$ is redundant, the coefficients are not unique, but any solution is appropriate. The projection is equal to:

$$\mathcal{P}_{\mathcal{W}_k}(-e_k \mathbf{f}_k) = -e_k \sum_{l \in S_k} c_{k,l} \mathbf{f}_l. \tag{5.44}$$

Consistent with section 5.3, we change step 3 of the algorithm to:

3. Update $\{a_l | l \in S_k\}$ to $a'_l = a_l - e_k c_{k,l}$, where $c_{k,l}$ satisfy (5.43).

Similarly, the residual is $-e_k \tilde{c}_k \mathbf{r}_k$, where

$$\tilde{c}_k = ||\mathbf{f}_k - \sum_{l \in S_k} c_{k,l} \mathbf{f}_l||, \text{ and} \tag{5.45}$$

$$\mathbf{r}_k = \frac{\mathbf{f}_k - \sum_{l \in S_k} c_{k,l} \mathbf{f}_l}{||\mathbf{f}_k - \sum_{l \in S_k} c_{k,l} \mathbf{f}_l||}, \tag{5.46}$$

consistent with (3.10) and (3.11) in page 41 respectively. In other words, we express $e_k \mathbf{f}_k$ as the direct sum of the vectors $e_k \tilde{c}_k \mathbf{r}_k \oplus e_k \sum_{l \in S_k} c_{k,l,S} \mathbf{f}_l$, and compensate only

for the second part of this sum. Note that $\tilde{c}_k$ and $\mathbf{r}_k$ are the same independent on what method is used to solve equation (5.43).

The modification to the equations for the total error and the corresponding cost functions are straightforward:

$$\mathcal{E} = \sum_{k=1}^{M} e_k \tilde{c}_k \mathbf{r}_k \tag{5.47}$$

$$E\{||\mathcal{E}||^2\} = \frac{\Delta^2}{12} \sum_{k=1}^{M} \tilde{c}_k^2, \text{ and} \tag{5.48}$$

$$||\mathcal{E}|| \leq \frac{\Delta}{2} \sum_{k=1}^{M} \tilde{c}_k. \tag{5.49}$$

When $S_k = \{l_k\}$ for $k < M$, this collapses to a tree quantizer. Similarly, when $S_k = \{k+1\}$, the structure becomes a sequential quantizer. Since the tree quantizer is a special case of the higher order quantizer, it is easy to show that for a given frame vector ordering a higher order quantizer can always achieve the cost of a tree quantizer. Note that S_M is always empty, and therefore $\tilde{c}_M = ||\mathbf{f}_M||$, which is consistent with the cost analysis for the first order quantizers.

For appropriately ordered finite frames in N dimensions, the first $M - N$ error coefficients $\tilde{c}_k$ can be forced to zero with an N^{th} or higher order quantizer. In this case, the error coefficients determining the cost of the quantizer are the remaining N ones—the error becomes $\sum_{k=M-N+1}^{M} e_k \tilde{c}_k \mathbf{r}_k$, with the corresponding cost functions modified accordingly. One way to achieve that function is to use all the unquantized coefficients to compensate for the quantization of coefficient a_k by setting $S_k = \{(k+1), \ldots, M\}$ and ordering the vectors such that the last N frame vectors span the space. This is not the only option; another example is discussed in the next section.

Unfortunately, the design space for higher order quantizers is quite large. The optimal frame vector ordering and S_k selection is still an open question and we do not attempt it in this work.

5.7 Experimental Results

To validate the theoretical results we presented above, in this section we consider the same example as was included in [4, 5]. We use the tight frame consisting of the 7^{th} roots of unity to expand randomly selected vectors in $\mathbb{R}^2$, uniformly distributed inside the unit circle. We quantize the frame expansion using $\Delta = 1/4$, and reconstruct the vectors using the corresponding synthesis frame. The frame vectors and

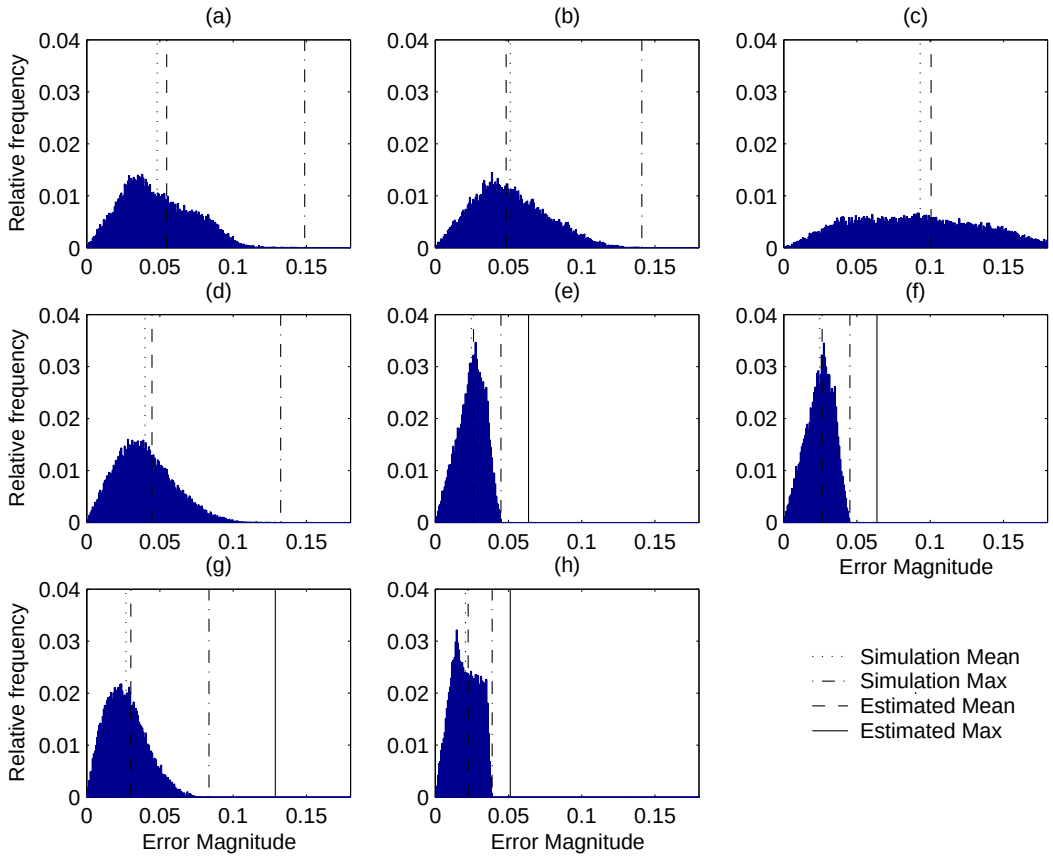


FIGURE 5-3: Histogram of the reconstruction error under (a) direct coefficient quantization, (b) natural ordering and error propagation without projections, (c) skip-two-vectors ordering and error propagation without projections. In the second row, natural ordering using projections, with (d) first, (e) second, and (f) third order error propagation. In the third row, skip-two-vectors ordering using projections, with (g) first and (h) second order error propagation (the third order results are similar to the second order ones but are not displayed for clarity of the legend).

the coefficients relevant to quantization are given by:

$$\underline{\mathbf{f}}_n = (\cos(2\pi n/7), \sin(2\pi n/7)), \tag{5.50}$$

$$\mathbf{f}_n = ((2/7)\cos(2\pi n/7), (2/7)\sin(2\pi n/7)), \tag{5.51}$$

$$c_{k,l} = \cos\left(2\pi(k-l)/7\right), \tag{5.52}$$

$$\tilde{c}_{k,l} = (2/7)|\sin\left(2\pi(k-l)/7\right)|. \tag{5.53}$$

For this frame the natural ordering is suboptimal given the criteria we propose. An

optimal ordering of the frame vectors is $(\mathbf{f}_1, \mathbf{f}_4, \mathbf{f}_7, \mathbf{f}_3, \mathbf{f}_6, \mathbf{f}_2, \mathbf{f}_5)$, and we refer to it as the skip-two-vectors ordering for the remainder of this section. A sequential quantizer with this optimal ordering meets the lower bound for the cost under both cost functions we propose. Thus, it is an optimal first order noise shaping quantizer for both cost functions. We compare this strategy to the one proposed in [4, 5] and also explored as a special case of section 5.6.1. Under that strategy, there is no projection performed, just error propagation. Therefore, based on the frame variation as described in [4, 5], the natural frame ordering is the best ordering to implement that strategy.

The simulations also examine the performance of higher order quantization, as described in section 5.6.2. Since the frame is two dimensional, a second order quantizer can perfectly compensate for the quantization of all but the last two expansion coefficients. Therefore, all the error coefficients of equation (5.47) are 0, except for the last two. A third order or higher quantizer will not improve the quantization cost. However, the ordering of frame vectors is still important, since the angle between the last two frame vectors to be quantized affects the total error, and should be as small as possible.

To visualize the results we plot the distribution of the reconstruction error magnitude. In figure 5-3(a) we consider the case of direct coefficient quantization. Figures 5-3(b) and (c) correspond to noise shaping using the natural and the skip-two-vectors ordering respectively, and the method proposed in [4, 5], i.e. without projecting the error. Figures 5-3(d), (e), and (f) use the projection method using the natural frame ordering, and first, second and third order projections, respectively. Finally, figures 5-3(g) and (h) demonstrate first and second noise shaping results, respectively, using projections on the skip-two-vectors ordering. For clarity of the legend we do not plot the third order results, although they are almost identical to the second order case. On all the plots dotted and dash-dotted lines indicate the average and maximum reconstruction error respectively. Dashed and solid lines are used to indicate the average and maximum error, as determined using the cost functions of section 5.4.[3]

The results show that the projection method results in smaller error, even when using the natural frame ordering. As expected, the results using the optimal frame vector ordering are the best among the simulations we performed. The simulations also confirm that in $\mathbb{R}^2$, noise shaping provides no benefit beyond second order and that the frame vector ordering affects the error even in higher order noise shaping, as predicted by the analysis. It is evident that the upper bound model is loose, as expected. The residual error vectors $\mathbf{r}_{i,j}$ are not collinear, and therefore the triangle inequality, on which the upper bound model is based, provides a very conservative bound. The error average, on the other hand, is surprisingly close to the simulation mean, although it usually overestimates it.

The results were similar for a variety of frame expansions on different dimensions, re-

[3] In some parts of the figure, the lines are out of the axis bounds. For completeness, we list the results here: (a) Estimated Max=0.25, (b) Estimated Max=0.22, (c) Estimated Max=0.45, Simulation Max=0.27, (d) Estimated Max=0.20.

dundancy values, vector orderings, and noise shaping orders, including non-orthogonal bases, validating the theory developed in the previous sections.

5.8 Noise Shaping with Complete Compensation

As described in section 5.6.2, when quantizing a finite frame, it is possible to force the error coefficients $\tilde{c}_k$ to zero for the first $M - N$ coefficients to be quantized. This can be done, for example, by ordering the frame vectors such that the last N vectors form a linearly independent set, and compensating for the error from quantizing coefficient a_k using all the subsequent coefficients $\{a_{k+1}, \ldots, a_M\}$. Even in this case, the ordering of the frame vectors affects the quantization error. The last N coefficients to be quantized correspond to linearly independent vectors, which can be chosen and ordered such that they are as aligned as much as possible and the corresponding error coefficients become as small as possible.

In this case it is possible to exploit the orthogonality properties of the residual vectors $\mathbf{r}_i$ in order to obtain a tighter expression on the upper bound on the residual error due to the use of projections. Using Gram-Schmidt orthogonalization it is also possible and computationally efficient to compute these vectors. In the subsequent development we assume that the last N coefficients are quantized in sequence $a_{M-N+k}, k = 1, \ldots, N$ and the error due to the quantization of a_{M-N+k} is compensated using all the remaining unquantized coefficients $\{a_{M-N+k+1}, \ldots, a_M\}$. The error due to the quantization of coefficients $\{a_1, \ldots, a_{M-N}\}$ is zero since the quantization of these coefficient can be perfectly compensated for. In denoting the relevant vectors and coefficients, we eliminate $S_{M-N+k} = \{M - N + k + 1, \ldots, M\}$ and $\mathcal{W}_{M-N+k} = \text{span}\{\mathbf{f}_l, l \in S_{M-N+k}\}$ from the subscripts since they are not ambiguous and they make the notation cumbersome.

5.8.1 Error Upper Bound

As noted in section 3.1, the residual vector, $\mathbf{r}_k$ is orthogonal to all the synthesis vectors $\{\mathbf{f}_l | l \in S_k\}$ used for the compensation of the error. Combined with the compensation ordering described above, this implies that:

$$\langle \mathbf{r}_k, \mathbf{f}_l \rangle = 0, \text{ for all } l > k > M - N. \tag{5.54}$$

But $\mathbf{r}_l$ is a linear combination of all the synthesis vectors $\mathbf{f}_i$ for $l \leq i \leq M$:

$$\mathbf{r}_l = \frac{\mathbf{f}_l - \sum_{l<i\leq M} c_{l,i}\mathbf{f}_i}{||\mathbf{f}_l - \sum_{l<i\leq M} c_{l,i}\mathbf{f}_i||}, \text{ for all } l > M - N. \tag{5.55}$$

Therefore, the last N residual vectors are orthogonal to each other:

$$\langle \mathbf{r}_k, \mathbf{r}_l \rangle = \langle \mathbf{r}_k, \frac{\mathbf{f}_l - \sum_{l<i\leq M} c_{l,i}\mathbf{f}_i}{||\mathbf{f}_l - \sum_{l<i\leq M} c_{l,i}\mathbf{f}_i||} \rangle \tag{5.56}$$

$$= \frac{\langle \mathbf{r}_k, \mathbf{f}_l \rangle - \sum_{l<i\leq M} \langle \mathbf{r}_k, c_{l,i}\mathbf{f}_i \rangle}{||\mathbf{f}_l - \sum_{l<i\leq M} c_{l,i}\mathbf{f}_i||} \tag{5.57}$$

$$= 0, \text{ for all } l > k > M - N. \tag{5.58}$$

The corresponding error becomes:

$$\mathcal{E} = \sum_{k=M-N+1}^{M} e_k \tilde{c}_k \mathbf{r}_k, \tag{5.59}$$

in which the $\mathbf{r}_k$ vectors are orthonormal and form a basis. Thus, the error energy follows from Parseval's equality:

$$||\mathcal{E}||^2 = \sum_{k=M-N+1}^{M} e_k^2 \tilde{c}_k^2 \tag{5.60}$$

$$\leq \frac{\Delta^2}{4} \sum_{k=M-N+1}^{M} \tilde{c}_k^2. \tag{5.61}$$

This is a tighter upper bound than the one described in section 5.4.2. It also proportional (with a proportionality constant $1/3$, independent of the frame) with the expected error power, as derived by the additive noise model in section 5.4.2. This provides further justification in the use of the additive noise model for the design of this system, not because it validates the noise model assumptions but because it provides the same results.

