Robust Compression of Multispectral Remote Sensing Data

by

Carlos R. Cabrera-Mercader

B.S., University of Puerto Rico (1991)
S.M., Massachusetts Institute of Technology (1993)

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1999

© Massachusetts Institute of Technology 1999. All rights reserved.

[Signature]

Department of Electrical Engineering and Computer Science
July 23, 1999

Certified by

David H. Staelin
Professor of Electrical Engineering
Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Departmental Committee on Graduate Students
Robust Compression of Multispectral Remote Sensing Data

by

Carlos R. Cabrera-Mercader

Submitted to the Department of Electrical Engineering and Computer Science
on July 23, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis develops efficient and robust non-reversible coding algorithms for multispectral remote sensing data. Although many efficient non-reversible coding algorithms have been proposed for such data, their application is often limited due to the risk of excessively degrading the data if, for example, changes in sensor characteristics and atmospheric/surface statistics occur. On the other hand, reversible coding algorithms are inherently robust to variable conditions but they provide only limited compression when applied to data from most modern remote sensors.

The algorithms developed in this work achieve high data compression by preserving only data variations containing information about the ideal, noiseless spectrum, and by exploiting inter-channel correlations in the data. The algorithms operate on calibrated data modeled as the sum of the ideal spectrum, and an independent noise component due to sensor noise, calibration error, and, possibly, impulsive noise. Coding algorithms are developed for data with and without impulsive noise. In both cases an estimate of the ideal spectrum is computed first, and then that estimate is coded efficiently. This estimator-coder structure is implemented mainly using data-dependent matrix operators and scalar quantization.

Both coding algorithms are robust to slow instrument drift, addressed by appropriate calibration, and outlier channels. The outliers are preserved by separately coding the noise estimates in addition to the signal estimates so that they may be reconstructed at the original resolution. In addition, for data free of impulsive noise the coding algorithm adapts to changes in the second-order statistics of the data by estimating those statistics from each block of data to be coded.

The coding algorithms were tested on data simulated for the NASA 2378-channel Atmospheric Infrared Sounder (AIRS). Near-lossless compression ratios of up to 32:1 (0.4 bits/pixel/channel) were obtained in the absence of impulsive noise, without preserving outliers, and assuming the nominal noise covariance. An average noise variance reduction of 12–14 dB was obtained simultaneously for data blocks of 2400–7200 spectra. Preserving outlier channels for which the noise estimates exceed three times the estimated noise rms value would require no more than 0.08 bits/pixel/channel provided the outliers arise from the assumed noise distribution. If contaminant outliers occurred, higher bit rates would be required. Similar performance was obtained for spectra corrupted by few impulses.

Thesis Supervisor: David H. Staelin
Title: Professor of Electrical Engineering
Acknowledgement

I wish to express my sincere gratitude to all those who helped me during the course of this work. Thanks to God for giving me strength to complete this thesis. Thanks to Prof. D. H. Staelin for his guidance of this research, his advice regarding both academic and professional matters, and his ever encouraging comments. Thanks to my thesis readers, Prof. J. S. Lim and Prof. G. W. Wornell, for enhancing the quality of this thesis through many suggestions. Thanks to the members of the Remote Sensing and Estimation Group: Phil Rosenkranz, Felicia Brady, Bill Blackwell, Junehee Lee, Fred Chen, Antonio Fuentes, and Vince Leslie for sharing their knowledge, and for all the fun during lunch break. Thanks to my family for their unconditional love, support, and encouragement.

I also wish to thank Ian Avruch, Dave and Kathy Backman, Karim and Maite Benitez, Felicia Brady, Hector and Monica Castillo, Maelo and Berti Castillo, Jorge Cortes, Marty Epstein, JuanMa and Songie Garcia, Abelardo and Isa Gutierrez, Seth Hall, Bill Kaliardos, Charlie Katz, Dominick Layfield, Sonia Miranda, Lelis and Wanda Nazario, Ruth Padilla, Mike and Susan Schwartz, and Noel Torres for their support and friendship.
# Contents

1 Introduction

1.1 Motivation and General Approach of the Thesis ..... 17
1.2 Problem Statement and Thesis Outline ..... 19

2 Background

2.1 Introduction ..... 27
2.2 Basics of Passive Remote Sensing ..... 27
2.3 Compression of Multispectral Data ..... 29
2.4 Atmospheric Infrared Sounder (AIRS) ..... 32
2.4.1 System Description ..... 33

3 Joint Linear Estimation and Transform Coding of Noisy Vectors ..... 37

3.1 Introduction ..... 37
3.2 Data Model and Codebook Definition ..... 39
3.3 The Case of Known Signal and Noise Statistics ..... 40
3.3.1 Optimization With Respect To Mean-Weighted-Square Error ..... 41
3.3.2 Estimator-Coder Structure of the Optimal Codebook ..... 48
3.3.3 Form of the Codebook for Noise-Adjusted Weighting ..... 49
3.3.4 Practical Implementation of the Optimal Codebook ..... 53
3.4 The Case of Unknown Signal Statistics and Known Noise Statistics ..... 55
3.4.1 Method for Empirical Codebook Design ..... 55
3.4.2 Codebook Design Using Maximum-Likelihood Estimates of the Signal Statistics ..... 58
3.4.3 Codebook Design Using Modified Maximum-Likelihood Estimates of the Signal Statistics ..... 69
3.5 Summary .............................................. 77

4 Calibration of Radiometric Data 79
4.1 Introduction ........................................ 79
4.2 Practical Linear Radiometric Calibration ........... 81
4.3 Linear MMSEE of Gain and Offset .................. 84
4.4 Approximate Stochastic Modeling of Gain and Offset Variations ........... 88
   4.4.1 Estimating the Statistics by Sample Averaging ......... 89
   4.4.2 Covariance Fitting by Approximate Stochastic Realization .... 91
4.5 Application: Statistical Characterization of Radiometric Calibration Errors on AIRS Data ........................................ 99
   4.5.1 Description and Pre-processing of the AIRS EM Test Data .... 100
   4.5.2 Estimates of AIRS Radiometric Calibration Error Statistics .... 107

5 Compression Algorithm for Multispectral Data in Additive Gaussian Noise 127
5.1 Introduction ........................................ 127
5.2 General Description of the Algorithm ................ 129
5.3 Setting the Codebook Overhead ...................... 131
   5.3.1 Approximating the Excess Distortion .................... 133
   5.3.2 Quantizing the Sample Mean ......................... 135
   5.3.3 Quantizing the Eigenvectors ......................... 136
   5.3.4 Quantizing the Variances of the Filtered NAPC ............ 137
   5.3.5 Operational Form of the Overhead Allocation Rule .......... 141
5.4 Bit Allocation Accounting for Codebook Overhead .......... 143
5.5 Tests with Simulated AIRS Data ...................... 146
   5.5.1 Details of the Tests ............................. 147
   5.5.2 Test 1: Comparison of Predicted and Actual Rate-Distortion Performance ..... 149
   5.5.3 Test 2: Rate-Distortion Performance with Codebook Overhead ...... 157
   5.5.4 Test 3: Rate-Distortion Performance with Different Noise Covariances 160
5.6 Coding Large Noise Estimates ........................ 167
5.7 Computational Burden .................................. 171
6 Compression Algorithm for Multispectral Data in Additive Gaussian and Impulsive Noise

6.1 Introduction ........................................ 179
6.2 The Mixture Noise Model .......................... 181
6.3 The Minimum Mean-Square Error Estimator ........ 182
6.4 Development of a Two-step Suboptimal Estimator ... 185
   6.4.1 The Detection Step .......................... 187
   6.4.2 Qualitative Discussion of the Detection Step ... 192
   6.4.3 The Estimation Step .......................... 194
6.5 Evaluation of the Two-step Estimator using AIRS Data ... 199
6.6 The Coding Algorithm ............................ 205
6.7 Summary and Additional Comments ................ 209

7 Conclusion ........................................ 211

7.1 Summary of the Thesis ........................... 211
   7.1.1 Main Contributions .......................... 213
7.2 Suggestions for Further Work .................... 216

A Proof of Proposition 3.1 .......................... 219

B Maximum-Likelihood Estimates of Gaussian Statistics from Noisy Observations .......... 221

C Proof of Proposition 3.2 .......................... 225

D Examples of Gain and Offset Covariances Estimated from AIRS EM Test Data ........ 233
List of Figures

1-1 Schematic diagram illustrating the process of generating the measurements. 20

2-1 Schematic diagram of the AIRS scan pattern. 34

3-1 Model of the measurement. 39
3-2 Structure of the codebook. 40
3-3 Structure of the optimal codebook. 50
3-4 Simplified representation of the optimal codebook when $W = C_n^{-1/2}$. 52
3-5 Method for empirical codebook design. 58
3-6 Simulation results for Example 1 in Table 3.1 using the MLDA. 64
3-7 Simulation results for Example 2 in Table 3.1 using the MLDA. 65
3-8 Simulation results for Example 3 in Table 3.1 using the MLDA. 66
3-9 Simulation results for Example 4 in Table 3.1 using the MLDA. 57
3-10 Amplitude of the MMSEE filtering matrix in NAPC space for Example 1 in Table 3.1 using the MLDA. 68
3-11 Estimated NAPC variance distribution for Example 1 in Table 3.1. 69
3-12 Empirical NAPC variance distributions in the absence of a signal as a function of sample size. 71
3-13 Simulation results for Example 1 in Table 3.1 using MMLDA. 73
3-14 Simulation results for Example 2 in Table 3.1 using MMLDA. 74
3-15 Simulation results for Example 3 in Table 3.1 using MMLDA. 75
3-16 Simulation results for Example 4 in Table 3.1 using MMLDA. 76

4-1 Schematic diagram of a total-power radiometer. 81
4-2 AIRS scan timing diagram. 101
4-3 Examples of data from “blinker” detectors on the AIRS EM. 102
4-4 Examples of data from “blinker” detectors on the AIRS EM. .............. 103
4-5 Examples of data from one of the AIRS EM test runs (file 640.dat). .... 104
4-6 Examples of data from one of the AIRS EM test runs (file 640.dat). .... 105
4-7 Examples of data from one of the AIRS EM test runs (file 640.dat). .... 106
4-8 Illustration of the data in Figure 4-5 after the observations of the cold target were corrected. ........................................... 107
4-9 Two possible explanations of the fluctuations of the observations of the hot target are illustrated. ........................................... 110
4-10 Smoothed estimates of the temperature of the hot target during both test runs.111
4-11 Estimated relative contributions of gain variations, offset variations, and thermal noise to variations in the test data from 754 channels of the AIRS EM. 114
4-12 Illustration of the use of the MMSE filter to obtain gain and offset estimates. 115
4-12 Estimated rms calibration error on individual spectral channels obtained using the MMSE filter of length fourteen scans. ................................. 118
4-14 Histograms of the ratio of rms calibration error to estimated NER @$301$ K for channels on four different detector modules. ................................. 119
4-15 Illustration of the estimated correlation between calibration errors for 754 channels in modules M-1, M-4, M-10, and M-12. ................................. 121
4-16 Periodograms of the calibration errors on two channels of the AIRS EM. 122
4-17 Periodograms of the calibration errors on four channels of the AIRS EM. 123
4-18 Examples of spectral channels with significant systematic variation in the estimated calibration accuracy as a function of scene view index. ........... 124

5-1 Block diagram of the adaptive NAPC-based algorithm. ..................... 130
5-2 Illustration of a block of multispectral data. .................................. 131
5-3 Fractional increase in quantizer distortion due to variance mismatch for Lloyd-Max quantizers designed for a zero-mean Gaussian input and various bit allocations. .................................................. 138
5-4 Signal-to-noise ratio of the NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the nominal AIRS noise covariance matrix.150
Eigenvectors corresponding to the first nine NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the nominal AIRS noise covariance matrix. .................................................. 151

Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. ................................. 153

Histograms of the first nine filtered NAPC evaluated using a set of 2400 noisy AIRS spectra. .......................................................... 154

Predicted and actual rate-distortion characteristics of the codebook designed with "knowledge" of the signal statistics. .................................................. 155

Filter amplitude estimated from data blocks of varying sizes. ..................... 157

Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. ......................... 159

Eigenvectors corresponding to the first nine NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the covariance matrix $C_{nn}$ described in page 160, with $\rho = 0.3$ and $S$ set equal to the nominal AIRS rms noise matrix for a 250 K target. ........................................................ 163

Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. ......................... 165

Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. ......................... 166

Model of the measurement incorporating a mixture of typical noise and impulsive noise. .......................................................... 181

Maximum amplitude of the cross-correlation coefficient of the prediction errors for all the spectral channels of AIRS. .................................................. 200

Root-mean-square prediction errors in the absence of impulsive noise for two different typical noise covariance matrices. ................................. 201

Operating characteristic of the detection step for singly-corrupted spectra. 203

Fractional increment in noise-adjusted distortion (all channels) caused by an impulse of amplitude equal to six times the rms prediction error. .............. 205

Example of estimator performance for a multiply-corrupted spectrum. .......... 206

Example of estimator performance for a multiply-corrupted spectrum. .......... 207
6-8 Block diagram of the coding algorithm for data corrupted by additive Gaussian noise and impulsive noise. ............................................ 268

D-1 Estimate of the joint covariance sequence of the gain and offset on channel 42.234
D-2 Estimate of the joint covariance sequence of the gain and offset on channel 140. ................................................................. 235
D-3 Estimate of the joint covariance sequence of the gain and offset on channel 815. ................................................................. 236
D-4 Estimate of the joint covariance sequence of the gain and offset on channel 1042. ............................................................... 237
D-5 Estimate of the joint covariance sequence of the gain and offset on channel 2089. ............................................................... 238
D-6 Estimate of the joint covariance sequence of the gain and offset on channel 2253. ............................................................... 239
## List of Tables

2.1 Spectral coverage of the twelve detector array modules of the AIRS infrared spectrometer. .................................................................................................................................................................................. 35

3.1 Parameters of the simulations used to evaluate the MLDA. The vector dimension was arbitrarily set to \( N = 100 \). The index \( i \) ranges from 1 to \( N \). .................................................................................................................. 63

3.2 Average absolute inner product between the first ten true and empirical NAPC basis vectors estimated from different sample sizes for Example 1 in Table 3.1. For the remaining NAPC the average absolute inner product never exceeded a value of 0.16. .......................................................................................................................... 68

4.1 Computational paradigm of calculating on-line the gain and offset estimates given by equation 4.21 for AIRS. .......................................................................................................................... 88

4.2 Estimated average decrease in calibration error variance (dB) using empirical MMSE calibration filters of different lengths. ........................................................................................................ 116

4.3 Estimated average decrease in calibration error variance (dB) using empirical MMSE calibration filters of different lengths. ........................................................................................................ 116

4.4 Mean and standard deviation of the ratio of rms calibration error to estimated NER at 301 K for each of four detector modules. ........................................................................................................ 120

4.5 Estimated average decrease in calibration error variance (dB) using zero-phase triangular FIR filters of different lengths. ........................................................................................................ 125

5.1 Predicted and actual quantizer distortion for the nine filtered NAPC distributions shown in Figure 5-7. ........................................................................................................ 152

5.2 Percent of the bit rate spent on codebook overhead as a function of block size and bit rate. ........................................................................................................ 158
5.3 Maximum average bit rate (bits/pixel/channel) required to code the noise estimates for AIRS as a function of the block size \( K \) and the threshold \( \xi \). \hspace{1cm} 171

5.4 Maximum average bit rate (bits/pixel/channel) required to code the noise estimates for AIRS as a function of the block size \( K \) and the threshold \( \xi \). \hspace{1cm} 171

5.5 Approximate operation count for the codebook design portion of the algorithm. \hspace{1cm} 172

5.6 Approximate operation count for the codebook application portion of the algorithm. \hspace{1cm} 173

5.7 Approximate computational burden of the coding algorithm when it is applied to AIRS data, assuming the noise covariance is full. \hspace{1cm} 174

5.8 Approximate computational burden of the coding algorithm when it is applied to AIRS data, assuming the noise covariance is diagonal. \hspace{1cm} 174

6.1 Number of operations (multiplications and additions) per equation for the detection algorithm. \hspace{1cm} 192
Chapter 1

Introduction

1.1 Motivation and General Approach of the Thesis

Multichannel data from airborne or spaceborne earth sensors are used for a variety of applications in areas such as environmental monitoring, weather forecasting, land use planning and mapping, and natural resource management. Applications in environmental monitoring include, for example, profiling atmospheric temperature and constituents, studying the structure of clouds and storms, monitoring land and ocean surface temperatures, and measuring wind over ocean. The area of natural resource management applications includes forestry, agriculture, water quality monitoring, and wildlife habitat management. Two attributes of a multispectral data set that determine its usefulness in many of these applications are its spatial resolution and spectral resolution. The former refers to the physical distance between adjacent observations at the earths' surface. The latter relates to the number of individual spectral bands or channels measured in a given spectral region. High spatial resolution is desirable for distinguishing physically small features such as roads or vegetation types in the case of land monitoring, or clouds in atmospheric sensing. High spectral resolution is desirable for isolating narrow line features in the absorption spectra of atmospheric constituents.

Increasing the spatial or spectral resolution of multispectral data requires increasing the rate at which the data are collected by the sensor. Other factors that increase the data rate are radiometric precision, area collection, sharpening, and stereo imagery [1]. For spaceborne sensors a higher data collection rate requires a downlink of higher bandwidth. For airborne sensors the requirement is higher onboard storage capacity.
It is foreseen by experts that future spaceborne remote sensors will be limited by downlink capacity [1, 2]. Also, experimenters' willingness to archive and access accumulated data may be limited by data set volume. For these reasons there is a strong interest in data compression techniques within the remote sensing community [3, 4, 5, 6, 2, 7, 8, 9]. While reversible or lossless coding techniques may be the preferred choice of many within that community, it is well known that only modest compression ratios can be obtained that way [10, 9, 11]. Recent studies have explained the observed limits of lossless compression of multispectral remote sensing data in terms of the entropy of the noise [12, 9], suggesting the limited need for requiring reversibility for that kind of data. Significantly higher compression can be obtained using non-reversible or lossy coding techniques at the expense of distorting or even discarding some of the data. The obvious disadvantage of lossy coding is that the original data can not be recovered exactly. In addition, there are practical difficulties associated with typical lossy coding algorithms:

1. The data may be excessively degraded if the coding algorithm is not flexible enough to recognize changes in the statistical behavior of the data.

2. Data outliers may not be reproduced accurately.

3. Instrument malfunctions, e.g. detector dropouts or amplifier failure, may cause serious data degradation depending on the coding algorithm [1].

Issues such as these should be addressed in the design of a robust lossy coding algorithm. In spite of these complications, the need for higher than lossless compression warrants the consideration of non-reversible coding techniques for remote sensing systems.

This thesis seeks to provide a compromise between traditional lossy and lossless coding by treating the problem of compressing multispectral remote sensing data as a noisy source coding problem. This approach explicitly recognizes the presence of noise in the data, and the fact that the signal, defined here as the radiance spectrum which would be obtained using an ideal, noiseless sensor, can not be recovered exactly even if the data are preserved intact. Since the signal is the quantity of interest, the error or distortion in the compressed data is not measured relative to the original data but rather relative to the signal. To implement this paradigm we rely on an estimator-coder structure: first the signal is estimated, and then its estimate is coded. Information and quantization theory provide the motivation for this structure [13, 14, 15, 16]. If designed appropriately, a coding algorithm of that form
can achieve good compression by accurately preserving variations in the data due mostly to signal, and reproducing less accurately, or discarding, variations due to noise. From this perspective, lossless coding and traditional lossy coding are both comparatively inefficient. Lossless coding achieves the optimal (nonzero) distortion at the expense of a relatively high data rate; it does not take into account the cost of achieving the optimal distortion. On the other hand, traditional lossy coding reduces the data rate inefficiently since it does not necessarily discriminate between signal and noise when deciding what to preserve.

The thesis focuses on developing coding algorithms for data from high-spectral-resolution sensors with hundreds or thousands of spectral channels. Two examples of such sensors are the 224-channel Airborne Visible and Infrared Imaging Spectrometer (AVIRIS) and the 2378-channel Atmospheric Infrared Sounder (AIRS). For multispectral data of even moderate spectral resolution it is known that the inter-channel correlations tend to be much stronger than the spatial correlations [1, 9]. This contrasts with visual color imagery, where spatial correlations afford most of the potential for data compression. The coding algorithms developed in this thesis seek to maximally exploit inter-channel correlations. For simplicity, the treatment is limited to coding algorithms consisting mostly of matrix operators and scalar quantization as opposed to full vector quantization. In an effort to address some of the potential problems associated with non-reversible coding techniques, the coding algorithms developed here provide some protection against changing signal statistics, e.g. due to an unanticipated spectral line, and outlier data characterized by unusually large deviations in a few spectral channels. Changes in instrument characteristics, e.g. gain and offset drift, are addressed by appropriately calibrating the data prior to coding. No formal attempt is made to address the possibility of instrument failures.

1.2 Problem Statement and Thesis Outline

The model for multispectral data adopted in this thesis is illustrated in Figure 1-1. The top schematic shows in simple form how the signal of interest is sensed and calibrated to produce a corresponding measurement. The combined effect of these processes is modeled as an independent noise source added to the signal. The signal, the noise, and the measurement are all vectors with identical numbers of spectral channels. The noise source accounts for sensor noise, calibration error, and, possibly, impulsive noise. In the absence of impulsive
Figure 1-1: Schematic diagram illustrating the process of generating the measurements.

noise, the noise is assumed to have a known jointly Gaussian distribution. When impulsive noise is present, the total noise is taken to be a mixture of jointly Gaussian noise and a higher-variance noise which is added with small probability. No specific distribution is assumed for the impulsive noise component.

The objective of the thesis is to develop efficient and robust coding algorithms for high-spectral-resolution multispectral data. Coding algorithms are developed for two different sets of assumptions. In the first case the measured spectra are assumed to be free of impulsive noise. As mentioned above, the non-impulsive noise is modeled as a jointly Gaussian random vector with known statistics. The signal is also modeled as a random vector but its statistics are left unspecified; they are estimated by the algorithm. In the second case the data may be corrupted by impulsive noise. The second-order statistics of the signal are assumed to be known in this case. The non-impulsive noise component is modeled as in the first case. No assumptions are made about the statistics of the impulsive noise, except that the noise itself occurs relatively infrequently.

Both coding algorithms have an estimator-coder structure. The basic building blocks employed to implement the algorithms are matrix operators and scalar quantization. Specifically, the algorithms make use of data-dependent statistical linear transforms as opposed to sinusoidal transforms, e.g. the discrete cosine transform (DCT), which generally are not
appropriate for exploiting the correlation structure of arbitrary vector data, and, in particular, of multispectral data. A mean-square "noise-adjusted" error criterion, where the error vector is normalized by a square-root of the (non-impulsive) noise covariance matrix, is used to quantify the deviation of the compressed data from the desired (noiseless) signal. This choice of distortion criterion arose from insights gained from the research itself and is discussed later. To the extent possible the algorithms are designed so that their performance is optimal or approaches that of an optimal system.

The thesis is organized as follows. Chapter 2 reviews some basic concepts of passive remote sensing and gives a high-level description of the NASA 2378-channel Atmospheric Infrared Sounder (AIRS). Simulated AIRS data are used as a test case for the coding algorithms developed in the thesis.

Chapter 3 treats the problem of optimal transform coding of noisy data, which provides the foundation upon which our coding algorithms are built. Throughout that chapter both the signal and noise components are assumed to be independent Gaussian random vectors. The codebooks considered there, shown in Figure 3-2, consist of an encoder matrix, a bank of scalar quantizers, and a decoder matrix. The design of the codebook is considered for two distinct cases: 1) when the statistics of both the signal and noise are known, and 2) when the statistics of the signal are not known.

In the case of known signal and noise statistics the codebook is optimized with respect to mean-weighted-square error. To make the problem tractable we impose an orthogonality constraint on the decoder matrix. In spite of the imposed constraint the optimal codebook is shown to have many of the properties implied by rate-distortion theory. Specifically, the optimal encoder matrix (see Figure 3-3) is the product of the linear minimum mean-square error estimator (MMSEE) matrix, the weight matrix, and a Karhunen-Loève (KL) transform. The optimal decoder matrix inverts the KL transform and the weight matrix. The optimal quantizers are Lloyd-Max quantizers. Although the same result was obtained previously by Malvar [17], our assumptions and derivation are different.

In Section 3.3.3 we derive and analyze the optimal codebook structure for the case in which the weight matrix is a square-root of the inverse noise covariance matrix. In that case the transform computed by the optimal codebook is the so-called noise-adjusted principal components (NAPC) analysis [18]. Accordingly, we refer to that distortion criterion as noise-adjusted distortion. The NAPC compact the signal energy per unit noise by selecting
uncorrelated linear combinations of the noisy data having maximal signal-to-noise ratio. Other interesting properties of NAPC are discussed as well in that section. NAPC reappear in the thesis a number of times since they are a natural representation for signals corrupted by uncorrelated additive noise.

In the case of unknown signal statistics the same optimization approach can not be used to design the codebook. The codebook must be designed using only samples of the noisy data, and knowledge of the noise statistics. The general design approach pursued in this case consists of estimating the statistics of the signal and the noisy data, and implementing the optimal codebook corresponding to those estimates of the statistics. It is shown in Appendix C that if the estimates are consistent, a codebook designed this way is asymptotically optimal as the number of data samples employed for its design approaches infinity. Although a similar result is known for vector quantizers under different assumptions [19], we do not know of a previous proof applying specifically to the transform coders considered in this work.

In addition to establishing conditions for asymptotically optimal performance, we evaluate the performance of specific codebooks at finite data-block lengths. Section 3.4.2 details the design of codebooks using Maximum-Likelihood (ML) estimates of the statistics, and evaluates their rate-distortion performance for some simple examples through numerical simulation. The ML estimates for Gaussian signal and noise are derived in Appendix B, where an interesting connection is found between the ML estimates and NAPC analysis. Namely, in NAPC space the signal covariance is estimated by subtracting unity from the eigenvalues of the sample covariance of the noisy data, and setting to zero any negative eigenvalues obtained in the process. Unfortunately, the codebooks designed using ML estimates are found to have poor performance even when the number of data samples used to design the codebook is several times the vector dimension. In a simulated example with low signal-to-noise ratio this occurred even when the number of samples was eight times that dimension.

In Section 3.4.3 an *ad hoc* modification of the ML estimates is proposed which leads to codebooks with improved rate-distortion characteristics. The ML estimates are modified in such a way that a better estimate of the linear MMSEE matrix is obtained. The modification depends on the (finite) number of data samples used to obtain the ML estimates. Since the modified estimates lead to a better estimate of the linear MMSE matrix, they can be
beneficial for noise filtering of vector data whenever the statistics of the signal are not known \textit{a priori} and a limited amount of noisy data is available for estimating those statistics.

Chapter 4 is dedicated to the problem of radiometric calibration, which refers to the conversion of the raw data collected by a radiometer into estimates of the radiance spectrum of the observed scene. The subject is relevant in the context of the thesis because the models and coding techniques considered here are intended for calibrated data. One motivation for this choice is that, whether or not data compression is performed, ultimately one wants to obtain from the raw data the best possible estimate of the ideal spectrum, i.e. one measured by an ideal, noiseless sensor that can be calibrated perfectly. Because this goal implies some form of calibration, it is simpler to design a coding algorithm by treating calibration separately as a preprocessing step. In addition, as argued later, the independent additive noise model of Figure 1-1 is more appropriate for calibrated data. In practice much of the risk to the user of compressed data arises from shifts in instrument characteristics revealed by calibration so optimum use of calibration data is important.

Section 4.2 describes an instrument model in terms of receiver gain and offset parameters, and reviews standard methods of estimating those parameters from periodic observations of two calibration targets. In Section 4.3 we develop a variation using linear MMSEE of the receiver gain and offset assuming these can be modeled as stationary random processes. Approximations needed to use these estimates for calibration of real data are discussed in Section 4.4.

In Section 4.5 we perform a statistical characterization of the calibration accuracy of AIRS data. Basically, this is an attempt to build the additive noise model of Figure 1-1 for AIRS, i.e. obtaining estimates of the mean and covariance matrix of the “noise” in the calibrated data. Preliminary test data from the AIRS Engineering Model were used for this purpose. The calibration techniques described in sections 4.2 and 4.3 were applied to those data. Unfortunately, the results of that exercise are of limited use for a number of reasons detailed there. In particular, the results should not be taken to be accurate estimates of the calibration accuracy of AIRS. Nonetheless, the estimated error statistics together with nominal noise statistics provided by the AIRS Science Team are used as a guide to generate reasonable test cases for the coding algorithms.

In Chapter 5 we develop a block-adaptive coding algorithm for data free of impulsive noise. The algorithm adapts to the signal statistics (mean and covariance) of each data-
block while the noise statistics are assumed to be relatively stationary. In practice the noise statistics would be obtained from accumulated calibration data, which would be transmitted intact to the user. The algorithm employs the codebook structure derived in Chapter 3 for the noise-adjusted distortion criterion. As indicated before, NAPC analysis is the optimal transform for that criterion. To use the codebook in an adaptive scenario we apply the codebook design algorithm developed in Section 3.4.3, which uses the modified ML estimates of the statistics of the signal and the noisy data. Two additional components are developed to make the algorithm better suited for adaptive use. In Section 5.3 an explicit rule is detailed for setting the codebook overhead for each data-block. Then in Section 5.4 the well-known log-variance bit allocation is modified to account for the codebook overhead. The combination of the overhead allocation rule and the modified bit allocation rule allows one to design a codebook for a finite data-block without exceeding a specified bit rate. In Section 5.5 three tests of the coding algorithm are conducted using simulated AIRS data. The tests address the following issues:

1. Comparing the actual and predicted rate-distortion performance.

2. Assessing the cost of codebook overhead.


Finally, in Section 5.6 we describe a strategy for separately preserving individual spectral channels in which the estimated noise, which normally would be discarded, is unusually large. Those channels are preserved by individually quantizing the noise estimate in addition to the estimated signal. This protects against the possibility of excessively degrading or discarding data outliers which later may be found to be of scientific value.

A coding algorithm for multispectral data corrupted by additive Gaussian noise and impulsive noise is developed in Chapter 6. In this case it is assumed that the second-order statistics of both the signal and the Gaussian noise component are known. Thus, the coding algorithm is not adaptive. The algorithm is also based on the NAPC-based codebook derived in Section 3.3.3, except that a different signal estimator is used instead of the linear MMSEE in order to minimize the impact of impulsive noise. The new signal estimator is developed using a mixture model for the total noise, as described in Section 6.2. Section 6.3 discusses the infeasibility of the MMSEE for the mixture noise model and provides motivation for a two-step suboptimal estimator developed in Section 6.4. The two-step estimator detects
channels which are likely corrupted by impulsive noise, and estimates the signal in a linear MMSE sense without using the corrupted channels. Section 6.5 evaluates the performance of the two-step estimator on AIRS data. Section 6.6 describes the complete coding algorithm.

A summary of the thesis and suggestions for further research are provided in Chapter 7.
Chapter 2

Background

2.1 Introduction

This chapter contains brief discussions of three topics of interest in the context of this thesis. Section 2.2 discusses the basic principles of passive remote sensing. A partial review of techniques used for compression of multispectral data is given in Section 2.3. Section 2.4 gives a brief description of the NASA Atmospheric Infrared Sounder (AIRS). Simulated AIRS data are used later as a test case for two coding algorithms developed in this thesis.

2.2 Basics of Passive Remote Sensing

Remote sensing techniques can be classified as passive or active according to the sources of electromagnetic radiation which they measure [20]. Active sensors generate an electromagnetic signal themselves and detect the echoes produced whenever part of the signal is reflected by the terrestrial atmosphere or surface. Passive sensors, on the other hand, measure radiation emitted by the terrestrial atmosphere and surface, and radiation emitted by celestial bodies and reflected by the earth. The most commonly used passive sensing techniques are radiometry and interferometry, which can be regarded as duals of each other. A radiometer measures the intensity, or radiance, of the incoming signal in one or multiple spectral channels over some portion of the electromagnetic spectrum. An interferometer indirectly measures the radiance spectrum over a range of frequencies by computing the interferogram or autocorrelation sequence of the incoming signal.

The theory of radiative transfer provides the physical basis for interpreting data observed
by passive remote sensing instruments [20]. Specifically, the equation of radiative transfer relates the upwelling radiance observed at a given altitude to the physical properties of the terrestrial surface and the intervening atmosphere. The appropriate form of the equation of radiative transfer that applies to a given set of atmospheric conditions and a given frequency range depends partly on whether there is significant scattering of radiation by the atmosphere or by a rough surface. For instance, to correctly interpret radiances observed at visible and infrared frequencies it is necessary to take into account scattering by clouds when they are present in the observation path. In contrast, at some microwave frequencies many types of clouds are fairly transparent so that scattering by clouds can often be ignored when interpreting such data.

In general, radiative transfer is simpler in the absence of scattering. Then the dominant effects are emission and absorption of radiation by the atmosphere and surface, and possibly reflection of cosmic background and solar radiation by the (smooth) surface [20]. Due to the combination of emission and absorption, the upwelling radiance observed by a remote sensor at the top of the atmosphere will have been emitted predominantly around some altitude in the atmosphere. At frequencies at which the atmosphere is relatively transparent the observed radiance is that emitted mainly by the surface and the adjacent atmosphere. Those “window” frequencies are ideal for monitoring human activity and geophysical processes at the surface of the earth. At frequencies in the neighborhood of absorption lines of atmospheric constituent molecules, e.g. O₂, H₂O, and CO₂, the atmosphere becomes opaque and the observed radiance will have been emitted higher in the atmosphere. By positioning the spectral channels of a radiometer in regions of varying degrees of absorption it is possible to obtain a sounding effect where radiances in different channels correspond to different levels in the atmosphere. This sounding capability enables the retrieval of both atmospheric temperature and composition profiles.

Many techniques for retrieving geophysical parameters, e.g. atmospheric temperature and composition profiles, from passive remotely sensed data exploit directly the physical relation established by the equation of radiative transfer [21, 22]. Other techniques use a priori statistics in combination with the physics of radiative transfer, for example [23, 24]. A third class of retrieval techniques are those based primarily on statistics, for example [25, 26].
2.3 Compression of Multispectral Data

Over the last few years there has been a lot of interest in compression of multispectral remote sensing data. This interest is in response to the tremendous increase in the amount of data generated by modern remote sensing systems in comparison to their predecessors. For example, the High Resolution Infrared Sounder (HIRS) that currently supports the efforts of the National Weather Service has a data rate of about 2.88 Kbps. By the end of the year 2000, the HIRS will be greatly surpassed in spectral resolution by the NASA Atmospheric Infrared Sounder (AIRS), which will have a data rate of 1.3 Mbps [27]; a 500-fold increase in data rate. For new and future systems the use of data compression would be beneficial in several ways [12, 7]:

1. to make efficient use of the data link from a spacecraft to a ground station, which in many cases must service multiple sensors,

2. to reduce the volume of data archives,

3. to reduce the time and cost of data distribution over a network.

Below we review briefly some of the more recent contributions in the area of multispectral data compression. Both lossless and lossy coding techniques are reviewed.

Lossless coding techniques are still the preferred choice within the remote sensing community. They afford the opportunity of recovering the original data exactly but at the cost of limited compression capability. Compression ratios typically do not exceed 3:1 [10, 9, 28, 11]. Most lossless coding algorithms for single or multispectral image data consist of a prediction step followed by efficient coding of the prediction residuals, for example [9, 28, 11]. Ideally the prediction residuals should be statistically independent to avoid redundancy. In practice the prediction of each pixel is usually obtained from a linear combination of other pixels adjacent in space or, for multispectral data, pixels on adjacent spectral channels. The prediction may be optimized in a mean-square error sense or it may be obtained by simple differentiation. In the former case the prediction is rounded-off so that the prediction error is integer-valued. For typical multispectral data it has been observed that predictors which exploit inter-channel correlations give higher compression than those which use spatial correlations exclusively [9]. Methods used to code the prediction residuals include Rice algorithms [9, 11], Huffman coding [9], and arithmetic coding [28].
Roger and Arnold recently explained the limited compression typically obtained using lossless coding in terms of the entropy of the additive noise in multispectral data [12]. Using simple probabilistic models for the noise they obtained upper bounds on compression which agree with practical experience. For example, for a single band of the Landsat Thematic Mapper (TM) they bound lossless compression at 4:1.

Lossy coding of multispectral data has received considerable attention recently. Proposed algorithms generally fall in one of three groups: 1) hybrid predictive coders, e.g. [4, 7], 2) transform coders, e.g. [6, 7, 2, 8], or 3) vector quantizers, e.g. [5].

The first group of algorithms uses prediction techniques, like differential pulse-code modulation (DPCM) and linear prediction, combined with transform coding or vector quantization. For example, Gupta and Gersho proposed coding a few of the spectral bands in a multispectral image using vector quantization for each band and then reconstructing the remaining bands from one of the quantized bands by means of linear or nonlinear prediction [4]. Testing their best algorithm on 7-band images from the Landsat TM they obtained a compression ratio of 22:1 with peak signal-to-noise ratios\(^1\) (PSNR) of 32–44 dB for each band. In those tests each spectral band was divided into 4 × 4 squares. Two bands were quantized and the remaining five bands were predicted. Another hybrid algorithm proposed by Aboussaleman et al. uses DPCM between adjacent spectral bands and codes the residual images in 8 × 8 blocks using a two-dimensional discrete cosine transform (DCT) followed by entropy-constrained trellis-coded quantization (ECTCQ) [7]. For a 140-band (out of a total of 224 bands) AVIRIS image 76:1 compression was obtained with a PSNR of 40 dB.

Transform-based coding algorithms for multispectral data use fast sinusoidal transforms like the DCT as well as data-dependent transforms like the Karhunen-Loève transform (KLT) [29], and combinations of both. Hoffman and Johnson used a variant of the KLT known as noise-adjusted principal components (NAPC) [18] to decorrelate a multispectral image along the spectral dimension. The transformed representation was truncated and uniformly quantized so that the rms coding residual for each band matched the corresponding rms noise. For AVIRIS images typical compression ratios of 50:1 were estimated. Aboussaleman et al. used a three-dimensional (two spatial dimensions and one spectral dimension) DCT to code AVIRIS imagery [7]. An image was divided into cubes of 8 × 8 × 8 pixels

---

\(^1\)Peak signal-to-noise ratio is defined as the ratio in dB of the square of the maximum data amplitude to the variance of the coding residual.
which were transformed using the three-dimensional DCT. The sequence of coefficients for each frequency was coded using ECTCQ. For a 140-band AVIRIS image they obtained 79:1 compression with a PSNR of 40 dB. Saghri et al. propose first using a KLT to produce a set of decorrelated images and coding each image using DCT-based JPEG [2]. For imagery from the 16-channel M7 sensor they observed "visually lossless" compression ratios of about 40:1.

Tretter and Bouman proposed several transform-based algorithms which for long image blocks approximate the performance of the optimal three-dimensional KLT [8]. Assuming a jointly stationary Gaussian process model they showed analytically that computing a spatial two-dimensional block frequency transform on each spectral band followed by a one-dimensional KLT across spectral bands for each spatial frequency is asymptotically equivalent in performance to the optimal the three-dimensional KLT. They also showed that when the covariance of the data is separable, that procedure is equivalent to the more commonly used procedure of computing first a one-dimensional KLT across bands followed by a two-dimensional frequency transform, as proposed in, for example, [2]. Although theoretically superior in performance, a practical disadvantage of the first procedure is that multiple KLT's, one for each spatial frequency, need to be computed. For example, if an \(N\)-band multispectral image is divided into cubes of \(8 \times 8 \times N\) pixels for the purpose of coding, then sixty-four KLT's would have to be computed. For sensors with hundreds of spectral channels the computational burden would be too high.

Vector quantization (VQ) has also been used to code multispectral data. To reduce the encoding and storage complexity some kind of structural constraint is usually imposed on the quantizer. In one example Gupta and Gersho used a multistage vector quantizer with greedy bit allocation to code Landsat TM imagery [5]. A 4-band input image was divided into cubes of \(4 \times 4 \times 4\) pixels which were first quantized with a single VQ stage. Subsequent VQ stages refined the quantized version of the input vector by quantizing the residual error resulting from the previous VQ stage. The number of VQ stages and, consequently, the number of bits used to quantize each 64-pixel vector was selected using a greedy algorithm that locally optimizes the rate-distortion trade-off. The VQ stages were designed using the Generalized Lloyd Algorithm and a training set of five Landsat TM images, and they were kept fixed when applying the multistage VQ to other data. For two 4-band Landsat TM test images, 25:1 compression was obtained with PSNR's of 35-43 dB for each band.
In a review paper [30] Gupta and Gersho compared the performance of the multistage VQ algorithm [5], the feature predictive VQ algorithm [4], and a KLT/DCT algorithm like the one in [2]. In tests with 4-band Landsat TM images the transform-based algorithms outperformed the VQ based algorithms at bit rates above 0.15 bits/pixel. This better performance of the KLT/DCT algorithm was partly due to the fact that smaller input cubes (4 x 4 x 4 vs. 8 x 8 x 4) were used for the VQ algorithms to keep the complexity at a reasonable level.