5.8.2 Determination of the Residual Vectors

In the case of complete compensation the residual vectors for the last N quantizations and the corresponding error coefficients can be efficiently computed using the Gram-Schmidt orthogonalization procedure [2] on the sequence $\{\mathbf{f}_M, \ldots, \mathbf{f}_{M-N+1}\}$ of the last N frame vectors reversed.

Starting from a set $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, Gram-Schmidt orthogonalization produces a sequence of orthonormal vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ using:

$$\mathbf{u}_k = \frac{\mathbf{y}_k - \sum_{j=1}^{k-1} \langle \mathbf{y}_k, \mathbf{u}_j \rangle \mathbf{u}_j}{\left|\left| \mathbf{y}_k - \sum_{j=1}^{k-1} \langle \mathbf{y}_k, \mathbf{u}_j \rangle \mathbf{u}_j \right|\right|}. \tag{5.62}$$

The algorithm guarantees that for any $k \leq N$ the vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ are orthonormal and have the same span as the vectors $\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$. Therefore, at step k, the sum $\sum_{j=1}^{k-1} \langle \mathbf{y}_k, \mathbf{u}_j \rangle \mathbf{u}_j$ is the projection of $\mathbf{y}_k$ onto the space spanned by $\{\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}\}$.

By setting $\mathbf{y}_k = \mathbf{f}_{M-k+1}$, and $\mathcal{W}_{M-k+1} = \text{span}\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$, it follows that the orthogonal vectors $\mathbf{u}_k$ produced by Gram-Schmidt satisfy (3.11) with the indices appropriately modified:

$$\mathbf{u}_k = \frac{\mathbf{f}_{M-k+1} - \mathcal{P}_{\mathcal{W}_{M-k+1}}(\mathbf{f}_{M-k+1})}{||\mathbf{f}_{M-k+1} - \mathcal{P}_{\mathcal{W}_{M-k+1}}(\mathbf{f}_{M-k+1})||} \tag{5.63}$$

$$= \mathbf{r}_{M-k+1}, \tag{5.64}$$

with the error, coefficients $\tilde{c}_{M-k+1}$ being the inverses of the normalization factors:

$$\tilde{c}_{M-k+1} = ||\mathbf{f}_{M-k+1} - \mathcal{P}_{\mathcal{W}_{M-k+1}}(\mathbf{f}_{M-k+1})|| \tag{5.65}$$

$$= \left|\left| \mathbf{y}_k - \sum_{j=1}^{k-1} \langle \mathbf{y}_k, \mathbf{u}_j \rangle \mathbf{u}_j \right|\right|. \tag{5.66}$$

Thus, Gram-Schmidt orthogonalization on the set $\{\mathbf{f}_M, \ldots, \mathbf{f}_{M-N+1}\}$ generates an orthonormal basis which is equal to the last N residual vectors, $\{\mathbf{r}_M, \ldots, \mathbf{r}_{M-N+1}\}$, of the compensation. The corresponding error coefficients are produced as a byproduct of the algorithm.

5.8.3 Noise Shaping on Finite Shift Invariant Frames

For a shift invariant frame, the equation to determine the compensation coefficients is simplified to (3.21). In this case, the solution can be efficiently computed using the Levinson-Durbin recursion [32, 25]. Compared to a general-purpose matrix inversion algorithm the use of the Levinson recursion has two advantages:

(a) The Levinson recursion has computational complexity $O(p^2)$ compared to the $O(p^3)$ complexity of general-purpose matrix inversion.

(b) The Levinson recursion is recursive in the matrix order. Therefore, it provides the solution to all intermediate problems without the need to perform separate matrix inversions. These intermediate solutions are needed to implement the projection of the error onto the remaining coefficients as the number of coefficient remaining unquantized decreases.

Specifically, the projection coefficients necessary to project the error due to quantizing the k^{th} coefficient onto the remaining $M-k$ frame vectors, are determined by solving the system of equation (3.21), with $p = M - k$. In order to project the error of each coefficient to all the remaining ones, in sequence, it is, therefore, necessary to solve equation (3.21) for $p = (M-1), \ldots, 1$. The Levinson recursion starts from the simple case of $p = 1$ and produces the compensation coefficients necessary to implement all the intermediate projections up to $p = M$ with overall computational complexity $O(M^2)$.

It should be noted that for certain shift invariant frames such as the harmonic frames [28, 43, 27] any subset of N coefficients span $\mathcal{W}$. Therefore, complete compensation for each quantization is possible using only the N coefficients subsequent to the

quantized coefficient. In this case, the solution to the $p = N$ problem can be used to fully compensate for the error due to the quantization of the first $M - N$ coefficients. Thus the overall complexity in this case is further reduced to $O(N^2)$.

For comparison, a general matrix inversion algorithm, applied independently to each of the $p = 1, \ldots, M$ problems requires $O(1^3 + 2^3 + \ldots + (M-1)^3) = O(M^4)$ computation. Although this computation is performed once at the design stage of the quantizer, the gains are significant, especially for large problems.

CHAPTER 6

Noise Shaping for Infinite Frame Representations

This chapter discusses the extension of Sigma-Delta noise shaping to arbitrary infinite frame expansions. Further emphasis is given on frames generated by LTI filters and filterbanks. In addition, two modifications to classical Sigma-Delta noise shaping are considered. In the first, the complexity of digital to analog conversion is reduced by eliminating the interpolation filter. In the second, the converter is tunable, depending on the needs of the particular application.

6.1 Extensions to Infinite Frames

When extending the results of chapter 5 to frames with countably infinite number of synthesis frame vectors, we let $M \to \infty$ and modify equations (5.22), (5.28), and (5.31) to reflect an error rate corresponding to average error per frame vector, or equivalently per expansion coefficient. As $M \to \infty$ the effect of the last term on the error rate tends to zero. Consequently, in considering the error rate, we replace

equations (5.22), (5.28), and (5.31) by

$$\overline{\mathcal{E}} = \lim_{M\to\infty} \frac{1}{M} \sum_{k=0}^{M-1} e_k \tilde{c}_{k,k+1} \mathbf{r}_{k,k+1}, \tag{6.1}$$

$$\overline{E\left\{||\mathcal{E}||^2\right\}} = \lim_{M\to\infty} \frac{1}{M} \frac{\Delta^2}{12} \left(\sum_{k=0}^{M-1} \tilde{c}_{k,k+1}^2 \right), \text{ and} \tag{6.2}$$

$$\overline{||\mathcal{E}||} \leq \lim_{M\to\infty} \frac{1}{M} \frac{\Delta}{2} \left(\sum_{k=0}^{M-1} \tilde{c}_{k,k+1} \right), \tag{6.3}$$

respectively, where $\overline{(\cdot)}$ denotes rate, and the frame vectors are indexed in $\mathbb{N}$. Similar modifications are straightforward for the cases of tree[1] and higher order quantizers, and for any countably infinite indexing of the frame vectors. At the design stage, the choice of frame should be such as to ensure convergence of the cost functions. In the remainder of this section we expand further on shift invariant frames, for which convergence of the cost functions is straightforward to demonstrate.

6.1.1 Infinite Shift Invariant Frames

As described in chapter 2, infinite shift-invariant reconstruction frames are infinite frames $\mathbf{f}_k$ for which the frame autocorrelation $R_{k,l} = \langle \mathbf{f}_k, \mathbf{f}_l \rangle$ is a function only of the index difference $m = k - l$: $R_m = \langle \mathbf{f}_k, \mathbf{f}_{k+m} \rangle$. Shift invariance implies that the reconstruction frame is uniform, with $||\mathbf{f}_k||^2 = \langle \mathbf{f}_k, \mathbf{f}_k \rangle = R_0$.

An example of such a frame is an LTI system: consider a signal $x[n]$ that is quantized to $\hat{x}[n]$ and filtered to produce $\hat{y}[n] = \sum_k \hat{x}[k]h[n-k]$. We consider the coefficients $x[k]$ to be the coefficients in a frame representation of $y[n]$, in which $h[n-k]$ are the reconstruction frame vectors $\mathbf{f}_k$. We rewrite the convolution equation as:

$$y[n] = \sum_k x[k]h[n-k] = \sum_k x[k]\mathbf{f}_k, \tag{6.4}$$

where $\mathbf{f}_k = h[n-k]$. Equivalently, we may consider $x[n]$ to be quantized, converted to continuous time impulses, and then filtered to produce $\hat{y}(t) = \sum_k \hat{x}[k]h(t - kT)$. We desire to minimize the quantization error after filtering, compared to the signals $y[n] = \sum_k x[k]h[n-k]$ and $y(t) = \sum_k x[k]h(t-kT)$, assuming the cost functions described. A filter forms a frame under the conditions discussed in detail in section 2.1.6.

For the remainder of this section we only discuss the discrete-time version of the problem since the continuous time development is identical. The corresponding frame autocorrelation functions are $R_m = R_{hh}[m] = \sum_m h[n]h[n-m]$ in the discrete-time case and $R_m = R_{hh}(mT) = \int h(t)h(t-mT)dt$ in the continuous-time case. A special case is the oversampling frame, in which $h(t)$ or $h[n]$ is the

[1] This is a slight abuse of the term, since the resulting infinite graph might have no root.

ideal lowpass filter used for the reconstruction, and $R_m = \mathrm{sinc}(m/r)$, where r is the oversampling ratio.

6.1.2 First Order Noise Shaping

Given a shift invariant frame, it is straightforward to determine the coefficients $c_{k,l}$ and $\tilde{c}_{k,l}$ that are important for the design of a first order quantizer. These coefficients are also shift invariant, so we denote them using $c_m = c_{k,k+m}$ and $\tilde{c}_m = \tilde{c}_{k,k+m}$. Combining equations (5.19) and (5.21) from section 5.3 and the definition of R_m above, we compute the relevant coefficients:

$$c_m = c_{-m} = \frac{R_m}{R_0} \tag{6.5}$$

$$\tilde{c}_m = \tilde{c}_{-m} = \sqrt{R_0(1 - c_m^2)} \tag{6.6}$$

For every coefficient a_k of the frame expansion and corresponding frame vector $\mathbf{f}_k$, the vector that minimizes the projection error is the vector $\mathbf{f}_{k\pm m_o}$, in which $m_o > 0$ minimizes $\tilde{c}_m$, or, equivalently, maximizes $|c_m|$, i.e. $|R_m|$. By symmetry, for any such m_o, $-m_o$ is also a minimum. Due to the shift invariance of the frame, m_o is the same for all frame vectors. Projecting to $\mathbf{f}_{k+m_o}$ or $\mathbf{f}_{k-m_o}$ generates a path with no loops, and therefore the optimal tree quantizer path, as long as the direction is consistent for all the coefficients. When $m_o = 1$, the optimal tree quantizer is also an optimal sequential quantizer. The optimality holds under both the additive noise model and the error upper bound model.

In the case of filtering, the noise shaping implementation is shown in figure 6-1, with $H_f(z) = c_{m_o} z^{-m_o}$. For the special case of the oversampling frame, $c_m = \mathrm{sinc}(m/r)$, and $m_o = 1$. Thus, the time sequential ordering of the frame vectors is optimal for the given frame.

6.1.3 Higher Order Noise Shaping

As we discuss in section 5.6.2, determining the optimal ordering for higher order quantization is not straightforward. Therefore, in this section we consider higher order noise shaping for the natural frame ordering, assuming that when a_k is quantized, the next p coefficients, $a_{k+1}, \ldots, a_{k+p}$, are used for compensation by updating them to

$$a'_{k+l} = a_{k+l} - e_k c_l, \; l = 1, \ldots, p. \tag{6.7}$$

The coefficients c_l project $\mathbf{f}_k$ onto the space $\mathcal{S}_k$ defined by $\{\mathbf{f}_{k+1}, \ldots, \mathbf{f}_{k+p}\}$. Because of the shift invariance property, these coefficients are independent of k. Shift

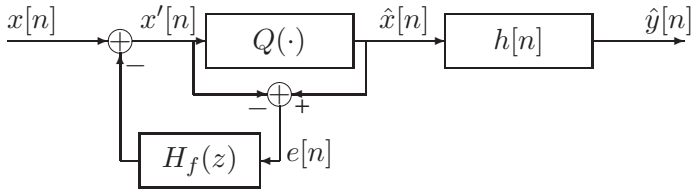


FIGURE 6-1: Noise shaping quantizer, followed by filtering

Noise Shaping order	Oversampling Ratio $r=2$	$r=4$	$r=8$	$r=16$	$r=32$	$r=64$
$p=1$	0.9	0.2	0.1	0.0	0.0	0.0
$p=2$	4.5	3.8	3.6	3.5	3.5	3.5
$p=3$	9.1	8.2	8.0	8.0	8.0	8.0
$p=4$	14.0	13.1	12.9	12.8	12.8	12.8

TABLE 6.1: Gain in dB in in-band noise power comparing p^{th} order classical noise shaping with p^{th} order noise shaping using projections, for different oversampling ratios r.

invariance also simplifies equation (5.43) to equation (3.21):

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_{p-1} \\ R_1 & R_0 & \cdots & R_{p-2} \\ \vdots & & \ddots & \vdots \\ R_{p-1} & & \cdots & R_0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_p \end{bmatrix}, \tag{3.21}$$

with R_m being the frame autocorrelation function.

The implementation for higher order noise shaping before filtering is shown in figure 6-1, with $H_f(z) = \sum_{l=1}^{p} c_l z^{-l}$, where the c_l solve (3.21). The feedback filter implements the projection and the coefficient update described in equation (6.7).