The coding algorithms developed later in this thesis fall in the group of transform-based algorithms. As one might expect, there are significant similarities between our algorithms and many of the algorithms mentioned above, particularly those which employ data-dependent transforms. The coding algorithm of Hoffman and Johnson [6] is closest to ours since it also uses the NAPC transformation along the spectral dimension to achieve data compression. The differences between their algorithm and ours are discussed in Section 5.2 (page 129). The motivation for using a data-dependent transform instead of the DCT along the spectral dimension is the highly irregular correlation structure which characterizes multispectral data sets with a large number of channels [1]. The DCT, which is known to be quasi-optimal for stationary first-order Markov processes with high inter-sample correlation [31], can not efficiently exploit the non-stationary correlation structure of high-spectral-resolution remotely-sensed data.

2.4 Atmospheric Infrared Sounder (AIRS)

The National Aeronautics and Space Administration’s Atmospheric Infrared Sounder (AIRS) is a high-spectral-resolution infrared spectrometer that will operate aboard the EOS-PM1 polar-orbiting platform scheduled for launch in the year 2000. AIRS is part of a three-instrument sounding suite, including the Advanced Microwave Sounding Unit (AMSU) and the Microwave Humidity Sounder (MHS). AIRS itself will provide spectral observations of the earth’s atmosphere and surface over the 4.2 µm and 15 µm carbon dioxide bands, and the 6.3 µm water vapor band, with a total of 2378 channels. Those data will be used to retrieve various geophysical parameters such as atmospheric temperature and humidity profiles, total precipitable water vapor, cloud fractional coverage, and ocean and land surface temperatures. The design of AIRS specifically responds to a study by the World Meteo-
rological Organization which determined that global temperature and humidity soundings with radiosonde accuracy are required to significantly improve weather forecasting [32]. That is equivalent to temperature profiles with a 1 K rms accuracy in one-kilometer-thick layers, and humidity profiles with an accuracy of twenty percent in the troposphere. That requirement can not be met by the satellite sounding system composed of the High Resolution Infrared Sounder (HIRS) and the Microwave Sounding Unit (MSU), which has now been in operation for about two decades.

In the context of this thesis AIRS data will be used both as a motivation and as a test case for several non-reversible data compression algorithms to be described in a subsequent chapter. With such a large number of spectral channels, AIRS is a particularly good example to illustrate the potential benefits of nonreversible compression of high-spectral-resolution remote sensing data. Although in the case of AIRS the spacecraft can handle its data rate of 1.3 Mbps, data compression would be desirable to reduce the cost of data storage and distribution once the data have been transmitted to earth.

2.4.1 System Description

The AIRS instrument consists of an infrared (IR) grating spectrometer, a visible/near-infrared (VIS/NIR) photometer, and various other subsystems which enable their use in space for the purpose of remote sensing of the earth’s atmosphere and surface [27, 32]. These include onboard radiometric, spectral, and photometric calibration targets; a scan mirror and optics which direct radiation from the scene and the calibration targets to the spectrometer and the photometer; a cryogenic cooler that controls the temperature of the IR focal plane; signal and data processing electronics that perform integration, digitization, 1/f noise mitigation, radiation circumvention, gain and offset correction, etc. Here we shall provide a brief description of the IR spectrometer and some of the related subsystems.

AIRS will measure the upwelling radiation from a polar orbit at an altitude of 705 km. As illustrated in Figure 2-1, the 1.1° instantaneous field of view is scanned by ±49.5° about nadir every 8/3 seconds. A two-speed scanning mirror scans through ninety scene footprints in two seconds, and completes a full revolution in the remaining 2/3 seconds making four successive views of cold space, and single views of a 310 K radiometric calibrator, a spectral reference target, and a photometric calibrator. The signal from all those sources is directed optically to the IR spectrometer.
The IR spectrometer consists primarily of an entrance pupil with eleven individual entrance slits, a four-mirror telescope, an echelle diffraction grating, and the focal plane assembly. Incoming radiation is first collimated by the telescope and then dispersed by the diffraction grating onto seventeen linear arrays of detectors located in the focal plane. The spectral separation is achieved by the combination of the diffraction grating and optical band-pass filters placed at each entrance slit and each detector array. Each filter isolates a specified grating order (spectral band) and the diffraction grating spatially disperses the energy within the selected grating order along one of the linear arrays of detectors. The seventeen detector arrays are grouped into twelve modules, each containing one, two, or four arrays. Table 2.1 indicates the grouping and the spectral coverage of each array. Each array is optimized for its corresponding spectral band. Arrays D-1a through D-10 are bilinear arrays of photovoltaic (PV) HgCdTe detectors, while D-11 and D-12 are linear arrays of photoconductive (PC) HgCdTe detectors. In the PV arrays the detectors are arranged in two rows along the dispersed direction. Each pair of PV detectors arranged along the cross-
<table>
<thead>
<tr>
<th>Module</th>
<th>Array</th>
<th>No. channels</th>
<th>Spectral Range (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-1</td>
<td>D-1a</td>
<td>118</td>
<td>3.7364–3.9169</td>
</tr>
<tr>
<td></td>
<td>D-1b</td>
<td>130</td>
<td>4.1100–4.3291</td>
</tr>
<tr>
<td>M-2</td>
<td>D-2a</td>
<td>116</td>
<td>3.9149–4.1100</td>
</tr>
<tr>
<td></td>
<td>D-2b</td>
<td>150</td>
<td>4.3271–4.6085</td>
</tr>
<tr>
<td>M-3</td>
<td>D-3</td>
<td>192</td>
<td>6.9356–7.4769</td>
</tr>
<tr>
<td>M-4</td>
<td>D-4a</td>
<td>104</td>
<td>6.2003–6.4934</td>
</tr>
<tr>
<td></td>
<td>D-4b</td>
<td>106</td>
<td>6.5504–6.8500</td>
</tr>
<tr>
<td></td>
<td>D-4c</td>
<td>94</td>
<td>7.4745–7.7921</td>
</tr>
<tr>
<td></td>
<td>D-4d</td>
<td>106</td>
<td>7.8605–8.2200</td>
</tr>
<tr>
<td>M-5</td>
<td>D-5</td>
<td>159</td>
<td>8.8073–9.4796</td>
</tr>
<tr>
<td>M-6</td>
<td>D-6</td>
<td>167</td>
<td>9.5650–10.2750</td>
</tr>
<tr>
<td>M-7</td>
<td>D-7</td>
<td>167</td>
<td>10.2750–10.9850</td>
</tr>
<tr>
<td>M-8</td>
<td>D-8</td>
<td>161</td>
<td>11.0704–11.7512</td>
</tr>
<tr>
<td>M-9</td>
<td>D-9</td>
<td>167</td>
<td>11.7431–12.6850</td>
</tr>
<tr>
<td>M-10</td>
<td>D-10</td>
<td>167</td>
<td>12.7989–13.7457</td>
</tr>
<tr>
<td>M-11</td>
<td>D-11</td>
<td>144</td>
<td>13.7377–14.5533</td>
</tr>
<tr>
<td>M-12</td>
<td>D-12</td>
<td>130</td>
<td>14.6672–15.4000</td>
</tr>
</tbody>
</table>

Table 2.1: Spectral coverage of the twelve detector array modules of the AIRS infrared spectrometer.

The dispersed direction is combined into one spectral channel. In the PC arrays there is only one row of detectors along the dispersed direction, and one detector per spectral channel. To reduce the $1/f$ noise on the PC detectors, the radiation incident on them is chopped at a rate of 357 Hz. The IR spectrometer has a total of 2378 channels and a nominal spectral resolution of $\lambda/\Delta\lambda = 1200$. A full 2378-channel spectrum is generated every 22.4 msec.

The signals from the PV detectors and the PC detectors are processed differently [27]. The readout circuits for the PV detectors perform signal integration and multiplexing on the focal plane. Twenty-six high-level analog output channels are buffered, resampled, and multiplexed before being digitized with 12-bit resolution. In contrast, each of the 274 low-level PC detector channels are connected to individual low-noise pre-amplifiers located off the focal plane. The outputs of the pre-amplifiers are integrated to match the integration time of the PV channels. The integrated outputs are then sampled, multiplexed, and digitized together with the PV outputs. After digitization both the PC and PV data are processed further. Sub-samples of each PC channel are differenced to attenuate $1/f$ noise in the chopped signal. The PV data is first processed by a double-differencing radiation circumvention algorithm. Then the outputs of the pair of detectors for each PV channel
are gain corrected to maximize the signal-to-noise ratio and added together.
Chapter 3

Joint Linear Estimation and Transform Coding of Noisy Vectors

3.1 Introduction

This chapter addresses the problem of coding noisy measurements of a vector source when the objective is to reconstruct the noiseless source output as closely as possible in a mean-square error sense, while imposing an upper limit on the number of bits used to communicate or store the observed measurements. This problem has been studied from an information theoretic perspective [13, 14]. The structure and properties of the optimal scalar and vector quantizers for noisy sources and mean-square error criterion have also been given in the literature [15, 16]. In this chapter we wish to consider results particular to the family of codes known as transform coders or block-quantizers [33, 31]. We focus on transform coders because we are interested in methods that can be applied to vector data with hundreds or thousands of elements at a modest computational cost. The simpler structure of transform coders compared to more general vector quantizers [34] makes them more practical for applications that generate data of that sort, e.g. multispectral remote sensing of the earth.

In the following sections we discuss both theoretical and practical aspects of the design and application of transform coders to the noisy source coding problem. The codebook and the data model are defined in Section 3.2. It is assumed throughout the chapter that the signal and noise are independent Gaussian random vectors. The rest of the chapter treats the design of the codebook in two different cases: 1) when the statistics of both the signal
and the noise are known, and 2) when the statistics of the signal are not known.

In Section 3.3 we treat the case of known signal and noise statistics. In that case we design the codebook by optimization with respect to mean-weighted-square error. A certain orthogonality constraint is imposed on the codebook in order to make the optimization problem tractable. Under that constraint it is shown that the optimal codebook can be described in terms of well-known results in statistical estimation and quantization, and has many of the properties implied by rate distortion theory; in particular, an estimator-coder structure. Reference is made to the work of Malvar [17], who obtained the same result under somewhat different assumptions. Details concerning the practical implementation of the optimal codebook are discussed in Section 3.3.4. Then, in Section 3.3.3 we specialize the results to the case of "noise-adjusted" distortion, where the weight matrix is a square-root of the inverse noise covariance matrix. We show that the optimal transform for that case computes so-called noise adjusted principal components which compact the signal energy per unit noise [18]. In addition, we derive a simplified structure for the optimal codebook for that case.

In Section 3.4 we consider the design of the codebook when the signal statistics are not known a priori. The noise statistics are assumed to be known. Because a complete probabilistic description of the signal and the noisy data is not known, by assumption, it is not possible to design the codebook following the same optimization approach. The general design approach pursued in this case is to first estimate the statistics of the signal and the noisy source from a finite set of (noisy) data samples, and then to implement the optimal codebook for those estimates. It is shown that if the estimates of the statistics are consistent, a codebook designed this way is asymptotically optimal as the number of data samples grows to infinity. Two design algorithms based on the proposed method are formulated and evaluated through numerical simulations. The first algorithm uses Maximum-Likelihood (ML) estimates of the statistics, and it is found to produce codebooks with poor rate-distortion characteristics even when the number of samples is many times greater than the vector dimension. The second algorithm uses ad hoc modified versions of the ML estimates, and it is found to produce better codebooks.
Figure 3-1: Model of the measurement.

3.2 Data Model and Codebook Definition

Figure 3-1 illustrates the data model used throughout this chapter. The \( N \)-dimensional measurement vector \( y(t) \) is obtained using an imperfect sensor idealized here as additive random noise \( n(t) \) independent of the signal \( x(t) \), which is also modeled as a random vector. Our objective is to find a discrete representation of \( y(t) \) from which \( x(t) \) can be estimated with minimum distortion.

The structure of the transform coder considered here is shown in Figure 3-2. The first operation is a matrix transformation of the input to the encoder. Then each element of the transformed vector \( v \) is digitized using scalar quantization [34]. In the figure each scalar quantizer is decomposed into an index generator, which resides at the encoder, and an index decoder residing at the decoder. This parsing of a quantizer is necessary when a digital channel exists between the encoder and the decoder. Consider that the \( i^{th} \) quantizer can be represented as a partition of the real line into \( L_i \) mutually exclusive sets \( \{ S_i(j) \}_{j=1}^{L_i} \) and a set of scalars \( \{ \phi_i(j) \}_{j=1}^{L_i} \), the so-called quantization levels. For an input \( v_i \), the index generator \( h_i(v_i) \) returns the index of the set \( S_i(j) \) containing \( v_i \), and the index decoder \( g_i(h_i(v_i)) \) returns the corresponding quantization level \( \phi_i(j) \). Note that only the index vector \( h(v) \),

\[
h(v) \triangleq [h_1(v_1) \ h_2(v_2) \ \ldots \ h_N(v_N)]^T,
\]

is transmitted from the encoder to the decoder. The superscript \( T \) denotes matrix transposition. Since this vector can assume at most \( L_1 \times L_2 \times \cdots \times L_N \) distinct values, it can
be transmitted using a binary code of length $B$ bounded by

$$
\sum_{i=1}^{N} \log_2 L_i \leq B < \sum_{i=1}^{N} \log_2 L_i + 1.
$$

(3.2)

For simplicity it is assumed throughout this chapter that the index vector is communicated to the decoder without error. The final step in the decoder is a matrix transformation of the quantized vector.

In contrast with traditional transform coders, no fixed relationship is assumed \emph{a priori} between the encoder and decoder matrices except that their dimensions are identical. A compelling reason for not requiring, for example, that one matrix be the inverse of the other is that for the problem of interest the output of the decoder is not to be an approximation of the input to the encoder, i.e. the noisy measurement, but instead it should approximate the noiseless signal.

### 3.3 The Case of Known Signal and Noise Statistics

When a complete probabilistic characterization of the signal and the noise is available it is possible to evaluate a codebook in terms of the expected distortion for some suitably defined distortion metric. It may be possible then to formulate an optimization problem in which the desired codebook minimizes the expected distortion subject to an upper limit on the bit rate. In general, this may be a difficult problem to solve. In our case if both the signal and the noise are (independent) Gaussian random vectors, the optimization problem for a mean-weighted-square error criterion becomes tractable. This problem is addressed
3.3.1 Optimization With Respect To Mean-Weighted-Square Error

In this section we derive the transform coder that minimizes the distortion

$$D_W \triangleq E\{\|W(x - \hat{x}(y))\|^2\} \tag{3.3}$$

subject to an upper limit of $B$ bits per vector when $y = x + n$, and $x$ and $n$ are independent zero-mean Gaussian random vectors. $\|\cdot\|$ is the Euclidean norm. From equation 3.2 the upper limit $B$ on the bit rate (bits/vector) translates to the constraint

$$\sum_{i=1}^{N} \log_2 L_i \leq B \tag{3.4}$$

where $L_i$ is the number of quantization levels of the $i^{th}$ quantizer. The matrix $W$ above can be chosen to weigh differently the coordinates of the error vector with respect to an arbitrary basis for $N$-dimensional real space. We require that $W$ is invertible so that $W(x - \hat{x}(y)) \neq 0$ if $x - \hat{x}(y) \neq 0$.

To solve for the optimal codebook we will impose one constraint on the decoder matrix $G$: we will require that its columns be conjugate with respect to $W^TW$. For $1 \leq i \leq N$ and $1 \leq j \leq N$,

$$G_{*,i}^T W^T W G_{*,j} = \begin{cases} 1, & i = j; \\ 0, & \text{else.} \end{cases} \tag{3.5}$$

where $G_{*,i}$ is the $i^{th}$ column of $G$. As we will see shortly, this constraint makes the optimization problem much simpler and leads to a solution which has many of the properties implied by the rate-distortion bound. We will show that under the above constraint on the decoder matrix the optimal transform codebook has the following characteristics:

1. The optimal codebook has the estimator-coder structure implied by the rate-distortion bound, i.e. the output of the codebook is generated by first computing $\hat{x}^*(y)$, the minimum mean-square error estimate (MMSEE) of the signal given the measurement, and then coding $\hat{x}^*(y)$ optimally in the same mean-weighted-square error sense.

---

1For any matrix $A$, $A_{*,i}$ denotes its $i^{th}$ column, and $A_{i,*}$ denotes its $i^{th}$ row
2. The optimal decoder matrix is $G = W^{-1}U$, where $U$ is a Karhunen-Loève basis for the scaled MMSE $W\hat{x}^*(y)$.

3. The optimal encoder matrix can be taken as $H = U^T W L_{xy}$, where $L_{xy}$ is the MMSE matrix. Since $Hy = U^T W\hat{x}^*(y)$, the optimal encoder matrix generates the Karhunen-Loève coefficients of $W\hat{x}^*(y)$.

4. The optimal scalar quantizers are Lloyd-Max quantizers for the Karhunen-Loève coefficients of $W\hat{x}^*(y)$.

5. The optimal quantizer level allocation or bit allocation is a function of the variances of the Karhunen-Loève coefficients of $W\hat{x}^*(y)$, and the rate-distortion characteristic of the Lloyd-Max quantizer for a unit-variance Gaussian random variable.

The optimal codebook obtained here was previously derived by Malvar [17] under different assumptions. In contrast to our derivation, Malvar did not impose any constraints on either the encoder or the decoder matrices, which he called the pre-filter and the post-filter, respectively. However, he did not simultaneously optimize the quantizers with the pre-filter and the post-filter. Instead, he derived the optimal pre-filter and post-filter for Lloyd-Max quantizers. He was able to derive the optimal filters by using a gain-plus-additive-noise model of scalar quantization. Malvar’s approach and ours are similar in that the assumptions made in each case simplify the problem by avoiding the explicit computation of the covariance matrix of the quantized variables. In our case this is done implicitly by considering only decoder matrices for which the distortion can be written in separable form (see equation 3.13) so that the cross-covariance between quantized variables is irrelevant. In Malvar’s case, the approximate model of scalar quantization used in his derivation leads to a tractable approximation of the aforementioned covariance. Malvar mentions other authors who obtained related results under more restrictive assumptions [17].

Before proceeding with the optimization of the codebook let us look at a particular decomposition of the expected distortion $D_W$ which begins to illustrate the relationship between the estimation and coding aspects of the problem. Note that the expected distortion $D_W$ can be written as

$$D_W = E\left\{\|W(x - \hat{x}^*(y))\|^2\right\} + E\left\{\|W(\hat{x}^*(y) - \hat{x}(y))\|^2\right\} \quad (3.6)$$
where $\hat{x}^*(y)$ is the MMSE of $x$ given $y$ [29]. The last equality follows from the orthogonality of the error $x - \hat{x}^*(y)$ with any function $f(y)$, $E\{(x - \hat{x}^*(y)) f(y)\} = 0$, which holds for any two random vectors $x$ and $y$ for which $\hat{x}^*(y)$ exists. The first term in equation 3.6 indicates that there is a finite bound on the performance of any codebook, imposed by the imperfect nature of the measurement. This component of the error is completely independent of any degradation introduced by the discrete nature of the codebook. Therefore, we might refer to this component of the distortion as the estimation distortion or estimation error. The second term in equation 3.6 arises as a result of coding the measurements. We will refer to this component as the coding distortion or coding error.

In the case of jointly Gaussian zero-mean random vectors it is well known that $\hat{x}^*(y)$ is linear in the observation [29];

$$\hat{x}^*(y) = L_{xy}y$$ (3.7)

where $L_{xy}$ satisfies

$$L_{xy}C_{yy} = C_{xy}$$ (3.8)

with $C_{xy} \triangleq E\{xy^T\}$, $C_{yy} \triangleq E\{yy^T\}$. In this case the expected distortion becomes

$$D_W = E\left\{\|W(x - L_{xy}y)\|^2\right\} + E\left\{\|W(L_{xy}y - \hat{x}(y))\|^2\right\},$$ (3.9)

and the optimal transform codebook is the one that minimizes the coding distortion $D^c_W$:

$$D^c_W \triangleq E\left\{\|W(L_{xy}y - Gq(Hy))\|^2\right\}.$$ (3.10)

**The Optimal Quantizers**

In general, the main difficulty in solving for the optimal transform codebook is determining the optimal scalar quantizers but this task becomes much easier when the constraint on the decoder matrix (equation 3.5) is enforced. The constraint on the decoder matrix $G$ implies that

$$G = W^{-1}U,$$ (3.11)
where \( \mathbf{U} \) is an arbitrary orthogonal matrix. Substituting this expression in equation 3.10 and performing some simple manipulations yields

\[
D_W' = E \left\{ \left\| \mathbf{U}^T \mathbf{W} \mathbf{L}_{xy} \mathbf{y} - q(\mathbf{H} \mathbf{y}) \right\|^2 \right\} \\
= \sum_{i=1}^{N} E \left\{ (z_i - q_i(v_i))^2 \right\} 
\]

(3.12)

(3.13)

with \( z \overset{\Delta}{=} \mathbf{U}^T \mathbf{W} \mathbf{L}_{xy} \mathbf{y} \) and \( v \overset{\Delta}{=} \mathbf{H} \mathbf{y} \). Thus, the optimal quantizers can be obtained by separately minimizing each term in equation 3.13. The following proposition proved in Appendix A shows that the optimal quantizers are Lloyd-Max quantizers for a particular scaling of the input.

**Proposition 3.1** Let \( z \) and \( v \) be zero-mean jointly Gaussian random variables with \( \sigma_z^2 \overset{\Delta}{=} E\{v^2\} > 0 \). The \( L \)-level quantizer \( q(\cdot) \) that minimizes \( E\{(z - q(v))^2\} \) is given by \( q^*(v) = \tilde{q}^* (\sigma_{zv}/\sigma_v^2 v) \) where, \( \sigma_{zv} \overset{\Delta}{=} E\{vz\} \), and \( q^*(\cdot) \) is the \( L \)-level Lloyd-Max quantizer for a zero-mean Gaussian random variable with variance \( \sigma_{zv}^2/\sigma_v^2 \). The optimal distortion is given by

\[
E\left\{ (z - q^*(v))^2 \right\} = \sigma_z^2 - \frac{\sigma_{zv}^2}{\sigma_v^2} (1 - d(L)),
\]

where \( \sigma_z^2 \overset{\Delta}{=} E\{z^2\} \) and \( d(L) \) is the distortion of the \( L \)-level Lloyd-Max quantizer for a unit-variance Gaussian random variable.

From the proposition and equation 3.13 it follows that for any given level allocation \( L_i \), \( 1 \leq i \leq N \), encoder matrix \( \mathbf{H} \), and decoder matrix \( \mathbf{G} \) satisfying equation 3.5, the coding distortion obtained using the optimal scalar quantizers is given by

\[
D_W = \sum_{i=1}^{N} \sigma_{z_i}^2 - \sum_{i \text{ s.t. } \sigma_{v_i} > 0} \frac{\sigma_{z_i v_i}^2}{\sigma_{v_i}^2} (1 - d(L_i)))
\]

(3.14)

where

\[
\sigma_{v_i}^2 = \mathbf{H}_{i,*} \mathbf{C}_{yy} \mathbf{H}_{i,*}^T, \\
\sigma_{z_i v_i} = \mathbf{A}_{i,*} \mathbf{C}_{yy} \mathbf{H}_{i,*}^T, \\
\sigma_{z_i}^2 = \mathbf{A}_{i,*} \mathbf{C}_{yy} \mathbf{A}_{i,*}^T, \\
\mathbf{A} \overset{\Delta}{=} \mathbf{U}^T \mathbf{W} \mathbf{L}_{xy}.
\]

(3.15)

(3.16)

(3.17)

(3.18)
The Optimal Encoder Matrix

The dependency of the coding distortion on the encoder matrix \( \mathbf{H} \) is through \( \sigma_{v_i}^2 \) and \( \sigma_{z,v_i}^2 \). For \( \sigma_{v_i}^2 > 0 \) one can write

\[
\frac{\sigma_{z,v_i}^2}{\sigma_{v_i}^2} = \frac{(A_{i,*}C_{yy}H_{i,*}^T)^2}{\mathbf{H}_{i,*}C_{yy}H_{i,*}^T} \quad (3.19)
\]

\[
= \left( A_{i,*}C_{yy}^{1/2}C_{yy}^{1/2}H_{i,*}^T \right)^2 \quad (3.20)
\]

\[
= \left( \tilde{A}_{i,*}\mathbf{H}_{i,*}^T \right)^2 \quad (3.21)
\]

where \( \tilde{A}_{i,*} \triangleq A_{i,*}C_{yy}^{1/2} \), \( \tilde{H}_{i,*} \triangleq H_{i,*}C_{yy}^{1/2} \), \( C_{yy}^{1/2} \) is a matrix square-root of \( C_{yy} \), and \( C_{yy}^{T/2} \triangleq (C_{yy}^{1/2})^T \). Substituting \( \sigma_{z_i}^2 = \| \tilde{A}_{i,*} \|^2 \) and equation 3.21 into equation 3.14 yields

\[
D_{W}^C = \sum_{i=1}^{N} \| \tilde{A}_{i,*} \|^2 - \sum_{i \text{ s.t. } \sigma_{v_i} > 0} \left( \tilde{A}_{i,*}\tilde{H}_{i,*}^T \right)^2 (1 - d(L_i)) \quad (3.22)
\]

So minimizing \( D_{W}^C \) with respect to \( \tilde{H}_{i,*} \) is equivalent to maximizing the second term in the equation above with respect to \( \tilde{H}_{i,*} \) with the constraint \( \| \tilde{H}_{i,*} \| = 1 \). Clearly, if \( \| \tilde{A}_{i,*} \| = 0 \) then \( \tilde{H}_{i,*} \) is arbitrary. Otherwise, the optimal \( \tilde{H}_{i,*} \) must satisfy

\[
2\tilde{H}_{i,*}\tilde{A}_{i,*}^T\tilde{A}_{i,*} - 2\lambda\tilde{H}_{i,*} = 0 \quad (3.23)
\]

where \( \lambda \) is a Lagrange multiplier. This implies \( \tilde{H}_{i,*} \propto \tilde{A}_{i,*} \) and because \( \| \tilde{H}_{i,*} \| = 1 \) one obtains

\[
\tilde{H}_{i,*} = \frac{\tilde{A}_{i,*}}{\| \tilde{A}_{i,*} \|} \quad (3.24)
\]

as a necessary condition for optimality. This condition is also sufficient because the continuity of the function and the compactness of the constraint set guarantee that an optimal solution exists. Since one is free to choose \( \tilde{H}_{i,*} \) as above for all \( i \) such that \( \| \tilde{A}_{i,*} \| > 0 \), it follows from the definitions of \( \tilde{H}_{i,*} \) and \( \tilde{A}_{i,*} \) that the optimal encoder matrix \( \mathbf{H} \) must satisfy

\[
H_{i,*}C_{yy}^{1/2} \propto A_{i,*}C_{yy}^{1/2} \quad (3.25)
\]
for all \( i \) such that \( \| \tilde{A}_{i,*} \| > 0 \). For \( i \) such that \( \| \tilde{A}_{i,*} \| = 0 \) the choice of \( H_{i,*} \) is arbitrary. Thus, the optimal encoder matrix is not unique. It is easy to show that if the covariance matrix of the noiseless data \( C_{xx} \) is positive definite then \( C_{yy} \) is also positive definite and \( \| \tilde{A}_{i,*} \| > 0 \) for \( 1 \leq i \leq N \). In that case \( H_{i,*} \) must be proportional to \( A_{i,*} \) for \( 1 \leq i \leq N \) according to equation 3.25. In any event, the optimal encoder matrix can always be taken as

\[
H = A = U^T W L_{xy} = G^{-1} L_{xy}.
\] (3.26)

Thus, the optimal encoder matrix applies the inverse of the decoder matrix to the MMSEE \( \hat{x}^* \) of the signal \( x \) given the measurement \( y \). It is also worth noting that with the optimal encoder matrix the optimal quantizers are exactly Lloyd-Max quantizers for the elements of \( G^{-1} \hat{x}^* \).

Finally, substituting equation 3.24 in equation 3.22 gives

\[
D^e_W = \sum_{i=1}^{N} \| \tilde{A}_{i,*} \|^2 d(L_i)
\] (3.27)

\[
= \sum_{i=1}^{N} A_{i,*} C_{yy} A_{i,*} d(L_i)
\] (3.28)

\[
= \sum_{i=1}^{N} U_{*,i}^T \left( W L_{xy} C_{yy} L_{xy}^T W^T \right) U_{*,i} d(L_i)
\] (3.29)

This coding distortion is achieved with the optimal encoder matrix and optimal quantizers for a given encoder matrix and quantizer level allocation \( L_i \), \( 1 \leq i \leq N \).

**The Optimal Decoder Matrix**

To derive the optimal decoder matrix \( G \) we first find the orthogonal matrix \( U \) that minimizes the coding distortion in equation 3.29 and then obtain \( G \) from equation 3.11. Specifically, the optimization problem for \( U \) is the following:

\[
\text{minimize } \sum_{i=1}^{N} U_{*,i}^T \left( W L_{xy} C_{yy} L_{xy}^T W^T \right) U_{*,i} d(L_i)
\]

subject to \( U_{*,i}^T U_{*,j} = \begin{cases} 1, & i = j; \\ 0, & \text{else}. \end{cases} \) (3.30)

It is well known that the solution to this problem is an orthogonal matrix of eigenvectors of the matrix \( W L_{xy} C_{yy} L_{xy}^T W^T \). Recall that \( d(L) \) is the distortion of the \( L \)-level Lloyd-Max
quantizer for a unit-variance Gaussian random variable. Therefore, \( d(L) \) is monotonically decreasing. Assuming that \( L_1 \geq L_2 \geq \cdots \geq L_N \) the columns of \( U \), the eigenvectors, must be ordered such that the corresponding eigenvalues are in decreasing order.

\[
U^T \left( WL_{xy} C_{yy} L_{xy}^T W^T \right) U = \Lambda \\
\Lambda = \text{diag} (\lambda_1, \lambda_2, \ldots, \lambda_N) \\
\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N
\] (3.31)

Once \( U \) has been computed the optimal decoder matrix can be obtained as \( G = W^{-1} U \). Noting that the matrix being diagonalized above is simply the covariance matrix of \( W \tilde{x}^* \) one can see that \( U \) generates the Karhunen-Loève expansion of the MMSEE of \( x \) scaled by the weighting matrix \( W \).

The coding distortion obtained with the optimal decoder and encoder matrices, the optimal quantizers and any quantizer level allocation \( L_1 \geq L_2 \geq \cdots \geq L_N \) is given by

\[
D_{\tilde{x}_W}^c = \sum_{i=1}^N \lambda_i d(L_i)
\] (3.34)

where \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \) are the eigenvalues of \( WL_{xy} C_{yy} L_{xy}^T W^T \) and \( d(L) \) is the quantizer distortion defined above.

**The Optimal Bit Allocation**

The remaining step in the derivation of the optimal transform codebook is choosing the allocation of quantizer levels \( L_i, 1 \leq i \leq N \) so that the coding distortion in equation 3.34 is minimized subject to the constraint

\[
\sum_{i=1}^N \log_2 L_i \leq B.
\] (3.35)

This problem is usually referred to as bit allocation because in practice the number of levels of each quantizer is taken to be an integer power of two so that the output of the \( i^{th} \) quantizer can be individually represented using a binary word of length \( b_i \triangleq \log_2 L_i \). In that case one finds \( b_i \) instead of \( L_i \) for each quantizer. Whether or not the latter constraint is imposed on the number of quantizer levels, an exact closed-form expression for the optimal
quantizer level allocation is not generally possible.

The known algorithms for finding the optimal quantizer level allocation or the optimal bit allocation are iterative in nature. Fox gave an algorithm based on marginal analysis that applies to our bit allocation problem [35]. Starting from a zero bit allocation for each quantizer, bits are allocated sequentially by incrementing at each point the allocation of the quantizer for which the decrease in distortion is largest. This is repeated until the available bit budget is exhausted. The final bit allocation obtained using this "greedy" algorithm is optimal whenever the quantizer distortion characteristic \( d(b) \) is convex, i.e. when the sequence of first-order differences of \( d(b) \) is strictly decreasing. In the case of a Lloyd-Max quantizer for a Gaussian input, although we do not know of a way to prove convexity, numerical evaluation of \( d(b) \) supports the hypothesis. A more general algorithm based on Lagrange multipliers was developed more recently by Shoham and Gersho [36]. Their algorithm is applicable to an arbitrary set of quantizers and, in particular, even when the number of quantizer levels is not an integer power of two. However, their algorithm is guaranteed to be optimal only when the distortion characteristics of all the quantizers are convex. Another algorithm by Riskin [37] finds the optimal bit allocation for an arbitrary set of quantizers by searching a particular kind of tree. When the rate-distortion characteristics of the quantizers are convex Riskin's algorithm is more efficient than the one by Shoham and Gersho.

It is also possible to obtain good solutions to the quantizer level allocation or bit allocation problem by relaxing some of the constraints in those problems and by approximating the quantizer distortion characteristic. A well-known approximation by Segall is discussed in Section 3.3.4.

### 3.3.2 Estimator-Coder Structure of the Optimal Codebook

In this section we summarize the results of the derivation of the optimal codebook and illustrate the estimator-coder structure of the solution. We reiterate that the optimality of the solution is subject to the assumption that the signal \( \mathbf{x} \) and the measurement \( \mathbf{y} \) are jointly-Gaussian random vectors, and to the constraint in equation 3.5 on the decoder matrix \( \mathbf{G} \). Recall that \( \hat{\mathbf{x}}^*(\mathbf{y}) = \mathbf{L}_{xy} \mathbf{y} \) is the minimum mean-square error estimate (MMSEE) of the signal given the measurement. In the derivation of the optimal codebook the following results were obtained:
• The optimal decoder matrix is $G = W^{-1}U$, where $U$ is a Karhunen-Loève basis for the scaled MMSE $W\hat{x}^*(y)$.

• The optimal encoder matrix can be taken as $H = U^TWL_{xy}$. Since $Hy = U^T W\hat{x}^*(y)$, the optimal encoder matrix generates the Karhunen-Loève coefficients of $W\hat{x}^*(y)$.

• The optimal scalar quantizers are Lloyd-Max quantizers for the Karhunen-Loève coefficients of $W\hat{x}^*(y)$.

• The optimal quantizer level allocation or bit allocation is a function of the variances of the Karhunen-Loève coefficients of $W\hat{x}^*(y)$ and the rate-distortion characteristic of the Lloyd-Max quantizer for a unit-variance Gaussian random variable.

Figure 3-3 shows the estimator-coder decomposition of the optimal codebook. This decomposition is obtained by recognizing that the optimal quantized estimate $\hat{x}$ is a function of the MMSE. In fact, the optimal codebook ensures that $\hat{x}$ approximates the MMSE as close as possible in a mean-weighted-square error sense.

The estimator-coder structure is also a property of the rate-distortion bound for a noisy source and a mean-weighted-square error fidelity criterion. Specifically, it can be shown that for a given bit rate the minimum distortion achievable by a codebook with the noisy measurement as the input and the noiseless source as the target is the same that can be achieved by a codebook with the MMSE of the noiseless source both as the input and as the target.

3.3.3 Form of the Codebook for Noise-Adjusted Weighting

The optimal transform coder for a Gaussian source corrupted by independent Gaussian noise can be further simplified when the weighting matrix $W$ is a square-root of the inverse of the noise covariance; i.e. $W^TW = C_{nn}^{-1}$. For this particular choice of error weighting it can be shown that the scaled signal $Wx$, the scaled noise $Wn$, and the scaled MMSE $W\hat{x}^*$ all have a common Karhunen-Loève (KL) basis. The transformation of the noisy data into this KL basis yields the so-called noise-adjusted principal components (NAPC) [18]. Because both the signal and noise components are simultaneously decorrelated, the MMSE matrix

---

2There are an infinite number of solutions $W$ [38]. Any two solutions with the same dimensions are related through an orthogonal matrix.
is diagonal in NAPC space. Thus, the NAPC transformation is a natural representation for
noise filtering and is also the optimal transformation for coding when the weighting matrix
is chosen as above. Below we discuss some interesting properties of NAPC and show the
representation of the optimal codebook in NAPC space.

The NAPC transformation is obtained by simultaneously diagonalizing the covariance
matrices of the signal and the noise in a particular way. The first step, as indicated above,
is to scale both the signal and the noise by a (square) matrix which is a square-root of the
inverse of the noise covariance $C_{nn}$. Let $W = C_{nn}^{-1/2}$ be such a matrix. The covariance of
the “noise-adjusted” measurements is then given by

$$E\left\{C_{nn}^{-1/2} y y^T \left(C_{nn}^{-1/2}\right)^T\right\} = C_{nn}^{-1/2} C_{yy} \left(C_{nn}^{-1/2}\right)^T,$$  \hspace{1cm} (3.36)

$$= C_{nn}^{-1/2} (C_{xx} + C_{nn}) \left(C_{nn}^{-1/2}\right)^T,$$  \hspace{1cm} (3.37)

$$= C_{nn}^{-1/2} C_{xx} \left(C_{nn}^{-1/2}\right)^T + I,$$  \hspace{1cm} (3.38)

which shows that in the “noise-adjusted” space the noise is uncorrelated and has unit
variance in every direction. The second and final step is to perform a KL transformation
of the “noise-adjusted” data. The KL basis in this case corresponds to the eigenvectors
of the covariance matrix in equation 3.38. Let $\Lambda$ be the diagonal matrix of eigenvalues
and $U$ be the corresponding matrix of eigenvectors, both ordered in such a way that the
eigenvalues appear in decreasing order along the diagonal of \( \Lambda \). From equation 3.38 the matrix of eigenvalues satisfies

\[
\Lambda = U^T \left( C_{nn}^{-1/2} C_{xx} \left( C_{nn}^{-1/2} \right)^T + I \right) U,
\]

(3.39)

\[
= U^T C_{nn}^{-1/2} C_{xx} \left( C_{nn}^{-1/2} \right)^T U + I,
\]

(3.40)

\[
\triangleq \Gamma + I.
\]

(3.41)

Equation 3.40 shows that both the signal covariance and the noise covariance are diagonal in NAPC space. From equations 3.40–3.41 the \( i \)th NAPC has a signal component of variance \( \gamma_i \) and a noise component of unit variance or, equivalently, \( \gamma_i \) is the signal-to-noise ratio of the \( i \)th NAPC. It can be shown that the sequence of ordered NAPC is the sequence of uncorrelated linear combinations of the data that maximizes the signal-to-noise ratio at each point in the sequence.

Now let us turn to the form of the optimal transform codebook when \( W = C_{nn}^{-1/2} \). Consider first the representation of the MMSEE matrix \( L_{xy} \) in NAPC space. Using \( U \) and \( \Gamma \) as defined above one can write

\[
L_{xy} = C_{xx} \left( C_{xx} + C_{nn} \right)^{-1}
\]

(3.42)

\[
= C_{xx} \left( C_{xx} + C_{nn}^{1/2} \left( C_{nn}^{1/2} \right)^T \right)^{-1}
\]

(3.43)

\[
= C_{xx} \left( C_{nn}^{-1/2} \left( C_{nn}^{-1/2} \right)^T \right) \left( C_{nn}^{-1/2} C_{xx} \left( C_{nn}^{-1/2} \right)^T + I \right)^{-1} C_{nn}^{-1/2}
\]

(3.44)

\[
= C_{nn}^{1/2} C_{nn}^{-1/2} C_{xx} \left( C_{nn}^{-1/2} \right)^T \left( C_{nn}^{-1/2} C_{xx} \left( C_{nn}^{-1/2} \right)^T + I \right)^{-1} C_{nn}^{-1/2}
\]

(3.45)

\[
= C_{nn}^{1/2} U \Gamma U^T \left( U \Gamma U^T + I \right)^{-1} C_{nn}^{-1/2}
\]

(3.46)

\[
= C_{nn}^{1/2} U \left( \Gamma + I \right)^{-1} U^T C_{nn}^{-1/2}
\]

(3.47)

From the last equality one can see that the MMSEE matrix first computes the NAPC of the noisy data, then scales the \( i \)th NAPC by \( \gamma_i / (\gamma_i + 1) \), and finally transforms the "filtered" NAPC back into regular space. This shows that the MMSEE matrix is diagonal in NAPC space. It remains to be shown that the NAPC transformation is the linear transformation performed by the optimal codebook. In Section 3.3.2 it was shown that the optimal codebook performs a KL expansion of the scaled MMSEE \( \hat{W} \hat{x}^*(y) \). Using the representation of the MMSEE matrix in NAPC space (equation 3.47) one can write the
Figure 3-4: Simplified representation of the optimal codebook when $W = C_{nn}^{-1/2}$. The matrix $U^T C_{nn}^{-1/2}$ computes the NAPC of the noisy data. The noise is filtered in NAPC space, where the MMSEE matrix is diagonal.

The scaled MMSEE of the signal as

$$
C_{nn}^{-1/2} \hat{x}^*(y) = C_{nn}^{-1/2} L_{xy} y, \quad (3.48)
$$

$$
= U (\Gamma + I)^{-1} U^T C_{nn}^{-1/2} y. \quad (3.49)
$$

Since $U^T$ generates the KL expansion of the “noise-adjusted” data $C_{nn}^{-1/2} y$ and multiplication by a diagonal matrix in KL space leaves the basis unchanged, it follows from the last equality that $U$ is a KL basis for $C_{nn}^{-1/2} \hat{x}^*(y)$.