For the special case of the oversampling frame, table 6.1 demonstrates the benefit of adjusting the feedback loop to perform a projection. The table reports the approximate dB gain in reconstruction error energy using the solution to (3.21) compared to the classical feedback loop implied by (5.41). For example, for oversampling ratios greater than 8 and third order noise shaping, there is an 8dB gain in implementing the projections. The gain figures in the table are calculated using the additive noise model of quantization.

The applications in this section can be extended for frames generated by oversampled filterbanks, a case extensively studied in [9]. In that work, the problem is posed in

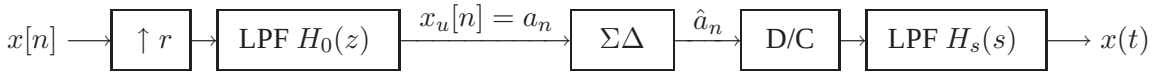


FIGURE 6-2: Classical Sigma-Delta DAC architecture

terms of prediction and quantization of the prediction error. Motivated by that work, we determined the solution to the filterbank problem using the projective approach. Setting up and solving for the compensation coefficients using equation (5.43) in section 5.6.2 corresponds exactly to solving equation (21) in [9], the solution to that setup under the white noise assumption.

It is comforting that the approach presented in this section, although different from [9] generates the same solution. Conveniently, the experimental results from that work apply in our case as well. Our theoretical results complement [9] by providing a projective viewpoint to the problem, developing a deterministic cost function and showing that even in the case of critically sampled biorthogonal filterbanks noise shaping can provide improvements compared to scalar coefficient quantization. On the other hand, it is not straightforward to use our approach to analyze and compensate for colored additive noise, as described in [9].

6.2 Multistage D/A Converters

The use of projections for error compensation only assumes a predetermined synthesis method using the frame synthesis equation. The method used to determine the representation coefficients has no effect on the methods and algorithms developed. Specifically in the case of noise shaping, this allows for more efficient implementation of classical Sigma-Delta noise shaping structures for the oversampling frame.

Figure 6-2 presents the typical structure of a classical Sigma-Delta digital to analog converter (*DAC*). The signal $x[n]$, to be converted to the signal $x(t)$ is being upsampled by an integer factor r to the intermediate representation $x_u[n]$, using the lowpass filter $H_0(z)$. The coefficients $x_u[n]$ are the frame representation coefficients of the signal using the r-times oversampling frame. This representation is subsequently quantized to the desired precision[2] using a Sigma-Delta quantizer of order p. The quantized representation is then used to reconstruct the signal using a low precision,

[2] We should use the term re-quantized to be precise, since this part of the system is implemented digitally, and the coefficients of $x[n]$ and $x_u[n]$ are digitally processed. Therefore they have already been quantized to a high precision. The Sigma-Delta quantizer re-quantizes them to a lower precision. For the purposes of this discussion, we can consider the original coefficients to be of infinite precision compared to the precision of the quantizer output.

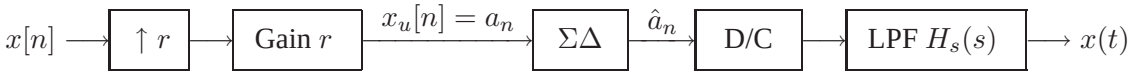


FIGURE 6-3: Simplified Sigma-Delta DAC architecture with the low-pass filter $H_0(z)$ replaced by a gain r. This architecture has the same performance as the one in figure 6-2.

oversampled DAC, followed by the low pass filter $H_s(s)$ to reject the out-of-band quantization noise. The combination of the DAC with the filter implements the synthesis equation for the frame. The quantizer should therefore be designed based on that filter, not on the analysis method.

6.2.1 Elimination of the Discrete-time Filter

As we discuss in chapter 2, the use of a frame decouples the analysis from the synthesis. In figure 6-2, the analysis is performed by the digital low-pass filter $H_0(z)$. The frame implied by this filter, assuming it is ideal, is the dual of the synthesis frame implied by the output filter $H_s(s)$. In principle, $H_0(z)$ can be replaced by a gain factor r, which implies a different analysis frame for $x[n]$. The resulting coefficients a_k are different, but represent the same signal, assuming the reconstruction is not modified. Thus, the implementation and the performance of the Sigma-Delta quantizer should not be affected by the change. For any particular input to the modified system, the output $\hat{a}_k$ is also different, but it represents the same signal, with the same error after reconstruction on average. The resulting system is simplified significantly, as shown in figure 6-3.

The elimination of the filter H_0 from the signal path has both advantages and disadvantages. If H_0 is designed to equalize H_s, its elimination poses tighter constraints on the design of H_s. On the other hand, if the output filter H_s is ideal or matches the system specifications, the elimination of H_0 removes a potential source of signal degradation.

One important role of the discrete time filter H_0 is the implementation of a sharp cutoff to eliminate the high-frequency components present in the signal due to the expansion operation. If H_0 is eliminated, even in the absence of quantization, the filter H_s at the output should reject all components above the bandwidth of the signal. In the classical implementation of figure 6-2 these components are rejected by H_0 in the digital domain. Thus, in the classical implementation, if the analog filter at the output H_s is not as sharp, the only side-effect is that some out-of-band quantization noise will pass through in the signal reconstruction.

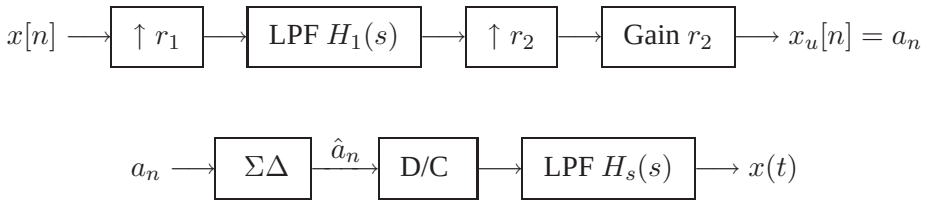


FIGURE 6-4: Two-stage simplified Sigma-Delta DAC architecture with the same performance as the one in figure 6-2.

6.2.2 Multistage Implementation

The design and manufacture of sharp continuous-time filters is not easy, and affects the total system cost. Thus, the elimination of the discrete-time filter has the potential of increasing the cost of the system since it transfers a sharp cutoff from the discrete-time domain to the continuous-time one. Furthermore the presence of the gain factor r after the expansion increases the likelihood that the quantizer will overflow if it has a finite number of levels. Alternatively, the gain can be placed after the D/C conversion stage which is equivalent to increasing the interval of the quantizer from Δ to $r\Delta$. In this case the resulting error magnitude will increase, although the quantizer is less likely to overflow.

It is also possible to implement a practical intermediate system structure that uses a continuous-time low-pass filter with loose specifications, and a discrete-time one with looser specifications compared to the classical case. The gain factor is also reduced, thus decreasing the probability of overflow. This two-stage expansion is demonstrated in figure 6-4, in which $r_1 r_2 = r$ so that the output rate to the DAC is the same as the previous systems. The discrete time filter H_1 can also be used to equalize the output filter H_s if this is necessary. Furthermore, the gain can be placed after the D/C conversion stage, as shown in figure 6-5. This increases the error compared to the system in figure 6-4, since the scaling of the quantizer is effectively changed but further reduces the probability of overflow.

The use of multiple stages to implement interpolation systems has been extensively studied in [16]. In that work it is demonstrated that the implementation of an interpolation system using multiple interpolation stages can decrease the computational complexity of the resulting system. This section further recognizes that in Sigma-Delta digital to analog conversion the functionality of the interpolation filter is replicated by the synthesis filter at the output of the quantizer.

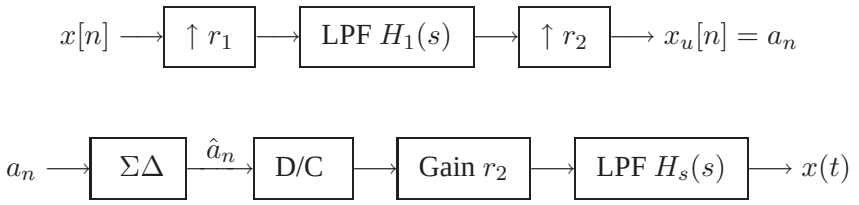


FIGURE 6-5: Two-stage simplified Sigma-Delta DAC architecture with the gain r_2 placed after the D/C converter. Moving the gain effectively modifies the quantization interval Δ of the quantizer, thus affecting the quantization performance.

6.2.3 Conversion Performance

The systems in figures 6-3, 6-4, and 6-5 can all be analyzed in the frequency domain using the classical noise model and be compared to the quantization performance of a direct quantization system. In the absence of the quantizer the system behaves as an ideal digital to analog converter. Using the additive white noise model and the Sigma-Delta modulator, the analysis follows the standard approach described in a variety of references [13, 3].

However, white noise is not a good model for quantization in these systems, especially in the case of direct scalar quantization. Specifically, the expansion by r without subsequent interpolation produces a sparse signal in which every non-zero coefficient is followed by $(r-1)$ zeros. Quantizing this signal produces a signal in which $(r-1)$ out of every r coefficients are the same and have the same error.[3] Therefore, depending on the error due the quantization of zero values, if noise shaping is not used, performance might deteriorate significantly.

The presence of the Sigma-Delta loop significantly improves the performance, and makes the white noise model more plausible. In addition, the upper bound introduced in section 6.1 provides an alternative cost measure that demonstrates that the worst case performance with or without the interpolation filter is the same. The use of projections minimizes the incremental error of quantization by taking the reconstruction into account. Thus, the error rejection of the system is determined by the shape and the nullspace of the output low-pass filter. The redundancy is introduced by the expansion operation and the existence of more coefficients to represent the signal, not by the interpolation that follows the expansion. Furthermore, the zeros introduced after the expansion provide a practical advantage. The quantizer is less likely to overflow on these coefficients since the error feedback from the Sigma-Delta modifies a zero coefficient, and not a coefficient that might be already near the overflow

[3] The error might in fact be zero for these coefficients, depending on whether zero is one of the quantization levels of the quantizer.

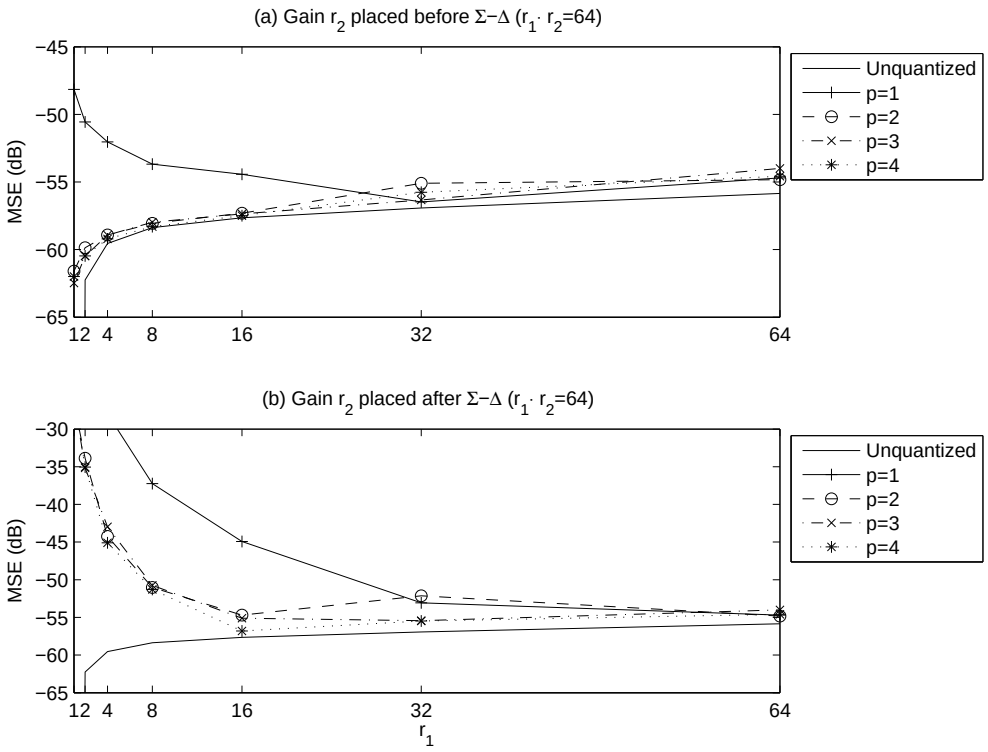


FIGURE 6-6: Performance of two-stage Sigma-Delta quantizers, with an interpolation filter used only after the first stage. In (a) the filter of the second stage is replaced by a gain factor r_2. In (b) the gain factor is placed in the system output. Note that the y-axis scale is different in the two plots.

boundary.

Figures 6-6(a) and (b) demonstrate simulation results for the systems presented in figures 6-4 and 6-5 respectively. The figures explore the simulation performance for $r = 64$ and $r_1 = 1, 2, 4, 8, 16, 32,$ and 64. The output filter is simulated in discrete-time using a 4097 point Hamming-window low-pass filter with cutoff at $\pi/64$. The interpolation filter H_1 is implemented using a 4097 point Hamming-window low-pass filter with cutoff frequency π/r_1. The cases of $r_1 = 64$ and $r_1 = 1$ correspond to the classical Sigma-Delta system in figure 6-2 and the system in figure 6-3, respectively. The figures plot the quantization performance of Sigma Delta quantizers designed optimally using (3.21) with the autocorrelation of the output filter H_0. In the simulations $\Delta = 1$ and the input signal tested is white noise uniformly distributed in ± 0.5. The solid line without markers represents the performance of the system without quantization, i.e. displays the distortion only due to the filters. It should be

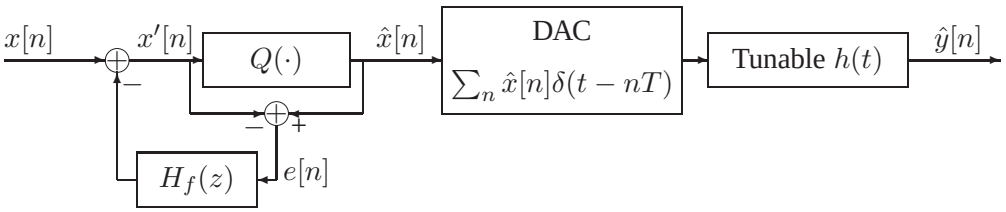


FIGURE 6-7: Tunable digital to analog converter.

noted that the y-axis scale is different in plots (a) and (b), to demonstrate the details of the plots.

In figure 6-6(a) it is evident that beyond 2^{nd} order quantization $(p \geq 2)$, the distortion due to the filters lower bounds the distortion due to the Sigma-Delta converter if the gain r_2 is placed before the quantizer. Therefore, as expected, the performance is improved by eliminating the filter using the systems in figures 6-3 and 6-4. On the other hand, a gain of 64, for example, requires a quantizer with a much higher overflow range, making this result impractical.

Figure 6-6(b) demonstrates that placing the gain after the DAC, using the system in figure 6-5, can still improve the error or reduce the complexity if the ratios r_1 and r_2 are chosen correctly. Specifically, for $r_1 \leq 8$, the effective increase of the quantization interval due to the gain increases the error significantly. On the other hand, for $r_1 \geq 16$ and $p \geq 2$ the performance is comparable to the system in figure 6-2.

These results demonstrate the potential to simplify practical DAC systems. However, the benefit depends on the design of the filter in the output of the converter, which forms the synthesis frame. In practical system design, further simulation and analysis is necessary to determine the exact tradeoff, if any.

6.3 Tunable Sigma-Delta Conversion

In an oversampled Sigma-Delta digital to analog converter the coarsely quantized output is low-pass filtered by a fixed low-pass filter. Similarly, the coarsely quantized output of an oversampled analog to digital Sigma-Delta converter is low-pass filtered and then decimated to produce a finely quantized signal at a lower rate. However, other filters can be applied at the output of the Sigma-Delta stage to perform reconstruction. In this case, the feedback loop is modified to implement the projection according to equation (3.21).

Band-pass Sigma-Delta converters sampling signals in a band of frequencies centered away from zero have been used in a variety of applications (for examples see [3] and references within). Tunable analog to digital converters have been introduced in [23],

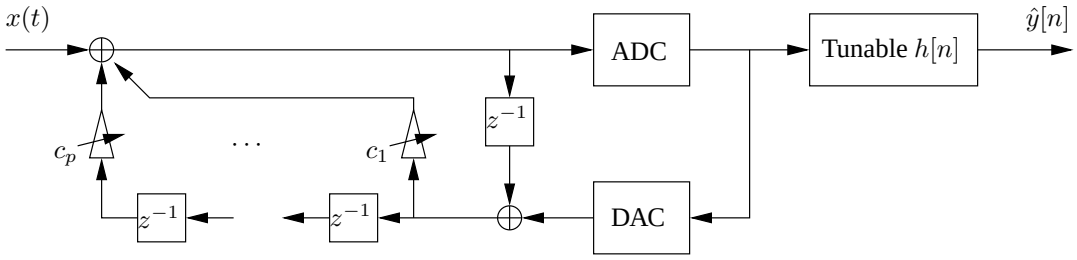


FIGURE 6-8: Tunable analog to digital converter. The gains c_i are tunable components. Their value is determined by inserting the autocorrelation of the tunable filter $h[n]$ in equation (3.21).

while tunable digital to analog converters have been mentioned in [35] but not further explored. This section presents these systems in the context of frames generated by arbitrary filters. Thus, the parameters in the feedback loop are determined using only the autocorrelation of the synthesis filter evaluated at the lags necessary to compute (3.21).

One application of such systems are tunable software radios in which signals of predetermined bandwidth should be acquired or generated at different center frequencies. For example a cellular phone operating in different countries or in different frequencies only needs to have one such converter. However, the flexibility of the systems allows the conversion of signals at varying bandwidths with varying fidelity in the representation. A wide-band or a multi-band signal can be acquired or generated with low precision at the output, while a narrow-band signal can be converted with higher precision.

It should be noted that the methods discussed in this section are not adaptive in real time. The converters presented are tuned before they start operating, according to the application. To modify their tuning their state should be reset, or some transient time period should be tolerated to reach steady state. In principle, it is possible to use the results of this chapter to design converters tunable in real-time without transients. In this case, the autocorrelation of the frame generated by the time-varying tunable (or adaptive) filter should be used in (3.21) to compute the time-varying parameters of the feedback loop at each time point. However, this is not an aspect explored in this section.

6.3.1 Tunable Digital to Analog Conversion

A tunable digital to analog converter is shown in figure 6-7, in which the filter $h(t)$ is tuned to the application requirements. Depending on the impulse response $h(t)$, the discrete-time filter $H_f(z)$ in the feedback loop should be adjusted to perform the

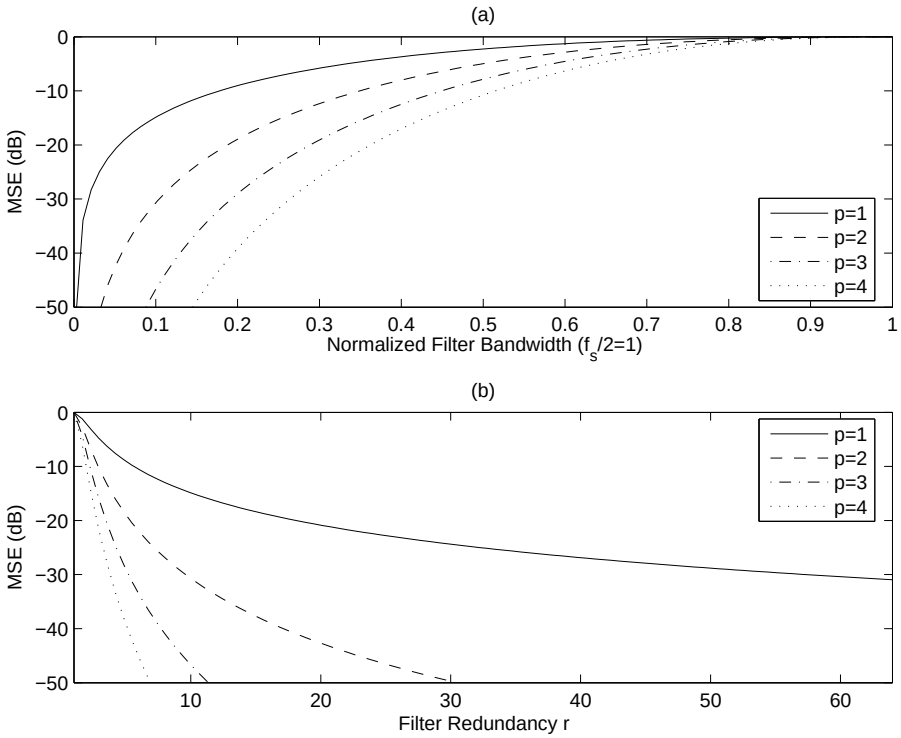


FIGURE 6-9: Tradeoff between error due to quantization and (a) filter bandwidth f_w or (b) filter redundancy r, assuming an ideal lowpass synthesis filter and optimal compensation of order p.

noise shaping projection as determined using (3.21). The impulse response $h(t)$ can in principle take any shape. In practice it is determined by a few variable components in the analog filter implementation.

The digital to analog conversion component converts the frame representation coefficients to continuous-time pulses at a high rate $f_s = 1/T$. The pulses are modeled as continuous time impulses. An arbitrary pulse shape can be incorporated into $h(t)$ by convolving the pulse shape $p(t)$ with the impulse response of the output filter. The DAC component has finite quantization precision, which cannot be adjusted.

6.3.2 Tunable Analog to Digital Conversion

A tunable analog to digital converter can be implemented using the system in figure 6-8. The feedback loop is implemented using switched capacitor delays and tunable gains. The system implements the feedback loop of figure 6-1 for an analog to digital converter system. The output is filtered by an impulse response $h[n]$ to

produce the acquired signal. All the components of the system operate at a high rate $f_s = 1/T$. Depending on the application, the system output can be subsequently converted to a lower rate. As with the digital to analog system, the analog to digital conversion component is implemented as a combination of sampling and quantization with finite precision, which is not adjustable.

6.3.3 Optimal Tuning and Quantization Precision

The application using the tunable systems has control over the tunable filters and the parameters of the feedback loop. To achieve optimal performance, according to the metrics presented in section 6.1, the application should set the feedback parameters to match the output filter using (3.21). In tuning these filters, however, there is a tradeoff between the output error and the range of frequencies in the pass band of the filter.

Specifically, in both systems, the digital to analog and the analog to digital components have finite quantization precision. The effect of quantization at the output of the two systems depends on the noise shaping loop and the reconstruction filters $h[n]$ and $h(t)$. In general, the larger the nullspace of the filters, the smaller the error due to quantization at the output. The tradeoff is difficult to quantify without further assumptions on the filters.

For example, figure 6-9 demonstrates the tradeoff between bandwidth and average or worst-case error assuming the tunable filter is an ideal lowpass filter with tunable bandwidth f_w. The error is normalized such that it is 0dB when the filter is all-pass. In the figure, (a) plots the error in dB as a function of the cutoff frequency f_w of the filter, normalized by half the sampling rate. Plot (b) shows the error as a function of the redundancy $r = f_s/2f_w$ of the filter. In the figure, p is the order of the system used to determine the optimal coefficients in (3.21).

CHAPTER 7

Compensation for Erasures

As discussed in chapter 2, the redundancy of frame representations decouples the analysis using inner products from the synthesis using the synthesis sum. The coefficients a_k that represent a vector $\mathbf{x}$ using a pre-specified synthesis frame $\{\mathbf{f}_k\}$ and the synthesis equation (2.7) can be determined in a variety of ways (for some examples, see [33, 27] and references within).

Similarly, the coefficients a_k of a vector analyzed using the analysis frame and equation (2.8) can be used in a variety of ways to synthesize the vector. For example, it is not necessary to use all the coefficients to reconstruct the signal. A subset of the coefficients is sufficient to represent the signal as long as the corresponding frame vectors still span the space. In this case, perfect reconstruction is possible, making the representation robust to erasures during transmission.

Consequently, most of the existing work on erasures on frame representations assumes that $\mathbf{x}$ is analyzed using inner products with an analysis frame. Under this assumption, the synthesis is modified to reconstruct the original signal. For example, linear reconstruction can be performed using a recomputed synthesis frame and equation (2.7) [27, 31]. Alternatively the erased coefficients can be re-computed using the non-erased ones, and used to fill in the coefficient stream. The vector is linearly synthesized using the recovered stream and the original synthesis frame [8, 7]. However, neither approach is possible without assuming an expansion using equation (2.8).

In this chapter, rather than assuming that the vector is analyzed using the analysis equation (2.8), we make no assumptions on how the representation coefficients a_k

are generated. We only assume that the synthesis is performed using a pre-specified synthesis frame and the synthesis sum of equation (2.7). The representation coefficients may be generated in a variety of ways, including the analysis equation, the use of the matching pursuit [33], or just coefficients to be used with the synthesis sum. Under this assumption, it is not possible to fill in the missing coefficients or appropriately modify the synthesis frame at the receiver.

We consider two cases. In the first case the transmitter is aware of the erasure events and uses the remaining coefficients in order to ensure that the synthesis using the pre-specified synthesis frame minimizes the reconstruction error. The receiver in this case only needs to perform the synthesis. In the second case the transmitter is not aware of the erasure event. Instead, the transmitter encodes the coefficients such that the receiver is able to recover the signal with as small error as possible if the erasure occurs.

In principle, it is possible to synthesize $\mathbf{x}$ at the transmitter using the synthesis frame and the synthesis sum of equation (2.7). Subsequently, a frame representation can be recomputed using an appropriate analysis frame. If the transmitter is aware of the erasures pattern, for example, it can expand the synthesized vector $\mathbf{x}$ using the dual of the remaining synthesis frame, taking that erasures pattern into consideration. Similarly, if the transmitter is not aware of the erasures, it can analyze $\mathbf{x}$ using any frame $\{\phi_k\}$ with the same redundancy and transmit these coefficients instead. The receiver receives some of the re-computed coefficients and synthesizes $\mathbf{x}$ using the dual of $\{\phi_k\}$ given the erasures pattern, as discussed in [27, 31, 8, 7]. This approach, however, requires significant computation and knowledge of most of the erasures pattern either at the transmitter or the receiver, which can generate significant delays in the reconstruction of the signal.

The algorithms described in this chapter, instead, modify the representation coefficients using orthogonal projections at the transmitter to properly compensate for an erasure. This assumes that the transmitter is aware that an erasure occurs, which is the first case considered. Even in the second case, in which only the receiver is aware that an erasure occurs, we demonstrate that a simple transmitter/receiver combination can implement the same compensation method. The transmitter modifies the frame representation assuming the erasures will occur, and the receiver undoes the changes if the erasures do not occur. The input-output behavior of the transmitter/receiver pair is identical to the input-output behavior of a transmitter which is aware of the erasure occurrence.

One advantage of using this approach is that the complete erasures pattern does not need to be known in advance. Furthermore, the representation coefficients may be generated in a variety of ways and it is not necessary to synthesize and re-analyze the signal $\mathbf{x}$ at the transmitter or the receiver. The drawback is that the causality constraints imposed in part of this development often allow only for partial compensation of the error. The approach described here is more appropriate for large or infinite frame setups, and streaming conditions, in which delay is important. For applications using small finite frames, in which delay is not an issue, this method is

not well suited.

The use of projections to compensate for erasures is similar to their use in chapter 5 to extend quantization noise shaping to arbitrary frame expansions. However, in that case, the quantization error is known at the transmitter—not necessarily the case with erasure errors. The use of redundancy to compensate for erasures assuming a fixed reconstruction method has also been considered in a different context in [21, 22]. In that work the error is again known at the transmitter and only the case of LTI reconstruction filters is considered. The problem is formulated and solved as a constrained optimization.

Projections can similarly be used at the transmitter to intentionally introduce erasures for the purpose of puncturing a dense representation. Erasures compensated for with projections can be the basis for algorithms that produce sparse representations from dense ones, a process we refer to as sparsification. They can also be combined with quantization, in which the combined error is projected to the remaining coefficients, as described in chapter 5, although not necessarily in a data-independent ordering. In that context, erasures can also be viewed as an extreme form of quantization, and can be compensated for accordingly.