Figure 3-4 shows a simplified representation of the optimal codebook using the results derived above. The MMSEE of the signal is computed in NAPC space, where the MMSEE matrix is diagonal. This implementation is particularly attractive when the noise covariance is diagonal. In that case the data can be “noise-adjusted” by simply dividing each component by the standard deviation of the corresponding noise. The optimal codebook is then completely specified by the eigenvectors and eigenvalues of the covariance matrix of the “noise-adjusted” data, and it is not necessary to solve a separate linear system of equations to obtain the MMSEE matrix.
3.3.4 Practical Implementation of the Optimal Codebook

Implementing an instance of the codebook in Figure 3-3 requires solving four problems: 1) solving $N$ systems of $N$ linear equations to obtain the MMSE matrix $L_{xy}$, 2) diagonalizing the $N \times N$ matrix $W L_{xy} C_{yy} (W L_{xy})^T$, 3) allocating the available bit budget among $N$ quantizers, and 4) designing and implementing the quantizers. The first two problems are standard in linear algebra. A discussion of methods for solving those problems can be found, for example, in [38]. In this section we discuss approximate solutions for the last two problems.

The quantizer level and bit allocation problems were posed in page 47 and iterative algorithms for their exact solution were briefly described. In practice it is often acceptable to settle for approximate solutions which do not require as much computation. A useful approximate solution to the bit allocation problem can be obtained if the integer constraint on the individual bit allocations $b_i$ is eliminated and the normalized distortion $d(b)$ of the quantizers is approximated by a decaying exponential. Segall gave the form of the optimal non-negative real bit allocation for a strictly convex rate-distortion characteristic with a continuous first derivative that tends to zero as the number of bits increases to infinity [34]. Segall's solution when the normalized distortion is approximated as $d(b) \approx 4^{-b}$ gives the well-known allocation rule:

$$b_i = \begin{cases} 
\frac{B}{k} + \frac{1}{k} \sum_{j=1}^{k} \log_4 \left( \frac{\lambda_i}{\lambda_j} \right), & 1 \leq i \leq k; \\
0, & \text{else;}
\end{cases}$$

(3.50)

where $k (\leq N)$ is the largest integer such that

$$B + \sum_{j=1}^{k} \log_4 \left( \frac{\lambda_k}{\lambda_j} \right) > 0.$$  

(3.51)

Generally this solution is used as a first guess and the final integer bit allocation is obtained by making small modifications following some ad hoc procedure. For instance, one might round off each $b_i$ and then add the surplus bits in a "greedy" fashion until the desired bit budget $B$ is met. Finally, note that Segall's solution can also be used to approximate the optimal integer quantizer level allocation by putting $L_i = 2^{b_i}$ and then modifying the $L_i$ in similar fashion to obtain integer values.
The Lloyd-Max quantizer for a scalar random input with a known distribution can be always obtained in iterative fashion as indicated by Lloyd [39] and Max [40]. However, when the number of quantizer levels is large those procedures are burdensome. Roe derived asymptotic approximations of the optimal quantizer when the distortion is the mean \( k^{th} \) power of the error magnitude and the input probability distribution is sufficiently (many times) differentiable [41]. Others have derived similar approximations [34]. The case of particular interest here is that of mean-square error distortion and a Gaussian input distribution. For a zero-mean unit-variance Gaussian distribution Roe approximated the decision levels of the optimal \( L \)-level quantizer as

\[
    l_i = \sqrt{6} \text{erf}^{-1} \left( \frac{2i - L}{L + 0.8532} \right), \quad 1 \leq i \leq L - 1, \tag{3.52}
\]

where \( \text{erf}^{-1}(\cdot) \) is the inverse error function, and \( \text{erf}(\cdot) \) is given by

\[
    \text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} \, dt. \tag{3.53}
\]

The reconstruction or output levels for the same case were approximated as

\[
    r_i = \sqrt{6} \text{erf}^{-1} \left( \frac{2i - 1 - L}{L + 0.8532} \right), \quad 1 \leq i \leq L. \tag{3.54}
\]

Roe found this approximation to be very accurate even for quantizers with a small number of levels. For example, for \( L = 6 \) he found the distortion to be within 0.2 percent of the value previously reported by Max [40].

The expressions above greatly simplify the implementation of the quantizer for a zero-mean unit-variance Gaussian input. To quantize an input value \( v \) one first inverts equation 3.52 with \( v \) in place of \( l_i \) and \( \hat{v} \) in place of \( i \) to obtain

\[
    \hat{v} = \left( \frac{L + 0.8532}{2} \right) \text{erf} \left( \frac{v}{\sqrt{6}} \right) + \frac{L}{2}. \tag{3.55}
\]

For an input \( v \) lying in the interval \( (l_i, l_{i+1}] \), \( 1 \leq i < L - 1 \), \( \hat{v} \) lies in \( (i, i+1] \). Similarly, for \( v \) in \( (-\infty, l_1] \) \( \hat{v} \) lies in \( (-0.4266, 1] \), and for \( v \) in \( (l_{L-1}, \infty) \) \( \hat{v} \) lies in \( (L - 1, L + 0.4266) \). Thus, the index generator of the quantizer, which gives the number of the interval containing \( v \),
is simply

$$h(v) = \begin{cases} 
1, & \hat{v} \leq 1, \\
L, & \hat{v} > L - 1, \\
\lceil \hat{v} \rceil, & 1 < \hat{v} \leq L - 1,
\end{cases} \quad (3.56)$$

where $\lceil \cdot \rceil$ rounds towards infinity. The quantized output is given equation 3.54 with $i = h(v)$. Note that this implementation of the quantizer does not require one to precompute and store the decision levels $l_i$ and the reconstruction levels $r_i$. These expressions can also be used to quantize a Gaussian input with arbitrary mean and variance by normalizing the input values prior to quantization and denormalizing the quantized values.

### 3.4 The Case of Unknown Signal Statistics and Known Noise Statistics

In this section we propose an algorithm for designing a transform coder for noisy data when the statistics of the signal are not known. Specifically, the codebook must be designed using a finite number of samples from the noisy source, and knowledge of the noise statistics. Because a complete probabilistic characterization of both the signal and noise is not available, it is no longer possible consider the minimization of a mean-square error criterion. It is possible, however, to formulate a consistent design algorithm; an algorithm for which the distortion of the empirical codebook approaches that of the optimal codebook for the true signal statistics, as the number of data samples grows to infinity. The design algorithm proposed here is based on using consistent estimates of the signal statistics. As before, we assume that the signal and noise are independent Gaussian random vectors. The mean and covariance of the noise are assumed to be known. In practice it is often possible to obtain the noise statistics by characterizing the sensor (channel) a priori or by periodically observing (transmitting) known test signals.

#### 3.4.1 Method for Empirical Codebook Design

The approach pursued here for empirical design of transform coders for noisy sources is based on the following observations:

1. In Section 3.3.1 it was shown that given the statistics of the signal and the noise, an optimal codebook with respect to mean-weighted-square error can be derived.
Furthermore, the optimal codebook is completely specified by the mean vectors and covariance matrices of the signal and the noisy source.

2. If the noise statistics are known, there exist consistent\(^3\) estimates of the statistics of both the signal and noisy source given an infinite sequence of samples of the noisy source.

These observations suggest an empirical design method consisting of first estimating the statistics of the signal and the noisy source from a finite set of samples of the noisy source, and computing the optimal codebook using the estimates in place of the true parameters. If the estimates of the statistics are consistent, one might expect the performance of the empirical codebook to approach that of the optimal codebook. It is relatively simple to show that this is indeed the case for any predetermined quantizer level allocation; this is shown in Appendix C. The implication of this result is that for a given bit quota, if one is able to guess the optimal allocation, a codebook arbitrarily close to optimal can be obtained if enough data is available. Although we suspect the optimal quantizer level allocation can also be determined given enough data, this has not been shown in our work. Below we describe the design method and give a precise statement of the consistency of the empirical codebook for a given quantizer level allocation. We close the section with some comments about the practical application of the method.

Let us now describe the general form of the design method. Let \( Y^{(k)} \) denote a sequence of \( k \) samples observed from the noisy source in Figure 3-1. Assume that the independent additive noise has mean zero and a positive definite covariance matrix \( C_{nn} \). The empirical codebook is obtained as follows:

1. Compute estimates \( \hat{m}_x^{(k)}, \hat{m}_y^{(k)}, \hat{C}_{xx}^{(k)}, \) and \( \hat{C}_{yy}^{(k)} \) of the mean vectors and covariance matrices of the signal and the noisy source, respectively, using \( Y^{(k)} \) and knowledge of the mean and covariance of the noise. \( \hat{C}_{xx}^{(k)} \) must be at least positive semi-definite and \( \hat{C}_{yy}^{(k)} \) must be positive definite.

2. Estimate the MMSE matrix as \( \hat{L}_{xy}^{(k)} = \hat{C}_{xx}^{(k)} \left( \hat{C}_{yy}^{(k)} \right)^{-1} \).

3. Diagonalize the matrix \( W \hat{L}_{xy}^{(k)} \hat{C}_{yy}^{(k)} \left( \hat{L}_{xy}^{(k)} \right)^{T} (W)T \). Let \( U^{(k)} \) be the matrix of eigenvectors and \( \Lambda^{(k)} \) the matrix of eigenvalues.

\(^{3}\)A sequence \( \hat{\theta}(i), i = 1, 2, \ldots, \) is a consistent estimate of a parameter \( \theta \) if \( \lim_{i \to \infty} \hat{\theta}(i) = \theta \) with probability one.
4. Let the decoder matrix be $G^{(k)} = W^{-1}U^{(k)}$, and the encoder matrix be $H^{(k)} = (U^{(k)})^T W\hat{L}_{xy}^{(k)}$.

5. Using the eigenvalues $\lambda_i^{(k)}$, $1 \leq i \leq N$, obtained in step 3 compute the optimal quantizer level allocation, or bit allocation if desired, for a quota of $B$ bits per vector.

6. Let $L_i$ denote the number of levels assigned in step 5 to the $i^{th}$ quantizer, $1 \leq i \leq N$. For each $i$ compute the $L_i$-level Lloyd-Max quantizer for a zero-mean Gaussian input with variance $\lambda_i^{(k)}$.

The reader should be able to recognize the two main parts of the method. First the statistics of the signal and the noisy data are estimated in step 1, and the optimal codebook assuming the estimates are correct is computed in steps 2–6. A graphical representation of the method is given in Figure 3-5. Note that it is not specified how the estimates in step 1 should be computed. The choice of estimates is left as part of the design process and the possible choices will lead to different algorithms patterned after the method outlined above. The following proposition indicates that for a predetermined quantizer level allocation the consistency of the estimates guarantees the asymptotic optimality of the codebook. The proposition is proved in Appendix C.

**Proposition 3.2** Let $y$ be the noisy source in Figure 3-1, where the signal $x$ and the noise $n$ are independent Gaussian random vectors. Let $m_x$ and $C_{xx}$ denote the mean and covariance of $x$, and $m_y$ and $C_{yy}$ denote the mean and covariance of $y$. Also let $D_W$ denote the expected distortion of the optimal transform codebook for a given weight matrix $W$, and a given quantizer level allocation $\{L_i\}_{i=1}^N$. Finally, for $k \in \{1, 2, \ldots\}$ let $\hat{m}_x^{(k)}$, $\hat{m}_y^{(k)}$, $\hat{C}_{xx}^{(k)}$, and $\hat{C}_{yy}^{(k)}$ be estimates of the mean vectors and covariance matrices of the signal and the noisy source, respectively, and denote by $D_W^{(k)}$ the expected distortion of the optimal codebook designed for the same quantizer level allocation $\{L_i\}_{i=1}^N$, and assuming the estimates above are the true parameters. If the estimates above are consistent, then $D_W^{(k)}$ converges to $D_W$ with probability one.

A practical design algorithm patterned after the general method outlined above must specify how to compute the optimal quantizer level allocation or bit allocation (step 5), and the optimal quantizers (step 6). Some of the algorithms available for both exact and approximate solution of the bit allocation problem are mentioned in Section 3.3.4. In the
case of the quantizers, Lloyd's Method I provides an iterative procedure for computing the optimal quantizer [39]. However, obtaining the optimal solution using this method generally requires an infinite number of iterations. In practice one must settle for an approximate solution either by stopping Lloyd's Method I after a finite number of iterations or by using some other approximation of the optimal quantizer. A high-resolution approximation of the optimal quantizer due to Roe [41] is discussed in Section 3.3.4.

3.4.2 Codebook Design Using Maximum-Likelihood Estimates of the Signal Statistics

In this section we discuss an algorithm for empirical codebook design using Maximum-Likelihood (ML) estimates of the mean and covariance of the signal. The algorithm is patterned after the general design method described above. For convenience, we shall refer to the algorithm as the Maximum-Likelihood Design Algorithm (MLDA). We evaluate and analyze the effectiveness of the algorithm by applying it to simple numerical examples.

Let us start the derivation of the MLDA by obtaining the ML estimates of the signal statistics. Let \( Y^{(k)} \) denote a sequence of \( k \) samples observed from the noisy source in Figure 3-1; \( Y^{(k)} = \{y(1), y(2), \ldots, y(k)\} \). The log-likelihood of \( Y^{(k)} \) is defined as the logarithm of the probability density of \( Y^{(k)} \) expressed as a function of the mean and covariance of the signal \( x \), denoted respectively by \( m_x \) and \( C_{xx} \). When the signal and noise are independent Gaussian random vectors the log-likelihood is given by

\[
\mathcal{L}\left(m_x, C_{xx}; Y^{(k)}\right) \triangleq -\frac{Nk}{2} \log 2\pi - \frac{k}{2} \log |C_{xx} + C_{nn}|
\]
\[-\frac{1}{2} \text{tr} \left[ (C_{xx} + C_{nn})^{-1} \sum_{t=1}^{k} (y(t) - m_x)(y(t) - m_x)^T \right] \] \quad (3.57)

where the noise is assumed to have mean zero and covariance \( C_{nn} \). The \( \text{tr} [\cdot] \) operator computes the trace of a matrix. The ML estimates of \( m_x \) and \( C_{xx} \) are simply the values of these parameters that maximize the log-likelihood for a given sequence of samples \( Y^{(k)} \). In Appendix B it is shown that the ML estimates in this case are given by

\[
m_x^{ML} = \bar{y} \overset{\Delta}{=} \frac{1}{k} \sum_{t=1}^{k} y(t), \quad (3.58)
\]

\[
\hat{C}_{xx}^{ML} = C_{nn}^{1/2} P \max (\Lambda - I, 0) P^T \left( C_{nn}^{1/2} \right)^T, \quad (3.59)
\]

where \( C_{nn}^{1/2} \) is a square matrix that satisfies \( C_{nn}^{1/2} \left( C_{nn}^{1/2} \right)^T = C_{nn} \), and \( P \) and \( \Lambda \) are the matrices of eigenvectors and eigenvalues, respectively, of the matrix \( C_{nn}^{-1/2} \hat{C}_{yy} \left( C_{nn}^{-1/2} \right)^T \), where

\[
\hat{C}_{yy} \overset{\Delta}{=} \frac{1}{k} \sum_{t=1}^{k} (y(t) - \bar{y}) (y(t) - \bar{y})^T. \quad (3.60)
\]

The ML estimate of \( m_x \) is simply the sample mean of the noisy data. An interpretation of the ML estimate of \( C_{xx} \) is in order. The matrix \( C_{nn}^{-1/2} \hat{C}_{yy} \left( C_{nn}^{-1/2} \right)^T \) is an estimate of the covariance of the noisy data after it has been transformed by \( C_{nn}^{-1/2} \) into a system of coordinates where the noise covariance is the identity matrix. The covariance estimate of the “noise-adjusted” data is then diagonalized and the identity matrix is subtracted from the matrix of eigenvalues. Those eigenvalues which are still positive are interpreted as the variances of signal components; the rest of the eigenvalues are set to zero. The ML estimate of the signal covariance is obtained by transforming the resulting matrix to the original coordinate system. From the discussion in Section 3.3.3 we recognize the above transformation performed on the data as an empirical noise-adjusted principal components (NAPC) analysis. The nonzero eigenvalues in equation 3.59 are estimates of the signal-to-noise ratios of the NAPC.

The MLDA has the six-step structure described in Section 3.4.1. To completely specify the algorithm it is necessary to specify the following: a) estimates of the mean and covariance of the noisy source, b) a procedure for computing/approximating the optimal non-negative integer bit allocation, and c) a procedure for computing the optimal quantizers. For the MLDA these items are chosen as follows:
a) The MLDA uses the ML estimates of the mean and covariance of the noisy source. Since \( \mathbf{m}_y = \mathbf{m}_x \) and \( C_{yy} = C_{xx} + C_{nn} \) by assumption, from equations 3.58 and 3.59 it follows that ML estimates of the statistics of the noisy source are given by

\[
\hat{\mathbf{m}}_{y}^{ML} = \hat{\mathbf{m}}_{x}^{ML}, \tag{3.61}
\]

\[
\hat{C}_{yy}^{ML} = \hat{C}_{xx}^{M^*} + C_{nn}. \tag{3.62}
\]

b) The optimal bit allocation is approximated by first computing the optimal continuous bit allocation given by equations 3.50 and 3.51. The resulting bit allocation is then rounded down and adjusted by performing as many “greedy” one-bit allocations as possible without exceeding the desired bit rate \( B \). A greedy one-bit allocation is performed by assigning the next available bit to the coefficient for which the distortion is reduced the most given the current bit allocation. For the greedy bit allocations the distortion of the Lloyd-Max quantizer for a unit-variance Gaussian input is used for bit allocations of 0–20 bits. For higher bit allocations the distortion is well approximated by scaling the distortion by \( 1/4 \) for every additional bit.

c) The approximation due to Roe [41] of the Lloyd-Max quantizer for a Gaussian input is used. This approximation is described in Section 3.3.4.

A concise listing of the MLDA is given below.

**Maximum-Likelihood Design Algorithm (MLDA)**


2. Put \( \hat{L}_{xy} = \hat{C}_{xx}^{ML} (\hat{C}_{yy}^{ML})^{-1} \).

3. Diagonalize the matrix \( W\hat{L}_{xy}\hat{C}_{yy}^{ML} (\hat{L}_{xy})^T (W)^T \). Let \( \mathbf{U} \) be the matrix of eigenvectors and \( \Lambda \) the matrix of eigenvalues.

4. The decoder matrix is \( \mathbf{G} = W^{-1} \mathbf{U} \), and the encoder matrix is \( \mathbf{H} = \mathbf{U}^T W\hat{L}_{xy} \).

5. For the eigenvalues obtained in step 3 approximate the optimal bit allocation by first computing Segall’s continuous bit allocation, rounding down the result, and doing greedy one-bit allocations until the bit quota \( B \) is met.
6. Compute the Lloyd-Max quantizers for the bit allocation obtained in step 5 using Roe's approximation.

**Evaluation of the MLDA**

To evaluate the effectiveness of the MLDA we applied the algorithm to four generic examples of the noisy source coding problem. The examples were designed to be simple and yet representative (at a basic level) of real multivariate data sets which can be appropriately modeled as being generated from a jointly Gaussian random distribution. In all examples the element of judgement used to evaluate the empirically-designed codebook is the increase in distortion (mean-weighted-square error) relative to the optimal codebook at the desired bit rate. This quantity can always be computed when one has knowledge of all the relevant statistics.

Let us begin by describing the examples used in the evaluation. To specify an instance of the noisy source coding problem it is necessary to specify the following quantities: the dimension of the signal and noise vectors, the mean and covariance of the signal, the mean and covariance of the noise, and the error weighting matrix \( W \). Here we set the dimension of the vectors \( N = 100 \). For the purpose of specifying the problem, one can assume without loss of generality that the mean values of the signal and noise are both zero. The actual empirical design does not make use of this assumption.

Consider first the selection of the signal and noise covariances, \( C_{xx} \) and \( C_{nn} \), respectively. For simplicity let us assume that the noise covariance is invertible. Then it is possible and convenient to effect the selection of the covariances using the NAPC representation (see Section 3.3.3). Using this representation one can write

\[
C_{xx} = P \Sigma \Gamma P^T \Sigma R^T \tag{3.63}
\]

\[
\Delta = C_{nn}^{1/2} P \Sigma P^T C_{nn}^{T/2} \tag{3.64}
\]

where \( P \) is an orthogonal matrix, \( \Gamma \) is diagonal, and \( P \Sigma^2 R^T \) is the eigen-decomposition of the noise covariance matrix \( C_{nn} \). The diagonal matrix \( \Gamma \) has the useful interpretation that its diagonal entries are the signal-to-noise ratios (SNR) of uncorrelated signal components. The diagonal matrix \( \Sigma \) has diagonal entries equal to the standard deviations of uncorrelated noise components. Note that if \( P \) is a permutation matrix and/or \( \Gamma \) is proportional to
the identity matrix then equation 3.63 implies that signal and the noise have the same correlation structure, i.e. $\mathbf{C}_{xx}$ and $\mathbf{C}_{nn}$ have the same eigenvectors. Since this is rather unlikely in practice we consider only cases in which the SNR’s are not identical. Table 3.1 lists the SNR distributions chosen for each of four simulations. Somewhat arbitrarily we have chosen the SNR distribution to be proportional to $i^{-4}$, where $1 \leq i \leq N$.\textsuperscript{4} The total SNR is about 1000 on two of the simulations and about ten on the other two simulations. On all the simulations we let $\mathbf{P}$ be a random orthogonal matrix. We set $\mathbf{R} = \mathbf{I} \left( \mathbf{C}_{nn}^{1/2} = \mathbf{\Sigma} \right)$ without loss of generality.

It remains to choose the diagonal RMS noise matrix $\mathbf{C}_{nn}^{1/2}$ and the weight matrix $\mathbf{W}$. Recall that the optimal transform codebook diagonalizes the matrix $\mathbf{WC}_{\mathbf{z}^*\mathbf{z}^*} \mathbf{W}^T$, where $\mathbf{C}_{\mathbf{z}^*\mathbf{z}^*}$ is the covariance of the MMSE of the signal given the noisy data. Using equation 3.64 and simple linear algebra we can write

$$\mathbf{WC}_{\mathbf{z}^*\mathbf{z}^*} \mathbf{W}^T = \mathbf{WC}_{nn}^{1/2} \mathbf{P} \mathbf{\Gamma}^2 (\mathbf{I} + \mathbf{I})^{-1} \mathbf{P}^T \left( \mathbf{WC}_{nn}^{1/2} \right)^T. \quad (3.65)$$

Since it is not practical for us to consider all the possible choices for $\mathbf{W}$ we would like to suggest two choices based on the following observation: for the optimal codebook the allocation of bits is done according to the eigenvalues of the matrix above. An obvious choice is $\mathbf{W} = \mathbf{I}$. From the left hand side of equation 3.65 this choice results in a codebook that allocates bits according to the variance of the signal estimate. Alternatively, the right hand side of the equation suggests one might choose the weighting matrix as $\mathbf{W} = \mathbf{C}_{nn}^{-1/2}$ so that the codebook allocates bits according to SNR since in NAPC space the signal “energy” is the SNR. This choice of the weighting matrix is also convenient because it makes the problem independent of the particular noise covariance. Referring to Table 3.1, we performed two different simulations for each choice of the weighting matrix. On the simulations where $\mathbf{W} = \mathbf{I}$ we arbitrarily chose the RMS noise matrix as $\mathbf{C}_{nn}^{1/2} = \text{diag} \left( i^{-1} \right)$, $1 \leq i \leq N$.

For each of the simulations in Table 3.1 we evaluated the expected distortion obtained with empirical codebooks designed from (noisy) data sequences of $N + 1$, $2(N + 1)$, $4(N + 1)$, and $8(N + 1)$ samples, where $N = 100$ is the vector dimension. For each sample size ten empirical codebooks were designed by applying the MLDA to different pseudo-random data.

\textsuperscript{4}A similar distribution has been empirically observed for simulated AIRS spectral data, which will be treated on a later chapter.
sequences, and the average performance of those ten codebooks was reported. The expected distortion for each empirical codebook was evaluated numerically by exercising the codebook on a separate pseudo-random data sequence of 5,000 samples. The expected distortion of the optimal codebook for each simulation was derived directly from the assumed parameters. In each case the distortion was evaluated at bit rates of 5, 10, ... , 50 bits/vector.

Figure 3-6 shows the results for the simulation corresponding to Example 1 in Table 3.1. Results for the optimal codebook are those corresponding to infinite sample size. In Figure 3-6(a) it is seen that the total distortion decreases at every bit rate for increasing sample size. However, for the smaller sample sizes the distortion does not always decrease as the bit rate increases. Some insight into the problem can be derived by considering the estimator-coder structure of the codebook. The estimation distortion shown in Figure 3-6(c) is the mean-weighted-square error between the signal and its linear estimate generated by the codebook. This component of the distortion is independent of bit rate and decreases with sample size as the estimates of the signal statistics become more accurate. Figure 3-6(b) illustrates the coding distortion, i.e. the distortion introduced by quantizing the linear estimate of the signal. This component is observed to decrease with bit rate, as expected, and approaches the corresponding component for the optimal codebook as the sample size increases. Although the estimation and coding errors are uncorrelated for the optimal codebook (see Section 3.3.1, p. 43), this is not necessarily the case for a suboptimal codebook. Figure 3-6(d) shows the correlation, i.e. the mean weighted inner product, between the estimation and coding errors for the empirical codebooks. This error correlation is termed here “cross-distortion.” Inspection of all four graphs suggests that the peculiar behavior of the total distortion is due mainly to estimation error. In particular, since the coding
distortion is relatively close to optimal, the consistently negative cross-distortion at all bit rates suggests that the signal is being over-estimated. The same behavior of the distortion components is observed in the results for Example 2 shown in Figure 3-7. The increase in distortion at higher bit rates is accentuated in this case due to a lower total SNR.

The corresponding results for examples 3 and 4 are shown in Figures 3-8 and 3-9, respectively. In those two examples the increase in total distortion with bit rate is almost imperceptible. However, one should keep in mind that for these two examples the error weighting is uniform in regular space ($W = I$). Because of the uniform weighting and the fact that the noise variance was chosen to be concentrated on a subspace of dimension much smaller than $N$ (see Table 3.1), the distortion criterion for examples 3 and 4 is much less sensitive to components of the error that lie on any subspace where the noise is small.
Put in other words, when using the weighting $W = C_{nn}^{-1/2}$, as in examples 1 and 2, error components are weighted in proportion to the RMS amplitude of the noise modes so that the sensitivity is independent of noise variance. Since the data simulated in examples 3 and 4 have the same SNR distribution as those in examples 1 and 2, respectively, the results of the former examples would have been the same as for the latter if the same weighting matrix had been used in both cases.

It was speculated above that the unusual behavior of the total distortion with bit rate observed in Figures 3-6(a) and 3-7(a) arises as a result of over-estimating the signal. Evidence supporting this claim can be obtained by comparing the empirical MMSEE matrix with the optimal one for a particular simulation. This is most easily done in NAPC space, where the aforementioned matrix is known to be diagonal (see Section 3.3.3). However,
some care must be taken in making the comparison since the empirical NAPC space is not identical to the true NAPC space. Table 3.2 lists the average (over ten trials) inner product between the first ten true and estimated NAPC basis vectors estimated for Example 1 from data samples of different size. The results indicate that only the first 4-5 empirical NAPC basis vectors are closely aligned with the true basis vectors. This would render the desired comparison useless if it were not for the fact that basis vectors for which the corresponding NAPC have low SNR are pretty much arbitrary. In this example the SNR is less than 0.5 for NAPC of order higher than six. Because the energy on the subspace spanned by those basis vectors is almost entirely due to noise, and thus nearly uniformly distributed, there is no practical basis for discriminating between directions (in terms of energy) within that subspace. Thus, in our view it is acceptable to make the comparison of the empirical and
optimal filters in their respective NAPC bases because they are approximately equal where it counts. Figure 3-10 shows the amplitudes of the coefficients of the optimal and empirical filtering matrices in NAPC space. The figure clearly shows that the empirical filters obtained for the four different sample sizes are not attenuating the data nearly as much as required. As a result, too much noise is "leaking" into the estimate of the signal. This explains why the total distortion was observed to increase with bit rate since the added bits were being used to reproduce noise.

In the next section we consider a modification of the MLDA that results in significant improvement of the empirical rate-distortion characteristic.
Table 3.2: Average absolute inner product between the first ten true and empirical NAPC basis vectors estimated from different sample sizes for Example 1 in Table 3.1. For the remaining NAPC the average absolute inner product never exceeded a value of 0.16.

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 samples</td>
<td>1.00</td>
<td>0.99</td>
<td>0.95</td>
<td>0.83</td>
<td>0.55</td>
<td>0.19</td>
<td>0.15</td>
<td>0.11</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>202 samples</td>
<td>1.00</td>
<td>0.99</td>
<td>0.98</td>
<td>0.92</td>
<td>0.74</td>
<td>0.37</td>
<td>0.21</td>
<td>0.12</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>404 samples</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>0.96</td>
<td>0.89</td>
<td>0.59</td>
<td>0.33</td>
<td>0.17</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>808 samples</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>0.98</td>
<td>0.93</td>
<td>0.79</td>
<td>0.46</td>
<td>0.19</td>
<td>0.13</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Figure 3-10: Amplitude of the MMSEE filtering matrix in NAPC space for Example 1 in Table 3.1 using the MLDA. Graph (a) shows the amplitudes of the optimal filtering coefficients for the true NAPC. Graph (b) shows the amplitudes of the empirical filtering coefficients in the empirical NAPC space.
Figure 3-11: Estimated NAPC variance distribution for Example 1 in Table 3.1.

3.4.3 Codebook Design Using Modified Maximum-Likelihood Estimates of the Signal Statistics

In this section we discuss a modification of the Maximum-Likelihood Design Algorithm (MLDA) that results in empirically-designed codebooks with much improved rate-distortion characteristics. The new algorithm is formulated taking into account the finding in Section 3.4.2 that the MLDA grossly over-estimates the signal. The modified algorithm is based on an *ad hoc* correction to the signal covariance estimate used by the MLDA (Equation 3.59). Substantial improvement in the accuracy of the signal estimate is observed in numerical simulations.

The correction of the signal covariance estimate proposed here involves a modification of the estimated signal variances in the empirical NAPC space. Recall that the ML estimate of the signal covariance matrix is given by

$$\hat{C}_{xx}^{ML} = C_{nn}^{1/2} P \max(A - I, 0) P^T \left( C_{nn}^{1/2} \right)^T,$$

(3.66)

where $P$ is the orthogonal matrix whose columns are the empirical NAPC basis vectors.
and $C^{1/2}$ is the matrix used to "noise-adjust" the noisy data. The diagonal matrix $\mathbf{A}$ has along its diagonal estimates of the variances of the NAPC. Since each of the NAPC contains, in principle, exactly one unit of noise variance it seems appropriate to estimate the variances of the signal components as above by subtracting unity from each of the diagonal elements of $\mathbf{A}$. However, when the NAPC variances are estimated from a finite data sample not all of them will be greater than unity so it becomes necessary to truncate the difference at zero as in Equation 3.66. As an illustration of the difference between empirical and true NAPC variance distributions Figure 3-11 shows the average (over ten trials) estimated NAPC variances for Example 1 in Table 3.1. The true NAPC variance distribution is the one corresponding to an infinite sample. For the smaller sample sizes the empirical distribution deviates significantly from the true distribution with many of the variances attaining values much less than unity. However, subtracting unity from the small variances and truncating the negative differences to zero is really not the problem since they most likely correspond to NAPC which are dominated by noise. The main problem is that subtracting unity from the larger NAPC variances clearly over-estimates the variances of the signal components. This explains why the codebooks designed using the MLDA in Section 3.4.2 tend to over-estimate the signal.

One would expect that the inflated NAPC variances observed in Figure 3-11 correspond to components which contain mostly noise. This can be illustrated by considering the empirical NAPC variance distribution obtained in the absence of a signal for a given finite sample size. In the absence of a signal the "noise-adjusted" data is simply a noise sequence $\tilde{n}(t), 1 \leq t \leq k$, of independent samples from the $N$-dimensional Gaussian distribution with mean zero and identity covariance matrix. Then the empirical NAPC variance distribution is the ordered (decreasing) sequence of eigenvalues $\phi^{(k)}_i, 1 \leq i \leq N$, of the biased sample covariance matrix

$$
\hat{C}_{\tilde{n}\tilde{n}} = \frac{1}{k} \sum_{t=1}^{k} (\tilde{n}(t) - \bar{n}) (\tilde{n}(t) - \bar{n})^T,
$$

(3.67)

where $\bar{n}$ is the sample mean. Because they are ordered, the mean values of the eigenvalues $\phi^{(k)}_i$ are not identical. Figure 3-12 illustrates average distributions for $N = 100$. Each of the distributions was obtained by averaging the results obtained from twenty-five pseudo-random data sequences with the indicated number of samples. A comparison with Figure 3-
Figure 3-12: Empirical NAPC variance distributions in the absence of a signal as a function of sample size. The distributions are averages of twenty-five simulations. Compare with Figure 3-11.

11 shows that both empirical NAPC variance distributions, with and without signal, are in good agreement except for those NAPC where there is significant signal content. This suggests using the empirical variance distributions obtained without the signal as the “noise floor” instead of the theoretical noise distribution.

Based on the previous observations we propose the following alternate estimates of the signal and noisy source covariance matrices. Let $\Phi^{(k)}$ be the diagonal matrix given by

$$\Phi^{(k)} \triangleq E\left\{ \text{diag}\left(\phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_N^{(k)}\right) \right\}, \quad (3.68)$$

where $\phi_i^{(k)}$, $1 \leq i \leq N$, are the ordered noise eigenvalues defined above. The modified ML estimates of the signal and noisy source covariance matrices are given by

$$\hat{C}_x^{MML} = C_{nn}^{1/2} \hat{P} \hat{P}^T \left(C_{nn}^{1/2}\right)^T, \quad (3.69)$$

$$\hat{C}_y^{MML} = \hat{C}_x^{MML} + C_{nn}, \quad (3.70)$$
where $\mathbf{P}$ and $\Lambda$ are the same as in equation 3.59, and $\hat{\Gamma}$ is the estimate of the diagonal SNR matrix given by

$$
\hat{\Gamma}_{i,i} = \begin{cases} 
\max \left( \Lambda_{1,1} - \Phi_{1,1}^{(k)}, 0 \right), & i = 1; \\
\min \left( \hat{\Gamma}_{i-1,i-1}, \max \left( \Lambda_{i,i} - \Phi_{i,i}^{(k)}, 0 \right) \right), & 1 < i \leq N.
\end{cases}
$$

(3.71)

The $\min(\cdot)$ operation is used to ensure that the SNR's are monotonically decreasing. This was motivated by simulations which showed small estimated SNR's are too noisy otherwise. The matrix of mean ordered eigenvalues $\Phi^{(k)}$ may be approximated by averaging the ordered eigenvalue distributions for several $k$-sample noise covariance matrices. We also note that an analytical expression for the joint probability density function (pdf) of the ordered eigenvalues $\phi^{(k)}$ of the sample covariance matrix is known for the case where the samples come from a multivariate Gaussian distribution with identity covariance matrix [42]. In principle, the joint pdf could be used to derive the matrix $\Phi^{(k)}$. However, the expression for the joint pdf is complicated and we have not found a way to carry out the derivation.

Based on these modified ML estimates we formulate the following alternate version of the MLDA.

**Modified Maximum-Likelihood Design Algorithm (MMLDA)**

1. Compute the estimates of the statistics of the signal and the noisy source given by equations 3.58, 3.61, 3.69, and 3.70.

2. Put $\hat{L}_{xy} = \hat{C}_{xx}^{\text{ML}} \left( \hat{C}_{yy}^{\text{ML}} \right)^{-1}$.

3. Diagonalize the matrix $\mathbf{W} \hat{L}_{xy} \hat{C}_{yy}^{\text{ML}} \hat{L}_{xy}^T \mathbf{W}^T$. Let $\mathbf{U}$ be the matrix of eigenvectors and $\Lambda$ the matrix of eigenvalues.

4. Do steps 4–6 of the MLDA.

**Evaluation of the MMLDA**

Below we show the results of applying the MMLDA to the simulations defined in Table 3.1. A comparison with the corresponding results for the MLDA in Section 3.4.2 clearly shows an improvement in the rate-distortion characteristics of the empirical codebooks obtained using the modified version of the algorithm. In particular, the increase in distortion with bit rate observed in the results for examples 1 and 2 using the MLDA is no longer observed.
Figure 3-13: Simulation results for Example 1 in Table 3.1 using MMLDA.
Figure 3-14: Simulation results for Example 2 in Table 3.1 using MMLDA.
Figure 3-15: Simulation results for Example 3 in Table 3.1 using MMLDA.
Figure 3-16: Simulation results for Example 4 in Table 3.1 using MMLDA.
3.5 Summary

In this chapter we considered the design and implementation of transform coders for noisy data. It was assumed throughout the chapter that the signal and additive noise are independent Gaussian random vectors. Two different design cases were considered: 1) when the statistics of both the signal and the noise are known, and 2) when the statistics of the signal are not known. In the first case the codebook is designed by optimization with respect to mean-weighted-square error. In the second case the statistics of the signal and the noisy data are estimated from a finite noisy data sample, and the optimal codebook for those estimates is implemented.

There are three main results in the chapter. The first result is the structure of the optimal codebook for mean-weighted-square error shown in Figure 3-3. Although the same result was derived previously by Malvar [17], our assumptions and derivation are different.

The other two results relate to the design of the codebook when the statistics of the signal are not known. First it was shown that whenever the estimates of the statistics are consistent, the codebook implemented for those estimates is asymptotically optimal as the number of data samples available for the design grows to infinity. Then empirical codebooks designed using Maximum-Likelihood (ML) estimates of the statistics were evaluated by means of simulations. Upon finding the rate-distortion characteristics of those codebooks to be poor for finite sample sizes a modification of the ML estimates was pursued. The ML estimates were modified in such a way that a better estimate of the linear MMSE matrix was obtained. The codebooks designed using the modified estimates were seen to have much better performance. Since the modified estimates lead to a better estimate of the linear MMSE matrix, they can be beneficial for noise filtering of vector data whenever the statistics of the signal are not known a priori and a limited amount of noisy data is available for estimating those statistics.
Chapter 4

Calibration of Radiometric Data

4.1 Introduction

Two kinds of calibration procedures are generally performed to convert the electrical output of a radiometer into a quantitative estimate of the radiance spectrum of a target scene. First a radiometric calibration is performed to obtain estimates of the radiance spectrum for the set of frequency bands observed by the sensor at the corresponding time. Radiometric calibration involves corrections to attenuate data variations arising as a result of changes in receiver gain and offset, and may include additional corrections for instrument-specific distortions, e.g. scan angle-dependent signal transmission. In practice gain and offset variations are estimated from periodic observations of multiple, usually two, calibration targets. After the data have been radiometrically calibrated, a spectral calibration is sometimes performed to convert the spectra to a standard set of frequencies. This may be done for satellite data for which the frequency bands drift due to system temperature changes along the orbital trajectory of the satellite. Converting the spectra to a standard set of frequencies facilitates subsequent data processing.

The subject of calibration is relevant to the thesis because the models and coding techniques we consider are intended for calibrated data. The motivation for this choice is that, whether or not data compression is performed, what is ultimately desired from the raw data is to obtain a good approximation of the ideal spectrum, i.e. one measured by an ideal, noiseless sensor that can be perfectly calibrated. If one set out to design a coding algorithm that minimizes a distortion criterion between the coded raw data and the ideal spectrum, then an implicit calibration would have to be performed by the coding algorithm.
itself since it must deliver an approximation of the ideal spectrum. Because calibration procedures tend to be *ad hoc* and are complicated by instrument-specific considerations, it is convenient, in order to simplify the design of a coding algorithm, to consider the calibration step separately, and let the coding algorithm operate on calibrated data. Provided that a good calibration procedure is used, the errors in the calibrated data may be reasonably modeled as being statistically independent of the ideal spectrum *and* the calibration data. This is the rationale for applying the independent additive noise model shown in Figure 1-1 to calibrated data as opposed to raw data. The use of such a simple model is beneficial since one need not be concerned with many instrument-specific details when designing the compression algorithm.

This chapter focuses on the problem of linear radiometric calibration, which pertains to all radiometers. Linear radiometric calibration is concerned with estimating the (approximate) transfer characteristic of the radiometer, from electrical output units to physical units, in terms of receiver gain and offset parameters which may vary with time. Other instrument-specific data corrections which may have to be included in a radiometric calibration algorithm are not discussed. The first two sections of the chapter describe calibration procedures. The instrument model in terms of gain and offset parameters, and the standard two-point (using two calibration targets) calibration approach are described in Section 4.2. In Section 4.3 we consider a variation of the standard calibration procedure using linear minimum mean-square error estimates of receiver gain and offset. For that purpose the gain and offset for each spectral channel of a radiometer are modeled as wide-sense stationary discrete-time random processes. Further approximations needed to use those estimates for calibration of real data are discussed in Section 4.4.

In Section 4.5 we perform a second-order statistical characterization of calibration errors on early data from an engineering model (EM) of the NASA Atmospheric Infrared Sounder (AIRS). Basically, this is an attempt to build the independent additive noise model of Figure 1-1 for AIRS, i.e. estimating the mean and covariance matrix of the "noise" in the calibrated data. For this purpose the calibration techniques described in sections 4.2 and 4.3 were applied to preliminary test data from the AIRS EM which were noisier than it is expected for the flight data. Unfortunately, the results of that exercise are of limited use for various reasons detailed there. In particular, the results should not be taken to be accurate estimates of the calibration accuracy of AIRS. In spite of these limitations,
Figure 4-1: Schematic diagram of a total-power radiometer.

the estimated error statistics together with nominal noise statistics provided by the AIRS Science Team are used later in the thesis as a guide to generate reasonable test cases for two coding algorithms.

4.2 Practical Linear Radiometric Calibration

In this section we discuss the common approach to linear calibration of radiometric data in a typical remote sensing application. We shall begin our discussion with a brief high-level description of the total-power radiometer.