The usefulness of redundant dictionaries of vectors as a synthesis set for approximate sparse representations has been shown for example in [28, 33], although the algorithms used to determine the sparse representation are different compared to our approach. In these papers the matching pursuit principle is used to produce a sparse representation starting from no representation at all. The set of coefficients and corresponding dictionary elements is augmented until the signal of interest is sufficiently approximated. In contrast, section 7.3 describes an algorithm that starts from a dense exact representation and removes dictionary elements and the corresponding coefficients until the signal is sparse enough or the maximum tolerable approximation error is reached.

The next section states the problem and establishes the notation. It is shown that the optimal solution is the orthogonal projection of the erasure error to the span of the remaining synthesis vectors, and some properties of sequential compensations are proven. A causal implementation is proposed in section 7.2.1, assuming the transmitter is aware of the erasure. Section 7.2.2 presents a transmitter that pre-compensates for the erasure and a receiver that undoes the compensation if the erasure does not occur. The use of projections to sparsify dense representations is explored in section 7.3.

7.1 Erasure Compensation Using Projections

After stating the problem and establishing notation, this section examines the compensation of a single erasure in the context of chapter 3. In section 7.1.3 the results are extended to the compensation of multiple erasures, and properties of sequential compensations are considered.

7.1.1 Problem Statement

We consider the synthesis of a vector $\mathbf{x}$ using (2.7):

$$\mathbf{x} = \sum_k a_k \mathbf{f}_k, \tag{2.7}$$

in which we make no assumptions on how the representation coefficients $\{a_k\}$ originate. The $\{a_k\}$ might even be data to be processed using the synthesis sum (2.7), such as a discrete-time signal to be filtered, not originating from the analysis of $\mathbf{x}$.

The coefficients $\{a_k\}$ are used to synthesize the signal using the pre-specified synthesis frame $\{\mathbf{f}_k\}$, subject to erasures known at the transmitter or the receiver. We model erasures as replacement of the corresponding a_k with 0, i.e. removal of the corresponding term $a_k\mathbf{f}_k$ from the summation in (2.7). Since the analysis method is not known, the goal is to compensate for the erasure as much as possible using the remaining non-erased coefficients.

Thru section 7.2.1 we assume that the transmitter anticipates an erasure and knows the value of the erased coefficient. Assuming coefficient a_i is erased, the transmitter is only allowed to replace the coefficients $\{a_k | k \in S_i\}$ with $\{\hat{a}_k | k \in S_i\}$ in order to compensate for the erasure, where $S_i = \{k_1, \ldots, k_p\}$ denotes the set of coefficient indices used for the compensation of a_i. The reconstruction is performed using equation (2.7) with the updated coefficients:

$$\hat{\mathbf{x}} = \sum_{k \in S_i} \hat{a}_k \mathbf{f}_k + \sum_{k \notin S_i, k \neq i} a_k \mathbf{f}_k, \tag{7.1}$$

such that $\hat{\mathbf{x}}$ minimizes the magnitude of the error $\mathcal{E} = \mathbf{x} - \hat{\mathbf{x}}$.

7.1.2 Compensation of a Single Erasure

The error due to the erasure of a single coefficient a_i and its subsequent compensation using the coefficients $\{a_k | k \in S_i\}$ can be rewritten using the synthesis sum:

$$\mathcal{E} = a_i \mathbf{f}_i + \sum_{k \in S_i} (a_k - \hat{a}_k) \mathbf{f}_k \tag{7.2}$$

The vectors $\{\mathbf{f}_k | k \in S_i\}$ span a space $\mathcal{W}_i$. Therefore, the error magnitude is minimized if the sum $\sum_{k \in S_i} (a_k - \hat{a}_k)\mathbf{f}_k$ is the orthogonal projection of $-a_i \mathbf{f}_i$ onto $\mathcal{W}_i$.

As described in chapter 3, we use the projection coefficients $c_{i,k}$, which satisfy:

$$\mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i) = \sum_{k \in S_i} c_{i,k} \mathbf{f}_k, \tag{7.3}$$

in which $\mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)$ is the projection of $\mathbf{f}_i$ onto $\mathcal{W}_i$. The projection coefficients are used

to optimally compensate for the erasure a_i by updating each of the a_k to:

$$\begin{aligned} \hat{a}_k &= a_k + a_i c_{i,k}, \text{ for all } k \in S_i && (7.4)\\ \Rightarrow \mathcal{E} &= a_i \mathbf{f}_i - a_i \sum_{k \in S_i} c_{i,k} \mathbf{f}_k && (7.5)\\ &= a_i (\mathbf{f}_i - \mathcal{P}_{\mathcal{W}_i}(\mathbf{f}_i)) && (7.6)\\ &= a_i \tilde{c}_i \mathbf{r}_i, && (7.7) \end{aligned}$$

in which $\tilde{c}_i$ and $\mathbf{r}_i$ are the error coefficient and the residual direction, as defined in (3.10) and (3.11) respectively.

As described in chapter 3, the projection coefficients $c_{i,k}$ satisfy equation (3.18):

$$\begin{bmatrix} R_{k_1,k_1} & \cdots & R_{k_1,k_p} \\ \vdots & \ddots & \vdots \\ R_{k_p,k_1} & \cdots & R_{k_p,k_p} \end{bmatrix} \begin{bmatrix} c_{i,k_1} \\ \vdots \\ c_{i,k_p} \end{bmatrix} = \begin{bmatrix} R_{i,k_1} \\ \vdots \\ R_{i,k_p} \end{bmatrix}$$
$$\Leftrightarrow \mathbf{R}\mathbf{c} = \rho, \qquad (3.18)$$

in which $R_{k,l} = \langle \mathbf{f}_k, \mathbf{f}_l \rangle$ is the frame autocorrelation function.

Satisfying (3.18) is equivalent to computing the frame expansion of $\mathbf{f}_i$ using $\{\mathbf{f}_k | k \in S_i\}$ as a synthesis frame. If the frame vectors $\{\mathbf{f}_k | k \in S_i\}$ are linearly dependent, the solution to (3.18) is not unique. All the possible solutions are optimal in terms of minimizing the error magnitude, given the constraint that only coefficients $\{a_k | k \in S_i\}$ can be modified. If the vector $a_i \mathbf{f}_i$ being compensated is in the span of the vectors $\{\mathbf{f}_k | k \in S_i\}$ used for the compensation (i.e. $\mathbf{f}_i \in \mathcal{W}_i$), then the erasure is fully compensated for. In this case the error is 0, and we call the compensation complete. In the development above we assume only one erasure, i.e. that none of the $\{a_k | k \in S_i\}$ are erased during the transmission.

7.1.3 Compensation of Multiple Coefficients

Projection-based compensation can be generalized to the sequential erasure of multiple expansion coefficients, allowing a subset of the remaining coefficients for each compensation. The sets S_i of coefficients used to compensate each of the erasures are part of the system design constraints. We assume that once a coefficient has been erased and compensated for, it is not used to compensate for subsequent erasures. Under these assumptions four properties of the compensation are derived. In formulating these, the term optimal is used if the compensation minimizes the error given the constraints and the term complete is used if the error after the compensation is exactly 0. These properties are:

(a) Compensation of the error is equivalent to projection of the data. Consider the vector $\mathbf{y}$ that can be synthesized from the erased coefficient a_i and the coefficients to be modified $\{a_k | k \in S_i\}$. Projecting $\mathbf{y}$ to the space $\mathcal{W}_i$, spanned by the frame vectors corresponding to the coefficients to be modified is equivalent

to compensating for the erasure. Specifically,

$$\mathbf{y} = a_i\mathbf{f}_i + \sum_{k\in S_i} a_k\mathbf{f}_k$$

$$\mathcal{P}_{\mathcal{W}_i}(\mathbf{y}) = \mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i + \sum_{k\in S_i} a_k\mathbf{f}_k) \tag{7.8}$$

$$= \mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i) + \mathcal{P}_{\mathcal{W}_i}(\sum_{k\in S_i} a_k\mathbf{f}_k) \tag{7.9}$$

$$= \mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i) + \sum_{k\in S_i} a_k\mathbf{f}_k \tag{7.10}$$

$$= \sum_{k\in S_i} \hat{a}_k\mathbf{f}_k. \tag{7.11}$$

This also implies that the error after the compensation is orthogonal to all the frame vectors used for compensation.

(b) Superposition. Using the linearity of projections it follows that:

$$\mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i + a_j\mathbf{f}_j) = \mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i) + \mathcal{P}_{\mathcal{W}_i}(a_j\mathbf{f}_j). \tag{7.12}$$

Furthermore, if $S_i = S_j$ then $\mathcal{W}_i = \mathcal{W}_j$. Thus, if the set of coefficients $S_i = S_j$ is used to separately compensate for the erasure of two different coefficients a_i and a_j, then the superposition of the individual compensations produces the same error as the erasure of a single vector $a_i\mathbf{f}_i + a_j\mathbf{f}_j$ followed by compensation using the same set of coefficients S_i.

(c) Sequential superposition. If $\mathcal{W}_j \subseteq \mathcal{W}_i$ then

$$\mathcal{P}_{\mathcal{W}_j}\left(\mathcal{P}_{\mathcal{W}_i}(\mathbf{y})\right) = \mathcal{P}_{\mathcal{W}_j}(\mathbf{y}). \tag{7.13}$$

Furthermore, if $S_j \subseteq S_i$ then $\mathcal{W}_j \subseteq \mathcal{W}_i$. Consider the case in which one of the updated coefficients $\hat{a}_j, j \in S_i$, used in the compensation of a_i, is subsequently erased and optimally compensated for using the remaining coefficients in S_i. Using (a) and (b) this becomes equivalent to the following projection sequence of the data:

$$\mathcal{P}_{\mathcal{W}_j}(\mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i + \sum_{k\in S_i} a_k\mathbf{f}_k)) = \mathcal{P}_{\mathcal{W}_j}(\mathcal{P}_{\mathcal{W}_i}(a_i\mathbf{f}_i + a_j\mathbf{f}_j + \sum_{k\in S_j} a_k\mathbf{f}_k)) \tag{7.14}$$

$$= \mathcal{P}_{\mathcal{W}_j}(a_i\mathbf{f}_i + a_j\mathbf{f}_j + \sum_{k\in S_j} a_k\mathbf{f}_k) \tag{7.15}$$

$$= \sum_{k\in S_j} \hat{a}_k\mathbf{f}_k, \tag{7.16}$$

in which $S_j = \{k \neq j | k \in S_i\}$ contains all the elements of S_i except for j, and $\{\hat{a}_k | k \in S_j\}$ is the set of the updated coefficients after both erasures of a_i

and of $\hat{a}_j$ have been compensated. Therefore, this is equivalent to optimally compensating both a_i and a_j using the coefficients in S_j.

(d) Sequential complete compensation. If an $a_j, j \in S_i$ used in the compensation of a_i is subsequently erased but completely compensated using the set S_j, the compensation of a_i is still optimal since the incremental error of the second compensation is zero.

If the compensation of a_i was complete, the total error after both compensations is zero. In this case:

$$a_i\mathbf{f}_i + a_j\mathbf{f}_j + \sum_{k\in S_{i,j}} a_k\mathbf{f}_k = \sum_{k\in S_{i,j}} \hat{a}_k\mathbf{f}_k \tag{7.17}$$

$$= \mathcal{P}_{\mathcal{W}_{i,j}} \left(\sum_{k\in S_{i,j}} \hat{a}_k\mathbf{f}_k \right) \tag{7.18}$$

$$= \mathcal{P}_{\mathcal{W}_{i,j}} \left(a_i\mathbf{f}_i + a_j\mathbf{f}_j + \sum_{k\in S_{i,j}} a_k\mathbf{f}_k \right), \tag{7.19}$$

in which $S_{i,j} = \{k \neq j | k \in (S_i \cup S_j)\}$ is the combined set of indices used to compensate for the erasure of a_i and a_j. $\mathcal{W}_{i,j}$ is the space spanned by the corresponding frame vectors. Therefore, using (a), the sequential complete compensation in this case is equivalent to optimally and completely compensating the erasure of both a_i and a_j using the set $S_{i,j}$.

7.2 Causal Compensation

In this section we examine the causal compensation of coefficient erasures using a transmitter aware of the erasure occurrence. We also develop a transmitter/receiver pair, which implements the same causal compensation method, yet only the receiver is aware of the erasure occurrence.

7.2.1 Transmitter-aware Compensation

For the remainder of this section we assume the coefficients are transmitted in sequence, indexed by k in (2.7). We focus on causal compensation in which only a finite number of coefficients subsequent to the erasure are used for compensation. The projections are straightforward to implement if the transmitter is aware of the erasure occurrence.

For clarity of the exposition we first develop the algorithm for a shift invariant frame. Such a frame has autocorrelation that is a function only of the index difference, i.e. satisfies $R_{i,j} = R_{i-j,0} \equiv R_{|i-j|}$. Thus, $c_{i,i+k} = c_{0,k} \equiv c_k$, and a transmitter aware of the erasure occurrence can be implemented using the system in figure 7-1, in which

the feedback system H is linear and time-invariant with impulse response:

$$h_n = \sum_{k=1}^{p} c_k \delta_{n-k}. \tag{7.20}$$

In the figure, e_k denotes a sequence of 1 and 0, which multiplicatively implements the erasures. The resemblance of the system to Sigma-Delta noise shaping systems is not accidental; projection-based compensation of errors is introduced in [11, 12] and used in chapters 5 and 6, to extend Sigma-Delta noise shaping to arbitrary frames.

The compensation is optimal if the erasures are rare such that there is only one erasure within p coefficients, or if p is such that the erasure compensation is complete. Otherwise it is only a locally optimal strategy which minimizes the incremental error after an erasure has occurred, subject to the design constraints.

For arbitrary, shift varying frames, the feedback system H is time varying with coefficients that satisfy (3.18) at the corresponding time point. Specifically, the output y_i of H should be:

$$y_i = \sum_{k=1}^{p} c_{i-k,i} x_{i-k}, \tag{7.21}$$

in which $x_i = a_i(1 - e_i)$ is the input.

The input and the output of the transmitter satisfy:

$$\tilde{a}_i = \sum_{k=1}^{p} (1 - e_{i-k}) c_{i-k,i} \tilde{a}_{i-k} + a_i \tag{7.22}$$

$$\hat{a}_i = \tilde{a}_i e_i \tag{7.23}$$

$$\Rightarrow a_i = \hat{a}_i + (1 - e_i)\tilde{a}_i - \sum_{k=1}^{p} (1 - e_{i-k}) c_{i-k,i} \tilde{a}_{i-k} \tag{7.24}$$

This is a recursive algorithm. Although an erasure of a_i is compensated using only the next p coefficients, another coefficient $a_j, j \leq i + p$ might be erased within p coefficients from the first one. In this case, the compensation of the second erasure attempts to compensate for the erasure of the modified coefficient $\hat{a}_j$, i.e. for the erasure of the original data, a_j, of the second erased coefficient and for the additive part due to the compensation of a_i. Thus, the feedback loop is potentially unstable. We explore some stability conditions in section 7.2.3.

7.2.2 Pre-compensation with Correction

In many systems, particularly in streaming applications, the transmitter is not aware of the erasure occurrence. In such situations it is possible to pre-project the error at the transmitter side, assuming an erasure will occur. If the erasure does not occur, the receiver undoes the compensation. It should be emphasized that the algorithm described in this section has identical input output behavior to the one described in

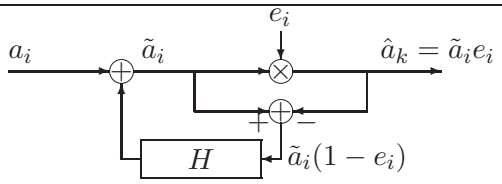
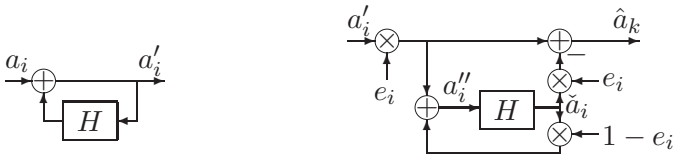


FIGURE 7-1: Erasure-aware transmitter projecting erasure errors.



(a) Transmitter (b) Receiver

FIGURE 7-2: Transmitter and receiver structure projecting erasure errors. Only the receiver is aware of the erasure.

section 7.2.1. Therefore all the analysis for that algorithm applies to this one as well.

To pre-compensate for the erasure, the transmitter at step i updates the subsequent coefficients $a_{i+1}, \ldots, a_{i+p}$ to:

$$a'_{i+k} = a_{i+k} + c_{i,i+k} a'_i, \tag{7.25}$$

where the $c_{i,i+k}$ satisfy (3.18). The a'_i used for the update is the coefficient as updated from all the previous iterations of the algorithm, not the original coefficient of the expansion, making the transmitter a recursive system. Depending on the frame, the transmitter might be unstable. This issue is separate from the stability of the compensation algorithm, raised in the previous section. Stability of this transmitter is also discussed in section 7.2.3.

If an erasure does not occur the receiver at time step i receives coefficient a'_i and sets $a''_i = a_i$. Otherwise it sets:

$$a''_i = \check{a}_i, \tag{7.26}$$

$$\text{with } \check{a}_i = \sum_{k=1}^{p} c_{i-k,i} a''_{i-k}, \tag{7.27}$$

which is the part of a'_i from equation (7.25) that is due to the projection of the non-erased coefficients. An erasure also erases the components of a'_i due to the projection of the previously received coefficients. The variables $\check{a}_i$ in (7.27) ensure that these

components can be removed from the subsequently received coefficients even when a'_i has not been received.

The receiver outputs $\hat{a}_i$, conditional on whether an erasure has occurred or not:

$$\hat{a}_i = (a'_i - \check{a}_i)e_i = \begin{cases} 0, & \text{if } e_i = 0 \\ a'_i - \check{a}_i, & \text{otherwise.} \end{cases} \tag{7.28}$$

This removes the projection of the previously received coefficients from a'_i.

To show that this system implements the same compensation method as the system in figure 7-1 we examine the evolution of the coefficients:

$$a_i = a'_i - \sum_{k=1}^{p} c_{i-k,i} a'_{i-k} \tag{7.29}$$

$$= a'_i - \sum_{k=1}^{p} c_{i-k,i} a'_{i-k} e_{i-k} - \sum_{k=1}^{p} c_{i-k,i} a'_{i-k}(1 - e_{i-k}), \tag{7.30}$$

$$\check{a}_i = \sum_{k=1}^{p} c_{i-k,i} a''_{i-k} \tag{7.31}$$

$$= \sum_{k=1}^{p} c_{i-k,i} a'_{i-k} e_{i-k} + \sum_{k=1}^{p} c_{i-k,i} \check{a}_{i-k}(1 - e_{i-k}). \tag{7.32}$$

Rearranging (7.30) and substituting into (7.32):

$$\check{a}_i = a'_i - \sum_{k=1}^{p} c_{i-k,i} a'_{i-k}(1 - e_{i-k}) - a_i + \sum_{k=1}^{p} c_{i-k,i} \check{a}_{i-k}(1 - e_{i-k}) \tag{7.33}$$

$$\Rightarrow a_i = a'_i - \check{a}_i - \sum_{k=1}^{p} c_{i-k,i}(a'_{i-k} - \check{a}_{i-k})(1 - e_{i-k}) \tag{7.34}$$

$$\Leftrightarrow a'_i - \check{a}_i = \sum_{k=1}^{p} c_{i-k,i}(a'_{i-k} - \check{a}_{i-k})(1 - e_{i-k}) + a_i \tag{7.35}$$

which holds for any input a_i and any signal e_i, not restricted to be an erasure pattern of zeros and ones. Comparing with (7.22), it follows that:

$$\tilde{a}_i = a'_i - \check{a}_i, \text{ for all } i. \tag{7.36}$$

Using (7.28) in (7.36), the output $\hat{a}_i$ is equal to:

$$\hat{a}_i = a'_i e_i - \check{a}_i e_i = \tilde{a}_i e_i, \tag{7.37}$$

which is the same as (7.23). Thus, the two systems are input-output equivalent.

The reconstruction in equation (7.28) undoes the recursive effects of (7.25) and en-

sures that the projection only affects the p coefficients subsequent to the erasure. The system looks like the one in figure 7-2, in which e_i, the sequence of ones and zeros denoting the erasures, is the same in all three locations in the figure. The systems H are the same as in figure 7-1.

In several applications, such as packetized transmissions, frame expansions are used for transmission of blocks of coefficients. In such cases the systems described can be modified using property (b) in section 7.1.3 to accommodate block erasures by projecting the whole vector represented by the transmitted block to the subsequent coefficients.

7.2.3 Compensation Stability

Depending on the frame and the erasure pattern, the system in figure 7-1 can become unstable. This section examines some aspects of the instability and provides a necessary condition and a sufficient condition for the systems to be stable. The conditions are presented assuming a shift-invariant frame. In this discussion, stability refers to bounded-input-bounded-output (BIBO) stability.

The evolution of the system variables is determined by equation (7.22). For a shift invariant frame this becomes:

$$\tilde{a}_i = \sum_{k=1}^{p} (1 - e_{i-k}) c_k \tilde{a}_{i-k} + a_i. \tag{7.38}$$

Consequently, $\tilde{\mu}_i$, the expected value of $\tilde{a}_i$ is:

$$\tilde{\mu}_i = \sum_{k=1}^{p} q c_k \tilde{\mu}_{i-k} + \mu_i, \tag{7.39}$$

in which $\mu_i = E\{a_i\}$, $\tilde{\mu}_i = E\{\tilde{a}_i\}$, and $q = P(e_i = 0)$ is the probability of erasures. Therefore, the compensation algorithm is stable in the mean if and only if the system $H(z) = 1/(1 - \sum_{k=1}^{p} q c_k z^{-k})$ is stable. Stability in the mean is a necessary but not sufficient condition for system stability.

The triangle inequality implies that the magnitude of the state has upper bound:

$$|\tilde{a}_i| \leq \sum_{k=1}^{p} |c_k| \cdot |\tilde{a}_{i-k}| + |a_i|. \tag{7.40}$$

Therefore, assuming a bounded input $|a_i|$, the stability of the algorithm is guaranteed for all q if the system $H(z) = 1/(1 - \sum_{k=1}^{p} |c_k| z^{-k})$ is stable. This is only a sufficient condition for stability. If it holds, this implies that the system is stable for any q, which also implies that the system described by (7.39) is also stable for all q. First order systems always have $|c_1| \leq 1$, which implies that first order optimal compensation algorithms for shift invariant frames are always stable.

The analysis above considers the stability of the compensation algorithm. The stabil-

ity of the transmitter in figure 7-2(a) is a separate issue. However, the output a_i' of the the transmitter in figure 7-2(a) follows the same dynamics as the expected value of the state in equation (7.39) with $q = 1$. Therefore the transmitter is a stable system if and only if the compensation algorithm is stable in the mean for $q = 1$. Otherwise it is not possible to implement the compensation algorithm using the transmitter/receiver combination described in section 7.2.2. Furthermore, BIBO stability of the compensation algorithm for $q = 1$ guarantees stability in the mean for $q = 1$, which implies that a separate transmitter and receiver system is also stable.

If both the transmitter and the compensation algorithm are stable for some probability of erasures q, then the receiver is also stable for the same q. The evolution of the receiver variables in (7.35) has the same dynamics as $\tilde{a}_i$ in (7.22). If the variable $\tilde{a}_i$ and the transmitter output a_i' are bounded, then the stability of the receiver state variable $\check{a}_i$ follows from (7.36). Furthermore, for any $q < 1$, the feedback loop in the receiver is reset to zero with probability 1 in a finite number of time steps after any erasure occurs. Therefore, the system does not exhibit any hidden instabilities, such as pole-zero cancellations, even in the case of parameter mismatch with the transmitter.

The solution to (3.18) might not provide coefficients that produce stable systems. In these cases, the equation can be modified to provide approximate solutions that balance the optimality of the projection with the stability of the system. Although we do not explore this issue, we should note that diagonal loading of the autocorrelation matrix $\mathbf{R}$ often leads to stable systems:

$$(\mathbf{R} + \alpha\mathbf{I})\mathbf{c} = \rho, \tag{7.41}$$

in which α is a small value. This is a simple method to implement, but it does not necessarily provide the best approximation tradeoff.

7.2.4 Simulation Results

In figure 7-3 simulation results are shown that demonstrate the performance of the algorithms in the case of i.i.d. erasures. The input a_i to the system is a white Gaussian process with unit variance and zero mean. The oversampling frame is approximated using a 4096th order, Hamming window FIR filter with cutoff π/r. The feedback coefficients are calculated using the filter autocorrelation. To compute the error, the output is compared to the unerased signal, as synthesized using the low-pass filter.

Four different cases are simulated. The systems in the top plots perform the optimal compensation of the erasures. The ones in the bottom plots use a diagonal loading factor $\alpha = 0.01$ in (3.18) to improve the stability at the expense of optimality. The oversampling rates are $r = 4$ and $r = 8$ for the left and the right plots, respectively. The plots display the mean squared error in dB against the probability of erasure q for various compensation orders $p = 0$ (i.e. no compensation) up to $p = 3$.

The figures demonstrate that increasing the compensation order improves the performance of the systems. They also show the tradeoff between stability and compen-

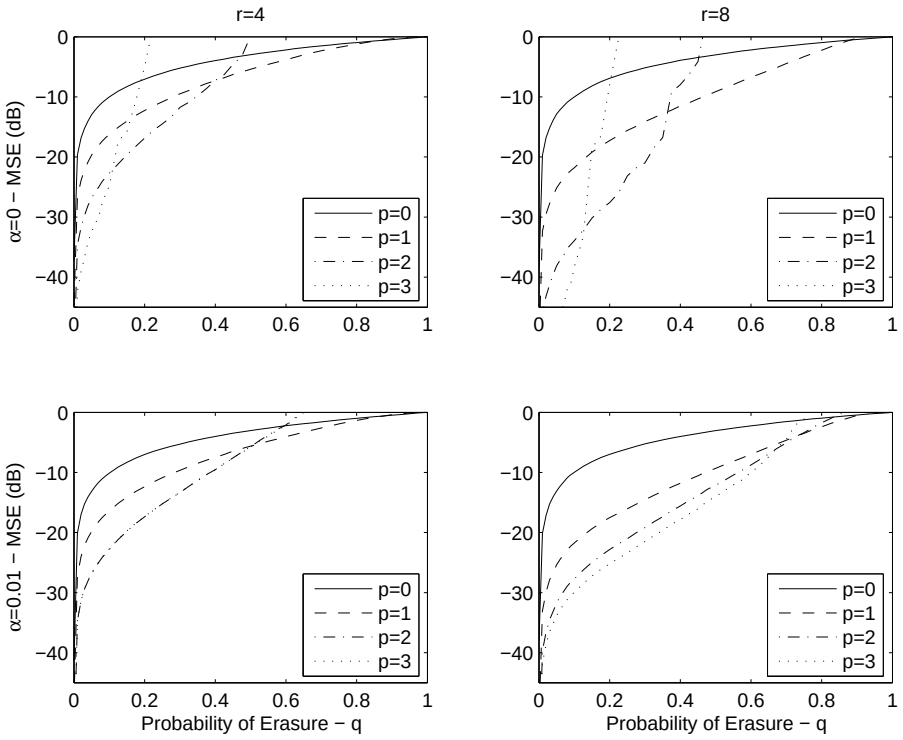


FIGURE 7-3: Performance of erasure compensation using projections for the uniform oversampling frame, with oversampling ratios $r = 4$ (left) and $r = 8$ (right). The top plots demonstrate the optimal (unstable) systems. In the bottom plots optimality is traded for stability. In the legend, p denotes the compensation order, and q the probability of erasure.

sation performance. It is evident in the top two plots that the second and third order systems become unstable at low probability of erasures—not the case in the bottom plots. On the other hand, especially for $r = 8$, there is an evident performance decrease to ensure stability. The plots also confirm that the $p = 0$ and $p = 1$ systems are stable.