A total-power radiometer or spectrometer is a receiver designed to measure the intensity of electromagnetic radiation or radiance of an observed target on multiple spectral bands. A schematic diagram of a total-power radiometer is shown in Figure 4-1. Radiation from a target observed by the antenna is first separated into different spectral bands. The band-pass signals are then square-law detected and integrated producing an electrical signal whose amplitude is (approximately) proportional the average power of the incoming radiation on each spectral band. In a practical system the integrated output of a each channel would be subsequently digitized. In microwave radiometers diodes are typically used as the square-law detectors whereas in infrared spectrometers photoconductive and photovoltaic solid-state detectors are most frequently used [43].

The output signal of a single-channel radiometer can be described using the following model. Let $r_a(t)$ be the radiance of a target observed by the antenna, and $g(t)$ and $b(t)$ be the receiver gain and offset, respectively, all at discrete-time $t$. The electrical output signal
at time $t$ can be modeled as

$$a(t) = g(t)r_a(t) + b(t) + n_a(t).$$  \hspace{1cm} (4.1)$$

where $n_a(t)$ is an additive noise term. Because in practice the output of a radiometer is
digital, $a(t)$ is often called the antenna counts. Here we will assume the antenna counts are
continuous in amplitude.\textsuperscript{1} In the above model the noise $n_a(t)$ shall account for the effect
of any and all sources of white noise, for example thermal or Johnson noise and shot noise
in solid-state detectors.\textsuperscript{2} The noise $n_a(t)$ is subscripted to indicate that its variance may
depend on the radiance of the observed target. All additive colored noises, i.e. noises having
non-flat spectra, for example fluctuations in background radiation due to slow instrument
temperature variations, are lumped in the offset term $b(t)$ for convenience. The offset
$b(t)$ may also include an electrical offset added as part of signal conditioning. The same
model can be applied to data from a multi-channel radiometer by making all quantities in
equation 4.1 dependent on spectral wavelength.

In remote sensing applications the radiometer is calibrated by estimating the receiver
gain and offset as a function of time using periodic observation of two or more reference
targets. Typically, an airborne or spaceborne radiometer observes a scene of interest by
scanning the antenna across the scene in a direction perpendicular to the aircraft or satel-
tite trajectory, as the case may be. In addition to the observations of the scene, observations
of the calibration or reference targets are also made during each scan cycle. Those calibra-
tion data are used to estimate the receiver gain and offset (as described shortly) so that
the observations of the scene can be converted to radiances. In most cases the reference
targets are designed to behave as black-bodies so that their radiance can be predicted from
knowledge of their physical temperatures by using Planck’s law of radiation. Cold space is
also sometimes used as a calibration reference.

Let us now describe the standard two-point calibration procedure. For the purpose of
the following discussion a scan cycle is idealized as a contiguous sequence of $N$ elementary
observation intervals or “dwell times.” Each observation of the scene or the reference targets

\textsuperscript{1}In practical instruments the digitization of the signal may introduce undesirable nonlinearities which
may need to be corrected as part of the calibration process in order to avoid consistent errors.

\textsuperscript{2}A detailed discussion of the physical mechanisms of noise in radiometric systems is beyond the scope
of this thesis. For an introductory discussion of noise on solid-state infrared detectors the reader is referred
to [43].
takes a single dwell time. We assume that the scan starts with single observations of “hot” and “cold” reference targets followed by a sequence of observations of the scene at different scanning angles. While successive observations of the scene at adjacent scan angles are separated by a single dwell time, the observations of the reference targets may be separated from each other and from the scene observations by multiple dwell times. Let \( h(kN) \) and \( c(kN + \Delta t) \) denote the output of the radiometer observed when viewing the hot and cold reference targets at times \( kN \) and \( kN + \Delta t \), respectively, where \( N \) is the number of dwell times per scan cycle, \( k \) is the scan index, and \( \Delta t < N \). Also let \( r_h(kN) \) and \( r_c(kN + \Delta t) \) be the radiances of the hot and cold calibration targets, respectively, at the indicated times. Using the model in equation 4.1 \( h(kN), r_h(kN), c(kN + \Delta t), \) and \( r_c(kN + \Delta t) \) satisfy

\[
\begin{align*}
h(kN) &= g(kN)r_h(kN) + b(kN) + n_h(kN), \quad (4.2) \\
c(kN + \Delta t) &= g(kN + \Delta t)r_c(kN+\Delta t) + b(kN + \Delta t) + n_c(kN + \Delta t), \quad (4.3)
\end{align*}
\]

where \( n_h(\cdot) \) and \( n_c(\cdot) \) represent additive white noises the variances of which may depend on the radiances of the corresponding reference target. Assuming that the receiver gain and offset do not change appreciably during a time interval of duration \( \Delta t \), \( g(kN) \) and \( b(kN) \) are estimated by setting \( g(kN) = g(kN + \Delta t) \) and \( b(kN) = b(kN + \Delta t) \) in equations 4.2 and 4.3 and inverting the equations ignoring the additive white noise terms. The resulting estimates \( \hat{g}(kN) \) and \( \hat{b}(kN) \) are given by

\[
\begin{align*}
\hat{g}(kN) &= \frac{h(kN) - c(kN + \Delta t)}{r_h(kN) - r_c(kN + \Delta t)}, \quad (4.4) \\
\hat{b}(kN) &= \frac{r_h(kN)c(kN + \Delta t) - r_c(kN + \Delta t)h(kN)}{r_h(kN) - r_c(kN + \Delta t)}. \quad (4.5)
\end{align*}
\]

To estimate the radiances of the scene observed when viewing a scene at time \( kN + l, \Delta t < l < N \), one first obtains estimates \( \hat{g}(kN + l) \) and \( \hat{b}(kN + l) \) by means of some kind of smoothing and/or interpolation of the two-point calibration estimates of receiver gain and offset in equations 4.4 and 4.5. A common procedure is to filter \( \hat{g}(kN) \) and \( \hat{b}(kN) \) (or \( \hat{g}(kN)^{-1} \) and \( \hat{b}(kN)\hat{g}(kN)^{-1} \)) using a zero-phase rectangular or triangular FIR filter to obtain smoothed estimates of the corresponding quantities, and to either use them as the estimates of receiver gain and offset for all views of the scene during the corresponding scan cycle or to linearly interpolate them to obtain estimates of the receiver gain and offset at the desired times.
Alternatively, one may first smooth and/or interpolate $h(kN)$, $c(kN + \Delta t)$, $r_h(kN)$, and $r_c(kN + \Delta t)$ and then obtain the estimates of the gain and offset from equations 4.4 and 4.5 using the smoothed/interpolated data. Finally, the calibrated scene radiance $\hat{r}_a(kN + l)$ is obtained by inverting equation 4.1 ignoring the additive white noise term and using $\hat{g}(kN + l)$ and $\hat{b}(kN + l)$ in place of the true gain and offset, i.e.

$$\hat{r}_a(kN + l) = \left( a(kN + l) - \hat{b}(kN + l) \right) / \hat{g}(kN + l). \tag{4.6}$$

### 4.3 Linear MMSEE of Gain and Offset

In this section we derive linear minimum mean-square error estimates (MMSEE) of receiver gain and offset obtained from a finite sequence of observations of two reference targets. The linear MMSEE can be used in place of other ad hoc estimates for the purpose of linear radiometric calibration. Because the estimates of receiver gain and offset typically used in practice are based on linear transformations of the calibration data, the use of the linear MMSEE could result in better calibrated radiances.

Our derivation of the linear MMSEE is based on the data model introduced in Section 4.2, embodied by equations 4.1, 4.2 and 4.3. We shall use the same notation except that the times (modulo $N$) at which the hot and cold calibration targets are observed within each scan will be denoted by $l_h$ and $l_c$, respectively. The receiver gain and offset are modeled as wide-sense stationary (WSS) discrete-time random processes [29]. It is assumed that the white noises $n_h(t)$ and $n_c(t)$ in equations 4.2 and 4.3 are independent of $g(\tau)$ and $b(\tau)$ for all $\tau$, and that their variances do not depend on the instantaneous values of the radiances of the calibration targets. Finally, it is assumed that the radiances of the calibration targets are known at all the necessary times.

We want to derive the linear MMSEE of the receiver gain and offset at the $l^{th}$ dwell time of the $k^{th}$ scan given the closest $2M$ pairs of calibration views. For all scene views within the $k^{th}$ scan, the closest $2M$ pairs of calibration views are precisely those on the current scan, the previous $M - 1$ scans, and the next $M$ scans. For compactness of notation it is convenient to vectorize these quantities. Let us start by defining a vector $y(k)$ containing the calibration views of the $k^{th}$ scan,

$$y(k) \triangleq [h(kN + l_h) \ c(kN + l_c)]^T, \tag{4.7}$$

84
and a vector \( x(k) \) containing the gain and offset values at the corresponding times,

\[
x(k) \triangleq [g(kN + l_h) \ b(kN + l_h) \ g(kN + l_c) \ b(kN + l_c)]^T.
\] (4.8)

The vectors \( x(k) \) and \( y(k) \) are related through equations 4.2 and 4.3. These two equations can be written in vector form by defining a matrix \( R(k) \),

\[
R(k) \triangleq \begin{bmatrix}
r_h(kN + l_h) & 1 & 0 & 0 \\
0 & 0 & r_c(kN + l_c) & 1
\end{bmatrix},
\] (4.9)

and a noise vector \( v(k) \),

\[
v(k) \triangleq [n_h(kN + l_h) \ n_c(kN + l_c)]^T,
\] (4.10)

so that

\[
y(k) = R(k) x(k) + v(k).
\] (4.11)

Now collect \( y(j), x(j), R(j), \) and \( v(j) \) for all \( j \) such that \( k - M + 1 \leq j \leq k + M \) into the following matrices:

\[
y^{(M)}(k) \triangleq \begin{bmatrix}
y(k + M)^T & y(k + M - 1)^T & \cdots & y(k - M + 1)^T
\end{bmatrix}^T,
\] (4.12)

\[
x^{(M)}(k) \triangleq \begin{bmatrix}
x(k + M)^T & x(k + M - 1)^T & \cdots & x(k - M + 1)^T
\end{bmatrix}^T,
\] (4.13)

\[
v^{(M)}(k) \triangleq \begin{bmatrix}
v(k + M)^T & v(k + M - 1)^T & \cdots & v(k - M + 1)^T
\end{bmatrix}^T,
\] (4.14)

\[
R^{(M)}(k) \triangleq \text{diag}(R(k + M) R(k + M - 1) \ldots R(k - M + 1))
\] (4.15)

With these definitions one can write equation 4.11 in vectorized form as

\[
y^{(M)}(k) = R^{(M)}(k) x^{(M)}(k) + v^{(M)}(k).
\] (4.16)

This equation relates the \( 2M \) pairs of calibration views \( (y^{(M)}(k)) \) closest to all scene views of the \( k^{th} \) scan to the values of receiver gain and offset \( (x^{(M)}(k)) \) at the corresponding times.

Using the definitions above, the problem of interest is to estimate the gain \( g(kN + l) \) and the offset \( b(kN + l) \) given the calibration data \( y^{(M)}(k) \). The linear MMSEE \( \hat{g}_{LS}(kN + l) \)
and $\hat{b}_{LS}(kN + l)$ are given by [29]

$$
\begin{bmatrix}
\hat{g}_{LS}(kN + l) \\
\hat{b}_{LS}(kN + l)
\end{bmatrix} = L(k, l) \bar{y}(M)(k) + E\{g(t)\} \\
E\{b(t)\},
$$

(4.17)

where $L(k, l)$ is the linear MMSEE matrix given by

$$
L(k, l) = E\left\{ \begin{bmatrix}
\hat{g}(kN + l) \\
\hat{b}(kN + l)
\end{bmatrix} \bar{y}(M)(k)^T \right\} E\{\bar{y}(M)(k)\bar{y}(M)(k)^T\}^{-1}.
$$

(4.18)

In these expressions tildes are used to denote zero-mean quantities.

As a function of the scene view index $l$, the MMSEE matrix $L(k, l)$ specifies the linear interpolation of the calibration data that produces the best estimates of gain and offset in a minimum mean-square error sense. As expected, the linear MMSEE matrix is a function of the auto-covariance of the calibration data $y(M)(k)$, and the cross-covariance between the calibration data and the gain $g(kN + l)$ and offset $b(kN + l)$. From equation 4.16 the cross-covariance is given by

$$
E\left\{ \begin{bmatrix}
\hat{g}(kN + l) \\
\hat{b}(kN + l)
\end{bmatrix} \bar{y}(M)(k)^T \right\} = E\left\{ \begin{bmatrix}
\hat{g}(kN + l) \\
\hat{b}(kN + l)
\end{bmatrix} \bar{x}(M)(k)^T \right\} R^{(M)}(k)^T,
$$

(4.19)

and the auto-covariance is given by

$$
E\{\bar{y}(M)(k)\bar{y}(M)(k)^T\} = R^{(M)}(k)E\{\bar{x}(M)(k)\bar{x}(M)(k)^T\} R^{(M)}(k)^T + \text{diag} (\sigma^2_{n_h}, \sigma^2_{n_e}, \ldots, \sigma^2_{n_h}, \sigma^2_{n_e}),
$$

(4.20)

where the diagonal matrix is the auto-covariance of the white noises. In practice the matrix expectations on the right hand side of equations 4.19 and 4.20, which depend on the second-order statistics of receiver gain and offset, and the white noise variances $\sigma^2_{n_h}$ and $\sigma^2_{n_e}$ would have to be estimated from real data. Unfortunately this will often limit the extent to which the linear MMSEE can be applied to real data. This is because usually one does not have appropriate data to estimate the covariance sequence of the gain and offset on the time scale defined by the scene view index $l$. Typically one can only estimate those statistics on the longer time scale of a scan period. To circumvent this limitation one
could compute the linear MMSEE of receiver gain and offset on the longer time scale and subsequently interpolate them using some reasonable method, e.g. linear interpolation, to obtain estimates of the gain and offset on the shorter time scale. The estimation of the second-order statistics of receiver gain and offset is discussed further in Section 4.4.

One should note that even though both the gain and offset are assumed to be WSS processes, the MMSEE matrix will generally depend on the absolute scan index $k$. This is because the calibration data $\mathbf{y}(k)$ are non-stationary whenever the radiances of the calibration targets vary with time (see equations 4.11 and 4.9). In particular, their covariance depends on the radiance of the calibration targets. Therefore, when variations in the temperatures of the calibration targets are anticipated, the covariance matrices in equations 4.19 and 4.20 are not completely determined a priori, and the MMSEE matrix has to be computed on-line in order to calibrate the data in real time. This would certainly be the case for those radiometers which use an ambient (unregulated) calibration target.

To conclude this section let us evaluate the computational load demanded by the estimates in equation 4.17. Assuming that the MMSEE matrix $\mathbf{L}(k,l)$ has to be computed on-line, it is advantageous to rewrite the estimates of the gain and offset as

$$
\begin{bmatrix}
\hat{g}_{LS}(kN + l) \\
\hat{b}_{LS}(kN + l)
\end{bmatrix} = E \left\{ \begin{bmatrix}
\hat{g}(kN + l) \\
\hat{b}(kN + l)
\end{bmatrix} \tilde{y}^{(M)}(k)^T \right\} \mathbf{u}(k) + \left[ E\{g(t)\} \\
E\{b(t)\} \right], \quad (4.21)
$$

where $\mathbf{u}(k)$ is the solution of the linear system

$$
E \left\{ \tilde{y}^{(M)}(k) \tilde{y}^{(M)}(k)^T \right\} \mathbf{u}(k) = \tilde{y}^{(M)}(k). \quad (4.22)
$$

This reformulation of the estimates is better because it does not require explicitly computing the matrix inverse appearing in equation 4.18, which would require about $64M^3$ multiplications and the same number of additions. In comparison, solving equation 4.22 requires approximately $64/3 M^3 + 40 M^2$ multiplications and a similar number of additions. The solution of equation 4.21 requires about $32 M^2$ multiplications and an equal number of additions. Note that these are number of computations required for a single channel.

As an example consider the real-time computation of the linear MMSEE of the gain and

---

3In fact, if one regards the radiances of the calibration targets (obtained in practice through thermometric measurements) as part of the calibration data, the estimates of receiver gain and offset given by equations 4.17 and 4.18 are strictly nonlinear.
Table 4.1: Computational load of calculating on-line the gain and offset estimates given by equation 4.21 for AIRS (2400 channels, 90 scene views per scan, 8/3 sec. scan period). Listed values are approximate total multiplications and additions in millions per second. The number of scans worth of calibration views used to derive the estimates in each case is $2M$.

offset for NASA's Atmospheric Infrared Sounder (AIRS). During a single scan lasting 8/3 seconds AIRS observes a scene at ninety scan angles. At each scan angle the raddiances of the scene on approximately 2400 different spectral channels are measured simultaneously. Table 4.1 lists the total number of operations per second required to compute the gain and offset estimates for all 2400 channels and for various values of $M$. For all the values of $M$ shown in the table the computational load is dominated by the computations required to solve equation 4.21 for all scene view indexes.

4.4 Approximate Stochastic Modeling of Gain and Offset Variations

To compute the linear MMSEE of receiver gain and offset derived above it is necessary to estimate the second-order statistics of variations in gain and offset, and the variances of the additive white noises in the calibration data. Specifically, one must supply estimates of the mean gain and offset, the auto-covariance sequences of the gain and offset, and the cross-covariance sequence between the gain and offset. In this section we address the problem of estimating those statistics from practical data. First it is argued here that if the receiver gain and offset are sufficiently slowly varying then the desired covariances can be reasonably approximated by sample averaging given a long enough training sequence of alternating radiometric observations of two constant reference targets. Secondly a procedure is described for "fitting" models to the covariance estimates by modeling the receiver gain and offset variations as the output of a minimum-phase linear time-invariant system driven by white noise. This may be beneficial when the training data is scarce in order to obtain smoother, less noisy, approximations of the covariance sequences.
4.4.1 Estimating the Statistics by Sample Averaging

Let us first consider estimating the second-order statistics of the gain and offset by sample averaging. Suppose one has the following training sequence of radiometric observations of two constant targets,

\[
h(0), c(\Delta t), h(p), c(p + \Delta t), \ldots, h(p(T - 1)), c(p(T - 1) + \Delta t). \tag{4.23}
\]

Here \(h(t)\) is the radiometer output obtained when observing the "hot" target at time \(t\), \(c(t)\) is the output obtained when observing the "cold" target at time \(t\), and \(p > \Delta t\) is the number of dwell times between successive observations of the same target. \(h(t)\) and \(c(t)\) are given by equations 4.2 and 4.3 with \(N = p\) and with the additional assumption that the radiances of the targets are constant, say \(r_h(t) = r_h\) and \(r_c(t) = r_c\), \(r_h \neq r_c\). From the training data sequence one can estimate the mean gain and offset as

\[
\begin{bmatrix}
\bar{g} \\
\bar{b}
\end{bmatrix} \triangleq \begin{bmatrix}
r_h & 1 \\
r_c & 1
\end{bmatrix}^{-1} \frac{1}{T} \sum_{t=0}^{T-1} \begin{bmatrix}
h(pt) \\
c(pt + \Delta t)
\end{bmatrix}, \tag{4.24}
\]

where \(\bar{g}\) is the estimate of the mean gain and \(\bar{b}\) the estimate of the mean offset. It can be easily verified that these estimates are unbiased.

Now consider estimating the joint covariance sequence of the receiver gain and offset, i.e. the sequence of \(2 \times 2\) matrices given by

\[
E\left\{ \begin{bmatrix}
\tilde{g}(t) \\
\tilde{b}(t)
\end{bmatrix} \begin{bmatrix}
\tilde{g}(t - p_i) \\
\tilde{b}(t - p_i)
\end{bmatrix}^T \right\} \tag{4.25}
\]

for \(i = 0, 1, 2, \ldots\) The tildes in this expression are used to denote zero-mean quantities. Note that we consider estimating the covariance sequence on the time scale on which each of the reference targets is observed. There are two basic difficulties in estimating the desired covariance sequence: 1) the gain and offset can vary between adjacent observations of the hot and cold the reference targets, 2) the calibration data contain variations not only due to receiver gain and offset but also due to additive white noise. To address the first problem suppose for the moment that the training data are noise-free \((n_h(t) = n_c(t) = 0)\). If both reference targets could be observed at the same time \((\Delta t = 0)\) then \(g(pt)\) and \(b(pt)\) for
\( t = 0, 1, \ldots, T - 1 \) could be determined exactly by

\[
\begin{bmatrix}
g(pt) \\
b(pt)
\end{bmatrix}
= \begin{bmatrix}
r_h & 1 \\
r_c & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
h(pt) \\
c(pt)
\end{bmatrix},
\]

(4.26)

and one could estimate the desired covariances by averaging outer products of these vectors. However, a radiometer can only observe one target at a time or, equivalently, one linear combination of \( g(t) \) and \( b(t) \) at any time \( t \), so that one can not determine \( g(t) \) and \( b(t) \) exactly even if the noise is zero. If one used \( c(pt + \Delta t) \) in place of \( c(pt) \) in the expression above one would obtain gain and offset estimates given by

\[
\begin{bmatrix}
\hat{g}(pt) \\
\hat{b}(pt)
\end{bmatrix}
= \begin{bmatrix}
r_h & 1 \\
r_c & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
h(pt) \\
c(pt + \Delta t)
\end{bmatrix},
\]

(4.27)

\[
= \begin{bmatrix}
r_h & 1 \\
r_c & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
r_h & 0 & 0 \\
0 & 0 & r_c & 1
\end{bmatrix}
\begin{bmatrix}
g(pt) \\
b(pt) \\
g(pt + \Delta t) \\
b(pt + \Delta t)
\end{bmatrix},
\]

(4.28)

\[
= \begin{bmatrix}
g(pt) \\
b(pt)
\end{bmatrix}
- \frac{1}{r_h - r_c}
\begin{bmatrix}
-r_c(g(pt) - g(pt + \Delta t)) - b(pt) + b(pt + \Delta t) \\
r_h r_c(g(pt) - g(pt + \Delta t)) + r_h (b(pt) - b(pt + \Delta t))
\end{bmatrix}.
\]

(4.29)

Thus, these estimates are reasonable if changes in receiver gain and offset on a time interval of duration \( \Delta t \) are small enough that the second term above can be neglected. In practice this seems to be a reasonable assumption. Additional errors in the estimates due to the presence of white noise in the training data can be mitigated by low-pass filtering the training data. This is also justified under the assumption that receiver gain and offset are slowly-varying processes. The cut-off frequency of the low-pass filter can be determined by computing the sampled Fourier spectra of the sequences \( h(pt) \) and \( c(pt + \Delta t) \) and locating a frequency above which their magnitudes are approximately flat. Let \( f(t) \) denote the impulse response of an appropriate low-pass filter, then the estimates of receiver gain and offset in
equation 4.27 become

\[
\begin{bmatrix}
\tilde{g}(pt) \\
\tilde{b}(pt)
\end{bmatrix}
= \begin{bmatrix}
r_h & 1 \\
r_c & 1
\end{bmatrix}^{-1} \begin{bmatrix}
f(t) \ast h(pt) \\
f(t) \ast c(pt + \Delta t)
\end{bmatrix}.
\] (4.30)

Using these estimates, the covariances in equation 4.25 for lags 0, p, 2p, \ldots, (T - 1)p can be estimated as

\[
E \left\{ \begin{bmatrix}
\tilde{g}(t) \\
\tilde{b}(t)
\end{bmatrix} \begin{bmatrix}
\tilde{g}(t - pi) \\
\tilde{b}(t - pi)
\end{bmatrix}^T \right\} \approx \frac{1}{T-1} \sum_{i=0}^{T-1-i} \begin{bmatrix}
\tilde{g}(pt) - \tilde{g} \\
\tilde{b}(pt) - \tilde{b}
\end{bmatrix} \begin{bmatrix}
\tilde{g}(p(t + i)) - \tilde{g} \\
\tilde{b}(p(t + i)) - \tilde{b}
\end{bmatrix}^T.
\] (4.31)

Equations 4.24 and 4.31 give estimates of the mean values and the joint covariance sequence of the gain and offset. The covariance sequence is estimated on a time scale determined by p, the time interval between successive observations of each reference target in the training data sequence. With regard to the linear MMSEE of the gain and offset derived in Section 4.3, this means that one can only compute those estimates on the time scale determined by p. Estimates of the receiver gain and offset on a shorter time scale could be obtained by further interpolating the MMSEE using, for example, linear interpolation.

The variances of the white noises \( n_h(t) \) and \( n_c(t) \), which are also needed to compute the linear MMSEE of the receiver gain and offset, can be estimated by inspecting the magnitude of the sampled Fourier spectra of \( h(pt) \) and \( c(pt + \Delta t) \), respectively, and noting the value of the high-frequency asymptote.

### 4.4.2 Covariance Fitting by Approximate Stochastic Realization

The estimates of the covariance sequence obtained by sample averaging may be too noisy, especially at large lags where there is not much data to average. One way to obtain estimates which appear less noisy is to impose the structure of a model with a reduced number of parameters. Here we discuss an algorithm for "fitting" a model to noisy estimates of a covariance sequence by modeling the stochastic process in question as the output of a minimum-phase linear time-invariant (LTI) system driven by white noise. The algorithm discussed here is a variation of the Unweighted Principal Components (UPC) algorithm of Arun and Kung [44]. In their paper Arun and Kung treated only the case of modeling a scalar stochastic process. We provide an extension of their algorithm for modeling vector
processes.

The UPC Algorithm

First let us briefly discuss the basis of the UPC algorithm. A detailed discussion of the algorithm is given in the original paper by Arun and Kung [44]. Let \( y(t), t \in \{ \ldots -1, 0, 1, \ldots \} \), be the \( n \)-dimensional vector process to be modeled. The UPC algorithm determines a minimum-phase LTI system and a white noise input which together generate an output process having correlations that approximate those of \( y(t) \). Assuming momentarily that \( y(t) \) is the output of such a system, \( y(t) \) can be expressed using the innovations representation [45] as

\[
\begin{align*}
  z(t + 1) &= A z(t) + K v(t), \\
  y(t) &= C z(t) + v(t),
\end{align*}
\]

(4.32)  
(4.33)

where \( v(t) \) is the innovation vector, and \( z(t) \) is the state vector, both at time \( t \), \( A \) is the one-step state-transition matrix, \( C \) is the observation matrix, and \( K \) is the steady-state Kalman gain. The innovations process \( v(t) \) is a stationary zero-mean white noise, and it is uncorrelated with \( z(\tau) \) for all \( \tau \leq t \). By manipulating the innovations representation it can be shown that the state vector \( z(t) \) is sufficient for linearly predicting the future output \( y_+ \),

\[
y_+ \triangleq [y(t)^T \ y(t + 1)^T \ \ldots ]^T,
\]

(4.34)

from the past output \( y_- \),

\[
y_- \triangleq [y(t - 1)^T \ y(t - 2)^T \ \ldots ]^T.
\]

(4.35)

Arun and Kung show that the minimum mean-square error (MMSE) prediction of \( y_+ \) given \( y_- \), denoted as \( y_+ \setminus y_- \), equals

\[
y_+ \setminus y_- = O \Psi y_- = Oz(t),
\]

(4.36)
where $O$ is the observability matrix,

$$
O = \begin{bmatrix}
C \\
CA \\
CA^2 \\
\vdots
\end{bmatrix},
$$

(4.37)

and $\Psi$ is given by

$$
\Psi = \left[ K \ (A - KC) \ K \ (A - KC)^2 \ K \ \ldots \right].
$$

(4.38)

Equation 4.36 implies that the dimension of the state vector equals the dimension of the space spanned by the optimal prediction $y_+ \setminus y_-$. However, in most cases the prediction $y_+ \setminus y_-$ will span an infinite subspace simply because a model with a state vector of finite dimension may not exist for $y(t)$. The UPC algorithm selects a "partial" state vector in such a way that most of the variance in the optimal prediction $y_+ \setminus y_-$ is retained. As discussed below, that choice is effected by preserving the most dominant principal components of $y_+ \setminus y_-$. Corresponding to that choice of partial state vector the UPC then generates an approximate stochastic realization of the form in equations 4.32 and 4.33. The covariance sequence of the output of the approximate system gives an approximation of the covariance sequence of the original process $y(t)$.

Let us now show how the partial state vector is selected using the UPC algorithm. First note that the optimal prediction $y_+ \setminus y_-$ can be computed using a basic result from linear MMSE estimation theory:

$$
y_+ \setminus y_- = HR^{-1}y_-,
$$

(4.39)

where $H \triangleq E\{y_+ y_+^T\}$ and $R \triangleq E\{y_- y_-^T\}$.\(^4\) This expression for $y_+ \setminus y_-$ is valid whenever $R$ is invertible, regardless of whether an exact stochastic realization, e.g. equations 4.32 and 4.33, exists. Using the expression above, the auto-covariance matrix of $y_+ \setminus y_-$ is easily shown to be $HR^{-1}H^T$. Suppose one wishes to select a partial state vector $\hat{z}(t)$ of dimension $m$. Then in analogy to equation 4.36 one defines the prediction of the future output $y_+$ from $\hat{z}(t)$ to be

$$
y_+ \setminus \hat{z}(t) = \hat{O} \Psi y_- = \hat{O} \hat{z}(t),
$$

(4.40)

---

\(^4\)In practice one would use finite estimates of $H$ and $R$ derived from a finite data sample.
where \( \hat{\Psi} \) is a matrix with \( m \) rows and \( \hat{\mathbf{O}} \) is a matrix with \( m \) columns. An optimal partial state of dimension \( m \) is then obtained by selecting \( \hat{\Psi} \) and \( \hat{\mathbf{O}} \) such that the prediction \( y_{+} \backslash \hat{z}(t) \) is optimal in a mean-square error sense, i.e. \( y_{+} \backslash \hat{z}(t) \) should be the best linear predictor of rank \( m \). Let \( \mathbf{U} \) and \( \Sigma^2 \) be the matrices of eigenvectors and eigenvalues of \( \mathbf{H}\mathbf{R}^{-1}\mathbf{H}^T \), with the eigenvalues sorted in descending order along the main diagonal of \( \Sigma^2 \). The optimal rank-\( m \) predictor is

\[
y_{+} \backslash \hat{z}(t) = \mathbf{U}_m \mathbf{U}_m^T \mathbf{H}\mathbf{R}^{-1} y_{-}, \tag{4.41}
\]

\[
y_{+} \backslash \hat{z}(t) = \mathbf{U}_m \mathbf{U}_m^T y_{+} \backslash y_{-}, \tag{4.42}
\]

where \( \mathbf{U}_m \) is the matrix having as its columns the first \( m \) eigenvectors. The last equality above indicates that the prediction based on an optimal partial state of dimension \( m \) is the projection of the prediction based on the complete past onto the subspace spanned by its first \( m \) Principal Components. Comparing equations 4.40 and 4.41 one can let \( \hat{\mathbf{O}} = \mathbf{U}_m \mathbf{D} \) and \( \hat{\Psi} = \mathbf{D}^{-1} \mathbf{U}_m^T \mathbf{H}\mathbf{R}^{-1} \), where \( \mathbf{D} \) is an arbitrary invertible \( m \times m \) matrix. Different choices of \( \mathbf{D} \) yield different optimal partial states \( \hat{z}(t) \). Arun and Kung suggest setting

\[
\mathbf{D} = \text{diag} \left( \Sigma_{11}^{1/2}, \Sigma_{22}^{1/2}, \ldots, \Sigma_{mm}^{1/2} \right), \tag{4.43}
\]

which leads to a so-called balanced realization. Then the partial state vector has auto-

covariance matrix

\[
\mathbf{P} \triangleq \mathbb{E} \{ \hat{z}(t)\hat{z}(t)^T \} = \text{diag} (\Sigma_{11}, \Sigma_{22}, \ldots, \Sigma_{mm}). \tag{4.45}
\]

Next let us discuss how to estimate the system and input parameters. For convenience let us write the approximate stochastic realization as

\[
\hat{z}(t + 1) = \hat{\mathbf{A}}\hat{z}(t) + \hat{\mathbf{K}}\hat{\mathbf{v}}(t), \tag{4.46}
\]

\[
\hat{y}(t) = \hat{\mathbf{C}}\hat{z}(t) + \hat{\mathbf{v}}(t), \tag{4.47}
\]

where all quantities are approximations of their counterparts in equations 4.32 and 4.33.
The covariance sequence of the output $\hat{y}(t)$ has the form

$$
E\{\hat{y}(t)\hat{y}(t - \tau)^T\} = (\hat{C}\hat{P}\hat{C}^T + Q)\delta(\tau) + \hat{C}\hat{A}^{\tau-1}Gu(\tau - 1) + \hat{C}\hat{A}^{\tau-1}Gu(-\tau - 1),
$$

(4.48)

where $u(\tau)$ is the unit step sequence, $\delta(\tau)$ is the unit sample sequence, as before $P$ is the auto-covariance matrix of $\hat{z}(t)$, $Q$ is the auto-covariance matrix of the innovation $\hat{v}(t)$, and $G$ is defined as

$$
G \triangleq \hat{A}\hat{P}\hat{C}^T + \hat{K}Q.
$$

(4.49)

Thus, a total of five matrices needs to be estimated to specify this covariance sequence.

Using the UPC algorithm, $P$, $\hat{C}$, $\hat{K}$, and $\hat{A}$ are obtained as follows:

1. $P = \text{diag}(\Sigma_{11}, \Sigma_{22}, \ldots, \Sigma_{mm})$ as discussed previously.

2. $\hat{C}$ and $\hat{K}$ are obtained from $\hat{O} = U_mD$ and $\hat{\Psi} = D^{-1}U_m^T\hat{H}R^{-1}$ with $D$ given by equation 4.43. By comparison with equations 4.37 and 4.38 one obtains

$$
\hat{C} = \text{first } n \text{ rows of } \hat{O},
$$

(4.50)

$$
\hat{K} = \text{first } n \text{ columns of } \hat{\Psi}.
$$

(4.51)

3. $\hat{A}$ is obtained from the estimate of the observability matrix $\hat{O}$. From equation 4.37 the observability matrix should have a special structure involving increasing powers of $A$. Specifically, let $O^{tn}$ equal $O$ minus its first $n$ rows. Then from equation 4.37 the following equality holds:

$$
OA = O^{tn}.
$$

(4.52)

However, in practice the approximate observability matrix $\hat{O}$ will not have the required structure and it will have only a finite number of rows. To obtain $\hat{A}$, let $\hat{O}^{tn}$ equal $\hat{O}$ minus its first $n$ rows and let $\hat{O}^{tn}$ equal $\hat{O}$ minus its last $n$ rows. $\hat{A}$ is obtained as the least-squares solution of

$$
\hat{O}^{tn}\hat{A} = \hat{O}^{tn}.
$$

(4.53)

To completely specify the covariance sequence in equation 4.48 one must determine the
input auto-covariance \( Q \). Because Arun and Kung only treated the case of modeling a scalar process, they did not specify how to obtain the input auto-covariance since in that case it is merely the variance of the input. Instead they obtained a normalized model for a unit variance input. In the case of an \( n \)-dimensional vector process \( Q \) would be an \( n \times n \) matrix and the concept of a normalized model would not make sense.

Extension of the UPC Algorithm for Modeling Vector Processes

To use the UPC algorithm to model a vector process it is necessary to estimate the input auto-covariance matrix \( Q \) in addition to the system parameters \( A, K, \) and \( C \). A straightforward estimate of \( Q \) can be obtained within the framework of the UPC algorithm. Note that the input (innovation) \( v(t) \) is, by definition, the error associated with the prediction \( y(t) \hat{\hat{a}}(t) \). In the UPC algorithm one computes the prediction \( y_+ \hat{\hat{a}}(t) \), which has an associated error covariance given by

\[
E \left\{ (y_+ - y_+ \hat{\hat{a}}(t)) (y_+ - y_+ \hat{\hat{a}}(t))^T \right\} \\
= E \left\{ y_+ y_+^T \right\} - E \left\{ (y_+ \hat{\hat{a}}(t)) (y_+ \hat{\hat{a}}(t))^T \right\} \\
= R - \hat{O} E \left\{ \hat{\hat{a}}(t) \hat{\hat{a}}(t)^T \right\} \hat{O}^T \\
= R - \hat{O} P \hat{O}^T. \tag{4.57}
\]

The covariance of \( v(t) \) is simply the top left \( n \times n \) sub-matrix of the error covariance matrix above. Thus, from equations 4.57 and 4.50 one obtains

\[
Q = R_n - \hat{\hat{C}} P \hat{\hat{C}}^T, \tag{4.58}
\]

where \( R_n \) is defined as the top left \( n \times n \) sub-matrix of \( R \). Referring to equation 4.48 one can see that with this choice of \( Q \) the value of the "fitted" covariance sequence at zero lag exactly matches \( R_n \), which is the given estimate of \( E \left\{ y(t) y(t)^T \right\} \).

Unfortunately, the proposed estimate of the input auto-covariance may not be consistent with the estimates \( P, \hat{A}, \hat{K} \) obtained by the UPC algorithm. Specifically, in order for the modeled process to be wide-sense stationary (WSS) the auto-covariance of the state vector...
\( \hat{z}(t) \) must be a constant, i.e.

\[
E\{ \hat{z}(t)\hat{z}(t)^T \} = P \text{ for all } t. \tag{4.59}
\]

Then, computing the auto-covariance matrix for the vectors on both sides of equation 4.46 one obtains

\[
P = \hat{A}P\hat{A}^T + \hat{K}Q\hat{K}^T. \tag{4.60}
\]

An equation of this type is called an algebraic Lyapunov equation [45]. This condition will generally not be satisfied by \( P, \hat{A}, \hat{K}, \text{ and } Q \) when they are estimated as described above. As a consequence, the “fitted” covariance sequence in equation 4.48 would not be guaranteed to be positive-real.\(^5\)

One way to produce a consistent set of parameters would be to obtain \( P \) and \( \hat{C} \) using the UPC algorithm, compute \( Q \) from equation 4.58, and then choose \( \hat{A} \) and \( \hat{K} \) so that equation 4.60 is satisfied. We propose the following procedure for obtaining \( \hat{A} \) and \( \hat{K} \). Pre-multiplying equation 4.60 by \( \hat{O}^{in} \) and post-multiplying it by \( (\hat{O}^{in})^T \) and rearranging terms one obtains

\[
\hat{O}^{in}\hat{K}Q\hat{K}^T (\hat{O}^{in})^T = \hat{O}^{in}P (\hat{O}^{in})^T - \hat{O}^{in}\hat{A}P\hat{A}^T (\hat{O}^{in})^T. \tag{4.61}
\]

Then, since one would like \( \hat{A} \) to satisfy \( \hat{O}^{in}\hat{A} = \hat{O}^{in} \) as closely as possible, we define an error vector \( e \) such that

\[
\hat{O}^{in}\hat{K}Q\hat{K}^T (\hat{O}^{in})^T = \hat{O}^{in}P (\hat{O}^{in})^T - \hat{O}^{in}P (\hat{O}^{in})^T + e. \tag{4.62}
\]

We choose \( \hat{K} \) to minimize the square-magnitude of the error \( e \). Noting that \( \hat{K}Q\hat{K}^T \) is symmetric positive semi-definite one can write

\[
\hat{K}Q\hat{K}^T = FF^T \tag{4.63}
\]

for some real \( m \times n \) matrix \( F \). Thus, the desired matrix \( \hat{K} \) can be obtained by minimizing \( ||e|| \) with respect to \( F \) and letting \( \hat{K} = FQ^{-1/2} \), where \( Q^{1/2} \) is an \( n \times n \) square-root of

\[^{5}\text{This condition is necessary and sufficient in order for a sequence to be the covariance sequence of a wide-sense stationary stochastic process [29].}\]
The minimization of $\|e\|$ with respect to $F$ can be effected by steepest descent or some other method of unconstrained minimization.

Once $\hat{K}$ has been obtained we determine $\hat{A}$ as follows. First we compute the Cholesky factorization of the matrix

$$\hat{A}P\hat{A}^T = P - \hat{K}Q\hat{K}^T,$$  \hspace{1cm} (4.64)

which yields an $n \times n$ lower-triangular matrix $L$ such that $\hat{A}P\hat{A}^T = LL^T$. By doing this we have determined $\hat{A}$ up to an orthogonal factor, i.e. $\hat{A} = LVP^{-1/2}$ for some orthogonal matrix $V$. Again, since we want $\hat{O}^{ln}\hat{A} = \hat{O}^{tn}$ to be satisfied as close as possible, we write

$$\hat{O}^{ln}\hat{A}P^{1/2} \approx \hat{O}^{tn}P^{1/2},$$ \hspace{1cm} (4.65)

$$\hat{O}^{ln}LV \approx \hat{O}^{tn}P^{1/2}. \hspace{1cm} (4.66)$$

We choose the orthogonal factor $V$ as the least-squares solution to the last expression above. It can be shown that the solution is

$$V = \Phi \Gamma^T,$$ \hspace{1cm} (4.67)

where $\Phi \Gamma^T$ is the Singular Value Decomposition [38] of the matrix $(\hat{O}^{ln}L)^T \hat{O}^{tn}P^{1/2}$.