7.3 Puncturing of Dense Representations

Coefficient erasures can also be introduced intentionally at the transmitter to sparsify dense representations. Sparse representations, in which most of the coefficients are zero, are useful in signal processing applications, such as compression, model order reduction, and feature selection. This section introduces an iterative sparsification algorithm based on the compensation using projections of intentional erasures.

The iterative algorithm is presented in 7.3.1. Section 7.3.2 uses the orthogonality of the projections to show that the incremental error at each iteration is orthogonal to the total error until that iteration, and, therefore, the total error magnitude is straightforward to compute on-line. Section 7.3.3 presents different approaches in determining the sequence of sparsifications to reach the desired sparsity. In section 7.3.4 the algorithm is extended to combine sparsification with quantization.

7.3.1 Puncturing Algorithm

In sparsifying a dense representation of a vector $\mathbf{x}$ we make no assumption on the origin of the representation $\{a_k\}$. The error introduced is measured against the synthesis of the dense representation. Specifically, if

$$\mathbf{x} = \sum_{k=1}^{M} a_k \mathbf{f}_k, \tag{7.42}$$

and the synthesis from the sparse representation is:

$$\hat{\mathbf{x}} = \sum_{k \in S} \hat{a}_k \mathbf{f}_k, \tag{7.43}$$

in which S denotes the indices of the coefficients remaining after the puncturing algorithm, and $\hat{a}_k$ denotes the updated remaining coefficients, then the error $\mathcal{E}$ introduced by the process is:

$$\mathcal{E} = \mathbf{x} - \hat{\mathbf{x}}. \tag{7.44}$$

At each iteration i of the iterative puncturing algorithm a number of coefficients are erased. The erasures are compensated using projections, as described in section 7.1, using all the remaining coefficients. Consistent with section 7.1, S_i denotes the set of indices of all the coefficients remaining in the representation after iteration i. Similarly, $\mathcal{W}_i$ denotes the vector space spanned by the corresponding frame vectors. The set $S_0 = \{1, \ldots, M\}$ contains the indices of all the frame vectors before the erasures are introduced. We refer to the sequence of S_i as the sparsification or erasures schedule.

The analysis of this algorithm does not assume that the schedule of erasures is predetermined. Specifically, the sequence of sets S_i may be adaptively determined during each iteration i using arbitrary rules. The number of coefficients erased at each iteration is also arbitrary, and depends on the rules specifying the sparsification schedule. The sets S_i satisfy:

$$S_i \subseteq S_{i-1} \Rightarrow \mathcal{W}_i \subseteq \mathcal{W}_{i-1}. \tag{7.45}$$

We use $S_i^c = \{k \in S_{i-1} | k \notin S_i\}$ to denote the complement of S_i in S_{i-1}, i.e. the set of all the coefficients to be erased at iteration i.

The algorithm stops according to some stopping rule after I iterations. The stopping condition is evaluated at the end of each iteration, and, therefore, I is in general data-dependent. For example the algorithm might stop once the desired sparsity or a

maximum tolerable error magnitude is reached.

In summary, the algorithm proceeds as follows:

1. Initialize $S_0 = \{1, \ldots, M\}$, $a_k^0 = a_k, k \in S_0$.

2. At iteration i determine $S_i^c \subseteq S_{i-1}$, the indices of coefficients to be erased, and the corresponding S_i, the set of coefficients to remain in the representation.

3. Update $a_k^i, k \in S_i$ using (7.4) for each coefficient $a_k^{i-1}, k \in S_i^c$ being erased.

4. If the stopping condition is met, stop and output $S = S_i, \hat{a}_k = a_k^i, k \in S_i$, and $I = i$. Otherwise increase i to $i+1$ and iterate from step 2.

In this algorithm we use $a_k^i, k \in S_i$ to denote the remaining non-erased coefficients as they have been modified after iteration i.

7.3.2 Error Evaluation

This section uses the orthogonality property of the projections to demonstrate that the incremental error introduced at each iteration of the algorithm is orthogonal to the error contributed from the other iterations. Therefore, the total error energy is straightforward to characterize as a sum of the incremental error energy.

We use $\mathbf{x}_i$ to denote the vector represented by the coefficients remaining after iteration i. Using (7.11) and (7.45):

$$\mathbf{x}_i = \sum_{k \in S_i} a_k^i \mathbf{f}_k \tag{7.46}$$

$$= \mathcal{P}_{\mathcal{W}_i}(\mathbf{x}_{i-1}) \tag{7.47}$$

$$= \mathcal{P}_{\mathcal{W}_i}(\mathbf{x}). \tag{7.48}$$

The error $\mathcal{E}_i$ contributed at each iteration is:

$$\mathcal{E}_i = \mathbf{x}_{i-1} - \mathbf{x}_i \tag{7.49}$$

$$= ||\mathcal{E}_i|| \mathbf{r}_i, \tag{7.50}$$

in which $\mathbf{r}_i = \mathcal{E}_i / ||\mathcal{E}_i||$ is the direction of the error vector at iteration i. The error $\mathcal{E}_i$, and therefore $\mathbf{r}_i$, is orthogonal to all the vectors in $\mathcal{W}_i$, which includes $\mathbf{x}_i$ and the remaining frame vectors $\mathbf{f}_k, k \in S_i$. Furthermore, $\mathcal{E}_i \in \mathcal{W}_{i-1}$. Using induction the orthogonality of the $\mathcal{E}_i$ follows:

$$\mathcal{E}_i \in \mathcal{W}_j, \text{ for all } j < i \tag{7.51}$$

$$\Rightarrow \quad \mathcal{E}_i \perp \mathcal{E}_j, \text{ for all } i \neq j \tag{7.52}$$

$$\Leftrightarrow \quad \langle \mathbf{r}_i, \mathbf{r}_j \rangle = \delta_{i,j} \tag{7.53}$$

The total error after k iterations is the sum of $\mathcal{E}_i, i = 1, \ldots, k$, which satisfies:

$$\sum_{i=1}^{k} \mathcal{E}_i = \mathbf{x} - \mathbf{x}_k \tag{7.54}$$

$$= \mathbf{x} - \mathcal{P}_{\mathcal{W}_k}(\mathbf{x}). \tag{7.55}$$

Using the orthogonality of $\mathbf{r}_i$, the magnitude of the error after k iterations can be determined from $||\mathcal{E}_i||$, the magnitude of the error at each iteration:

$$\sum_{i=1}^{k} \mathcal{E}_i = \sum_{i=1}^{k} ||\mathcal{E}_i|| \mathbf{r}_i \tag{7.56}$$

$$\Rightarrow \left|\left| \sum_{i=1}^{k} \mathcal{E}_i \right|\right|^2 = \sum_{i=1}^{k} ||\mathcal{E}_i||^2 \tag{7.57}$$

$$\Rightarrow ||\mathcal{E}||^2 = \sum_{i=1}^{I} ||\mathcal{E}_i||^2. \tag{7.58}$$

Thus the magnitude square of the total error is additive in the magnitude square of the incremental error at each iteration.

The additivity of the error is a particularly useful feature of this algorithm. It provides a simple performance measure at run time, which can guide the determination of the S_i at each iteration, before the sparsification is performed. It can also be useful in the evaluation of the stopping rule.

Since the frame is redundant, up to $M - N$ coefficient erasures can be tolerated without any error. In principle, these erasures can all be performed in the initial iterations of the algorithm, but this choice depends in general on the sparsification schedule and the rules used to determine it. Using certain frames, such as the harmonic frames [43, 27], any subset of N frame vectors spans the space, and the first $M - N$ erasures can be tolerated without error, independent of the sparsification schedule.

While the sparsification schedule and the stopping rule can be arbitrary, the frame vectors $\{\mathbf{f}_k | k \in S\}$ remaining at the conclusion of the algorithm should be linearly independent, and we assume this is the case. Otherwise, the representation can be further sparsified without increasing the error. Under this assumption, the expansion:

$$\hat{\mathbf{x}} = \sum_{k \in S} \hat{a}_k \mathbf{f}_k = \mathcal{P}_{\mathcal{W}}(\mathbf{x}), \tag{7.59}$$

is unique given $\mathbf{x}$ and S, independent of the intermediate steps taken. In this expression $\mathcal{W} = \text{span}\{\mathbf{f}_k, k \in S\}$ is the span of the frame vectors remaining after the sparsification.

The uniqueness of (7.59) implies that if two different sparsification schedules conclude with the same S at the output, then the resulting $\{\hat{a}_k\}$ are going to be equal.

This holds independent of which method is used to compute the left inverse of equation (3.18) at each iteration. However, the choice of left inverse to be used can influence the schedule determination algorithm, and therefore influence the final output S and the corresponding error.

7.3.3 Sparsification Schedule

In general, the determination of the sparsification schedule depends on the application. In this section we indicate certain heuristics that can be used at each iteration to determine the sparsification schedule. These heuristics, based on a greedy approach, select the coefficients that reduce the incremental error in each iteration, while taking into consideration the complexity of evaluating the error and the effect of the erasure to the remaining coefficients. We assume a frame in which any set of $M-N$ erasures can be tolerated without error, such as the harmonic frame [43, 27].

We consider two separate stages in the evolution of the algorithm. During the first stage more than N coefficients remain in the representation and any erasure can be completely compensated for. In this stage the incremental error $\mathcal{E}_i$ is always zero, and other criteria are used to determine the coefficients to be erased at each iteration. In the second stage less than N coefficients remain for compensation, and, therefore, each erasure cannot be fully compensated for. In this stage, we use the magnitude square of the incremental error to erase the coefficients with the smallest contribution to the total error.

For the first stage, we identify three different rules, presented from the most to the least complex to implement. Although the incremental error in the representation is zero at this stage, each rule affects the remaining coefficients, and therefore the scheduling algorithm, in a different way.

(a) Each iteration erases the coefficient whose compensation least affects the remaining coefficients in a mean square sense. Any coefficient erasure can be fully compensated using equation (7.4). The coefficient selected is the a_i for which the corresponding $a_i^2 \sum_{k \in S_i} c_{i,k}^2$ is as small as possible. An implication of using this rule is that the Moore-Penrose pseudoinverse should be used to determine the $c_{i,k}$ in equation (3.18). Thus, implementation of this heuristic can be computationally expensive.

(b) Each iteration erases the coefficient with the smallest magnitude. This heuristic is simpler to implement since it avoids the computation of the Moore-Penrose pseudoinverse for each of the coefficients considered. However, it might lead to an iteration that significantly affects the remaining coefficients, thus confusing the subsequent iterations of the algorithm.

(c) Erase the $M - N$ smallest coefficients in one iteration. This approach is the simplest and most greedy one. It has the advantage that it is the least expensive computationally. On the other hand, it has the potential to erase in one step a large number of small coefficients that together are significant for the signal representation and thus affect subsequent performance of the algorithm.

For the second stage, the incremental error is non-zero and can be used to guide the heuristics determining the schedule. In this stage we also identify three heuristics in decreasing order of complexity:

(a) Each iteration erases the coefficient that least contributes to the magnitude of the error, as computed in equation (7.57). As with heuristic (a) in the first stage, this is the least greedy and the most computationally expensive method.

(b) Each iteration erases the smallest coefficient from the ones remaining. This approach uses the magnitude of each coefficient as a proxy for the error due to their erasure. While it is computationally simpler than (a), it has the potential to erase a coefficient that contributes more to the incremental error than (a).

(c) In one iteration erase all the coefficients necessary to achieve the desired sparsity. The coefficients erased are the ones with the smallest magnitude. This is the simplest approach of the three, but the most greedy one. It has the further disadvantage that there is no control on the erasure error, only on the sparsity of the representation. Thus, it is not possible to use this approach to sparsify a representation up to a maximum tolerable error.

Heuristics (c) in both stages can also be combined to erase in one single step all the coefficients necessary to achieve the desired sparsity. The coefficients selected to be erased are the smallest in magnitude.

7.3.4 Quantization Combined with Sparsity

The sparsification algorithm can be combined with quantization to produce a sparse quantized representation. In addition to the coefficients to be erased, at each iteration i the algorithm determines which coefficients should be quantized, and how severe the quantization of each coefficient should be. Erasures can also be considered an extreme form of quantization to a single level, $p = 0$.

All the coefficients that have not been quantized or erased yet are used for the compensation of the total error. Consistent with the previous sections, we use S_i to denote the set of indices of these coefficients, and $\mathcal{W}_i$ to denote the space spanned by the corresponding frame vectors. The nesting described in equation (7.45) still holds, which implies that the error evaluation results of section 7.3.2 also hold. The only exception is the independence of the representation (7.59) from the erasure and quantization schedule. Due to the quantization of intermediate iterations, the final representation might be differently quantized for different schedules, even if the final set S is the same.

One difference in this case is that there is no stopping rule. The algorithm continues until all the coefficients have been quantized or erased and $S_I = \emptyset$. The algorithm keeps all the coefficients $\hat{a}_k$. The set S of the non-erased coefficients can be determined from the nonzero coefficients using $S = \{k|\hat{a}_k \neq 0\}$.

In summary, the algorithm is modified as follows:

1. Initialize $S_0 = \{1, \ldots, M\}$, $a_k^0 = a_k, k \in S_0$.

2. At iteration i determine $S_i^c \subseteq S_{i-1}$, the indices of coefficients to be erased or quantized, and the corresponding S_i, the set of coefficients to remain and be used for compensation

3. Set $\hat{a}_k = Q(a_k^{i-1}), k \in S_i^c$, according to the quantization and erasure schedule determined in step 2.

4. Update $a_k^i, k \in S_i$ using (7.4) to compensate for each coefficient $a_k^{i-1}, k \in S_i^c$, being quantized or erased.

5. If all the coefficient have been quantized or erased, stop and output $S = \{k|\hat{a}_k \neq 0\}$, $\hat{a}_k, k \in S$, and $I = i$. Otherwise increase i to $i+1$ and iterate from step 2.

In this algorithm $Q_k(\cdot)$ denotes the quantization or erasure function for the k^{th} coefficient, as determined in step 2.

CHAPTER 8 *Conclusions and Future Work*

This section summarizes the most important contributions of this thesis and suggests possible research directions indicated by the results.

8.1 Error Compensation Using Projections

The main tool used through this thesis is the compensation of errors by modifying frame coefficients. The modifications are such that the error is projected to the space spanned by the selected frame vector and subtracted from the corresponding coefficients. The use of projections assumes a pre-specified synthesis frame, but makes no assumptions on the analysis method.

In comparison, most of the existing work on frame representations assumes a pre-specified analysis method, using inner products [28, 38, 39, 7, 8, 34]. This assumption is used to determine the synthesis method depending on the error type. The redundancy of the analysis frame creates dependencies in the values of the representation coefficients. The synthesis algorithms exploit these dependencies to reduce the synthesis error. In contrast, assuming only a pre-specified synthesis frame provides no information on the values of the frame coefficients. Instead, the use of projections to compensate for errors exploits the existence of a nullspace and the spectral shape of the singular values of the synthesis operator.

One significant exception is the use of the matching pursuit principle to determine sparse and quantized sparse representations [33, 28]. Similar to the compensation using projections, the matching pursuit principle assumes a pre-specified synthesis

frame. However, the implicit assumption is that there no prior representation, i.e. that the matching pursuit principle is used as an analysis method. Thus it cannot be used to compensate for errors such as erasures or to quantize existing representations. Instead, the matching pursuit is useful in determining a representation, which includes the compensation coefficients used to project the error.

8.2 Quantization Limits

Although the inefficiency of scalar quantization was known [20, 39], in chapter 4 this inefficiency is quantified deterministically, independent of the frame used, for any finite level quantizer. Specifically, a lower bound is derived on both the bit waste and the quantization error reduction as a function of the frame redundancy. Consistent reconstruction methods [38, 28] are known to achieve that lower bound. The lower bound on the error growth is also derived in the context of the oversampling frame in [39], assuming a uniform infinite-level quantizer. In principle it is applicable to any finite frame. However, the general result in chapter 4 also quantifies the bit use, and assumes an arbitrary finite-level quantizer. The implications are important:

- The optimality of consistent reconstruction methods is demonstrated in terms of the reconstruction error decay as a function of the redundancy.
- A target rate for subsequent entropy coding is provided, independent of the scalar quantizer used. An optimal entropy coder subsequent to a scalar quantizer should represent the signal at a rate lower or equal to what the bound suggests.
- Smarter quantization on the encoder side is motivated. Since it is known how to achieve optimal error decay using consistent reconstruction, increasing the complexity of the synthesis method provides no benefit in that sense. The alternative, increasing the complexity of the analysis method, keeping the synthesis simple is more promising since the optimal error magnitude decay in this case can exponential instead of linear in the redundancy. Sigma-Delta noise shaping and the quantized matching pursuit are examples of such analysis methods, although they still do not achieve exponential decay.
- A benchmark is provided for frame design. Although the lower bound is proven for any frame, ill-designed frames might achieve much worse performance. A well-designed frame should achieve that lower bound. The oversampling frame and the harmonic frame do so [38, 39, 28].

8.3 Generalization of Sigma Delta Noise Shaping

Extensions of classical noise shaping to bandpass filter and reconstruction filterbanks have been explored in the literature [9, 3, 35, 23]. Recent work also considers extending Sigma-Delta noise shaping to finite frame expansions by subtracting the error from subsequent coefficients [5, 4].

Chapters 5 and 6 contribute a novel view of Sigma-Delta noise shaping as a projection of the error to the subsequent frame vectors. Furthermore, two cost functions are identified, one based on the additive white noise model, and one based on a deterministic upper bound. These contributions are important for several reasons:

- Extension of Sigma-Delta noise shaping to arbitrary frames is possible, with improvements in average and worst case error performance, independent of the frame vector ordering. The work in [5, 4], instead, requires the proper frame vector ordering to improve the quantization error.
- Alternative noise shaping algorithms are introduced, such as the tree quantizer. These allow for more flexibility in the quantizer design, further reducing the average error.
- The cost functions provide an algorithmic method to determine the ordering of frame vectors. The design of an optimal sequential quantizer is shown to be equivalent to solving the traveling salesman problem, while the design of an optimal tree quantizer is equivalent to the minimum spanning tree problem.

8.4 Compensation for Erasures

Error compensation using projections is also considered in the case of erasure errors. The main advantage, compared to existing work on erasures in frame representations, is that this compensation method does not make any assumptions on the origin of the frame expansion coefficients. The contributions of chapter 7 are significant:

- Projection of the erasure error to the remaining coefficients is shown to be equivalent to projection of the data.
- A transmitter that causally projects the erasure error to the subsequent coefficients is developed, assuming the transmitter is aware of the erasure events.
- It is demonstrated that the transmitter can be separated into two stable systems: a linear transmitter that encodes the data by pre-projecting the error assuming an erasure occurs, and a received that undoes the compensation depending on whether the error occurs. The input-output behavior of the separated systems is equivalent to the behavior of the transmitter that projects the erasure error only when an erasure occurs.
- A puncturing algorithm is derived that generates approximate sparse representations starting from dense exact ones.

8.5 Suggested Research Directions

This section identifies possible research directions, related to the topics discussed this thesis. This is only a biased sample of the possible research problems that exist in the field.

- Although error compensation using projections is examined for the case of quantization and erasures, the space of errors has not been exhausted. For

example, they can be used to compensate for additive random noise, white or colored.

- Chapter 3 develops the computation of the compensation coefficients to a certain extent. When the equation is underdetermined, however, several solutions exist, all of them optimal. However, each of the solution has further properties that can affect other aspects of the system performance, an aspect not examined in the thesis.
- Chapter 4 demonstrates the limits of the frame analysis using inner products, followed by scalar quantization. It is also shown that more involved analysis methods followed by linear synthesis can improve the error decay as a function of oversampling. However, there is no known general algorithm that achieves exponential error decay using linear reconstruction. It is also not known if such error decay is feasible, even with arbitrary complexity.
- Chapter 5 discusses the optimal ordering of frame vectors for first order noise shaping. The optimal ordering and grouping for higher order quantization is a difficult problem that is not addressed.
- The asymptotic performance of Sigma-Delta noise shaping using projections as a function of the frame redundancy has not been analyzed for arbitrary frames.
- Chapter 7 discusses the compensation of erasures, and the tradeoff between system stability and compensation performance. This tradeoff has not been explored in this thesis.
- The algorithms in chapters 5 through 7 can be generalized to vector quantizers and erasures of vectors.
- The determination of the schedule for the puncturing algorithm in chapter 7 is based on heuristics that have not been explored neither theoretically nor experimentally.
- The complexity of some of the heuristics in the puncturing schedule can be reduced by exploiting the structure of the problem. This is not something we analyze in this thesis.

Of course, this is only the tip of the research iceberg in this rich topic.

Bibliography

[1] D. Anastassiou. Error Diffusion Coding for A/D Conversion. *IEEE Transactions on Circuits and Systems*, 36(9):1175–1186, September 1989.

[2] S. Axler. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition, 1997.

[3] P. M. Aziz, H. V. Sorensen, and J. Van Der Spiegel. An Overview of Sigma-Delta Converters. *IEEE Signal Processing Magazine*, 13(1):61–84, January 1996.

[4] J. J. Benedetto, A. M. Powell, and O. Yilmaz. Sigma-Delta Quantization and Finite Frames. *IEEE Transactions on Information Theory*, submitted for publication. Available: http://www.math.umd.edu/~jjb/ffsd.pdf.

[5] J. J. Benedetto, O. Yilmaz, and A. M. Powell. Sigma-Delta Quantization and Finite Frames. In *Proceedings of IEEE ICASSP 2004*, Montreal, Canada, May 2004. IEEE.

[6] W. R. Bennett. Spectra of Quantized Signals. *Bell System Technical Journal*, 27:446–472, July 1948.

[7] R. Bernardini, M. Durigon, and R. Rinaldo. Low-delay Reconstruction of Punctured Frame-coded Streams. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1519– 1523. IEEE, Nov. 2003.

[8] R. Bernardini, A. Tonello, and R. Rinaldo. Efficient Reconstruction of Frames Based Joint Source-Channel Coded Streams in the Presence of Data Loss. In *Proceedings of the Symposium on Wireless Personal Multimedia Communications (WPMC)*, Yokosuka, Kanagawa, Japan, October 19-22 2003.

[9] H. Bolcskei and F. Hlawatsch. Noise Reduction in Oversampled Filter Banks Using Predictive Quantization. *IEEE Transactions on Information Theory*, 47(1):155–172, Jan 2001.

[10] H. Bolcskei, F. Hlawatsch, and H. G. Feichtinger. Frame-theoretic Analysis of Oversampled Filter Banks. *IEEE Transactions on Signal Processing*, 46(12):3256–3268, Dec 1998.

[11] P. Boufounos and A. V. Oppenheim. Quantization Noise Shaping on Arbitrary Frame Expansions. In *Proceedings of IEEE ICASSP 2005*, Philadelphia, PA, USA, March 2005. IEEE.

[12] P. Boufounos and A. V. Oppenheim. Quantization Noise Shaping on Arbitrary Frame Expansions. *EURASIP Journal on Applied Signal Processing, Special issue on Frames and Overcomplete Representations in Signal Processing, Communications, and Information Theory*, 2005. (accepted for publication).

[13] J.C. Candy and G. C. Temes, editors. *Oversampling Delta-Sigma Converters*. IEEE Press, 1992.

[14] P. G. Casazza and J. Kovacevic. Equal-Norm Tight Frames with Erasures. *Advances in Computational Mathematics, special issue on Frames*, pages 387–430, 2002.

[15] H. T. Cormen et al. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2nd edition, 2001.

[16] R. E. Crochiere and L. R. Rabiner. Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow-Band Filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(5):444–456, October 1975.

[17] Z. Cvetkovic and M. Vetterli. Overcomplete Expansions and Robustness. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, 1996.*, pages 325–328, Jun 1996.

[18] Z. Cvetkovic and M. Vetterli. Error-rate characteristics of oversampled analog-to-digital conversion. *IEEE Transactions on Information Theory*, 44(5):1961–1964, Sep 1998.

[19] Z. Cvetkovic and M. Vetterli. Tight Weyl-Heisenberg Frames in $l^2(Z)$. *IEEE Transactions on Signal Processing*, 46(5):1256–1259, May 1998.

[20] I. Daubecies. *Ten Lectures on Wavelets*. CBMS-NSF regional conference series in applied mathematics. SIAM, Philadelphia, PA, 1992.

[21] S. R. Dey, A. I. Russell, and A. V. Oppenheim. Digital Pre-compensation for Faulty D/A Converters: The "Missing Pixel" Problem. In *Proceedings of IEEE ICASSP 2004*, Montreal, Canada, May 2004. IEEE.

[22] S. R. Dey, A. I. Russell, and A. V. Oppenheim. Pre-Compensation for Anticipated Erasures in LTI Interpolation Systems. *IEEE Transactions in Signal Processing*, 54(1):325–335, Jan 2005.

[23] C. Dick and F. Harris. FPGA signal processing using Sigma-Delta modulation. *IEEE Signal Processing Magazine*, 17(1):20–35, January 2000.

[24] R. J. Duffin and A. C. Schaeffer. A Class of Nonharmonic Fourier Series. *Trans. Amer. Math. Soc.*, 72(2):341–366, Mar 1952.

[25] J. Durbin. The fitting of time-series models. *Revue de l'Institut International de Statistique*, 28(3):233–244, 1960.

[26] A. Gersho. Asymptotically Optimal Block Quantization. *IEEE transactions on Information Theory*, 25(4):373–380, July 1979.

[27] V. K. Goyal, J. Kovacevic, and J. A. Kelner. Quantized Frame Expansions with Erasures. *Applied and Computational Harmonic Analysis*, 10:203–233, 2001.

[28] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\mathbb{R}^N$: analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, Jan 1998.

[29] R. Gray. Oversampled Sigma-Delta Modulation. *IEEE Transactions on Communications*, 35(5):481–489, May 1987.

[30] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, October 1998.

[31] J. Kovacevic, P. L. Dragotti, and V. K. Goyal. Filter Bank Frame Expansions with Erasures. *IEEE Trans. on Information Theory*, 48(6):1439–1450, June 2002.

[32] N. Levinson. The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction. *Journal of Mathematics and Physics*, 25(4):261–278, Jan 1947.

[33] S. G. Mallat and Z. Zhang. Matching Pursuits with Time-frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[34] M. Puschel and J. Kovacevic. Real, Tight Frames with Maximal Robustness to Erasures. In *Prococeedings Data Compression Conference (DCC) 2005*, pages 63–72, Snowbird, UT, March 2005. IEEE.

[35] R. Schreier and M. Snelgrove. Bandpass Sigma-Delta Modulation. *Electronics Letters*, 25(23):1560–1561, November 1989.

[36] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[37] N. T. Thao. Vector Quantization Analysis of $\Sigma\Delta$ Modulation. *IEEE Transactions on Signal Processing*, 44(4):808–817, April 1996.

[38] N. T. Thao and M. Vetterli. Reduction of the MSE in R-times oversampled A/D conversion $O(1/R)$ to $O(1/R^2)$. *IEEE Transactions on Signal Processing*, 42(1):200–2003, Jan 1994.

[39] N. T. Thao and M. Vetterli. Lower Bound on the Mean-Squared Error in Oversampled Quantization of Periodic Signals Using Vector Quantization Analysis. *IEEE Transactions in Information Theory*, 42(2):469–479, March 1996.

[40] G. Walker. On Periodicity in Series of Related Terms. *Proceedings of the Royal Society of London, Series A*, 131(818):518–532, Jun 1931.

[41] G. U. Yule. On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers. *Philosophical Transactions of the Royal Society of London, Series A*, 226:267–298, 1927.

[42] R. Zamir and M. Feder. On Lattice Quantization Noise. *IEEE Transactions on Information Theory*, 42(4):1152–1159, 1996. July.

[43] G. Zimmerman. Normalized Tight Frames in Finite Dimensions. In W. Haussmann, K. Jetter, and M. Reimer, editors, *Recent Progress in Multivariate Approximation*, volume 137 of *International Series of Numerical Mathematics*, pages 249–252. Birkhauser Verlag Basel, 2001.