We should point out that there is one problem with the procedure outlined above for computing $\hat{K}$ and $\hat{A}$. Namely, the right-hand side of equation 4.64 is not guaranteed to be positive semi-definite, as it should be, when one chooses $\hat{K}$ as the least-squares solution to equation 4.62. If $P - \hat{K}Q\hat{K}^T$ is not positive semi-definite then equation 4.64 does not make sense (since the left-hand side must be positive semi-definite for $P$ positive definite) and one can not solve for $\hat{A}$. Thus, strictly speaking one should enforce that constraint when solving for $\hat{K}$. Finding the least-squares solution to equation 4.62 under that constraint is more complicated and will not be discussed here. In practice we have found this not to be a serious limitation. In most cases the unconstrained least-squares solution $\hat{K}$ also yields a matrix $P - \hat{K}Q\hat{K}^T$ which is positive semi-definite. If it does occur that $P - \hat{K}Q\hat{K}^T$ is not positive semi-definite, it is often possible to circumvent the problem by reducing the dimension of the state vector ($m$) and re-computing the parameters.

On the other hand, we should also point out an important advantage of using equation 4.64 to derive the state-transition matrix $\hat{A}$. If that equation holds then any matrix
\( \hat{A} \) that satisfies it does not have any eigenvalues of magnitude greater than unity. To show this let \( x \) be an eigenvector of \( \hat{A}^T \) with eigenvalue \( \lambda \). The following sequence of inequalities shows that \( |\lambda| \leq 1 \):

\[
 x^H \hat{A} P \hat{A}^T x = x^H P x - x^H \hat{K} \hat{Q} \hat{K}^T x, \quad (4.68)
\]
\[
 |\lambda|^2 x^H P x = x^H P x - x^H \hat{K} \hat{Q} \hat{K}^T x, \quad (4.69)
\]
\[
 |\lambda|^2 x^H P x \leq x^H P x, \quad (4.70)
\]
\[
 |\lambda|^2 \leq 1, \quad (4.71)
\]

where \( x^H \) denotes the conjugate transpose of \( x \). This implies that the "fitted" covariance sequence in equation 4.48 is guaranteed to be bounded. This would not be the case if one used the matrix \( \hat{A} \) obtained using the UPC algorithm (see equation 4.53); that matrix may have eigenvalues of magnitude greater than unity.

In the next section the extension of the UPC algorithm proposed here is used to model the joint covariance sequence of receiver gain and offset variations in radiometric data from NASA's Atmospheric Infrared Sounder.

4.5 Application: Statistical Characterization of Radiometric Calibration Errors on AIRS Data

To use AIRS data as a test case for our data compression algorithms we need to obtain a second-order (mean and covariance) statistical characterization of radiometric calibration errors on those data. The objective of this section of the thesis is to obtain such a characterization using preliminary test data from the AIRS Engineering Model (EM).\(^6\) The EM test data and pre-processing performed to correct some defects present on these data are described in Section 4.5.1. In Section 4.5.2 estimates of the mean and covariance of calibration errors on AIRS data are obtained by performing two different linear calibrations of the EM test data. One calibration used the linear minimum mean-square error (MMSE) filters described in Section 4.3. The other calibration used a zero-phase triangular finite impulse response filter to smooth instantaneous estimates of the gain and offset obtained using the calibration data from each scan. Estimates of the calibration error statistics were

\(^6\)Similar data from the AIRS Protoflight Model was not available at the time this analysis was conducted.
obtained for different filter lengths. Both calibrations gave practically identical calibration accuracies.

Before proceeding with this analysis it is appropriate to point out some limitations of the results reported below. First, we were not able to estimate the radiometric calibration accuracy for all of the AIRS channels because only five out of twelve detector array modules were populated on the AIRS EM. In addition, the estimates of calibration accuracy obtained here are partial because they only account for the effect of gain and offset variations, and thermal noise. In particular, residual errors due to non-linearities in the response of the infrared detectors and instrument electronics, and scan-angle-dependent effects, both of which are expected to be significant in the case of AIRS [46], are not quantified here. According to the AIRS Instrument Calibration Plan [46], errors due to non-linearities will be negligible at the beginning of the mission but will dominate the radiometric calibration error after five years in orbit; they are caused by the radiation environment in orbit. Residual errors due to scan-angle-dependent effects are expected to amount to a fraction of the rms thermal noise.

4.5.1 Description and Pre-processing of the AIRS EM Test Data

The AIRS EM is a prototype of the AIRS instrument built for testing purposes only. In particular, the performance of the final AIRS Flight Unit is expected to be better than that of the EM. The EM was provided with infrared detectors for only 1122 spectral channels out of the 2378 channels of the full AIRS instrument. Table 2.1 lists the spectral ranges covered by each of twelve semiconductor (HgCdTe) detector array modules on the full AIRS instrument. The detectors on modules M-1 through M-10 are photovoltaic detectors and those on modules M-11 and M-12 are photoconductors. Only modules M-1, M-4, M-7, M-10, and M-12 were populated on the AIRS EM.

The preliminary EM test data\(^7\) used in this study consists of two two-minute sequences of radiometric observations of two reference targets at nominal temperatures of 301 K and 166 K, respectively. During each test sequence the reference targets were observed repeatedly following the timing of the AIRS scan pattern shown in Figure 4-2. No actual scanning took place during the tests, however. The dwell time of each observation was 224

---

\(^7\)The AIRS EM test data was provided by the AIRS Science Team via anonymous ftp to ir.jpl.nasa.gov. The files used are 640.dat and 641.dat in the directory /pub/EM.data/radiometric.
msec. The target at 301 K was observed during the calibration and scene views of the scan pattern. During the cold space views the aperture through which this target was observed was closed by a shutter cooled to a nominal temperature of 166 K, thus providing a second reference target. Each scan cycle of observations lasted 8/3 sec for a total of 45 cycles on each two-minute test run.

Unfortunately, a substantial part of the test data was unusable for various reasons. There were 104 channels in which the data for the hot target (301 K) or for both targets was constant throughout the tests. In a few cases this was due to saturation of the analog-to-digital converter but in most of those channels the cause seemed to be a dead detector. In addition, there were 205 channels in which the signal for the 301 K target was very close to or smaller than the signal from the 166 K target, implying a very small or negative gain on those channels. Among these were all the channels from module M-7. Another kind of aberration observed on some channels was the occurrence of noticeable step-like variations in the signal. Examples of this phenomenon are illustrated in Figures 4-3 and 4-4. Detectors which exhibit this behavior are often referred to as "blinkers" [47]. A total of 58 channels were identified as having "blinker" detectors. As seen in the examples the characteristics of the blinking behavior observed in different channels varied significantly. All the "blinkers" found in the test data correspond to photovoltaic detectors or photodiodes, which are used on modules M-1 through M-10 of AIRS. In photodiodes this behavior is thought to be caused by defects in the p-n junction (Reference needed.). None of the photoconductive detectors in module M-12 of the AIRS EM exhibited noticeable signs of blinking.

Ignoring all data having the problems described above left a total of 754 channels for subsequent analysis. However, there was still a minor defect in the views of the cold shutter on all of the remaining channels. Figures 4-5 through 4-7 show data from one of the test runs for six of the remaining channels. One can observe in the figures that at certain times
Figure 4-3: Examples of data from “blinker” detectors on the AIRS EM.
Figure 4-4: Examples of data from "blinker" detectors on the AIRS EM.
the views of the cold shutter are biased high. The cause appears to be that the shutter did not close properly at all times thus leaving the hot target partially exposed [48]. The data are consistent with this explanation since the corrupted cold views occur synchronously on all the channels.

To correct the corrupted cold views the following *ad hoc* procedure was used. First we noted the times at which the cold views were corrupted on each test run. For example, in Figure 4-5 we observe six different times on which the cold views are too high. In each instance four consecutive cold views were corrupted. The grouping of four arises from the repetition of the AIRS scan pattern, which has two cold views both at the beginning and the end of each scan (see Figure 4-2). To correct each group of four consecutive corrupted cold views first the average value of the group was subtracted from each of them. Then the
Figure 4-6: Examples of data from one of the AIRS EM test runs (file 640.dat).
Figure 4-7: Examples of data from one of the AIRS EM test runs (file 640.dat).
Figure 4-8: Illustration of the data in Figure 4-5 after the observations of the cold target were corrected.

closest groups of four non-corrupted cold views to the left and to the right of the corrupted views were located and the average of those eight non-corrupted cold views was added to the each of the corrupted views. Figure 4-8 shows the data in Figure 4-5 after the cold views were corrected.

4.5.2 Estimates of AIRS Radiometric Calibration Error Statistics

Estimates of the mean and covariance of radiometric calibration errors are obtained in this section for some of the AIRS channels. Estimates were obtained for those (754) channels for which there was good data in the AIRS EM test runs described in Section 4.5.1. The radiometric calibration accuracy of AIRS is estimated here by calibrating the AIRS EM test data and evaluating the calibration errors on those data. Specifically, only those views of the
hot target (≈ 301 K) corresponding to scene views in the AIRS scan pattern (see Figure 4-2) were calibrated. The views of the hot and cold targets corresponding to calibration target views and cold space views were used to estimate the gain and offset.

To calibrate the EM test data first it was necessary to estimate the true radiances of the hot and cold targets. Assuming both targets behave as black-bodies their radiances can be determined from their physical temperatures. Planck's Law predicts that a black-body at an absolute temperature \( T \) (K) emits electromagnetic radiation of radiance \( \rho_\lambda(T) \) given by

\[
\rho_\lambda(T) = \frac{2 \times 10^5 h c^2}{\lambda^3 (e^{hc/k\lambda T} - 1)} \text{ (mW \cdot m}^{-2} \cdot \text{str}^{-1} \cdot \text{cm)}
\]  

(4.72)

where \( c \) (m/s) is the speed of light, \( h \) (J-s) is Planck's constant, \( k \) (J/K) is Boltzmann's constant, and \( \lambda \) (m) is the wavelength in free space. It was also necessary for us to make some assumption about the temperatures of the targets since these were not monitored during the tests. One possibility is to assume that the hot and cold targets remained at their nominal temperatures, 301 K and 166 K, throughout the tests. However, the EM test data display variations which suggest that the temperature of the hot target may have varied slowly during the test runs. Specifically, the observations of the hot target on most of the channels from modules M-1 and M-4 display the same slowly-varying pattern. This is seen on three of the channels shown in Figures 4-5 and 4-6. Coincidentally, the same slow drift is less noticeable at the longer wavelengths where the radiance is not as sensitive to changes in the temperature of the target. If one made the assumption that the temperature of the hot target was constant that would attribute the fluctuations observed in the data to gain variations; they could not be additive noise because the same pattern is not observed in the observations of the cold target.

The following simple procedure was performed to determine if either temperature or gain variations alone could account for the slow fluctuations in the observations of the hot target. First a smoothed interpolated version of the cold target observations was subtracted from the hot target observations to remove any significant offset variations. The cold target observations in the test data varied sufficiently slowly that a linear fit to the data seemed like a reasonable guess. Assuming that the offset was exactly cancelled we had obtained at
all times \( t \) where the hot target was observed a quantity \( \Delta(t) \) given by

\[
\Delta(t) = g(t) (r_h(t) - r_c(t)) + n_h(t), \tag{4.73}
\]

\[
= g(t) (\rho_\lambda(T_h(t))) - \rho_\lambda(T_c)) + n_h(t), \tag{4.74}
\]

where \( T_h(t) \) denotes the temperature of the hot target and \( T_c \) denotes the temperature of the cold target. \( T_c \) was set to its nominal value of 166 K. Let \( \bar{\Delta} \) denote the average value of \( \Delta(t) \) on a given test run and let \( \bar{g} = \bar{\Delta}/(\rho_\lambda(301) - \rho_\lambda(165)) \). To explain the fluctuations in the observations of the hot target as gain variations we set \( T_h(t) = 301 \) in equation 4.74 and estimated the gain variation \( (g(t) - \bar{g})/\bar{g} \) as

\[
\frac{(g(t) - \bar{g})}{\bar{g}} \approx \frac{(\Delta(t) - \bar{\Delta})}{\Delta}. \tag{4.75}
\]

On the other hand, to explain the same fluctuations as the result of temperature variations we set \( g(t) = \bar{g} \) in equation 4.74 and estimated the temperature variation \( T_h(t) - 301 \) as

\[
T_h(t) - 301 \approx \frac{(\Delta(t) - \bar{\Delta})}{\bar{g} \frac{d\rho_\lambda}{dT} \bigg|_{T=301}}, \tag{4.76}
\]

where \( d\rho_\lambda/dT \) is the derivative with respect to temperature of the Planck function in equation 4.72. The estimates of gain variations and temperature variations obtained using data from two different channels on one of the test runs are shown in Figure 4-9. The raw data for the same channels are shown in Figures 4-5 and 4-6. The estimates of gain variations on the two different channels are clearly different. Since it seems unlikely that different gain variations would give rise to fluctuations in the data which look so similar, this not a plausible explanation for those fluctuations. On the other hand, the estimates of the temperature variation, although noisy, seem to be consistent with each other. Estimates of the temperature variation obtained from many other channels looked very similar to the ones shown in Figure 4-9.

Based on the comparison described above the explanation that the slow fluctuations in the observations of the hot target are due to variations in its temperature was favored. Figure 4-10 shows smoothed estimates of the temperature of the hot target during both test runs. The estimates were obtained by first computing the estimates on equation 4.76 using data from 162 channels in the spectral range of 3.73 \( \mu \text{m} \)-4.33 \( \mu \text{m} \). Low-order polynomials
Figure 4-9: Two possible explanations of the fluctuations of the observations of the hot target are illustrated. The two graphs on top show estimated variations in the temperature of the target that would explain the fluctuations in the data. The two graphs on the bottom show estimated gain variations that would account for the fluctuations in the data. The estimates were obtained using data from the indicated channels; which are shown in Figures 4-5 and 4-6.
Figure 4-10: Smoothed estimates of the temperature of the hot target during both test runs.

were then fit to the averages of the 162 estimates from each test run. The estimates shown in Figure 4-10 were used to estimate the radiances of the hot target on both test runs using equation 4.72. The estimated radiances were subsequently used in place of the true radiances to evaluate calibration errors.

The next step toward characterizing the radiometric calibration accuracy of AIRS was to estimate the mean values and the joint covariance sequence of gain and offset variations, and the thermal noise variances. But first the calibration views were corrected to eliminate the effect of temperature variations of the hot target. This was accomplished by performing the following transformation of the calibration data;

\[
\begin{bmatrix}
    h'_\lambda(t) \\
    c'_\lambda(t-6)
\end{bmatrix} = \begin{bmatrix}
    \rho_\lambda(301) & 1 \\
    \rho_\lambda(166) & 1
\end{bmatrix}^{-1} \begin{bmatrix}
    h_\lambda(t) \\
    c_\lambda(t-6)
\end{bmatrix}, \quad (4.77)
\]

\[
= \begin{bmatrix}
    \frac{\rho_\lambda(301)-\rho_\lambda(166)}{\rho_\lambda(T_\lambda(t))-\rho_\lambda(166)} h_\lambda(t) + \frac{\rho_\lambda(T_\lambda(t))-\rho_\lambda(301)}{\rho_\lambda(T_\lambda(t))-\rho_\lambda(166)} c_\lambda(t-6) \\
    c_\lambda(t-6)
\end{bmatrix}, \quad (4.78)
\]

where \( h_\lambda(t) \) denotes the hot calibration view of each scan and \( c_\lambda(t-6) \) denotes the second
cold view of each scan (see Figure 4-2) at wavelength \( \lambda \).

The thermal noise variances \( \sigma_h^2 \) and \( \sigma_z^2 \) for each channel were estimated by averaging the square-magnitude of the Fast Fourier Transform (FFT) of the temperature-compensated calibration data, i.e. equation 4.78 for \( t = 8, 8 + 119, \ldots, 8 + 44 \times 119 \), over the frequency interval \([\pi/2, \pi]\) and then averaging over both test runs. Then the temperature-compensated calibration data were filtered to reduce the effect of thermal noise using a fifth-order Butterworth low-pass filter with a 3-dB frequency of 0.2226\( \pi \). Passband and stopband frequencies of \( \omega_p = 2\pi/9 \) and \( \omega_s = \pi/3 \), respectively, were chosen by inspecting the square-magnitude of the FFT of the sequences. The maximum attenuation in the passband was set to \( \delta_p = 3 \) dB and the minimum attenuation in the stopband was set to \( \delta_s = 20 \) dB. The filter order and 3-dB frequency were determined from \( \omega_p, \omega_s, \delta_p, \delta_s \) using the `buttord` MATLAB function.

Next the sample mean and covariance sequence of the filtered temperature-compensated calibration data vectors

\[
\begin{bmatrix}
  h_n^{\prime}(t) \\
  e_n^{\prime}(t-6)
\end{bmatrix}, \quad t = 8, 8 + 119, \ldots, 8 + 44 \times 119,
\]

(4.79)

were computed for each test run and averaged over both test runs. A low-order rational model was fit to the estimated covariance sequence for each channel as described in Section 4.4. The order of the model for each channel was chosen by trial-and-error until a reasonable fit was obtained. Examples of the fitted models can be found in Appendix D.

Finally, estimates of the mean and covariance sequence of the gain and offset for each channel were obtained from the corresponding estimates of mean and covariance of the filtered temperature-compensated calibration data by assuming the following linear transformation holds exactly:

\[
\begin{bmatrix}
  h_n^{\prime}(t) \\
  e_n^{\prime}(t-6)
\end{bmatrix} = \begin{bmatrix}
  \rho(301) & 1 \\
  \rho(166) & 1
\end{bmatrix} \begin{bmatrix}
  g(t) \\
  b(t)
\end{bmatrix}.
\]

(4.80)

Examples of the estimates of the covariance of the gain and offset are also given in Appendix D.

We should note that the estimates of the thermal (white) noise variances obtained here may be strongly contaminated by gain and/or offset variations. Because the thermal noise variances were estimated using data observed on a time scale of one scan-length, the aliasing of gain and/or offset variations occurring on smaller time scales will cause the white noise
to be overestimated. Analysis of the calibration errors obtained below shows that this was indeed the case for the AIRS EM test data. Nonetheless, for the purpose of deriving filters to estimate gain and offset on a time scale of one scan-length these estimates are appropriate.

For illustrative purposes Figure 4-11 shows the relative contributions of gain variations, offset variations, and thermal noise to the test data from the AIRS EM. Each individual contribution is quantified in terms of the standard deviation. The top graph shows that the rms thermal noise is very signal dependent at the shorter wavelengths. The average ratio $\sigma_{nc}/\sigma_{nh}$ for modules M-1, M-4, M-10 and M-12 is 45 percent, 88 percent, 88 percent, and 102 percent, respectively. The contributions of gain and offset variations were compared to the rms thermal noise at 301 K. The rms offset variations ($\sigma_{b}$) are on the average 6.5 dB lower than the rms thermal noise $\sigma_{nh}$ on module M-1 and within 2.6 dB (average over each module) of $\sigma_{nh}$ on all other modules. The contribution of gain variations was evaluated for a black-body target at 200 K, 301 K, and 350 K. The two extreme temperatures delimit the nominal dynamic range of the AIRS instrument. The contribution of gain variations is seen to vary widely over the dynamic range of the instrument especially at the shorter wavelengths. For the channels on module M-1 the rms variation $\sigma_{g}$ is on the average 8 dB higher than $\sigma_{nh}$ at the high end of the dynamic range and 26 dB lower at the lower end of the dynamic range. We should note, however, that a comparison of the individual contributions of gain, offset, and thermal noise in terms of variances only is not strictly appropriate because gain and offset variations generally have a strongly colored (non-flat) spectrum.

To estimate the calibration accuracy of AIRS two different linear calibrations of the AIRS EM test data were performed. The first calibration used the linear MMSE filter described in Section 4.3. To calibrate the data observed during a given scan, the filter was used to estimate the gain and offset at the time of the calibration view of that scan and the following scan, as illustrated in Figure 4-12. The filter operated on the calibration view and the second cold space view (see Figure 4-2) of a number of adjacent scans before and after the scan to be calibrated. Then the estimates of gain and offset obtained with the MMSE filter were linearly interpolated to obtain estimates of the same at the times of the scene views. Finally, the test data observed during the scene views of the given scan were calibrated using the basic calibration equation 4.6 with the interpolated gain and offset estimates. This was repeated using MMSE filters of various lengths derived from
Figure 4-11: Estimated relative contributions of gain variations, offset variations, and thermal noise to variations in the test data from 754 channels of the AIRS EM. $\sigma_g$ and $\sigma_b$ denote the standard deviation of fluctuations in the gain and offset, respectively. The contribution of gain variations was evaluated for a black-body target at three different temperatures. The two extreme temperatures delimit the nominal dynamic range of AIRS. $\rho_\lambda(T)$ is the Planck function (see equation 4.72), $\sigma_{nc}$ is the rms thermal noise in the cold views, $\sigma_{nh}$ is the rms thermal noise in the hot views, $\sigma_b$ is the rms offet variation, and $\sigma_g$ is the rms gain variation.
Figure 4-12: Illustration of the use of the MMSE filter to obtain gain and offset estimates. \( \hat{g} \) and \( \hat{b} \) denote the estimates of the gain and offset at the appropriate times.

The estimated joint covariance sequence of the gain and offset variations for each spectral channel.

The second calibration performed is of the standard type described in Section 4.2, and it is very similar to the calibration procedure described in the AIRS Instrument Calibration Plan [46]. In this case the gain and offset were first estimated locally at the beginning of each scan by inverting the calibration data (the calibration view and the second cold space view). Then the sequences of gain estimates and offset estimates were individually smoothed using a zero-phase triangular FIR filter normalized so that the values of the coefficients added up to unity. The smoothed estimates were taken to be the values of the gain and offset at the time of the calibration view of each scan. As in the first calibration, these estimates were linearly interpolated to obtain estimates of the gain and offset at the times of the scene views. The two-point calibration equation was then applied to the test data observed during the scene views. A discussion of the estimates of calibration accuracy derived from both calibrations follows.

The average calibration error variance for modules M-1, M-4, M-10, and M-12 obtained
Table 4.2: Estimated average decrease in calibration error variance (dB) using empirical MMSE calibration filters of different lengths. The reference chosen is the calibration error variance obtained using the MMSE filter of length two scans. The decrease in error variance (in dB) was averaged over spectral channels on each module and over all scene view indexes. The MMSE filters were derived from low-order rational models fitted to the estimated joint covariance sequence of the calibration data as described in the text.

<table>
<thead>
<tr>
<th>AIRS Module</th>
<th>1</th>
<th>4</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>6</td>
<td>0.66</td>
<td>0.58</td>
<td>0.20</td>
</tr>
<tr>
<td>length</td>
<td>10</td>
<td>0.83</td>
<td>0.69</td>
<td>0.24</td>
</tr>
<tr>
<td>(scans)</td>
<td>14</td>
<td>0.88</td>
<td>0.69</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.92</td>
<td>0.73</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 4.3: Estimated average decrease in calibration error variance (dB) using empirical MMSE calibration filters of different lengths. The reference chosen is the same as in Table 4.2 to facilitate comparison of both sets of results. The decrease in error variance (in dB) was averaged over spectral channels on each module and over all scene view indexes. The MMSE filters were derived from first-order (one state variable) rational models fitted to the estimated joint covariance sequence of the calibration data.

<table>
<thead>
<tr>
<th>AIRS Module</th>
<th>1</th>
<th>4</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>2</td>
<td>-0.011</td>
<td>-0.076</td>
<td>0.0027</td>
</tr>
<tr>
<td>length</td>
<td>6</td>
<td>0.62</td>
<td>0.22</td>
<td>0.19</td>
</tr>
<tr>
<td>(scans)</td>
<td>10</td>
<td>0.78</td>
<td>0.29</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>0.83</td>
<td>0.31</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>0.87</td>
<td>0.37</td>
<td>0.27</td>
</tr>
</tbody>
</table>
using empirical MMSE calibration filters is shown in Table 4.2 for different filter lengths. The average calibration error variance for each module and each filter length is expressed in decibels relative to the average error variance obtained using the filter of length two scans. As expected, the error variance was observed to decrease with increasing filter length. The fractional decrease in error variance for each module was less than one percent when the filter length was increased from fourteen to eighteen scans. Since the correlation of the gain and offset variations was observed to decrease monotonically with time lag for most spectral channels, we infer that a filter of length fourteen scans achieves most of the potential error reduction.\(^8\)

Because the order of the covariance model used to derive the MMSE filters was chosen by trial-and-error (see discussion on p. 112) there was a concern that overfitting the covariance estimates could have had an adverse effect on the predicted calibration accuracy since the covariance estimates are rather noisy due to the small size of the data set. For comparison, MMSE filters were also derived from first-order models fitted to the covariance sequences of the calibration data. The average calibration error variances obtained using those filters are shown in Table 4.3. The calibration errors are seen to be higher than before. The fits obtained using first-order models were notably worse for most channels on module M-4, which is the module for which performance was degraded the most.

Unless stated otherwise, the calibration error results discussed from this point on were obtained using the MMSE filter of length fourteen scans derived from the variable-order covariance models.

Figure 4-13 shows the estimated rms calibration error for individual channels of AIRS using the MMSE calibration filter of length fourteen scans. To obtain those results the calibration error variance for each channel was averaged over all scene view indexes. We note that for this same case the estimated average calibration errors were observed to be small compared to the rms calibration errors on all channels, 11.8 dB lower on the average, so that they can be ignored. Also shown in Figure 4-13 is the noise-equivalent radiance (NER) predicted (by the AIRS Science Team) for all the AIRS channels for a black-body target at 250 K. The NER in a given channel is defined to be the radiance perturbation equivalent to the standard deviation of the noise when observing a black-body target of a

\[^8\text{The fitted covariance model for some channels exhibited oscillations but their amplitudes were small compared to the value of the covariance at lag zero.}\]
Figure 4-13: Estimated rms calibration error on individual spectral channels obtained using the MMSE filter of length fourteen scans. The calibration error variance on each channel was averaged over all scene view indexes. Also shown is the AIRS Noise-Equivalent Radiance (NER) specification at 250 K.

specified temperature. The NER assumes that the receiver gain and offset are stable, so that fluctuations in the data are due to random noise, e.g. Johnson noise and shot noise. Thus, the NER provides a lower bound for the calibration error since that noise can not be removed by calibration. Note that in Figure 4-13 some of the estimated rms calibration errors are lower than the NER. This is probably due to errors in those estimates.

To get an idea about how much of the calibration error is due to gain and offset variations the rms calibration errors were compared to the NER. Since the data that were calibrated are observations of a target at 301 K, we compared the rms calibration error to the NER for a target at the same temperature. We did not have a specification for the NER at 301 K so we estimated it by dividing our estimate of $\sigma_{n_h}$ for each channel by the corresponding average gain, in that way converting $\sigma_{n_h}$ from counts to radiance units. Figure 4-14 shows histograms of the ratio of the rms calibration noise to estimated NER at 301 K over the four detector modules M-1, M-4, M-10, and M-12. Table 4.4 lists the average and standard deviation of the same ratio over each module. These results suggest that for most chan-
Figure 4-14: Histograms of the ratio of rms calibration error to estimated NER @ 301 K for channels on four different detector modules. The NER was estimated for each channel by dividing $\sigma_{n_h}$ by the average gain.

...nals, residual gain and offset variations make only a small contribution to the calibration error. However, as noted before, if there were significant gain and/or offset fluctuations at frequencies above half the scan rate (3/16 Hz) they would inflate our estimate of the NER. Our previous observation is only valid under the assumption that the estimates of NER are not inflated.

Next we evaluated the correlations between calibration errors in different channels. Figure 4-15 shows the magnitude of the sample correlation coefficients of the calibration errors for the 754 channels in question. The matrix is divided into blocks corresponding to detector array modules M-1, M-4, M-10, and M-12. The relatively high correlation between the
<table>
<thead>
<tr>
<th>AIRS Module</th>
<th>1</th>
<th>4</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>1.06</td>
<td>0.96</td>
<td>1.03</td>
<td>1.05</td>
</tr>
<tr>
<td>std. dev.</td>
<td>0.12</td>
<td>0.12</td>
<td>0.14</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 4.4: Mean and standard deviation of the ratio of rms calibration error to estimated NER at 301 K for each of four detector modules.

calibration errors on some channels indicate that those errors are not dominated by random thermal noise. In particular, significant correlations are observed between channels located in the same detector array. In module M-4 there is a block of 153 channels\(^9\) which show higher correlations than the rest of the channels in the same module; 56 of them have a correlation coefficient of more than 0.5 with at least one other channel (see channel 815 in Figure 4-6 for an example). Those same channels from module M-4 have errors which are also highly correlated with the errors on about thirty-five channels from module M-1 and a few channels from module M-12. The group of thirty-five channels\(^10\) from module M-1 also exhibits high correlations within itself. The cause of the abnormally high correlations between calibration errors in different channels is not known.

The time behavior of the calibration errors was investigated briefly. Periodograms of the calibration errors were computed for various channels by computing the FFT of the error sequence along each scan, computing the square magnitude of the spectrum, and averaging over all scans. Figures 4-16 shows the periodograms of the calibration errors for two of the channels with highly correlated (between channels) errors. It can be seen that those calibration errors are significantly non-white. Note that, except for the component at zero frequency, these variations occur on time scales shorter than a scan cycle so that they can not be effectively removed by calibration. Figure 4-17 shows the periodograms for four channels in which the calibration errors were not significantly correlated with errors on other channels. Most of these calibration errors appear to have flatter spectra than the previous ones.

The calibration errors were also evaluated on each channel as a function of scene view index to determine if the calibration accuracy varied significantly along a scan. In general, one would expect better calibration accuracy towards both ends of the scans, where calibrations for all these channels are physically located in arrays D-4a and D-4b.

\(^9\)The detectors for all these channels are physically located in arrays D-4a and D-4b.

\(^{10}\)The detectors for all these channels are physically located in array D-1b.
Figure 4-15: Illustration of the estimated correlation between calibration errors for 754 channels in modules M-1, M-4, M-10, and M-12. The base-10 logarithm of the absolute value of the correlation coefficient between all pairs of channels is shown. Values less than $10^{-2}$ are colored black. In the figure wavelength increases downward along the vertical axis and to the right along the horizontal axis. The calibration errors were obtained using the MMSE filter of length fourteen scans.
Figure 4-16: Periodograms of the calibration errors on two channels of the AIRS EM. The calibration errors on these channels were significantly correlated with calibration errors on other channels.

...tion data are observed. Unfortunately, because of the small size of the data set used here, the estimates of calibration error for individual scene views are very noisy so a quantitative analysis was not pursued. However, a visual inspection of the estimates did not reveal significant variations of the type expected for most of the channels. About fifteen channels (out of 754) were found which did show significant variations in calibration accuracy along a scan. Some examples are shown in Figure 4-18. Most of those channels were found on module M-10 which, as suggested by Figure 4-13, seems to have higher noise than is expected for the AIRS flight unit. In conclusion, it seems that using statistics averaged over the scene view indexes to characterize the calibration errors on AIRS data is reasonable.

The estimates of calibration accuracy obtained using the second type of calibration (using a triangular FIR filter to smooth instantaneous estimates of the gain and offset) of the AIRS EM test data were virtually identical with those discussed above. Therefore, detailed results are omitted. Table 4.5 lists the average improvement in calibration accuracy over each module as a function of the length of the smoothing filter. For module M-12, which has channels which do not show any significant gain or offset variations (see Figure 4-7 for a typical example), the calibration error variance is worse when short triangular filters are used because of insufficient noise averaging. (Although the error variance is being compared with that obtained using a short MMSE filter, one should note that the MMSE filter operates
Figure 4-17: Periodograms of the calibration errors on four channels of the AIRS EM. None of the calibration errors on these channels were significantly correlated with calibration errors on other channels.
Figure 4-18: Examples of spectral channels with significant systematic variation in the estimated calibration accuracy as a function of scene view index. The calibration error variances were obtained using the MMSE filter of length fourteen scans.
<table>
<thead>
<tr>
<th>AIRS Module</th>
<th>1</th>
<th>4</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>5</td>
<td>0.55</td>
<td>0.54</td>
<td>0.02</td>
</tr>
<tr>
<td>length</td>
<td>9</td>
<td>0.81</td>
<td>0.75</td>
<td>0.11</td>
</tr>
<tr>
<td>(scans)</td>
<td>13</td>
<td>0.91</td>
<td>0.80</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.99</td>
<td>0.84</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 4.5: Estimated average decrease in calibration error variance (dB) using zero-phase triangular FIR filters of different lengths. The filters were used to smooth local estimates of the gain and offset at the beginning of each scan. The decrease in error variance (in dB) was averaged over spectral channels on each module and over all scene view indexes. The reference chosen is the same as in Table 4.2 to facilitate comparison of both sets of results.

on mean-centered calibration data to generate estimates of the gain and offset variations, and then adds the mean gain and offset. Those mean values were estimated by averaging all the calibration data in both test runs.) The results in Table 4.5 can be compared to those in Table 4.2.

Summary of Results

Here we summarize the most important results of the characterization of calibration errors. These results apply only to data from the 754 channels analyzed above.

1. When using MMSE calibration filters it was found that a filter fourteen scans long provided most of the calibration variance reduction. Increasing the length to eighteen scans reduced the error variance by less than one percent on the average. Similar calibration error variances, typically within five percent, were obtained using zero-phase triangular FIR filters instead of the MMSE filters.

2. The rms calibration errors were found to be no more than forty percent higher (worst case) than the estimated noise-equivalent radiance. This suggests that residual gain and offset variations are not the main contributors to the calibration error.

3. Relatively high correlations—greater than fifty percent—were observed between calibration errors in different channels. That phenomenon was confined mostly to channels physically located in the same detector array, especially in modules M-1 and M-4. The correlated calibration errors were also found to have more strongly colored
spectra than otherwise. A significant fraction of the colored portion of the spectrum occurs at frequencies higher than half the scan rate. Those error components can not be removed effectively by calibration.

4. For most channels the calibration error variance did not depend strongly on the scene view index so it seems reasonable to use scan-averaged statistics to characterize the calibration errors.
Chapter 5

Compression Algorithm for Multispectral Data in Additive Gaussian Noise

5.1 Introduction

In this chapter we describe a block-adaptive transform coding algorithm for multispectral data corrupted by additive Gaussian noise. The algorithm is based on the assumption that the independent additive noise data model in Figure 3-1 can be applied to the data once they have been calibrated, i.e. converted to physical units. Thus, the algorithm is designed to operate on previously calibrated data. Specifically, the noise is used to model errors in the calibrated data arising as a result of both instrument noise and imperfect calibration. Throughout the chapter we shall refer to those errors as "noise." The coding algorithm is able to estimate the statistics (mean and covariance matrix) of the noiseless spectrum for each block of noisy data assuming the noise statistics are known or can be estimated from another source. Throughout the chapter we assume that the noise statistics are known. In practice those statistics may be estimated from accumulated calibration data, which would be transmitted intact to the ground.

The coding algorithm described below employs the structure of the NAPC-based codebook for noisy data derived in Section 3.3.3. As shown in that section, if the signal and the noise are uncorrelated and have known Gaussian distributions, the NAPC-based codebook
is optimal for a distortion criterion where the error is weighted by a square root of the inverse covariance matrix. That weighting normalizes errors in the decoded data by the rms noise along the noise eigen-directions, so that the distortion can be interpreted as an error-to-noise ratio. We choose to use this "noise-adjusted" distortion criterion because of an appealing property of the NAPC: they compact the signal energy per unit noise energy in the data. Many other algorithms for image coding are evaluated in terms of peak signal-to-noise ratio\(^1\) (PSNR) \([4, 7, 5]\). PSNR provides a measure of quality relative to the original noisy data. In contrast, our performance criterion measures quality relative to the noiseless radiance spectrum.

An alternative way of evaluating the quality of a compressed data set would be in terms of a use-dependent criterion. For example, in the case of data from an atmospheric sounder like AIRS one could evaluate the error in temperature or humidity profiles retrieved from the decoded radiances as a function of the compression ratio. In the case of surface imaging data, the probability of correct classification and the percent of preserved edges are often of interest \([49]\). One should note that in these examples the quality attributed to the compressed data will depend partly on the effectiveness of the particular retrieval or classification algorithm applied to them.

The rest of the chapter is organized as follows. Section 5.2 describes the overall structure of the algorithm. Most of the discussion there draws on material developed in Chapter 3. Two new components are developed in order to make the coding algorithm better suited for adaptive use. In Section 5.3 we detail a rule for setting the codebook overhead associated with each encoded data block. Procedures for quantizing the required codebook parameters are described, and bounds on the quantization resolution are given so that the increase in distortion due to quantizing the codebook is acceptable. Then in Section 5.4 we modify the usual log-variance bit allocation to account for the codebook overhead. The combination of the overhead allocation rule and the modified bit allocation rule allows one to design a codebook for a finite data block without exceeding a specified bit rate. Section 5.5 documents several tests conducted using simulated AIRS data to evaluate the performance and robustness of the coding algorithm. Finally, in Section 5.6 we describe a strategy for preserving individual channels of a given spectrum in which the estimated noise is

---

\(^1\)Peak signal-to-noise ratio is defined as the ratio in dB of the square of the maximum data amplitude to the variance of the coding residual.
unusually high. This provides protection against the possibility of excessively degrading outlier channels.

5.2 General Description of the Algorithm

A block diagram of the coding algorithm is shown in Figure 5-1.\(^2\) The encoder occupies the top portion of the figure and the decoder occupies the bottom. Starting on the top left corner of the figure, the sequence of calibrated spectra is first grouped into blocks of \(K\) spectra. Figure 5-2 shows how each data block is formed. For each data block an empirical NAPC-based codebook is designed and subsequently used to code the data block itself. Functions pertaining to the design of the codebook are illustrated to the right of the vertical dashed line in Figure 5-1. Note that those codebook parameters which are continuous in amplitude and are needed at the decoder, i.e. the eigenvectors \(U\), the sample mean \(\bar{m}_y\), and the variances \(\Sigma\) of the filtered NAPC, are quantized before being transmitted from the encoder to the decoder through a digital channel.

For each data block, the codebook is designed using essentially the Modified Maximum-Likelihood Design Algorithm (MMLDA) described in Section 3.4.3. The only differences lie in the bit allocation for the Lloyd-Max quantizers used to quantize the filtered NAPC coefficients, and in the design of the quantizers themselves. In our adaptive coding algorithm the Lloyd-Max quantizers are designed using the quantized variances of the filtered NAPC rather than the continuous amplitude (or double-precision floating-point) values. This is done so that the quantizer index generator \(h\) and the index decoder \(g\) which reside on opposite ends of the channel, are consistent with each other. The bit allocation is performed taking into account the overhead required to transmit to the decoder all the required codebook parameters, so that the desired bit rate constraint is satisfied. The bit allocation also uses the quantized variances of the filtered NAPC. The precise manner in which the codebook overhead is determined, and the bit allocation rule used are described in Sections 5.3 and 5.4, respectively. In Figure 5-1 both the determination of the overhead and the bit allocation are represented by the block labeled “Rate Allocation.”

The coding algorithm shown in Figure 5-1 is similar to one proposed by Hoffman and Johnson [6]. They too used the NAPC decomposition to code multispectral data. How-

\(^2\)Most of the symbols used in the diagram are defined in Chapter 3.
Figure 5-1: Block diagram of the adaptive NAPC-based algorithm.
ever, there are many conceptual and practical differences between our algorithm and theirs. Although their method used the NAPC representation, a clear connection between NAPC and the noise-adjusted distortion criterion was not made in their work. In addition, their method attempts to separate the noise from the signal only by truncating the NAPC, which is theoretically inferior to linear least-squares filtering. To effect the separation they truncate the NAPC expansion at the point at which the mean-square value of the residuals just falls below unity, which is the value of the mean-square noise in the noise-adjusted space. In the "lossy" version of their method the residuals are discarded. They suggested using uniform entropy-coded quantizers to quantize the retained NAPC coefficients. Accordingly, they estimated the data compression achieved by their method as the sum of the first-order entropies of the quantized NAPC coefficients. Although they illustrated the dependency of compression on the truncation point and the quantization step size, they did not explicitly specify a rule for simultaneously choosing those parameters. In contrast, using our method the truncation of the NAPC expansion is done indirectly through the bit allocation process, and the bit allocation is done optimally, under an appropriate set of assumptions, for a desired bit rate. Our bit allocation rule also takes into account the codebook overhead.

5.3 Setting the Codebook Overhead

In this section we describe the component of our coding algorithm that selects the overhead in bits used to communicate the required codebook parameters to the decoder. As illustrated in Figure 5-1, certain codebook parameters must be known at the decoder in order to
recover the signal estimate from its quantized representation in NAPC space. In this sense
the encoded representation of the signal estimate includes not only the quantized NAPC
coefficients but also some of the codebook parameters. Because some of those parameters
are ideally continuous in amplitude it is necessary to quantize them in order to transmit
them through a channel of limited capacity. In what follows we derive a rule that is used to
determine the overhead (bits) required to represent those quantized codebook parameters.

The idea behind the overhead allocation rule derived here is to quantize the relevant
codebook parameters finely enough that the excess distortion arising as a result of their
quantization is negligible. This general approach is used in many practical coding algo-
rithms, e.g. [6, 8, 2]. The approach is particularly convenient in two respects. First, it
allows one to neglect the quantized nature of the decoder when predicting and analyzing
the performance of the coding algorithm. Second, it is simple to carry out. The rigorous
formulation and solution of the overhead allocation problem seems too difficult to pursue
analytically. An obvious disadvantage of the stated approach is that it does not attempt
to be efficient. In practice this inherent inefficiency is circumvented whenever possible by
making the codebook adaptive over long blocks of data, so that the codebook overhead
becomes an insignificant fraction of the total bit budget per block.

Although the general approach for overhead allocation pursued here is commonly used,
an explicit well-defined rule is seldom given for its implementation. We shall derive such
a rule within the context of our coding algorithm. Our rule may also be used with similar
transform-based coding algorithms. Although it is ad hoc, the rule derived here is useful, if
for nothing else, because it is simple and the assumptions behind it are clearly stated.

Let us begin the derivation of our overhead allocation rule by making some definitions.
Let \( \hat{x}(t) \) be the signal estimate obtained with the quantized codebook parameters, as illustrated in Figure 5-1. Also, let \( \hat{x}(t) \) be the signal estimate obtained without quantizing any of the parameters of the codebook. These two estimates are given by

\[
\hat{x}(t) = \hat{m}_y + C_{nn}^{1/2} \tilde{U} \hat{q}(t), \tag{5.1}
\]

\[
\hat{x}(t) = \hat{m}_y + C_{nn}^{1/2} Uq(t), \tag{5.2}
\]

where \( \hat{m}_y \) is the quantized sample mean of the input data block, \( \tilde{U} \) is the quantized matrix of eigenvectors, and \( q(t) \) and \( \hat{q}(t) \) are two different quantized versions of the filtered NAPC.
The difference between $q(t)$ and $\tilde{q}(t)$ is that the former is obtained using quantizers designed for the exact (estimated) variances of the filtered NAPC, while the latter is obtained using quantizers designed for quantized values of those variances. For convenience we define

\begin{align*}
\hat{e}(t) & \triangleq x(t) - \hat{x}(t), \\
\bar{e}(t) & \triangleq x(t) - \bar{x}(t),
\end{align*}

(5.3)  
(5.4)

\begin{align*}
\hat{D} & \triangleq E\left\{ \hat{e}(t)^T C_{nn}^{-1} \hat{e}(t) \right\} = E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\|^2 \right\}, \\
\bar{D} & \triangleq E\left\{ \bar{e}(t)^T C_{nn}^{-1} \bar{e}(t) \right\} = E\left\{ \left\| C_{nn}^{-1/2} \bar{e}(t) \right\|^2 \right\},
\end{align*}

(5.5)  
(5.6)

where $\| \cdot \|$ denotes the Euclidean norm. $\hat{D}$ and $\bar{D}$ denote the mean distortions associated with $\hat{x}(t)$ and $\bar{x}(t)$, respectively.

5.3.1 Approximating the Excess Distortion

The first part of the derivation consists of computing an approximation of the excess distortion $\hat{D} - \bar{D}$. In the end we want to obtain an expression for $\hat{D} - \bar{D}$ in terms of the errors $\hat{m}_y - \hat{m}_y$, $\hat{U} - U$, and $\hat{q}(t) - q(t)$. The first two terms are the quantization errors of $\hat{m}_y$ and $U$. The last term can be related directly to the quantization of the diagonal covariance matrix $\Sigma$ (see Figure 5-1). The expression involving those three error terms will be used to select the quantization resolution of $\hat{m}_y$, $U$, and $\Sigma$.

To obtain an approximation for the excess distortion $\hat{D} - \bar{D}$ we first compute an upper bound for $\hat{D}$. By direct application of the triangle inequality one can write

\begin{equation}
\left\| C_{nn}^{-1/2} \hat{e}(t) \right\| \leq \left\| C_{nn}^{-1/2} \bar{e}(t) \right\| + \left\| C_{nn}^{-1/2} (\hat{e}(t) - \bar{e}(t)) \right\|. 
\end{equation}

(5.7)

Squaring both sides of the expression and taking the expected value gives

\begin{align*}
E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\|^2 \right\} & \leq E\left\{ \left\| C_{nn}^{-1/2} \bar{e}(t) \right\|^2 \right\} + E\left\{ \left\| C_{nn}^{-1/2} (\hat{e}(t) - \bar{e}(t)) \right\|^2 \right\} + \\
& \quad 2 E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\| \left\| C_{nn}^{-1/2} (\hat{e}(t) - \bar{e}(t)) \right\| \right\}.
\end{align*}

(5.8)

Finally, applying the Schwartz inequality gives

\begin{align*}
E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\|^2 \right\} & \leq E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\|^2 \right\} + E\left\{ \left\| C_{nn}^{-1/2} (\hat{e}(t) - \bar{e}(t)) \right\|^2 \right\} + \\
& \quad + E\left\{ \left\| C_{nn}^{-1/2} (\hat{e}(t) - \bar{e}(t)) \right\|^2 \right\}.
\end{align*}
\[ 2E\left\{ \left\| C_{nn}^{-1/2} \hat{e}(t) \right\|^2 \right\}^{1/2} E\left\{ \left\| C_{nn}^{-1/2} (\hat{e}(t) - \hat{\epsilon}(t)) \right\|^2 \right\}^{1/2}. \quad (5.9) \]

Using the definitions of \( \hat{\epsilon}(t), \hat{e}(t), \hat{D}, \) and \( \hat{D}, \) the expression above gives us the following bound for \( \hat{D}: \)

\[ \hat{D} \leq \hat{D} + 2\hat{D}^{1/2} E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\}^{1/2} + E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\}. \quad (5.10) \]

Thus, one can guarantee that the excess distortion \( \hat{D} - \hat{D} \) does not exceed \( \alpha \hat{D}, \) for some \( \alpha > 0, \) by requiring that

\[ 2\hat{D}^{1/2} E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\}^{1/2} + E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\} \leq \alpha \hat{D}. \quad (5.11) \]

For small \( \alpha \) this expression is approximately equivalent to

\[ E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\} \leq \frac{\alpha^2}{4} \hat{D}. \quad (5.12) \]

Instead of directly enforcing the last inequality, we pursue an approximation of the left-hand side expression in terms of the errors \( \hat{m}_y - \hat{m}_y, \hat{U} - U, \) and \( \hat{q}(t) - q(t). \) Using equations 5.1 and 5.2 we can write

\[ C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) = C_{nn}^{-1/2} (\hat{m}_y - \hat{m}_y) + (\hat{U} - \hat{U}) \hat{q}(t) + U(q(t) - \hat{q}(t)). \quad (5.13) \]

So the mean-square Euclidean norm of \( C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \) equals

\[ E\left\{ \left\| C_{nn}^{-1/2} (\hat{\epsilon}(t) - \hat{\epsilon}(t)) \right\|^2 \right\} = \left\| C_{nn}^{-1/2} (\hat{m}_y - \hat{m}_y) \right\|^2 + E\left\{ \left\| (U - \hat{U}) \hat{q}(t) \right\|^2 \right\} + E\left\{ \left\| U(q(t) - \hat{q}(t)) \right\|^2 \right\} + 2(\hat{m}_y - \hat{m}_y)^T \left(C_{nn}^{-1/2}\right)^T (U - \hat{U}) E\{\hat{q}(t)\} + 2(\hat{m}_y - \hat{m}_y)^T \left(C_{nn}^{-1/2}\right)^T U E\{(q(t) - \hat{q}(t))\} + 2E\{\hat{q}(t)^T (U - \hat{U})^T U (q(t) - \hat{q}(t))\}. \quad (5.14) \]

This expression is valid for a particular quantization of \( m_y \) and \( U. \) We want to approximate the above expression by ignoring the last three terms. To do that we model the quantization errors in \( m_y \) and \( U \) as independent random vectors. Then taking the expectation in
equation 5.14 with respect to all quantizations of $\hat{m}_y$ and $U$ gives

\[
E\left\{\left\|C_{nn}^{-1/2} (\hat{x}(t) - \tilde{x}(t))\right\|^2\right\} = E\left\{\left\|C_{nn}^{-1/2} (\hat{m}_y - \tilde{m}_y)\right\|^2\right\} + E\left\{\left\|(U - \tilde{U}) \tilde{q}(t)\right\|^2\right\} + E\left\{\left\|U (q(t) - \tilde{q}(t))\right\|^2\right\},
\]

(5.15)

\[
= E\left\{\left\|C_{nn}^{-1/2} (\hat{m}_y - \tilde{m}_y)\right\|^2\right\} + E\left\{\left\|(U - \tilde{U}) \tilde{q}(t)\right\|^2\right\} + E\left\{\left\|q(t) - \tilde{q}(t)\right\|^2\right\},
\]

(5.16)

where in the last expression we used the fact that $U$ is an orthogonal matrix. We use this expression as the basis of our overhead allocation rule.

At this point we can state the general form of the overhead allocation rule: to obtain an excess distortion $\tilde{D} - \hat{D}$ of at most $\alpha \tilde{D}$, quantize $\hat{m}_y$, $U$, and $\Sigma$ finely enough so that

\[
E\left\{\left\|C_{nn}^{-1/2} (\hat{m}_y - \tilde{m}_y)\right\|^2\right\} + E\left\{\left\|(U - \tilde{U}) \tilde{q}(t)\right\|^2\right\} + E\left\{\left\|q(t) - \tilde{q}(t)\right\|^2\right\} \leq \frac{\alpha^2}{4} \tilde{D}.
\]

(5.17)

The rule is approximate because of the assumptions used to obtain equation 5.16. To actually use this rule to determine the codebook overhead one must specify how the parameters $\hat{m}_y$, $U$, $\Sigma$ are to be quantized. We specify how to quantize those parameters next.

### 5.3.2 Quantizing the Sample Mean

The sample mean $\hat{m}_y$ is quantized uniformly in the noise-adjusted space, i.e. $\hat{m}_y$ is obtained by uniformly quantizing the elements of $C_{nn}^{-1/2} \hat{m}_y$ and then pre-multiplying by $C_{nn}^{1/2}$. Using a uniform quantizer with step size $\Delta_m$ and modeling the quantization errors as zero-mean uniformly distributed random noise one obtains

\[
E\left\{\left\|C_{nn}^{-1/2} (\hat{m}_y - \tilde{m}_y)\right\|^2\right\} = N \frac{\Delta_m^2}{12}.
\]

(5.18)

To determine the overhead required to transmit the quantized sample mean for a given quantization step size $\Delta_m$ one must specify the range of $C_{nn}^{-1/2} \hat{m}_y$. For convenience let $\mu = C_{nn}^{-1/2} \hat{m}_y$. If the $i^{th}$ element of the input data lies in the range $\min y_i \leq y_i \leq \max y_i$,
the $i^{th}$ element of $\mu$ is bounded by

$$\max \mu_i = \sum_{j \text{ s.t.} \left( C_{nn}^{-1/2} \right)_{ij} > 0} \left( C_{nn}^{-1/2} \right)_{ij} \max y_j + \sum_{j \text{ s.t.} \left( C_{nn}^{-1/2} \right)_{ij} < 0} \left( C_{nn}^{-1/2} \right)_{ij} \min y_j,$$

$$\min \mu_i = \sum_{j \text{ s.t.} \left( C_{nn}^{-1/2} \right)_{ij} < 0} \left( C_{nn}^{-1/2} \right)_{ij} \max y_j + \sum_{j \text{ s.t.} \left( C_{nn}^{-1/2} \right)_{ij} > 0} \left( C_{nn}^{-1/2} \right)_{ij} \min y_j.$$  \hfill (5.19)

To uniformly quantize $\mu_i$ with a stepsize of at most $\Delta_m$, the number of bits required is

$$b_{\mu_i} = \left\lceil \log_2 \left( \frac{\max v_i - \min v_i}{\Delta_m} \right) \right\rceil,$$  \hfill (5.21)

where $\lceil \cdot \rceil$ rounds toward infinity. The total overhead required to quantize the sample mean is $\sum_{i=1}^{N} b_{\mu_i}$.

5.3.3 Quantizing the Eigenvectors

The elements of the eigenvectors are quantized using a uniform quantizer with step size $\Delta_U$. Modeling the quantization errors as zero-mean uniformly distributed random noise gives

$$E\left\{ \left\| (U - \bar{U}) \tilde{q}(t) \right\|^2 \right\} = E\left\{ \text{tr} \left[ (U - \bar{U})^T (U - \bar{U}) \tilde{q}(t)\tilde{q}(t)^T \right] \right\},$$

$$= \text{tr} \left[ E\left\{ (U - \bar{U})^T (U - \bar{U}) \right\} E\left\{ \tilde{q}(t)\tilde{q}(t)^T \right\} \right],$$

$$= \text{tr} \left[ N \frac{\Delta_U^2}{12} \mathbf{I} \ E\left\{ \tilde{q}(t)\tilde{q}(t)^T \right\} \right],$$

$$= N \frac{\Delta_U^2}{12} \sum_{i=1}^{m} E\left\{ \tilde{q}_i(t)^2 \right\},$$

$$\approx N \frac{\Delta_U^2}{12} \sum_{i=1}^{m} \Sigma_{ii},$$  \hfill (5.22)

where $m$ is the number of NAPC that received non-zero bit allocations.

Determining the number of bits required to quantize the eigenvectors using a step size
ΔU is straightforward. Since each eigenvector has unit magnitude by definition, each element \( U_{ji} \) lies in the interval \(-1 \leq U_{ji} \leq 1\). Thus, the number of bits required to quantize each element of each eigenvector is given by

\[
b_U = \left\lfloor \log_2 \left( \frac{2}{\Delta U} \right) \right\rfloor. \quad (5.24)
\]

The total overhead required for quantizing all \( m \) eigenvectors is \( mN b_U \).

### 5.3.4 Quantizing the Variances of the Filtered NAPC

Quantizing the variances (\( \Sigma \)) of the filtered NAPC would increase the distortion because the quantizers applied to the filtered NAPC themselves would not be matched to their estimated variances. To decide how to quantize those variances we first evaluate the increase in the distortion of a Lloyd-Max quantizer when the variance of its input is perturbed by a small amount. That will allow us to obtain an estimate of the distortion \( E\{\|q(t) - \bar{q}(t)\|^2\} \), and it will give us an idea of how to quantize the variances to keep that distortion small.

The excess distortion of a Lloyd-Max quantizer caused by a perturbation to the variance of its input can be evaluated numerically as follows. Let \( \{r_i\}_{i=1}^{2^b} \) and \( \{l_i\}_{i=0}^{2^b} \), denote the set of output levels and the set of decision levels, respectively, of a \( b \)-bit Lloyd-Max quantizer designed for a zero-mean unit-variance random variable (not necessarily Gaussian). Let the actual random input \( v \) to the quantizer have the assumed distribution, a mean value of zero, and variance \( \sigma_v^2 \). For the set of decision levels \( \{l_i\}_{i=0}^{2^b} \), the optimal output levels \( \{r_i^*\}_{i=1}^{2^b} \) for quantizing \( v \) are known to be the centroids of the quantization intervals [39, 34], i.e.

\[
r_i^* = E\{v / l_{i-1} \leq v < l_i\}, \quad 1 \leq i \leq 2^b. \quad (5.25)
\]

The set of optimal output levels would differ from \( \{r_i\}_{i=1}^{2^b} \) if \( \sigma_v^2 \neq 1 \). From the optimality of the centroids \( \{r_i^*\}_{i=1}^{2^b} \) one can easily show that the distortion obtained when quantizing \( v \) using the suboptimal output levels \( \{r_i\}_{i=1}^{2^b} \) is given by

\[
d(b; \sigma_v^2) \triangleq \sigma_v^2 + \sum_{i=1}^{2^b} r_i (r_i - 2r_i^*) \Pr \{l_{i-1} \leq v < l_i\}, \quad (5.26)
\]

where \( \Pr \{\cdot\} \) denotes probability. From this expression one can also obtain the distortion of the optimal \( b \)-bit quantizer for \( v \). Note that \( d(b; 1) \) is the distortion of the optimal \( L \)-level
Figure 5-3: Fractional increase in quantizer distortion due to variance mismatch for Lloyd-Max quantizers designed for a zero-mean Gaussian input and various bit allocations. Also shown is the approximation given by equation 5.34.

quantizer for an input with unit variance. Thus, the distortion of the optimal \( b \)-bit quantizer for \( v \) is simply

\[
d^{*}(b; \sigma_v^2) \triangleq \sigma_v^2 d(b; 1) .
\] (5.27)

Equations 5.26 and 5.27 may be evaluated numerically for a given input distribution and number of quantizer levels. Using those two expressions, the fractional increase in the distortion of a \( b \)-bit Lloyd-Max quantizer (for a given probability distribution) when the ratio of the actual input variance to the assumed variance equals \( \gamma \) can be obtained as

\[
\frac{d(b; \gamma) - d^{*}(b; \gamma)}{d^{*}(b; \gamma)} .
\] (5.28)

Figure 5-3 shows the fractional increase in quantizer distortion as a function of the ratio of actual to assumed variance for a Gaussian input and various bit allocations. To obtain this graph we evaluated equation 5.26 numerically. The decision levels were computed using
Roe's approximation [41];

\[ l_i = \sqrt{6} \text{erf}^{-1} \left( \frac{2i - L}{L + 0.8532} \right), \quad 1 \leq i \leq L - 1, \quad (5.29) \]

where \( \text{erf}^{-1}(\cdot) \) is the inverse error function, and \( \text{erf}(\cdot) \) is given by

\[ \text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (5.30) \]

The quantizer output levels were computed as

\[ r_i = \sqrt{\frac{2}{\pi}} \left( \frac{e^{-l_{i-1}^2/2} - e^{-l_i^2/2}}{\text{erf}(l_i/\sqrt{2}) - \text{erf}(l_{i-1}/\sqrt{2})} \right), \quad (5.31) \]

\[ r_i^* = \sigma_v \sqrt{\frac{2}{\pi}} \left( \frac{e^{-l_{i-1}^2/2\sigma_v^2} - e^{-l_i^2/2\sigma_v^2}}{\text{erf}(l_i/\sqrt{2}\sigma_v) - \text{erf}(l_{i-1}/\sqrt{2}\sigma_v)} \right), \quad (5.32) \]

for \( 1 < i < L \), where \( l_0 = -\infty \) and \( l_L = \infty \). From Figure 5-3 it appears that the fractional increase in quantizer distortion is roughly independent of bit allocation for small variance ratios. For convenience we approximate the fractional increase in distortion by the second-order term of its Taylor expansion evaluated at a fixed bit allocation \( b_0 = 16 \);

\[ \frac{d(b; \gamma)}{d^*(b; \gamma)} - 1 \approx \frac{1}{2} \left. \frac{d^2}{d\gamma^2} d(b_0; \gamma) \right|_{\gamma=1} (\gamma - 1)^2, \quad (5.33) \]

\[ \approx \frac{9}{4} (\gamma - 1)^2. \quad (5.34) \]

This approximation, also shown in Figure 5-3, is reasonable for the purpose of bounding the excess distortion arising as a result of \( \gamma \) deviating from unity.

Figure 5-3 suggests to us uniformly quantizing the logarithm of the variances of the filtered NAPC so as to impose a common bound on the excess fractional distortion of all the quantizers. Quantizing \( \Sigma_{ii} \) to within a fraction \( \delta \) using a logarithmic quantizer means

\[ \log_e(1 - \delta) \leq \log_e \left( \frac{\tilde{\Sigma}_{ii}}{\Sigma_{ii}} \right) \leq \log_e(1 + \delta), \quad (5.35) \]

where \( \tilde{\Sigma}_{ii} \) is the quantized variance. For small \( \delta \) (\( \leq 0.2 \)) this is approximately equivalent to

\[ -\delta \leq \log_e \tilde{\Sigma}_{ii} - \log_e \Sigma_{ii} \leq \delta, \quad (5.36) \]
so a quantization step size of $\Delta_\Sigma = 2\delta$ would be needed. If the set of variances lie in the interval $[\Sigma_{\text{min}}, \Sigma_{\text{max}}]$ then the number of bits $b_\Sigma$ required to quantize them logarithmically with step size $\Delta_\Sigma$ is given by

$$b_\Sigma = \left\lfloor \log_2 \left( \frac{1}{\Delta_\Sigma} \log_e \left( \frac{\Sigma_{\text{max}}}{\Sigma_{\text{min}}} \right) \right) \right\rfloor.$$  \hspace{1cm} (5.37)

The quantized variance $\tilde{\Sigma}_{ii}$ would be obtained using the relation

$$\log_e \left( \frac{\tilde{\Sigma}_{ii}}{\Sigma_{\text{min}}} \right) = \Delta_\Sigma \left\lfloor \frac{1}{\Delta_\Sigma} \log_e \left( \frac{\Sigma_{ii}}{\Sigma_{\text{min}}} \right) \right\rfloor,$$  \hspace{1cm} (5.38)

where $\left\lfloor \cdot \right\rfloor$ denotes rounding to the nearest integer. To avoid having to estimate the range of variances a priori we make the quantizer adaptive at the cost of adding a very small overhead. We effect this by setting

$$\Sigma_{\text{min}} = 2^{\nu},$$  \hspace{1cm} (5.39)

$$\Sigma_{\text{max}} = \tilde{\Sigma}_{11},$$  \hspace{1cm} (5.40)

where $\nu = \left\lfloor \log_2 (\Sigma_{\text{min}}) \right\rfloor$, and $\tilde{m}$ will be determined shortly (see equation 5.55). The integers $\nu$ and $b_\Sigma$ would have to be transmitted to the decoder in order to recover the quantized variances.

From the above discussion we can estimate $E\{\|q(t) - \tilde{q}(t)\|^2\}$ as

$$E\{\|q(t) - \tilde{q}(t)\|^2\} = \sum_{i=1}^{m} E\{(q_i(t) - \tilde{q}_i(t))^2\},$$  \hspace{1cm} (5.41)

$$= \sum_{i=1}^{m} \tilde{\Sigma}_{ii} d\left(b^*_i; \Sigma_{ii}/\tilde{\Sigma}_{ii}\right) - d^*(b^*_i; \Sigma_{ii}),$$  \hspace{1cm} (5.42)

$$= \sum_{i=1}^{m} \left( \frac{\tilde{\Sigma}_{ii} d\left(b^*_i; \Sigma_{ii}/\tilde{\Sigma}_{ii}\right)}{d^*(b^*_i; \Sigma_{ii})} - 1 \right) d^*(b^*_i; \Sigma_{ii}),$$  \hspace{1cm} (5.43)

$$= \sum_{i=1}^{m} \left( \frac{d\left(b^*_i; \Sigma_{ii}/\tilde{\Sigma}_{ii}\right)}{d^*(b^*_i; \Sigma_{ii}/\tilde{\Sigma}_{ii})} - 1 \right) d^*(b^*_i; \Sigma_{ii}),$$  \hspace{1cm} (5.44)

$$\approx \sum_{i=1}^{m} \frac{g}{4} \left( \Sigma_{ii}/\tilde{\Sigma}_{ii} - 1 \right)^2 d^*(b^*_i; \Sigma_{ii}),$$  \hspace{1cm} (5.45)

where $b^*_i$ is the bit allocation for the $i^{th}$ filtered NAPC, and $m$ is the number of NAPC that
received non-zero bit allocations. If the variances $\Sigma_{ii}$ are quantized as specified above then

$$\left| \frac{\Sigma_{ii}}{\tilde{\Sigma}_{ii}} - 1 \right| \approx \left| \frac{\hat{\Sigma}_{ii}}{\Sigma_{ii}} - 1 \right| \leq \delta = \Delta_\Sigma/2,$$

(5.46)

and one obtains the approximate bound

$$E\left\{ \|q(t) - \hat{q}(t)\|^2 \right\} \leq \frac{9}{16} \Delta_\Sigma^2 \sum_{i=1}^{m} d^*(b_i^*; \Sigma_{ii}).$$

(5.47)

To summarize, this expression quantifies the excess distortion caused by quantizing the estimated variances $\Sigma_{ii}$, $1 \leq i \leq N$. It assumes that $\Sigma_{ii}$ is the correct variance of the input to the $i^{th}$ quantizer. In addition, it assumes Gaussian input distributions.

5.3.5 Operational Form of the Overhead Allocation Rule

Now we are in a position to specify the operational form of the overhead allocation rule. Substituting the expressions obtained in equations 5.18, 5.23, and 5.47 into the general form of the rule, equation 5.17, gives

$$N \frac{\Delta_m^2}{12} + N \frac{\Delta_U^2}{12} \sum_{i=1}^{m} \Sigma_{ii} + \frac{9}{16} \Delta_\Sigma^2 \sum_{i=1}^{m} d^*(b_i^*; \Sigma_{ii}) \leq \frac{\alpha^2}{4} \hat{D}.$$ 

(5.48)

For any choice of quantization step sizes $\Delta_m$, $\Delta_U$, and $\Delta_\Sigma$ that satisfies this constraint the corresponding codebook overhead would be

$$\text{Total Overhead} = \sum_{i=1}^{N} b_{\mu_i} + N m b_U + m b_\Sigma \text{ (bits)},$$

(5.49)

where $b_{\mu_i}$, $b_U$, and $b_\Sigma$ are determined using equations 5.21, 5.24, and 5.37. A reasonable way to choose the quantization step sizes is to split the constraint equally among the three terms on the left-hand side, i.e.

$$N \Delta_m^2 = \alpha^2 \hat{D},$$

(5.50)

$$N \Delta_U^2 \sum_{i=1}^{m} \Sigma_{ii} = \alpha^2 \hat{D},$$

(5.51)

$$\Delta_\Sigma^2 \sum_{i=1}^{m} d^*(b_i^*; \Sigma_{ii}) = \frac{4}{27} \alpha^2 \hat{D}.$$ 

(5.52)
Equations 5.50–5.52 give almost the final form of the allocation rule. Although these equations could be used, we want to modify them so that the choice of the quantization step sizes $\Delta_m$, $\Delta_U$, and $\Delta_S$ does not depend on the bit allocation $b^*$. The advantage of doing this is that one can determine the codebook overhead before performing the bit allocation for the NAPC. This opens up the possibility of taking into account the codebook overhead when the bit allocation is performed. Such a bit allocation procedure will be described next in Section 5.4.

To make the choice of $\Delta_m$, $\Delta_U$, and $\Delta_S$ independent of the bit allocation $b^*$ we assume that the rate-distortion operating point of the algorithm will be chosen to be very close to the high-rate distortion asymptote. The distortion asymptote $\hat{D}_a$ is given by

$$\hat{D}_a = \sum_{i=1}^{N} L_{ii},$$  \hfill (5.53)

where $L$ is the filtering matrix in NAPC space. Recall that $\hat{D}_a$ equals the distortion of the non-quantized linear signal estimate. Let us first modify Equation 5.51. That expression must be modified because $m$, the number of NAPC with non-zero bit allocations, clearly depends on the bit allocation $b^*$. We obtain a bound for $m$ by truncating the NAPC representation of the signal estimate so that the distortion asymptote is degraded only slightly. If the NAPC representation is truncated at the $m^{th}$ coefficient, the high-rate distortion becomes

$$\hat{D}_a(m) = \hat{D}_a + \sum_{i=m+1}^{N} \Sigma_{ii}.$$  \hfill (5.54)

The truncation point $\bar{m}$ is chosen as

$$\bar{m} = \min \left\{ m \mid 1 \leq m \leq N, \hat{D}_a(m) - \hat{D}_a \leq \frac{\beta}{2} \hat{D}_a \right\},$$  \hfill (5.55)

for some small $\beta > 0$. Then the distortion operating point $\hat{D}$ is set to

$$\hat{D} = \left( 1 + \frac{\beta}{2} \right) \hat{D}_a(\bar{m}),$$  \hfill (5.56)

$$\approx \left( 1 + \frac{\beta}{2} \right)^2 \hat{D}_a,$$  \hfill (5.57)

$$\approx (1 + \beta) \hat{D}_a.$$  \hfill (5.58)
for small $\beta$. This choice of $\hat{D}$ implies

$$
\sum_{i=1}^{m} d^e(b_i^*; \Sigma_{ii}) = \hat{D} - \hat{D}_a(\tilde{m})
$$

(5.59)

$$
= \frac{\beta}{2} \hat{D}_a(\tilde{m}),
$$

(5.60)

which makes equation 5.52 independent of the bit allocation $b^*$. With these modifications equations 5.50–5.52 become

$$
N \Delta_m^2 = \alpha^2 (1 + \beta) \hat{D}_a,
$$

(5.61)

$$
N \Delta_U^2 \sum_{i=1}^{\tilde{m}} \Sigma_{ii} = \alpha^2 (1 + \beta) \hat{D}_a,
$$

(5.62)

$$
\Delta_\Sigma^2 = \frac{4}{27} \alpha^2 (2/\beta + 1).
$$

(5.63)

The operational form of the overhead allocation rule is given by equations 5.61–5.63. The quantization step sizes $\Delta_m$, $\Delta_U$, and $\Delta_\Sigma$ are determined from these equations. Once the quantization step sizes have been determined, the codebook overhead is given by equation 5.49. $\hat{D}_a$ and $\tilde{m}$ are given by equations 5.53 and 5.55, respectively. The parameters $\alpha$ and $\beta$ control the excess distortion, which should be approximately bounded by $\alpha (1 + \beta) \hat{D}_a$.

### 5.4 Bit Allocation Accounting for Codebook Overhead

As discussed in Section 5.3, when using our adaptive NAPC-based coding algorithm, part of the bit budget allocated for coding a given data block is spent on coding certain parameters of the codebook. Specifically, it is necessary to communicate to the decoder the estimate of the mean signal ($\tilde{m}_y$), and the eigenvectors ($U$) and variances ($\Sigma$) of the filtered NAPC coefficients. Those parameters form part of the encoded representation of the signal estimate. When the size of the data block and the bit budget per block are small, the overhead spent in coding those parameters (mainly the eigenvectors) can be a significant fraction of the bit budget. Then it becomes necessary to take into account this overhead when performing the bit allocation for the rest of the encoded representation, i.e. the NAPC coefficients, in order not to exceed the specified bit rate.

In this section we derive a bit allocation rule which takes into account codebook overhead. The rule can be applied whenever the codebook overhead has the following two com-
ponents: 1) a fixed component which is independent of how many transform coefficients are retained, and 2) a variable component which is directly proportional to the number of transform coefficients retained. In the context of our compression algorithm the fixed overhead component is associated with the estimate of the mean signal, and the variable overhead component is associated with the eigenvectors and variances of the NAPC coefficients. The new bit allocation rule is based on a modification of the familiar log-variance rule. Its derivation involves the use of simple concepts in constrained optimization.

Let us first briefly review the classical bit allocation problem. That problem was treated in Section 3.3.1 as part of the derivation of an optimal transform codebook for jointly-Gaussian noisy data. For convenience we shall reproduce here the statement of the problem. It was shown in Section 3.3.1 that for the optimal encoder and decoder matrices, and the optimal quantizers, the optimal bit allocation with a bit rate constraint $B$ solves the problem

$$
\text{minimize } D_W^e = \sum_{i=1}^{N} \lambda_i d(b_i)
$$

subject to $b_i \geq 0, 1 \leq i \leq N$

$$
\sum_{i=1}^{N} b_i \leq B,
$$

(5.64)

where $D_W^e$ is the coding distortion, $b_i$ is the bit allocation for the $i^{th}$ quantizer, $d(b_i)$ is the quantizer distortion for a unit-variance input, $\lambda_i$ is the variance of the input to the $i^{th}$ quantizer, and $B$ is the bit rate (bits/vector). For practical quantizers the bit allocations $b_i$ must also be integers. However, a closed-form solution is not known when that constraint is enforced. A useful approximate solution can be obtained by ignoring the integer constraint and taking $d(b) \propto 4^{-b}$. Then the solution is the well-known log-variance rule given by equations 3.50 and 3.51. In practice the log-variance rule is used as a first guess and the final integer bit allocation is obtained by modifying it using some reasonable procedure.

We consider the following variation of the classical bit allocation problem:

$$
\text{minimize } \sum_{i=1}^{N} \lambda_i d(b_i)
$$

subject to $b_i \geq 0, 1 \leq i \leq N,$

$$
\sum_{i=1}^{N} b_i + O_e 1_+(b_i) \leq B - O_f,
$$

(5.65)

where

$$
1_+(b) = \begin{cases} 
1, & b > 0; \\
0, & \text{else}, 
\end{cases}
$$

(5.66)
and we assume \(d(b) = 4^{-b}\). Here \(O_f\) and \(O_c\) denote the fixed overhead and the overhead per transform coefficient, respectively. The fundamental difference between this bit allocation problem and the classical one lies in the variable overhead component; the fixed overhead component does not add any complexity by itself. For our algorithm the overhead components are given by

\[
O_f = \frac{1}{K} \sum_{i=1}^{N} b_{\mu_i}, \quad (5.67)
\]

\[
O_c = \frac{1}{K} (N b_U + b_{\Sigma}), \quad (5.68)
\]

where \(K\) is the length of the data block, and \(b_{\mu_i}, b_U\), and \(b_{\Sigma}\) are given by equations 5.21, 5.24, and 5.37, respectively.

Let \(b^* = [b^*_1, b^*_2, \ldots, b^*_N]\) denote the solution to the new bit allocation problem and put \(n^* = \sum_{i=1}^{N} 1_{+}(b^*_i)\). To find \(b^*\) we first solve for the optimal solution \(b^*(n)\) assuming the number of non-zero bit allocations equals \(n\), for \(n = 1, 2, \ldots, N\). Thus, we solve the following set of problems:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} \lambda_i \ d(b_i) \\
\text{subject to} & \quad b_i \geq 0, \ 1 \leq i \leq n, \\
& \quad b_i = 0, \ n < i \leq N, \\
& \quad \sum_{i=1}^{N} b_i \leq B - O_f - n O_c,
\end{align*}
\]

(5.69)

for \(n = 1, 2, \ldots, N\). That \(b^*\) is the solution to one of these problems follows from the following observations:

a) If a bit allocation vector \(b = [b_1, b_2, \ldots, b_N]\) is a feasible point, i.e. it satisfies the constraints, of problem 5.69 for a given \(n\), it is also a feasible point for the problem in equation 5.65. Therefore, the constraint set of the former problem is a subset of the constraint set of the latter problem.

b) Let \(b^*(n)\) be the solution to the problem in equation 5.69 for a given \(n\). From the previous observation it follows that

\[
\sum_{i=1}^{N} \lambda_i d(b_i^*(n)) \geq \sum_{i=1}^{N} \lambda_i d(b_i^*). \quad (5.70)
\]
c) $b^*$ is a feasible point of problem 5.69 when $n = n^*$.

From these observations it follows that $b^* = b^*(n^*)$ for some integer $n^*$, $1 \leq n^* \leq N$. The solution $b^*(n)$ to the problem in equation 5.69 is given by the log-variance rule in equations 3.50 and 3.51 by making the substitutions $N \leftarrow n$ and $B \leftarrow B - O_f - n O_c$, i.e.

$$
   b_i^*(n) = \begin{cases} 
   \frac{B - O_f - n O_c}{k(n)} - \frac{1}{k(n)} \sum_{j=1}^{k(n)} \log_4 \left( \frac{\lambda_j}{\lambda_i} \right), & 1 \leq i \leq k(n), \\
   0, & \text{else}, 
\end{cases}
$$

(5.71)

where $k(n) (\leq n)$ is the largest integer such that

$$
   B - O_f - n O_c + \sum_{j=1}^{k(n)} \log_4 \left( \frac{\lambda_k(n)}{\lambda_j} \right) > 0.
$$

(5.72)

The desired bit allocation $b^*$ can be found by computing $b^*(n)$ for $n = 1, 2, \ldots, N$ using equations 5.71 and 5.72, and choosing the one that yields the minimum distortion $\sum_{i=1}^{N} \lambda_i 4^{-b_i^*(n)}$. Since the final bit allocation must be integer-valued we modify $b^*$ by first rounding down all its elements and by performing as many "greedy" one-bit allocations as possible without exceeding the desired bit rate $B$. The "greedy" one-bit allocations are done as in the MLDA (see page 60). The final bit allocation is generally suboptimal.

Finally, we would like to point out that, in general, to find the desired bit allocation $b^*$ it is necessary to compute $b^*(n)$ only for a subset of all possible values of $n$. To see this, note that if the number of non-zero bit allocations $k(n)$ in $b^*(n)$ is strictly less than $n$, then $b^*(n)$ can not be the optimal solution of problem 5.65. If $k(n) < n$, $b^*(k(n))$ gives a lower distortion than $b^*(n)$ because the number of bits allocated to the transform coefficients equals $B - O_f - k(n) O_c$ in the former case versus $B - O_f - n O_c$ in the latter case. Now, from equation 5.72 one can see that $k(n + 1) \leq k(n)$. So it is only necessary to compute $b^*(n)$ starting with $n = 1$ until $k(n) < n$ or $n = N$, whichever condition is satisfied first.

5.5 Tests with Simulated AIRS Data

We will now report the results of three tests of the coding algorithm described above conducted using simulated noiseless spectra from NASA's Atmospheric Infrared Sounder (AIRS). The main issues addressed in the tests are:

2. Assessing the cost of codebook overhead.


By performance we mean the rate-distortion characteristic of the algorithm. Of these three points, the first one is particularly important in the context of the noisy coding problem.

An important difference between the noisy and noiseless coding problems is that in practical instances of the latter problem it is not possible to evaluate the distortion directly for a given set of (noisy) input data. The reason is simply that the desired output of the codebook, the noiseless signal, is not available; otherwise one would not bother coding the noisy data. The implication is that in practice one must either rely on a prediction of the distortion between the output of the codebook and the noiseless signal, or use a different metric that can be evaluated directly from the input data. For our coding algorithm it is possible to predict the distortion by assuming that the model of the noisy data is correct and the empirically designed codebook is optimal. Specifically, the average distortion obtained by applying the coding algorithm to a given noisy data set is predicted by computing the expected distortion assuming that:

- the signal and noise are independent Gaussian random vectors,

- the noise mean and covariance are known,

- the estimates of the signal mean and covariance obtained from the noisy data are correct.

On the other hand, because we do have access to the noiseless signal in the tests conducted here, we can compute the average distortion directly and compare it to the predicted value in each case.

5.5.1 Details of the Tests

Before proceeding with the discussion of the tests it is appropriate to provide some relevant details about the data set used, and detail how the predicted and actual distortions were evaluated in the tests. To test our coding algorithm we used a data set consisting of 7496 simulated noiseless AIRS spectra provided by the AIRS science team. The spectra
were simulated over land during nighttime. Cloudy conditions were also simulated. A corresponding set of noisy measurements was generated by adding pseudo-random Gaussian noise to the set of noiseless spectra. Two different noise covariance matrices were used: the nominal AIRS (diagonal) noise covariance, and a noise covariance derived from the analysis of the AIRS Engineering Model data in Section 4.5. The coding algorithm was then applied to data blocks extracted from the noisy data set, i.e. for each data block an empirical codebook was designed and used to code the same data, and the predicted distortion and the actual distortion were evaluated in each case. The predicted distortion is given by

\[ D_{\text{predicted}} = \sum_{i=1}^{N} L_{ii} + \sum_{i=1}^{N} \Sigma_{ii} d(b^*_i), \]  

(5.73)

where \( L \) is the (diagonal) filtering matrix in NAPC space, \( \Sigma \) is the (diagonal) covariance matrix of the filtered NAPC, \( b^* \) is the bit allocation given by equations 5.71 and 5.72, and \( d(\cdot) \) is the quantizer distortion for a unit-variance input. The expression above is based on the assumption that the data model is correct (see items 1–3 in the previous paragraph) and that the empirical codebook is optimal. The actual distortion was obtained as

\[ D_{\text{actual}} = \frac{1}{k} \sum_{t=1}^{k} (x(t) - \hat{x}(t))^T C_{nn}^{-1} (x(t) - \hat{x}(t)), \]  

(5.74)

where \( k \) is the data block size, \( x(t) \) is a noiseless spectrum, and \( \hat{x}(t) \) is the corresponding estimate obtained using the empirical codebook.

Because in a remote sensing application the accuracy of the compressed data is the most important consideration, we are mostly interested in the near-lossless performance of the algorithm. By that we mean the regime where the distortion in the compressed data becomes limited by the noise (due to calibration errors and instrument noise) in the measured data, and not by the specified bit rate. That will occur at sufficiently high bit rates where the distortion is determined by the accuracy of the (non-quantized) signal estimate.

A meaningful way to quantify the high-rate or estimation distortion is in terms of the equivalent noise reduction. Note that the noise-adjusted distortion can be written as

\[ D = E\left\{ (\hat{x}(t) - x(t))^T C_{nn}^{-1} (\hat{x}(t) - x(t)) \right\}, \]

\[ = \sum_{i=1}^{N} E\left\{ \left( C_{nn}^{-1/2} (\hat{x}(t) - x(t)) \right)_i^2 \right\}, \]  

(5.75)

(5.76)
where $C_{nn}$ is the noise covariance matrix, $x(t)$ is the signal (noiseless spectrum), $\hat{x}(t)$ is the signal estimate generated by our algorithm, and the subscript $i$ denotes the $i^{th}$ element of the vector in parenthesis. If no filtering and coding of the noisy data $y(t)$ were done, then $\hat{x}(t) = y(t)$ and

$$D = \sum_{i=1}^{N} E \left\{ \left( C_{nn}^{-1/2} (y(t) - x(t)) \right)_i^2 \right\}, \quad (5.77)$$

$$= \sum_{i=1}^{N} E \left\{ \left( C_{nn}^{-1/2} n(t) \right)_i^2 \right\}, \quad (5.78)$$

$$= N. \quad (5.79)$$

The last equality follows by noting that the elements of $C_{nn}^{-1/2} n(t)$ all have unit variance. This simply states that when no processing of the noisy data is performed the distortion equals the total noise energy in noise-adjusted space. Thus, the noise reduction (NR) obtained using our coding algorithm can be defined as

$$NR = 10 \log_{10} \left( \frac{N}{D} \right). \quad (5.80)$$

where $D$ is the noise-adjusted distortion.

5.5.2 Test 1: Comparison of Predicted and Actual Rate-Distortion Performance

The purpose of this first test is to investigate two closely related issues: a) the agreement between the predicted distortion and the actual distortion of the empirical codebooks, and b) the cost in terms of rate-distortion performance when estimating the signal statistics from finite blocks of noisy data. To simplify interpretation of the results the codebook overhead was ignored in this test. The cost of codebook overhead is quantified in the next test.

In this test the nominal AIRS (diagonal) noise covariance matrix was used. The nominal noise-equivalent radiance (NER) or rms noise as a function of wavelength is shown in Figure 4-13. For the purpose of illustration, the signal-to-noise ratios of the NAPC of the set of 7496 simulated AIRS spectra are shown in Figure 5-4 for that choice of noise covariance matrix. The eigenvectors corresponding to the first nine NAPC are shown in
Figure 5-4: Signal-to-noise ratio of the NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the nominal AIRS noise covariance matrix.

Figure 5-5. The eigenvectors and the signal-to-noise ratios (eigenvalues) were computed by "noise-adjusting," i.e. normalizing, the radiance in each spectral channel by its corresponding rms noise and diagonalizing the sample covariance matrix of the set of "noise-adjusted" spectra.

The main results of the test are illustrated in Figure 5-6, which shows the predicted and actual rate-distortion performance of the algorithm when applied to blocks of noisy data of three different sizes. The bit rate is given in bits per pixel per channel, where one pixel is the 2378-channel spectrum observed at a given time and location. Let us first comment on the agreement between the predicted and actual curves. As expected, the curves approach a horizontal asymptote for high enough bit rates. In that regime the signal estimate is quantized at such high resolution that quantization errors are negligible relative to errors in the estimate itself, so that the total distortion is dominated by the latter errors. The value of the asymptote is what we termed the estimation distortion in Section 3.3.1. It is seen that the predicted and actual curves agree very well in their asymptotic behavior. In addition, the value of the asymptote is observed to decrease as the size of the data block increases.
Figure 5-5: Eigenvectors corresponding to the first nine NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the nominal AIRS noise covariance matrix. The horizontal axis on all the plots is wavelength (μm).
Table 5.1: Predicted and actual quantizer distortion for the nine filtered NAPC distributions shown in Figure 5-7. The bit allocation used corresponds to a bit rate of 0.051 bits/pixel/channel; the lowest bit rate shown in Figure 5-6.

<table>
<thead>
<tr>
<th>Number of filtered NAPC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
</table>

This would be expected for stationary data because the empirical signal estimate obtained using the coding algorithm would converge to the linear least-squares estimate as the block size increases. In all cases the asymptote was practically reached (within 10 percent) at a bit rate of 0.2 bits/pixel/channel. The raw (uncalibrated) AIRS data is digitized using 12–14 bits, the average being about 13 bits/pixel/channel. The plots in Figure 5-6 show that high compression ratios (CR) (32:1 at 0.4 bits/pixel/channel) can be achieved without degrading the non-quantized signal estimate more than three percent in distortion.

The agreement between the predicted and actual curves is not so close at low bit rates, where the distortion decreases rapidly toward the asymptote. That seems to occur mostly because the distributions of the NAPC are not exactly Gaussian, which causes the distortion of the quantizers to exceed their predicted values. Figure 5-7 shows histograms of the first nine filtered NAPC evaluated using a set of 2400 noisy spectra. The figure shows that the distributions of the third, sixth, and seventh filtered NAPC deviate noticeably from the corresponding Gaussian density. The predicted and actual quantizer distortion for all nine filtered NAPC are listed in Table 5.1. The bit allocation used for those results listed in the table corresponds to a bit rate of 0.051 bits/pixel/channel; the lowest bit rate shown in Figure 5-6. The mismatch between the predicted and the actual quantizer distortion correlates well with the mismatch between the empirical and the assumed distribution. For the third filtered NAPC, whose distribution appears to deviate the most from a Gaussian, the actual distortion is more than sixty times the predicted value. That alone accounts for most of the deviation between the predicted and actual distortion at 0.051 bits/pixel/channel in Figure 5-6. Interestingly, those NAPC whose empirical distributions deviate the most from a Gaussian appear to be associated with eigenvectors (Figure 5-5) in which one or very few spectral channels are weighted heavily compared to the rest.

The good agreement at high bit rates between the predicted and actual distortion of
Figure 5-6: Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. These simulations used the nominal AIRS noise covariance matrix. The codebook overhead was ignored (set to zero). The graphs corresponding to a block size of 2400 spectra show results averaged over three of such data blocks. The other two plots show results obtained from one data block only. CR stands for compression ratio.
Figure 5-7: Histograms of the first nine filtered NAPC evaluated using a set of 2400 noisy AIRS spectra. The corresponding eigenvectors (essentially those shown in Figure 5-5) and the filtering matrix were computed from 7200 noisy spectra. The noise covariance matrix used is the nominal AIRS noise covariance. Also shown for comparison are the zero-mean Gaussian densities with the corresponding variances. The amplitude of each density was scaled so that its integral would match the area of the corresponding histogram.
Figure 5-8: Predicted and actual rate-distortion characteristics of the codebook designed with “knowledge” of the signal statistics. To approximate that condition, the mean vector and covariance matrix of the signal were estimated from the complete set of 7496 noiseless spectra. The NAPC-based codebook was designed for those signal statistics and the nominal AIRS noise covariance matrix. The actual distortion was evaluated using all 7496 pairs of noisy and noiseless spectra. CR stands for compression ratio.

our algorithm owes partly to the fact that in that regime the assumption that the signal and noise are Gaussian is irrelevant. As the data block size is increased, the distortion asymptote will approach the distortion of the linear least-squares estimator, which depends only on the second-order statistics of the signal and the noise, and not on their specific distributions. Thus, the ability to predict the distortion of our coding algorithm at high bit rates depends on the accuracy of the estimates of the signal mean and covariance matrix, and the accuracy of the assumed noise mean and covariance matrix. The results in Figure 5-6 above give some evidence that our coding algorithm can obtain good estimates of those statistics even with relatively little data.

Now let us comment on the absolute performance of the empirical codebooks. It would be desirable to compare the performance of the empirical codebooks shown in Figure 5-6 with the performance of the theoretically optimal codebook to assess the penalty in performance. Even though it is not possible for us to obtain the optimal codebook exactly
(assuming it exists) because we only have a finite data set, we can obtain an approximation by designing a codebook using the whole set of 7496 noiseless spectra. That would at least give us a reference to assess the penalty incurred by having to design the codebook from noisy data because the signal statistics are not known a priori. Figure 5-8 shows the predicted and actual rate-distortion characteristics of the codebook designed from the noiseless data. Again, the predicted distortion assumes a jointly Gaussian signal. The actual distortion was evaluated using all 7496 pairs of noisy and noiseless spectra. A comparison of Figure 5-8 with Figure 5-6 shows that the main difference is a higher estimation distortion for the codebooks designed from noisy data. The codebook designed from 7496 noiseless spectra achieved an estimation distortion of 66 while the codebooks designed from noisy spectra achieved an estimation distortion of 98–153.3 depending on the block size. This corresponds to a high bit rate noise reduction of 15.5 dB for the former codebook versus 11.9–13.8 dB for the latter codebooks. The predominant cause for the lower noise reduction is the departure of the empirical filtering matrix from the optimal one. Figure 5-9 provides evidence of such departure by comparing the filter amplitude in empirical NAPC space derived from noisy and noiseless data. It is seen that the filter amplitudes derived from noisy data are too high for the high-order NAPC. This is in spite of the improvement obtained by using the empirical noise-eigenvalue distribution to estimate the filter amplitude, as described in Section 3.4.3.

To put in perspective the compression results obtained above it is informative to compare them with the maximum lossless compression possible for the same data. We can obtain a lower bound on the minimum average number of bits needed to code the simulated AIRS data reversibly in terms of the entropy of the (digitized) noise in those data [12]. To compute the entropy of the noise we assume a dynamic range for each channel equal to twice the black-body radiance at 350 K. The specified scene dynamic range for AIRS is 195–357 K [46] so our assumption is fairly conservative. Furthermore, we assume that prior to analog-to-digital conversion the noise in each channel is Gaussian and independent from the noise in other channels. The rms value of the noise is taken to be the nominal NER at 250 K shown in Figure 4-13. Then using the analog-to-digital resolution for each channel, which ranges from 12 to 14 bits, we estimate the entropy of the digitized noise using the approximation in [12]. Under these assumptions one predicts a lower bound of 2–4 bits for coding the noise alone, depending on the channel. Adding up the lower bounds for all the channels
Figure 5-9: Filter amplitude estimated from data blocks of varying sizes. The noise covariance matrix used is the nominal AIRS noise covariance.

one obtains a maximum lossless compression ratio of 4.7:1, which is only achievable if zero bits are required to code the signal. The near-lossless compression ratios of 32:1 obtained with our algorithm compare favorably with the predicted maximum lossless compression.

5.5.3 Test 2: Rate-Distortion Performance with Codebook Overhead

This test evaluates the impact of codebook overhead on the rate-distortion performance of the coding algorithm. The impact of codebook overhead will be significant when the bit budget per data block is small enough. Since the bit budget per block equals the product of bit rate (bits/pixel) and block size, the overhead becomes a limitation when either the bit rate or the block size is small.

The test consisted of evaluating the rate-distortion performance of the algorithm as in the previous test, but this time including the codebook overhead as part of the specified bit rate. As before, the nominal AIRS noise covariance matrix was used to generate the noisy data. The amount of overhead for each data block was determined using the allocation rule described in Section 5.3. The parameters α and β, which control the excess distortion
<table>
<thead>
<tr>
<th>Bit rate (bits/pixel/channel)</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2400</td>
</tr>
<tr>
<td>0.05</td>
<td>76.6</td>
</tr>
<tr>
<td>0.10</td>
<td>77.1</td>
</tr>
<tr>
<td>0.20</td>
<td>77.5</td>
</tr>
<tr>
<td>0.41</td>
<td>77.8</td>
</tr>
<tr>
<td>0.81</td>
<td>75.9</td>
</tr>
<tr>
<td>1.63</td>
<td>63.9</td>
</tr>
<tr>
<td>3.25</td>
<td>32.0</td>
</tr>
</tbody>
</table>

Table 5.2: Percent of the bit rate spent on codebook overhead as a function of block size and bit rate. The results corresponding to a block size of 2400 spectra are averages over three of such data blocks.

caus[ed by quantizing the codebook parameters, were set to $\alpha = \beta = 0.1$. This choice of $\alpha$ and $\beta$ should yield an excess distortion which does not exceed ten percent of the lowest achievable distortion.

Table 5.2 shows the codebook overhead as a percentage of the specified bit rate for empirical codebooks designed for block sizes of 2400, 4800, and 7200 spectra. For most of the cases evaluated in the table the overhead is a significant fraction of the bit rate. The distribution of the overhead among the codebook parameters for all these codebooks was as follows: $b_{\mu}$, the number of bits for each element of the sample mean, varied from 15 to 19 bits; $b_{\Omega}$, the number of bits for each element of each eigenvector, was 19 bits; $b_{\Sigma}$, the number of bits for the variance of each filtered NAPC, was 7 bits.

The predicted and actual rate-distortion characteristics for the three different block sizes are displayed in Figure 5-10. Comparing the curves with the ones obtained in the previous test, shown in Figure 5-6, it is seen that the distortion increased significantly at very low bit rates. For a block size of 2400 spectra the distortion increased substantially at bit rates below 0.8 bits/pixel/channel, while for block sizes of 4800 and 7200 spectra the increase in distortion was significant only at bit rates below 0.4 bits/pixel/channel. In all cases the increase in distortion decreased monotonically with bit rate. The value of the distortion asymptote actually decreased slightly compared to the results of the previous test. This is not a contradiction since the empirical codebooks depart from the “true” optimal codebook, so there is always some room for improvement. One factor that likely contributed to decreasing the value of the distortion asymptote is the truncation of the
Figure 5-10: Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. These simulations used the nominal AIRS noise covariance matrix. The codebook overhead was included in the indicated bit rate. The graph corresponding to a block size of 2400 spectra shows results averaged over three of such data blocks. The other two plots show results obtained from one data block only. CR stands for compression ratio.
NAPC representation done as part of the overhead allocation (see page 142). The truncation discards some of the high-order NAPC that are not attenuated enough by the empirical filtering matrix (see Figure 5-9).

5.5.4 Test 3: Rate-Distortion Performance with Different Noise Covariances

Here we evaluate the rate-distortion performance of the compression algorithm when the data are corrupted by noise having a covariance matrix different from the nominal one. The purpose of these simulations is to illustrate how the algorithm would perform on AIRS data with various "reasonable" noise covariance matrices. The main performance indicators of interest are the high-rate distortion asymptote, or equivalently the maximum noise rejection, and the minimum bit rate at which this asymptote is (approximately) reached. In place of the nominal AIRS noise covariance we use synthetic noise covariances. The selection of the alternate covariances is based partly on the estimate of the calibration "noise" covariance obtained in Section 4.5 for a subset of the AIRS channels. In particular, we shall use non-diagonal covariance matrices to study the impact of inter-channel noise correlations.

First we evaluate the performance of the compression algorithm for a noise covariance matrix which incorporates a simple form of inter-channel noise correlations. Based on the results of the AIRS calibration error analysis performed in Section 4.5 it seems reasonable to expect non-zero error correlations between channels physically located on a common detector array. We choose to model such correlations using a block-diagonal covariance matrix $C'_{nn}$:

$$C'_{nn} = \text{diag} \left( C'_{nn}(1), C'_{nn}(2), \ldots, C'_{nn}(m) \right),$$  \hspace{1cm} (5.81)

where $m$ is the number of detector arrays, and the matrix block corresponding to the $i^{th}$ array is given by

$$C'_{nn}(i) = S(i) \left( (1 - \rho) I + \rho uu^T \right) S(i).$$  \hspace{1cm} (5.82)

Here $S(i)$ is the diagonal matrix of rms noises for the $i^{th}$ detector array, $u = [1 \ 1 \ \ldots \ 1]^T$, and $I$ is the identity matrix. Although not indicated explicitly, the size of the block $C'_{nn}(i)$ varies according to the number of channels of the corresponding detector array. From the definition of $C'_{nn}(i)$ it is easy to verify that $\rho$ is the noise correlation coefficient for any pair
of channels on the same detector array\(^3\). Thus, our simple model assumes that the noise is uniformly correlated within each detector array.

For the noise covariance model \( C'_nn \) described above it is easy to anticipate how the noise-adjusted data will differ from the case of a diagonal noise covariance (\( \rho = 0 \)). Note that the matrix of noise correlation coefficients for those channels located in \( i^{th} \) detector array can be decomposed as

\[
S^{-1}(i)C'_nn(i)S^{-1}(i) = (1 - \rho) \left\{ I + \frac{\rho}{1 - \rho} uu^T \right\}^{1/2} \\
= (1 - \rho) \left\{ \frac{u}{\sqrt{N(i)}} \begin{bmatrix} 1 + N(i) \frac{\rho}{1 - \rho} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 + N(i) \frac{\rho}{1 - \rho} \end{bmatrix} \begin{bmatrix} u \\ \sqrt{N(i)} \end{bmatrix} \right\}^T,
\]

where \( N(i) \) is the number of channels in the \( i^{th} \) detector array, \( 0 \) is a vector with zero-valued elements, and \( P \) is a \( N(i) \times (N(i) - 1) \) matrix with orthonormal columns, all of them orthogonal to \( u \). This decomposition clearly shows that for \( \rho > 0 \) the unit-variance noises are preferentially correlated along the "common-mode" direction \( u \). Using the above expression one can verify that a valid noise-adjusting matrix is given by

\[
C'_nn(i)^{-1/2} = (1 - \rho)^{-1/2} \left\{ I - \frac{1}{N(i)} \left( 1 - \left( 1 + N(i) \frac{\rho}{1 - \rho} \right)^{-1/2} \right) uu^T \right\} S(i)^{-1}, \tag{5.85}
\]

i.e. the product \( C'_nn(i)^{-1/2}C'_nn(i) \left( C'_nn(i)^{-1/2} \right)^T \) equals the identity matrix. Let \( \bar{y}(i) \) denote the vector of mean-centered radiances for those channels with detectors located on the \( i^{th} \) array. The corresponding vector of noise-adjusted radiances is then

\[
C'_nn(i)^{-1/2} \bar{y}(i) = \left(1 - \rho \right)^{-1/2} \left\{ S(i)^{-1} \bar{y}(i) - \left( 1 - \left( 1 + N(i) \frac{\rho}{1 - \rho} \right)^{-1/2} \right) \overline{S(i)^{-1} \bar{y}(i) u} \right\}, \tag{5.86}
\]

where \( \overline{S(i)^{-1} \bar{y}(i)} \) denotes the average taken over the elements of \( S(i)^{-1} \bar{y}(i) \). Thus, the radiances for the channels on the \( i^{th} \) array are noise-adjusted (after subtracting the sample mean) by first normalizing them with the diagonal matrix of rms noises \( S(i) \), then sub-

---

\(^3\)In order for \( C'_nn(i) \) to be a valid covariance matrix it is necessary that \( \rho \geq -1/(N(i) - 1) \), where \( N(i) \) is the number of rows and columns of the matrix.
tracting a fraction \(1 - \left(1 + N(i)\frac{\rho}{1-\rho}\right)^{-1/2}\) of the average value over the set of channels, and finally scaling the result by \((1 - \rho)^{-1/2}\).

There are two differences between the case \(\rho > 0\) and the case of zero noise correlation. First, the average or “common-mode” component of the normalized radiances is attenuated relative to all other components along directions orthogonal to the common-mode direction \(u\). Second, scaling the normalized radiances by \((1 - \rho)^{-1/2}\) increases the signal-to-noise ratio within the subspace orthogonal to the common-mode direction. This occurs because a larger fraction of the total noise energy is concentrated in the common-mode, so the noise energy in the remaining modes decreases accordingly. The opposite occurs for \(\rho < 0\).

As an example let us consider the case where \(S_i\)

\[
S = \text{diag}(S(1), S(2), \ldots, S(m)),
\]

(5.87)
equals the nominal AIRS rms noise matrix for a 250 K target, and \(\rho = 0.3\). Figure 5-11 shows the eigenvectors corresponding to the nine dominant NAPC. A comparison with the eigenvectors for \(\rho = 0\) shown in Figure 5-5 clearly shows the relative attenuation of the common-mode component across detector arrays. Recall that AIRS has seventeen detector arrays. Each array measures an average of 140 different spectral channels, so for \(\rho = 0.3\) the common-mode component is attenuated to \(100 \times \left(1 + 140 \cdot 0.3\right)^{-1/2} = 12.8\) percent of its value, on the average over all arrays. The signal-to-noise ratio (SNR) distribution in NAPC space is very similar in shape to the one for \(\rho = 0\), shown in Figure 5-4. However, the total SNR is now lower; \(4.7 \times 10^6\) versus \(3.1 \times 10^7\).

Figure 5-12 shows the performance of the algorithm for the same noise covariance matrix \(C_{nn}'\), i.e. with \(\rho = 0.3\) and \(S\) set equal to the nominal AIRS rms noise matrix for a 250 K target. It was assumed that the noise covariance is known exactly, so the data were noise-adjusted correctly. The most noticeable difference between this case and that of the nominal noise covariance, shown in Figure 5-10, is a much larger deviation between the predicted and actual rate-distortion curves. Again, this occurs because the distributions of the NAPC deviate from the assumed Gaussian form. The noise rejection at high bit rates in this case ranged from 12.0 dB to 13.8 dB, which is virtually identical (within 2 percent) to that obtained for the nominal noise covariance matrix.

Note that even though the total SNR decreased by about an order of magnitude com-
Figure 5-11: Eigenvectors corresponding to the first nine NAPC of the set of 7496 simulated AIRS spectra corrupted by additive noise having the covariance matrix $C_{nn}$ described in page 160, with $\rho = 0.3$ and $S$ set equal to the nominal AIRS rms noise matrix for a 250 K target. The horizontal axis on all the plots is wavelength (\mu m).
pared to the case of zero noise correlation, the ability to separate the noise from the signal was virtually unchanged. This can be explained by noting that the noise rejection depends on the sum $\sum_{i=1}^{N} \lambda_i/(\lambda_i + 1)$, where $\lambda_i$ is the SNR of the $i$th NAPC. When $\lambda_i$ is several orders of magnitude larger than unity, as is the case for the dominant NAPC for these data, the corresponding term in the sum is relatively insensitive to small changes in $\lambda_i$.

Next we evaluate the performance of the algorithm for a noise covariance matrix obtained partially from the AIRS calibration error analysis in Section 4.5. In that section we obtained an estimate of the calibration error covariance for a subset of the AIRS infrared channels. For the present purpose we combined those covariance estimates with the nominal AIRS noise covariance to obtain a noise covariance for the complete set of AIRS infrared channels. Because the calibration error covariance estimates obtained in Section 4.5 are rather noisy, they were not used directly but instead were modified as follows. Refer to Figure 4-13, which shows the estimated rms calibration errors for a subset of the channels along with the nominal rms noise corresponding to a 250 K target. For the two groups of channels around 4 $\mu$m and 13 $\mu$m the nominal rms values were scaled until their average value in dB matched the average of the corresponding estimated rms values. The scaled values were used as the rms noise values for those channels. For all other channels the nominal rms noise values were used. The correlation coefficients of the noise on different spectral channels were selected as follows. Refer now to figure 4-15, which shows the estimated correlation coefficients for the same subset of the AIRS channels. From the depicted matrix we chose to use the estimated correlations only for the two obvious blocks of channels, i.e. the block on the top left within module M-1, and the block containing channels from modules M-1 and M-4. The noise correlations between all other channels were set to zero. We denote the noise covariance constructed this way by $C''_{nn}$.

Figure 5-13 shows the rate-distortion performance of the coding algorithm for AIRS data corrupted by noise having the covariance matrix $C''_{nn}$ described above. Again, it was assumed that the noise covariance is known exactly, so the data were noise-adjusted appropriately. These results can be compared to those obtained for the nominal noise covariance matrix, shown in Figure 5-10. In this case the value of the asymptotic distortion varied between 94.9 and 142, which corresponds to a noise reduction of 12.2-14.0 dB. Thus, the noise reduction is essentially identical to that obtained with the nominal noise covariance matrix. Also, as before the distortion asymptote was essentially reached at a bit rate of
Figure 5-12: Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. These simulations used the noise covariance $C'_{nn}$ described on page 160, with $\rho = 0.3$ and $S$ set equal to the nominal AIRS rms noise matrix for a 250 K target. The codebook overhead was included in the indicated bit rate. The graph corresponding to a block size of 2400 spectra shows results averaged over three of such data blocks. The other two plots show results obtained from one data block only. CR stands for compression ratio.
Figure 5-13: Predicted and actual rate-distortion characteristics of empirical codebooks derived from and applied to data blocks of varying sizes. These simulations used the noise covariance $C''_{nn}$ described on page 164. The codebook overhead was included in the indicated bit rate. The graph corresponding to a block size of 2400 spectra shows results averaged over three of such data blocks. The other two plots show results obtained from one data block only. CR stands for compression ratio.
0.4 bits/pixel/channel except at the smaller block size of 2400 spectra. The only noticeable
difference between both sets of results is that in Figure 5-13 the deviation between the
predicted and actual curves at low bit rates is slightly larger. Again, this occurs because
the distributions of the NAPC coefficients depart from the assumed Gaussian shape.

5.6 Coding Large Noise Estimates

The coding algorithm described thus far in this chapter (Section 5.2) codes the signal
estimate for each channel and discards the noise estimates or residuals, i.e. the difference
between the noisy data and the signal estimate. Although it is generally desirable to discard
the noise in order to achieve good data compression, there may be outlier data points for
which the residuals or noise estimates are unusually large. For those abnormal data points,
coding the noise estimates in addition to the signal estimates affords the opportunity of
reconstructing the data and conducting later a detailed analysis to investigate their origin.
For example, the combination of the signal estimates and the preserved noise estimates may
reveal an unanticipated spectral feature occurring only in isolated geographic locations. The
noise estimates may be discarded subsequently if no useful information is obtained from
them.

In this section we describe a strategy for individually coding noise estimates which are
larger than typical. Our strategy is designed to preserve all noise estimates whose am-
plitudes exceed \( \xi \) times the rms value of the noise estimates on each channel, where the
threshold \( \xi \) is specified by the user. This strategy would be incorporated to the aforemen-
tioned coding algorithm to protect against the possibility of excessively distorting outlier
channels.

To apply our coding strategy to a block of \( K \) spectra first the following two steps are
performed:

1. For each spectrum \( y(t) \) in the current data block a noise estimate \( \hat{n}(t) \) is computed
   by taking the difference \( y(t) - \hat{x}(t) \), where \( \hat{x}(t) \) is the signal estimate obtained after
   quantization (see Figure 5-1).

2. For each channel, the rms value of the noise estimates is computed over the current
data block. To prevent the rms value from being inflated by outliers, any noise
estimates with amplitudes larger than three times the \textit{a priori} rms noise (given by
\( \sqrt{(C_{nn})_{i,i}} \) for the \( i^{th} \) channel) are excluded from the computation. Thus, the rms value \( \sigma_{\hat{n}_i} \) of the noise estimates for the \( i^{th} \) channel is estimated as

\[
\sigma_{\hat{n}_i}^2 \triangleq \frac{1}{|A_i|} \sum_{t \in A_i} \hat{n}_i(t)^2,
\] (5.88)

where \( A_i = \{ t \text{ such that } \hat{n}_i(t)^2 < 9 (C_{nn})_{i,i} \} \) and \( |A_i| \) is the number of elements in the set \( A_i \).

After performing these two steps one has an \( N \times K \) matrix of noise estimates for the current data block, and an \( N \times 1 \) vector of rms values for the same.

Our coding strategy codes, for each channel, all the noise estimates with amplitudes larger than \( \xi \) times their rms value computed as indicated above. For each channel those noise estimates are encoded by providing: 1) the number of noise estimates to be coded, 2) the time indexes (relative to the current data block) of those noise estimates, and 3) the uniformly quantized values of those noise estimates. This information is encoded using the following simple procedure:

1. For the \( i^{th} \) channel, record a binary flag indicating if there are any noise estimates to be coded.

2. If there are noise estimates to be coded, record the following:
   a. The number \( c_i \) of noise estimates to be coded. For a data block of length \( K \) this number is represented with a binary word of length \( \lceil \log_2 K \rceil \), where \( \lceil \cdot \rceil \) rounds toward infinity.
   b. The number \( c_i \) of bits required to uniformly quantize the noise estimates at the nominal resolution \( \Delta_i = DR_i / 2^{b_i} \), where \( DR_i \) and \( b_i \) are the nominal dynamic range and the A/D precision of the \( i^{th} \) channel, respectively. \( c_i \) is determined by the noise estimate \( \hat{n}_i(t) \) of largest amplitude within the current data block;

\[
c_i = \left\lfloor \log_2 \left( \max_{1 \leq t \leq K} \left\lfloor \frac{\hat{n}_i(t)}{\Delta_i} \right\rfloor \right) \right\rfloor + 1,
\] (5.89)

where the additional bit is for coding the sign of each noise estimate. Since \( c_i - 1 \) can vary from zero to \( b_i \), \( c_i \) itself can be represented by a binary word of length \( \lceil \log_2 (b_i + 1) \rceil \).
b. The time indexes of the noise estimates to be preserved. For a data block of length \( K \) each index requires a binary word of length \( \lceil \log_2 K \rceil \).

c. The values of the noise estimates quantized uniformly with resolution \( \Delta_i \). Each quantized value is a binary word of length \( c_i \).

Using the procedure above in conjunction with the coding algorithm in Section 5.2, the total bit rate (bits/pixel/channel) required for coding for a block of \( K \) spectra is given by

\[
R_T = R_0 + \frac{1}{K} + \sum_{i=1}^{N} \left( \frac{\lceil \log_2 K \rceil + \lceil \log_2 (b_i + 1) \rceil}{NK} \right) 1_+ (a_i) + \sum_{i=1}^{N} \frac{\lceil \log_2 K \rceil + c_i}{NK} a_i,
\]

where \( R_0 \) is the bit rate allocated to the signal estimate (the NAPC and the codebook overhead), and

\[
1_+ (x) = \begin{cases} 
1, & x > 0; \\
0, & \text{else.}
\end{cases}
\]\n
For long data blocks \( (K \) large) the total bit rate should be dominated by \( R_0 \) and the last term on the right hand side.

Although not indicated explicitly in the above expression, the total bit rate \( R_T \) must depend on the threshold \( \xi \). An expression relating \( R_T \) and \( \xi \) would be useful in practice as a guide for selecting an appropriate value for the latter. In general, \( R_T \) should increase whenever \( \xi \) is decreased. However, for any given data block the total bit rate is determined not only by \( \xi \) but also by the noise estimates. A relationship between \( R_T \) and \( \xi \) that does not involve the values of noise estimates is only possible in an average sense, using a suitable probabilistic model.

Consider estimating the expected bit rate under the assumption that the noise estimates \( \hat{n}_i(t), 1 \leq t \leq K \) are independent samples of a zero-mean Gaussian distribution with variance \( \sigma_{\hat{n}_i}^2 \). This model is reasonable if there are no outlier noise estimates other than those coming from the nominal noise distribution, which we assume to be Gaussian. Under this assumption, the number \( a_i \) of noise estimates to be coded is a binomial random variable with \( K \) Bernoulli trials and probability \( P_0 = 2\Phi(\xi) - 1 \) that the outcome of any trial equals zero, where \( \Phi(\cdot) \) is the cumulative probability distribution of a zero-mean unit-variance Gaussian random variable. It is then easy to show that

\[
E\{1_+ (a_i)\} = 1 - P_0^K,
\]
\[ E\{a_i\} = K(1 - P_0). \] (5.93)

To compute an exact expression for \( E\{R_T\} \) one would also have to evaluate \( E\{c_i a_i\} \), which is not easily evaluated. However, since \( c_i \leq b_i + 1 \) one can instead use the bound

\[
E\{c_i a_i\} \leq (b_i + 1) E\{a_i\}.
\]

Accordingly, the expected value of \( R_T \) is bounded above by

\[
E\{R_T\} \leq R_0 + \frac{1}{K} + (1 - P_0^K) \sum_{i=1}^{N} \frac{[\log_2 K] + [\log_2 (b_i + 1)]}{NK} + (1 - P_0) \sum_{i=1}^{N} \frac{[\log_2 K] + b_i + 1}{N},
\] (5.94)

where \( P_0 = 2\Phi(\xi) - 1 \), as indicated above.

The bound derived above clearly indicates that the simple encoding procedure used here is relatively inefficient for arbitrarily long block lengths. In particular, that expression contains a term proportional to \( [\log_2 K] \), which increases monotonically with the block length \( K \). That term corresponds to the number of bits used to individually represent the time index for each noise estimate. This can be improved if the time indexes are encoded collectively instead of individually. Specifically, note that for each channel the set of time indexes of the noise estimates to be coded can be represented as a binary sequence of length \( K \) with ones at the positions of the noise estimates to be coded and zeros everywhere else. Shannon's information theory asserts that it is possible to code such a sequence using an average number of bits per symbol arbitrarily close to, but not less than the entropy \( H(p) \), provided that the symbols in the sequence are statistically independent, the probability that any symbol equals one is \( p \), and the sequence is arbitrarily long [50]. These assumptions are consistent with the ones used to derive the bound above when \( p = 1 - P_0 \). Thus, if such an efficient encoding method is used to code the time indexes the expected bit rate \( R'_T \) is bounded by

\[
E\{R'_T\} \leq R_0 + H(1 - P_0) + (1 - P_0) \sum_{i=1}^{N} \frac{b_i + 1}{N}
\] (5.95)

for arbitrarily large \( K \). This expression can be used as a reference against which to compare equation 5.94 to assess the inefficiency of our encoding procedure.

Finally, let us apply the bounds derived above to the case of AIRS. The A/D precision \( b_i \) for AIRS varies from 12 to 14 bits depending on the channel, with the average being
about 13 bits. Table 5.3 lists the maximum average bit rate increment $E\{R_T\} - R_0$ as a function of the threshold $\xi$ and the block length $K$ assuming $b_t = 13$ for all channels. $K = 90 \times 56$ corresponds to 2.5 minutes of data, $K = 90 \times 112$ to 5 minutes of data, and so on. For a base bit rate $R_0 = 0.4$ bits/pixel/channel, the bit rate increments for $\xi = 2$, $\xi = 3$, and $\xi = 4$ are about 300 percent, 20 percent, and 0.75 percent, respectively, for the block lengths listed in the table. These results suggest that choosing $\xi \geq 3$ would not alter significantly the level of compression achieved by the coding algorithm, under normal conditions. From a scientific point of view, a threshold of $\xi = 3$ should be sufficient in practice since it would be increasingly difficult to detect any sporadic features of smaller amplitude even if the original data were preserved exactly. Table 5.4 lists the corresponding bit rate increments when the time indexes are entropy coded, as assumed in equation 5.95. Those results bound the efficiency of our simple encoding procedure.

### 5.7 Computational Burden

In this section the computational burden of the coding algorithm illustrated in Figure 5-1 is evaluated.

Table 5.5 lists a partial operation count for the codebook design portion of the algorithm.
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Mults.</th>
<th>Adds.</th>
<th>Total</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{m}_y$</td>
<td>0</td>
<td>$NK$</td>
<td>$NK$</td>
<td></td>
</tr>
<tr>
<td>$\hat{C}_{yy}$</td>
<td>$N^2K$</td>
<td>$N^2K$</td>
<td>$2N^2K$</td>
<td></td>
</tr>
<tr>
<td>$C_{nn}^{-1/2}\hat{C}<em>{yy}\left(C</em>{nn}^{-1/2}\right)^T$</td>
<td>$2N^3$</td>
<td>$2N^3$</td>
<td>$4N^3$</td>
<td>Assuming $C_{nn}^{-1/2}$ is precomputed</td>
</tr>
<tr>
<td></td>
<td>$2N^2$</td>
<td>0</td>
<td>$2N^2$</td>
<td>Assuming $C_{nn}$ is diagonal</td>
</tr>
<tr>
<td>$U, \Lambda$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Depends on matrix size and eigenvalue distribution</td>
</tr>
<tr>
<td>$L$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Negligible</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Negligible</td>
</tr>
<tr>
<td>$b_\mu, b_U, b_\Sigma$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Negligible</td>
</tr>
<tr>
<td>$b^*$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Negligible</td>
</tr>
<tr>
<td>$\bar{m}_y, U, \Sigma$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Negligible</td>
</tr>
</tbody>
</table>

Table 5.5: Approximate operation count for the codebook design portion of the algorithm. $N$ is the number of channels, and $K$ is the number of spectra used to compute the quantities.

All the dominant computations are covered with the exception of the eigenvectors $U$ and eigenvalues $\Lambda$. The number of operations required to compute those depends not only on the dimensions of the matrix $C_{nn}^{-1/2}\hat{C}_{yy}\left(C_{nn}^{-1/2}\right)^T$ to be diagonalized but also on its eigenvalue structure. For the simulated AIRS data used in this work, the operation count for computing the eigenvectors and eigenvalues using MATLAB was estimated at $8 \times 10^{10}$ floating-point operations (FLO) using the MATLAB flops function. For long data blocks ($K$ equal to several times $N$) the dominant computation is that of the covariance matrix $\hat{C}_{yy}$.

Table 5.6 lists the operation count for the application of the codebook to a single $N$-channel spectrum. When applying the codebook the dominant computation depends on whether or not the noise covariance $C_{nn}$ is diagonal or sparse. If it is not, the dominant computations are the multiplication by $C_{nn}^{-1/2}$ and $C_{nn}^{1/2}$ to noise-adjust and de-adjust the data, respectively. Otherwise, the dominant computations are the forward and inverse transformations with the matrices $U$ and $\hat{U}$.

From the tables, the total number of operations required to apply the algorithm (both designing and applying the codebook) to a block of $K$ spectra is approximately

$$FLO_{\text{total}} \approx 6N^2K + 4N^3 + (4N + 150)mK + 3NK + FLO(U, \Lambda)$$  \hspace{1cm} (5.96)

172
<table>
<thead>
<tr>
<th>Step</th>
<th>Mult.</th>
<th>Add.</th>
<th>Total</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y(t) - \hat{m}_y$</td>
<td>0</td>
<td>$N$</td>
<td>$N$</td>
<td></td>
</tr>
<tr>
<td>$C_{nn}^{-1/2} (y(t) - \hat{m}_y)$</td>
<td>$N^2$</td>
<td>$N^2$</td>
<td>$2N^2$</td>
<td>Assuming $C_{nn}^{-1/2}$ is precomputed</td>
</tr>
<tr>
<td></td>
<td>$N$</td>
<td>0</td>
<td>$N$</td>
<td>Assuming $C_{nn}$ is diagonal</td>
</tr>
<tr>
<td>$\mathbf{U}^T C_{nn}^{-1/2} (y(t) - \hat{m}_y)$</td>
<td>$mN$</td>
<td>$mN$</td>
<td>$2mN$</td>
<td></td>
</tr>
<tr>
<td>$v(t) = \mathbf{L} \mathbf{U}^T C_{nn}^{-1/2} (y(t) - \hat{m}_y)$</td>
<td>$m$</td>
<td>0</td>
<td>$m$</td>
<td>Estimated using MATLAB flops</td>
</tr>
<tr>
<td>$h(v(t))$</td>
<td>-</td>
<td>-</td>
<td>50$m$</td>
<td></td>
</tr>
<tr>
<td>$q(t) = g(h(v(t)))$</td>
<td>-</td>
<td>-</td>
<td>100$m$</td>
<td>Estimated using MATLAB flops</td>
</tr>
<tr>
<td>$Uq(t)$</td>
<td>$Nm$</td>
<td>$Nm$</td>
<td>$2Nm$</td>
<td></td>
</tr>
<tr>
<td>$C_{nn}^{1/2} Uq(t)$</td>
<td>$N^2$</td>
<td>$N^2$</td>
<td>$2N^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$N$</td>
<td>0</td>
<td>$N$</td>
<td>Assuming $C_{nn}$ is diagonal</td>
</tr>
<tr>
<td>$C_{nn}^{1/2} Uq(t) + \hat{m}_y$</td>
<td>0</td>
<td>$N$</td>
<td>$N$</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6: Approximate operation count for the codebook application portion of the algorithm. $N$ is the number of channels, and $m$ is the number of NAPC with non-zero bit allocations.

when the noise covariance is not diagonal. Here FLO($\mathbf{U}, \Lambda$) denotes the number of operations required to compute the eigenvectors and eigenvalues. If the noise covariance is diagonal, the total number of operations becomes

$$\text{FLO}_{\text{total}} \approx 2N^2K + 2N^2 + (4N + 150)mK + 5NK + \text{FLO}(\mathbf{U}, \Lambda).$$

In either case, for long blocks ($K > N$) the dominant term should be the one proportional to $N^2K$.

Tables 5.7 and 5.8 list the computational burden of the coding algorithm when it is applied to AIRS data. For the results in Table 5.7 it was assumed that the noise covariance matrix is a full matrix, and for the results in Table 5.8 that matrix was assumed to be diagonal. For both sets of results we used $N = 2378$, FLO($\mathbf{U}, \Lambda$) = $8 \times 10^{10}$, and an average sampling rate of 34 spectra/sec. It was also assumed that all the required computations for a given data block would be executed by a dedicated processor while the next data block is being collected. Note that in all cases we included the operations required to both encode and decode the signal estimates in Figure 5-1 in real time. Decoding the signal
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
$K(\times90)$ & $m$ & GFLOPS \\
\hline
112 & 100 & 1.63 \\
\hline
112 & 200 & 1.66 \\
\hline
225 & 100 & 1.40 \\
\hline
225 & 200 & 1.43 \\
\hline
\end{tabular}
\end{center}

Table 5.7: Approximate computational burden of the coding algorithm when it is applied to AIRS data, assuming the noise covariance is full. $K$ is the number of spectra per block, and $m$ is the number of NAPC with non-zero bit allocations. GFLOPS stands for Giga floating-point operations per second.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
$K(\times90)$ & $m$ & GFLOPS \\
\hline
112 & 100 & 0.68 \\
\hline
112 & 200 & 0.72 \\
\hline
225 & 100 & 0.55 \\
\hline
225 & 200 & 0.58 \\
\hline
\end{tabular}
\end{center}

Table 5.8: Approximate computational burden of the coding algorithm when it is applied to AIRS data, assuming the noise covariance is diagonal. $K$ is the number of spectra per block, and $m$ is the number of NAPC with non-zero bit allocations. GFLOPS stands for Giga floating-point operations per second.

estimates in real time is necessary in order to compute the noise estimates and code them as in Section 5.6. The computational cost of coding the noise estimates was not included but it is negligible compared to that of coding the signal estimates.

It can be seen from the tables that the computational cost of the coding algorithm is very high in the case of AIRS. However, when the noise covariance matrix is diagonal or nearly so, about sixty percent of the computational cost is due to computing the covariance matrix of the input data. That can be reduced by computing the covariance matrix using only a subset of the data in each data block. For example, for $K = 90 \times 225$ and $m = 100$, if only a quarter of each data block is used to compute the covariance matrix, the computational load is reduced from 0.55 GFLOPS to 0.26 GFLOPS. Even greater computational savings could be obtained by dividing each spectrum into several blocks of channels and applying the coding algorithm to each of them separately. The loss in compression performance of these alternatives could be the subject of a future study.
5.8 Summary and Additional Comments

In this chapter we developed a block-adaptive transform coding algorithm for noisy multispectral data, and tested it on simulated AIRS data. The algorithm was designed with two main objectives in mind: 1) to obtain efficient near-lossless compression of the useful content in the data, and 2) to be robust with respect to changes in statistics and outliers. Both of these considerations are important in remote sensing applications because of the high cost of generating high-quality data.

The first objective was addressed by using a codebook with a particular estimator-coder structure. The calibrated spectrum is modeled as the linear superposition of the signal (the noiseless, perfectly calibrated spectrum) and independent Gaussian noise. The codebook transforms each calibrated spectrum into noise-adjusted principal components (NAPC) which compact the signal energy per unit noise in the data, filters out the noise, and quantizes the filtered NAPC using Lloyd-Max quantizers for Gaussian distributions. This structure affords efficient near-lossless compression of the useful data content by using the allocated bit rate to efficiently quantize the best linear estimate of the signal given the noisy data.

The second objective was addressed by making the algorithm adaptive to changes in signal statistics, and by preserving individual outlier channels. The algorithm adapts to the signal statistics of successive data blocks. For each block of noisy spectra to be coded, the codebook (filtering matrix, transform matrix, and quantizers) is designed using estimates of the second-order statistics of the signal and the noisy spectrum. Those estimates are derived from each data block, and require knowledge of the second-order statistics of the noise. In practice the latter statistics can be estimated from accumulated calibration data transmitted intact to the user. The length of the data blocks should be chosen small enough that the algorithm can track changes in the statistics, and consistent with the system memory and data latency requirements. In general, data with higher spatial resolution may require shorter blocks. However, in all cases it is recommended that the block length be chosen to be greater than the number of spectral channels; at least two or three times larger. If shorter blocks are desired, one could divide the spectrum into multiple segments and code those separately.

Our coding algorithm preserves outlier channels in each spectrum by estimating the noise
(noisy data minus signal estimate) in all channels and comparing the amplitudes of the noise estimates to their rms values. Noise estimates whose amplitude exceeds nominally three times their rms value are uniformly quantized to the nominal resolution of the original data and transmitted to the user. The user may later reconstruct (approximately) the outliers by combining the noise and signal estimates to investigate their origin.

Two novel components were developed to make the coding algorithm more amenable for adaptive use. For this purpose we developed a rule for setting the codebook overhead, and a bit allocation procedure which takes into account the codebook overhead. Together these two components allow the design of the codebook for a finite data block without exceeding a specified bit rate.

The algorithm was applied to simulated AIRS data to illustrate its performance. For those data it was found that for blocks sizes (number of spectra) on the order of two or three times the number of spectral channels it is possible to obtain near-lossless compression ratios of up to 32:1 (0.4 bits/pixel/channel), including the codebook overhead. Those compression ratios are near-lossless in the sense that the total distortion or, equivalently, the average noise variance reduction is practically the same as that obtained when the data are not coded. The noise reduction obtained at those high compression ratios was 12–14 dB (higher for longer data blocks). At lower bit rates the distortion was found to be higher than predicted due to the fact that the distributions of some of the NAPC coefficients depart from the assumed Gaussian shape; the quantizer performance is quite sensitive to that assumption. This was found to be more serious in simulations in which the additive noise was assumed to be correlated between channels. Better models of the distributions of the NAPC are needed to remedy this problem. Nonetheless, near-lossless compression ratios on the order of 32:1 compare favorably with a predicted maximum lossless compression ratio of 4.7:1 for the same data.

The computational cost of the coding algorithm was also evaluated. The operation count of the algorithm depends on whether or not the noise covariance matrix may be assumed to be diagonal. If the covariance matrix is diagonal, the operation count is roughly \((2N^2 + 4Nm + 150m)K\), where \(N\) is the number of channels, \(m\) is the number of NAPC with non-zero bit allocations, and \(K\) is the number of spectra per data block. The term \(2N^2K\) accounts for the computation of the covariance matrix of the input data. On the other hand, if the noise covariance matrix is full, the operation count is about \((6N^2 + 4Nm + 150m)K\).
Both estimates become increasingly accurate when $K$ is much larger than $N$. For example, for AIRS, with a data block of 225 scans-worth (10 minutes) of data and $m = 100$, the computational cost is about $0.55 \times 10^9$ floating-point operations per second (FLOPS) if the noise covariance is diagonal, and $1.4 \times 10^9$ FLOPS if the noise covariance is full. The computational cost of the coding algorithm could be reduced, for example, by using only a fraction of the input data to compute the covariance matrix, or by dividing each spectrum into several blocks of channels and applying the algorithm to each of them separately.

Finally, we should note that in order to apply our coding algorithm to real AIRS data additional provisions must be made if there are "blinker" detectors like the ones shown in Figures 4-3 and 4-4. The algorithm discussed here would not operate effectively on data with that kind of anomaly. If there are any channels which exhibit blinking behavior, they should be excluded from being coded by our algorithm. Those channels could be identified by inspecting the calibration data, which would be transmitted intact to earth.
Chapter 6

Compression Algorithm for Multispectral Data in Additive Gaussian and Impulsive Noise

6.1 Introduction

In this chapter we develop a compression algorithm for data corrupted by both additive Gaussian noise and impulsive noise. The Gaussian noise component models errors due to typical instrument noise and calibration errors. Impulsive noise may occur, for example, due to the radiation environment in orbit in the case of a spaceborne sensor. As in the rest of the thesis, an estimation-coding approach is followed. However, in contrast to the previous cases, the new algorithm is not claimed to be optimal in any particular way. The idea behind the algorithm is to first identify those channels in a given spectrum which likely have been corrupted by impulsive noise, to estimate the noiseless spectrum from the non-corrupted channels, and finally to code the estimate of the spectrum in the NAPC domain as before. Thus, the only difference compared to the case of data free of impulsive noise is in the way the signal component is estimated. Since the only new element in the coding algorithm is the signal estimator, the focus in this chapter is on developing the estimator and characterizing its performance.

Unfortunately, there is a non-trivial complication in the implementation of an adaptive algorithm. Due to the presence of impulsive noise it is not possible to consistently estimate
the statistics of the signal component from the noisy data as before. For simplicity we assume throughout this chapter that both the statistics of the signal and the Gaussian noise are known \textit{a priori}. Consequently the coding algorithm to be described is not adaptive.

The new signal estimator is derived using a mixture model for the total noise present in the data. Within that model, impulsive noise is obtained by adding with small probability a second noise component of higher variance than the "typical" noise. While the typical noise is assumed to be Gaussian, no particular distribution is assumed for the impulsive noise component. In fact, the proposed estimator does not depend on the distribution of the impulsive noise component. A mixture model was used by Efron \textit{et. al} to study the problem of \textit{signal detection} in a correlated Gaussian noise plus impulsive noise environment [51]. It is argued below that the mixture noise model is also useful in the \textit{signal estimation} problem because the minimum mean-square error (MMSE) estimator displays a structure which, although not directly practical in general, suggests a way of obtaining practical suboptimal estimators.

Another model of impulsive noise that has received much attention in the literature uses symmetric $\alpha$-stable distributions [52, 53]. These are generalizations of the Gaussian distribution that have tails which decay slower. The decay is controlled by a parameter $\alpha$ lying in the interval $(0, 2]$. For example, setting $\alpha = 2$ gives the Gaussian density, and $\alpha = 1$ corresponds to the Cauchy density. A smaller value of $\alpha$ accentuates the impulsive character of the corresponding random variable. While they are effective for modeling impulsive noise, symmetric $\alpha$-stable distributions are difficult to manipulate mathematically, and their moments of order $\alpha$ and higher are infinite. In particular, their covariances are infinite. Because of these difficulties we did not use symmetric $\alpha$-stable distributions to model impulsive noise.

The rest of the chapter is organized as follows. The mixture noise model is described in further detail in Section 6.2. Next, in Section 6.3 we study a decomposition of the MMSE signal estimator in terms of \textit{conditional} MMSE estimators given knowledge of which elements in the measured vector are corrupted by impulsive noise. That discussion serves to illustrate the practical difficulty of implementing the MMSE estimator, and provides motivation for our suboptimal estimator, which is developed in Section 6.4. An efficient implementation of the estimator is detailed there. In Section 6.5 the performance of the estimator is evaluated using simulated AIRS data. Finally, in Section 6.6 the complete
Figure 6-1: Model of the measurement incorporating a mixture of typical noise and impulsive noise. Impulsive noise \( u(t) \) is generated by forming the element-wise product of the random vectors \( r(t) \) and \( s(t) \). The latter is a binary-valued vector, with elements equal to zero or one, that selects for the corresponding instance of the signal vector \( x(t) \) which individual signal elements are corrupted by noise \( r(t) \) of higher variance than the typical noise \( n(t) \).

coding algorithm is described.

6.2 The Mixture Noise Model

The mixture noise model employed here is illustrated pictorially in Figure 6-1. The typical noise \( n(t) \) has non-zero elements with probability one, while the impulsive noise \( u(t) \) has elements which take non-zero values with small probability. The typical noise and the impulsive noise are assumed to be statistically independent. The impulsive noise is generated by forming the element-wise product of \( s(t) \) and \( r(t) \), where the former is a binary-valued random vector, with elements equal to zero or one, and the latter is a real-valued random vector of higher variance than the typical noise \( n(t) \). Thus, the \( i^{th} \) element of the signal vector \( x(t) \) is corrupted by impulsive noise when the corresponding element of \( s(t) \) equals one. Since \( s(t) \) acts as a switch we shall refer to it as the switch vector.

The noise model shown in Figure 6-1 is called a mixture because the marginal density of the total noise \( n_{\text{total}}(t) \triangleq n(t) + u(t) \) is given by the superposition of the conditional densities
given all possible values of the vector \( s(t) \) weighted by their corresponding probabilities,

\[
p_{n_{\text{total}}} (n_{\text{total}}(t)) = \sum_{s'} \Pr (s(t) = s') p_{n_{\text{total}}/s} (n_{\text{total}}(t)/s'), \quad (6.1)
\]

where \( \Pr(\cdot) \) denotes probability, \( p_{n_{\text{total}}/s} (\cdot/s') \) is the conditional density of \( n_{\text{total}}(t) \) given \( s(t) = s' \), and \( p_{n_{\text{total}}/s} (\cdot) \) is the marginal density of \( n_{\text{total}}(t) \). So with probability \( \Pr (s(t) = s') \) the total noise \( n_{\text{total}}(t) \) is generated from the probability density \( p_{n_{\text{total}}/s} (\cdot/s') \). When the impulsive noise \( u(t) \) and the typical noise \( n(t) \) are statistically independent it can be shown that

\[
p_{n_{\text{total}}/s} (n_{\text{total}}(t)/s') = p_{u/s} (n_{\text{total}}(t)/s') * p_n (n_{\text{total}}(t)), \quad (6.2)
\]

where \( p_{u/s} (\cdot/s') \) is the conditional density of \( u(t) \) given \( s(t) = s' \), \( p_n (\cdot) \) is the marginal density of \( n(t) \), and \( * \) denotes the convolution operation. It is possible to express \( p_{u/s} (\cdot/s') \) in terms of \( s(t) \) and the marginal density of \( r(t) \) but the expression is complicated and we shall have no use for it.

In what follows we shall characterize the typical noise \( n(t) \) by a known covariance matrix, and a mean value of zero. No particular functional form for the probability density of the impulsive noise shall be assumed. In fact, the estimation method to be developed in Section 6.4 will not make use of the statistics of the impulsive noise.

### 6.3 The Minimum Mean-Square Error Estimator

Here we shall study the MMSE estimator of the random signal \( x(t) \) given the noisy measurement \( z(t) \) generated by the mixture noise model described above. The discussion in this section is intended to illustrate both the complexity of computing the MMSE estimator, and the motivation behind the suboptimal estimator developed in Section 6.4. First we derive a general expression for the MMSE estimator in terms of conditional MMSE estimators for all possible values of the switch vector \( s(t) \). For the purpose of illustration, the resulting expression is specialized to the case of a Gaussian signal, Gaussian typical noise, and Gaussian impulsive noise. We conclude by discussing the complexity of computing of the MMSE estimate via its decomposition into conditional estimators. Although not practical by itself, this decomposition of the MMSE estimator suggests ways of implementing reasonable suboptimal estimators.
It is well known that the estimator \( \hat{x}(z(t)) \) that minimizes the mean-square estimation error

\[
E\left\{ (x(t) - \hat{x}(z(t)))^T (x(t) - \hat{x}(z(t))) \right\}
\]

is given by \( \hat{x}(z(t)) = E\{x(t)/z(t)\} \), the conditional mean of \( x(t) \) given \( z(t) \) [29]. An expression for \( E\{x(t)/z(t)\} \) in terms of optimal conditional estimators can be obtained for any mutually exclusive, collectively exhaustive collection of events \( \{A_i\}_{i=1}^{M} \) defined in the joint space of \( x(t), s(t), n(t), \) and \( r(t) \). By mutually exclusive we mean that

\[
Pr\left( A_{i(1)} \cap A_{i(2)} \cap \cdots \cap A_{i(m)} \right) = 0
\]

for any sub-collection of such events, where \( A_i \cap A_j \) denotes the intersection of events \( A_i \) and \( A_j \). By collectively exhaustive we mean that

\[
Pr\left( A_1 \cup A_2 \cup \cdots \cup A_M \right) = 1,
\]

where \( A_i \cup A_j \) denotes the union of events \( A_i \) and \( A_j \). For such a collection of events one can write

\[
E\{x(t)/z(t)\} = \sum_{i=1}^{M} Pr\{A_i/z(t)\} E\{x(t)/z(t), A_i\},
\]

where \( E\{x(t)/z(t), A_i\} \) is the conditional mean of \( x(t) \) given \( z(t) \) and the event \( A_i \). For the mixture noise model described in Section 6.2 a convenient collection of events is given by the set \( S \) of all possible values of the \( N \)-dimensional switch vector \( s(t) \). The collection of events \( \{s(t) = s'\}_{s' \in S} \) is clearly mutually exclusive and collectively exhaustive, so one can write

\[
E\{x(t)/z(t)\} = \sum_{s' \in S} Pr\{s(t) = s'/z(t)\} E\{x(t)/z(t), s(t) = s'\},
\]

\[
= \frac{\sum_{s' \in S} Pr\{s(t) = s'/z(t)\} p_{z/s} (z(t)/s') E\{x(t)/z(t), s(t) = s'\}}{\sum_{s' \in S} Pr\{s(t) = s'/z(t)\} p_{z/s} (z(t)/s')},
\]

where the last equality is a direct application of Bayes' rule. Thus, the MMSE signal estimator can be obtained by computing the conditional MMSE estimators for all possible combinations of elements of the measurement vector that could have been corrupted by impulsive noise, and adding together all the estimates weighted by the \textit{a posteriori} probability
that the corresponding combination of data elements was in fact corrupted by impulsive noise.

As an example, let us specialize the expression in equation 6.8 to the case of a Gaussian signal, zero-mean Gaussian typical noise, and zero-mean Gaussian impulsive noise. We start by writing \( z(t) \) as

\[
z(t) = x(t) + n(t) + s(t) \odot r(t),
\]

(6.9)

where \( \odot \) denotes the element-wise product of the operands. Note that when \( s(t) \) is given, the vectors \( x(t) \) and \( z(t) \) are jointly Gaussian since \( x(t) \), \( n(t) \) and \( r(t) \) are jointly Gaussian by assumption. So the conditional estimator \( E\{x(t)/z(t), s(t)\} \) is the linear MMSE estimator of \( x(t) \) given \( z(t) \) and \( s(t) \);

\[
E\{x(t)/z(t), s(t)\} = C_{xx}C_{zz/s(t)}^{-1}(z(t) - m_x) + m_x,
\]

(6.10)

where the auto-covariance of \( z(t) \) given \( s(t) \) is given by

\[
C_{zz/s(t)} = C_{xx} + C_{nn} + (s(t)s(t)^T) \odot C_{rr},
\]

(6.11)

where \( C_{xx} \), \( C_{nn} \), and \( C_{rr} \) are the covariances of \( x(t) \), \( n(t) \), and \( r(t) \), respectively. The conditional density of \( z(t) \) given \( s(t) \) is the Gaussian density with mean \( m_x \) and covariance \( C_{zz/s(t)} \);

\[
p_{z/s}(z(t)/s(t)) = (2\pi)^{-N/2} \left| C_{zz/s(t)} \right|^{-1/2} e^{-1/2(z(t) - m_x)^T C_{zz/s(t)}^{-1}(z(t) - m_x)}. \]

(6.12)

Equations 6.10, 6.11 and 6.12 together with the probability distribution of \( s(t) \) completely specify the MMSE estimator in this case.

Computing the MMSE estimator using equation 6.8 is probably not practical in most cases. Even if each conditional estimator and the corresponding a posteriori probability could be readily computed, the number of terms in the sum is \( 2^N \) for an \( N \)-dimensional measurement vector \( z(t) \). The fact that each conditional estimator is weighted by an associated a posteriori probability suggests the possibility of computing only those conditional estimators for which the corresponding probability is high enough. Unfortunately, that simplification may not be practical either since the a posteriori probabilities \( \Pr(s(t) = s') \) generally would have to be computed using Bayes' rule as in equation 6.8, which implies
that \( p_{z/s}(z(t)/s') \) would have to be evaluated for all \( 2^N \) values of the switch vector \( s(t) \). Another difficulty is that the probability distribution of the impulsive noise must be specified in order to compute the MMSE estimator. Even when the functional form of the distribution is assumed, it is likely that some of the parameters of the distribution will need to be estimated. Obtaining accurate parameter estimates may be difficult in practice since the impulsive noise would generally occur relatively infrequently.

In spite of its practical limitations, the structure of equation 6.8 provides motivation for pursuing suboptimal estimators involving a detection step and an estimation step. The objective of the detection step would be to identify which elements of the measurement are corrupted by impulsive noise. The estimation step would then estimate the signal taking into account the result of the detection step. Simplified implementations of the detection and estimation steps can be made in order to obtain a practical estimator. In the next section we develop one such two-step estimator.

### 6.4 Development of a Two-step Suboptimal Estimator

Here we develop a practical alternative to the MMSE estimator given by equation 6.7. Our suboptimal estimator involves two basic steps:

1. detecting which elements of the measurement vector \( z(t) \) are likely corrupted by impulsive noise, and

2. estimating the signal \( x(t) \) using only those elements of the measurement vector which, according to the previous step, are not corrupted by impulsive noise.

The detection-estimation structure of our estimator is suggested by the structure of the MMSE estimator displayed in equation 6.7, which results from the use of the mixture noise model described in Section 6.2. The detection step is implemented using an iterative procedure in which each scalar element is predicted from the other elements of the measurement vector. A thresholding operation is performed on the prediction errors to decide if there are any corrupted elements. The signal is estimated in a linear MMSE sense, assuming all the elements corrupted by impulsive noise were correctly identified in the detection step.

Efron et. al previously proposed the idea of using a two-step procedure like the one outlined above for signal detection in a correlated Gaussian noise plus impulsive noise en-
vironment [51]. In particular, they proposed first detecting samples corrupted by impulsive noise and deleting them from the observed sequence, and then performing the signal detection using the remaining samples. However, Efron et. al did not detail a specific method for detecting impulsive noise. Instead their work focused on the structure of the so-called “clairvoyant” matched filter for deterministic signals and the “clairvoyant” power detector for Gaussian signals, both of which can only be implemented when the location of the data samples corrupted by impulsive noise are known. They used these ideal systems to show that in many practical cases one is justified in deleting those samples corrupted by impulsive noise prior to detecting the signal, assuming the corrupted samples can be identified. They also showed that the same result could be obtained by replacing the corrupted samples with their linear MMSE estimates obtained from the non-corrupted data samples. The idea behind the two-step approach of Efron et. al was used to develop our signal estimator.

Before describing our estimator let us introduce our notation for denoting sub-vectors and sub-matrices, and derive two useful matrix equalities that we will employ in the subsequent development. Let us start with the notation. For a given positive integer $N$ let $A$ be a subset of the set of integers from one to $N$, and let $A(i)$ denote the $i^{th}$ element of $A$ when its elements are arranged in order of increasing value. It is common use to denote the $i^{th}$ element of an $N$-dimensional vector $v$ as $v_i$. We would like to extend that notation so that $v_A$ denotes the sub-vector of $v$ indexed by the elements of the set $A$, i.e.

$$v_A \triangleq [v_{A(1)} \ v_{A(2)} \ \cdots \ v_{A(|A|)}]^T,$$  

where $|A|$ denotes the number of elements in $A$. The same notation may be applied to an arbitrary $N \times M$ matrix $C$. Again, for such a matrix it is common to denote the element in the $i^{th}$ row and the $j^{th}$ column as $C_{i,j}$. Let $A$ be defined as above, and $B$ be a subset of the set of integers from one to $M$. The sub-matrix $C_{A,B}$ is defined as

$$C_{A,B} \triangleq \begin{bmatrix} C_{A(1),B(1)} & C_{A(1),B(2)} & \cdots & C_{A(1),B(|B|)} \\ C_{A(2),B(1)} & C_{A(2),B(2)} & \cdots & C_{A(2),B(|B|)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{A(|A|),B(1)} & C_{A(|A|),B(2)} & \cdots & C_{A(|A|),B(|B|)} \end{bmatrix}.$$  

(6.14)
Finally, for convenience we shall use the symbol \( \ast \) to denote the set of all indexes along either direction. For example, \( C_{\ast, i} \) would be the \( i \)th column of \( C \).

Now suppose \( C \) is an \( N \times N \) invertible matrix, and let \( A \) and \( B \) be mutually exclusive sets, and let their union equal the set of integers from one to \( N \). In the development of our algorithm we shall find useful two matrix equalities relating the sub-matrices of a matrix \( C \) and its inverse obtained by partitioning both matrices according to a pair of sets \( A \) and \( B \) such as the one described above. Specifically, when \( C \) is invertible we can write

\[
\begin{bmatrix}
C_{A,A} & C_{A,B} \\
C_{B,A} & C_{B,B}
\end{bmatrix}
\begin{bmatrix}
(C^{-1})_{A,A} & (C^{-1})_{A,B} \\
(C^{-1})_{B,A} & (C^{-1})_{B,B}
\end{bmatrix}
= \begin{bmatrix}
I & 0 \\
0 & I
\end{bmatrix}.
\]

(6.15)

There are four equations above, each corresponding to one of the four sub-matrices of the identity matrix on the right-hand side. The equations corresponding to the top sub-matrices are, after multiplying the terms on the left-hand side,

\[
C_{A,A} (C^{-1})_{A,A} + C_{A,B} (C^{-1})_{B,A} = I, \quad (6.16)
\]

\[
C_{A,A} (C^{-1})_{A,B} + C_{A,B} (C^{-1})_{B,B} = 0. \quad (6.17)
\]

From the last equation one can obtain

\[
C_{A,B}^{-1} C_{A,B} = -(C^{-1})_{A,B} (C^{-1})_{B,B}^{-1}. \quad (6.18)
\]

In our notation the indexing operation takes precedence over inversion so that, for example, \( C_{A,A}^{-1} \) is the inverse of \( C_{A,A} \). Finally, premultiplying equation 6.16 by \( C_{A,A}^{-1} \) and substituting equation 6.18 gives

\[
C_{A,A}^{-1} = (C^{-1})_{A,A} - (C^{-1})_{A,B} (C^{-1})_{B,B}^{-1} (C^{-1})_{B,A}. \quad (6.19)
\]

Equations 6.18 and 6.19 will be useful in the sequel.

### 6.4.1 The Detection Step

Our algorithm for detecting impulsive noise is based on the idea of removing from each element of the measurement vector \( z(t) \) both the signal and the typical noise components so as to expose the impulsive noise component. The signal and typical noise components on
any given element of \( z(t) \) are removed by subtracting their best linear prediction based on all of the other elements, and the predictions are computed assuming the impulsive noise vector \( u(t) \) is the zero vector. The hypothesis is then tested by comparing the magnitudes of the prediction error for all the elements of \( z(t) \) with their rms values when the impulsive noise is indeed zero. If the magnitude of any of the prediction errors is greater that \( \eta \) times its rms value, for a suitable threshold \( \eta \), it is concluded that \( u(t) \) is not zero. Then that element of \( z(t) \) having the prediction error with the largest magnitude relative to its rms value is labeled as “corrupted.” The process is repeated on the sub-vector of \( z_{\bar{C}}(t) \) which excludes the “corrupted” element, i.e. the prediction errors are computed for \( z_{\bar{C}}(t) \) and the hypothesis that \( u_{\bar{C}}(t) \) equals zero is tested. Whenever the hypothesis is rejected another element is labeled as “corrupted,” and the process is repeated on the sub-vector which excludes all the “corrupted” elements. The detection algorithm stops when the magnitudes of all the prediction errors for the current sub-vector \( z_{\bar{C}}(t) \) are less than the desired threshold. The details of the algorithm are derived below, and an efficient implementation is given. A qualitative discussion of the algorithm is given in Section 6.4.2.

Let us start by deriving an expression for the prediction errors. Let \( \bar{C} \) be a non-empty subset of the set of integers from one to \( N \). Under the assumption that \( u_{\bar{C}}(t) \) is the zero vector, \( z_{\bar{C}}(t) = y_{\bar{C}}(t) \). Accordingly, for each index \( \bar{C}(i) \in \bar{C} \) we compute the linear MMSE prediction of \( y_{\bar{C}(i)}(t) \) given \( y_{\bar{C}(i)}(t) = z_{\bar{C}(i)}(t) \), where the set \( \bar{C} - \bar{C}(i) \) includes all elements of \( \bar{C} \) except \( \bar{C}(i) \). The linear MMSE prediction of \( y_{\bar{C}(i)}(t) \) given \( y_{\bar{C}(i)}(t) = z_{\bar{C}(i)}(t) \), as obtained by solving the Wiener-Hopf equation, is given by

\[
\hat{y}_{\bar{C}(i)} \left( z_{\bar{C}(i)}(t) \right) = \\
(m_y)_{\bar{C}(i)} + (C_{yy})_{\bar{C}(i),\bar{C}(i)} \left( C_{yy}^{-1} \right)_{\bar{C}(i),\bar{C}(i)} \left( z_{\bar{C}(i)}(t) - (m_y)_{\bar{C}(i)} \right), \quad (6.20)
\]

where \( m_y \) and \( C_{yy} \) are the mean and covariance of \( y(t) \). Substituting the transpose of equation 6.18 with \( C = (C_{yy})_{\bar{C},\bar{C}}, \ A = \{1, 2, \ldots, i - 1, i + 1, \ldots, |\bar{C}|\}, \) and \( B = i \), into equation 6.20 gives

\[
\hat{y}_{\bar{C}(i)} \left( z_{\bar{C}(i)}(t) \right) = \\
(m_y)_{\bar{C}(i)} - \left( (C_{yy})_{\bar{C},i}^{-1} \right)_{i,i} \left( (C_{yy})_{\bar{C},i}^{-1} \right)_{i,A} \left( z_{\bar{C}(i)}(t) - (m_y)_{\bar{C}(i)} \right). \quad (6.21)
\]

188
The last expression above, although somewhat complicated, leads to a convenient expression for the prediction error

$$e_i (z(t), \bar{C}) \triangleq z_{\bar{C}(i)}(t) - \hat{y}_{\bar{C}(i)} \left( z_{\bar{C}C(i)}(t) \right), \quad (6.22)$$

namely,

$$e_i (z(t), \bar{C}) = \left( ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} \right)_{i,i} \left( ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} \right)_{i,*} (z_{\bar{C}}(t) - (m_{\bar{y}})_{\bar{C}}), \quad (6.23)$$

where $\left( ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} \right)_{i,*}$ denotes the $i^{th}$ row of $((C_{y\bar{y}})^{-1}_{\bar{C},\bar{C}})$. Thus, the vector of prediction errors for $z_{\bar{C}}(t)$ is computed by pre-multiplying $z_{\bar{C}}(t) - (m_{\bar{y}})_{\bar{C}}$ by the inverse of the covariance matrix $((C_{y\bar{y}})^{-1}_{\bar{C},\bar{C}})$, and dividing the $i^{th}$ element of the resulting vector by the $i^{th}$ diagonal element of the inverse covariance matrix.

Next let us derive the second-order statistics of the prediction errors $e (z(t), \bar{C})$ conditioned on $u_{\bar{C}}(t)$. Using equation 6.23 and $z(t) = u(t) + y(z)$ write the vector of prediction errors as

$$e (z(t), \bar{C}) = S ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} u_{\bar{C}}(t) + S ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} (y_{\bar{C}}(t) - (m_{\bar{y}})_{\bar{C}}), \quad (6.24)$$

where $S$ is the diagonal matrix given by

$$S = \text{diag} \left( \left( ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} \right)_{1,1}, \ldots, \left( ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} \right)_{|\bar{C}|,|\bar{C}|} \right). \quad (6.25)$$

Since $y_{\bar{C}}(t)$ is independent of $u_{\bar{C}}(t)$ by assumption, the mean and covariance conditioned on $u_{\bar{C}}(t)$ of the prediction errors for $z_{\bar{C}}(t)$ are given by

$$E \{ e (z(t), \bar{C}) / u_{\bar{C}}(t) \} = S ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} u_{\bar{C}}(t), \quad (6.26)$$

$$\text{cov} \{ e (z(t), \bar{C}) / u_{\bar{C}}(t) \} = S ((C_{y\bar{y}})_{\bar{C},\bar{C}})^{-1} S, \quad (6.27)$$

From these expressions one can see that the conditional mean-square value of each prediction error, which equals the sum of the squared conditional mean and the conditional variance, is an increasing quadratic function of $u_{\bar{C}}(t)$. When the impulsive noise vector equals the zero vector, the conditional mean-square value is simply the conditional variance, which we denote as

$$\sigma^2_t (\bar{C}) \triangleq \text{var} \{ e_i (z(t), \bar{C}) / u_{\bar{C}}(t) \}, \quad (6.28)$$
= \left( (C_{yy})^{-1}_{\tilde{C},\tilde{C}} \right)^{-1}_{i,i}.

(6.29)

Now let us detail an efficient implementation of our detection algorithm. To facilitate
the discussion let \( k = 1, 2, \ldots \) be the iteration index, let \( C^{(k)} \) be the set of indexes of
the elements of \( z(t) \) which have been labeled as "corrupted" prior to the \( k^{th} \) iteration,
and let \( \tilde{C}^{(k)} \) be the complement of \( C^{(k)} \). The dominant computation of the algorithm is
that associated with the prediction errors given by equation 6.23, which are recomputed
every time an impulsive noise burst is detected in one of the elements of the measurement
vector. To facilitate the implementation it is desirable to redefine the vector of prediction
errors so that its size does not change from one iteration to the next. Accordingly, redefine
\( e \left( z(t), \tilde{C}^{(k)} \right) \) as

\[
e_i \left( z(t), \tilde{C}^{(k)} \right) \triangleq \begin{cases} 
0, & i \in C^{(k)}, \\
\left( (C_{yy})^{-1}_{\tilde{C}(k),\tilde{C}(k)} \right)^{-1}_{j,j} \left( (C_{yy})^{-1}_{\tilde{C}(k),\tilde{C}(k)} \right)^{-1}_{j,*} (z_{\tilde{C}(k)}(t) - (m_y)_{\tilde{C}(k)}), & i = \tilde{C}^{(k)}(j),
\end{cases}
\]

(6.30)

for \( 1 \leq i \leq N \). In this definition the prediction errors for those elements which are already
labeled as "corrupted" are set to zero, whereas before those errors simply were not included
in the vector. If one were to compute the prediction errors by computing the inverse
of \( (C_{yy})_{\tilde{C}(k),\tilde{C}(k)} \) at each iteration, about \( \left| \tilde{C}^{(k)} \right|^3 \) multiplications and an equal number of
additions would be required. Since the size of \( \tilde{C}^{(k)} \) equals \( N - k + 1 \), the computational
load per iteration would be on the order of \( N^3 \) for large \( N \).

A more efficient way of computing the prediction errors can be obtained by using the
matrix equality in equation 6.17. With \( C = C_{yy}, A = \tilde{C}^{(k)}, \) and \( B = C^{(k)} \) one can write

\[
(C_{yy})^{-1}_{\tilde{C}(k),\tilde{C}(k)} = (C_{yy}^{-1})_{\tilde{C}(k),\tilde{C}(k)} - (C_{yy}^{-1})_{C^{(k)},\tilde{C}(k)} \left( (C_{yy})_{\tilde{C}(k),\tilde{C}(k)}^{-1} \right) \left( (C_{yy})_{C^{(k)},C^{(k)}}^{-1} \right) \left( (C_{yy})_{C^{(k)},\tilde{C}(k)}^{-1} \right).
\]

(6.31)

Note that this expression only involves sub-matrices of the inverse of \( C_{yy} \), which has to be
computed only once instead of once per iteration. Although the sub-matrix \( (C_{yy}^{-1})_{C^{(k)},\tilde{C}(k)} \)
has to be inverted, that sub-matrix is much smaller than \( (C_{yy})_{\tilde{C}(k),\tilde{C}(k)} \) when only a few
elements of the measurement vector are corrupted by impulsive noise. Combining equa-
tions 6.30 and 6.31 one can compute the vector of prediction errors as

\[
e_i (z(t), \tilde{C}^{(k)}) = \begin{cases} 
0, & i \in C^{(k)}, \\
((C_{yy})^{-1}_{\tilde{C}^{(k)}, \tilde{C}^{(k)}})_{j,i} \left( \tilde{e}_i^{(k)} - e_i^{(k)} \right), & i = \tilde{C}^{(k)}(j),
\end{cases} 
\] (6.32)

for \(1 \leq i \leq N\), where \(\tilde{e}^{(k)}\) and \(\tilde{e}^{(k)}\) are given by

\[
\tilde{e}^{(k)} = (C_{yy})^{-1}_{z, \tilde{C}^{(k)}} \left( z_{\tilde{C}^{(k)}}(t) - (m_x)_{\tilde{C}^{(k)}} \right), \quad (6.33)
\]

\[
\tilde{e}^{(k)} = (C_{yy})^{-1}_{*, \tilde{C}^{(k)}} \left( C_{yy}^{-1}_{\tilde{C}^{(k)}, \tilde{C}^{(k)}} \tilde{e}^{(k)} \right), \quad (6.34)
\]

Furthermore, it is easy to show that \(\tilde{e}^{(k)}\) can be computed recursively as

\[
\tilde{e}^{(k+1)} = \tilde{e}^{(k)} - (C_{yy})^{-1}_{*i^{(k)}(t)} \left( z_{i^{(k)}}(t) - (m_x)_{i^{(k)}} \right), \quad k \geq 1, \quad (6.35)
\]

where \(i^{(k)}\) is the index of the "corrupted" element detected in the \(k^{th}\) iteration. The diagonal entries of \((C_{yy})^{-1}_{\tilde{C}^{(k)}, \tilde{C}^{(k)}}\), which are needed in equation 6.32, can be computed using equation 6.31.

Equations 6.31–6.35 completely specify the computation of the prediction errors. Table 6.1 lists the number of operations associated with each equation. If \(K\) iterations are required for a given measurement vector \(z(t)\), the total number of operations is approximately

\[
\text{Total No. of Operations} = 2N(N + K) + 2N(K - 1) \left( \frac{1}{3} K^2 + \frac{5}{6} K + 1 \right) + (K - 1)K - \frac{1}{6} (K - 1)^2 K^2. \quad (6.36)
\]

When \(N \gg K\), the total number of operations is about \(2N^2 + \frac{2}{3} N(K - 1)K^2\). In contrast, if one used equation 6.33 directly, the number of operations for \(K\) iterations would be about \(2N^3 K\).

To conclude let us give a complete listing of the algorithm. Let \(z(t)\) be the current \(N\)-dimensional measurement vector. Let \(k\) be the iteration index, which initially is set to one. Also let \(\tilde{C}^{(1)}\) be the set of all integers from one to \(N\), and let \(C^{(1)}\) be the empty set.

At the \(k^{th}\) iteration do the following:

1. If \(k = 1\), compute \(\tilde{e}^{(k)}(t)\) using equation 6.33; otherwise compute \(\tilde{e}^{(k)}(t)\) using equa-
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Equation</th>
<th>Number of operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{e}^{(1)}$</td>
<td>6.33</td>
<td>$2N^2$</td>
</tr>
<tr>
<td>$\mathbf{e}^{(k)}$, $k &gt; 1$</td>
<td>6.35</td>
<td>$2N + 1$</td>
</tr>
<tr>
<td>$\mathbf{e}^{(k)}$, $k \geq 1$</td>
<td>6.34</td>
<td>$2N \left</td>
</tr>
<tr>
<td>$\left( (\mathbf{C}<em>{yy}^{-1} \mathbf{C}^{(k)})</em>{i,i} \right)$, $1 \leq i \leq \left</td>
<td>\mathbf{C}^{(k)} \right</td>
<td>$</td>
</tr>
<tr>
<td>$\mathbf{e}^{(k)}$, $k \geq 1$</td>
<td>6.32</td>
<td>$2 \left</td>
</tr>
</tbody>
</table>

Table 6.1: Number of operations (multiplications and additions) per equation for the detection algorithm.

2. Compute $\mathbf{e}^{(k)}(t)$ using equation 6.34.

3. Compute $\left( (\mathbf{C}_{yy}^{-1} \mathbf{C}^{(k)})_{i,i} \right)$, $1 \leq i \leq \left| \mathbf{C}^{(k)} \right|$, using equation 6.31.

4. Compute the prediction error vector $\mathbf{e} \left( \mathbf{z}(t), \bar{\mathbf{C}}^{(k)} \right)$ using equation 6.32.

5. For $1 \leq i \leq N$ put

$$
\sigma_i^{(k)} = \begin{cases} 
1, & i \in \mathbf{C}^{(k)}, \\
\left( (\mathbf{C}_{yy}^{-1} \mathbf{C}^{(k)})_{i,j}^{-1/2} \right), & i = \mathbf{C}^{(k)}(j).
\end{cases} \tag{6.37}
$$

6. If $\left| \mathbf{e}_i \left( \mathbf{z}(t), \bar{\mathbf{C}}^{(k)} \right) \right| \leq \eta \sigma_i^{(k)}$ for $1 \leq i \leq N$, then stop; otherwise go to 7.

7. Find $i^{(k)} = \arg\max_{1 \leq i \leq N} \left| \mathbf{e}_i \left( \mathbf{z}(t), \bar{\mathbf{C}}^{(k)} \right) \right|$.

8. Put $\bar{\mathbf{C}}^{(k+1)} = \bar{\mathbf{C}}^{(k)} - i^{(k)}$ and $\mathbf{C}^{(k+1)} = \mathbf{C}^{(k)} \cup i^{(k)}$.

9. $k \leftarrow k + 1$. Go to 1.

6.4.2 Qualitative Discussion of the Detection Step

The objective of the following discussion is to develop some insight regarding the effectiveness of the detection algorithm described above. It is argued that the algorithm is most effective whenever the inverse of the covariance matrix of the signal-plus-typical-noise is nearly diagonal. Cases in which that condition is satisfied can be characterized using the noise-adjusted principal components (NAPC) representation of the signal-plus-typical noise.
Recall equation 6.24, which expresses the prediction error vector as a function of the impulsive noise vector and the signal-plus-typical-noise vector. It can be seen from that equation that the inverse of the covariance matrix \((C_{yy})^{-1}_{C,C}\) determines how the prediction errors are perturbed by impulsive noise. To illustrate the point more clearly, using equation 6.27 one can show that the correlation coefficient \(\rho_{i,j}\) conditioned on \(u_C(t)\) between the \(i^{th}\) and \(j^{th}\) prediction errors is given by

\[
\rho_{i,j}(C) = \sigma_i(C) \left((C_{yy})^{-1}_{C,C}\right)_{i,j} \sigma_j(C),
\]

(6.38)

where \(\sigma_i(C)\) is the standard-deviation of the \(i^{th}\) prediction error (see equation 6.29). One can write the \(i^{th}\) prediction error in terms of the correlation coefficients as

\[
e_i(z(t), \bar{C}) = \sigma_i(C) \sum_{j=1}^{|C|} \rho_{i,j}(C) u_{C(j)}(t) / \sigma_j(C) + \sigma_i(C) \sum_{j=1}^{|C|} \rho_{i,j}(C) \left(y_{C(j)}(t) - (m_y)_{C(j)}\right) / \sigma_j(C).
\]

(6.39)

At least two interesting observations can be made regarding the expression above. First note that the perturbation caused by an impulsive noise burst \(u_{C(j)}(t)\) on any of the prediction errors is inversely proportional to the typical rms value \(\sigma_j(C)\), i.e. in the absence of impulsive noise, of the prediction error \(e_j(z(t), \bar{C})\). Thus, an impulsive noise burst will be observed as long as it is relatively large compared to the typical rms value of the corresponding prediction error. Secondly, the perturbation caused by \(u_{C(j)}(t)\) on the \(i^{th}\) prediction error is proportional to the correlation coefficient between the \(i^{th}\) and \(j^{th}\) prediction errors. This implies that, in general, a single impulsive noise burst will perturb all the prediction errors. In order to effectively detect an impulsive noise burst, say in the \(i^{th}\) element of \(z_C(t)\), by testing the corresponding prediction error \(e_i(z(t), \bar{C})\), it is beneficial if the correlation coefficients of all the pairs of prediction errors have small magnitudes. Otherwise, a single noise burst would cause perturbations of similar magnitude in many of the prediction errors, making its detection more difficult.

From equation 6.38, the correlation coefficients between all pairs of prediction errors are zero if and only if \((C_{yy})^{-1}_{C,C}\) is diagonal. Aside from special cases which are not of practical interest, this can only happen if the elements of both the signal sub-vector \(x_C(t)\) and the
typical noise sub-vector \( n_C(t) \) are uncorrelated. This does not apply to most multi-channel signals encountered in practice, and in particular to remote sensing data. However, there are conditions which are often satisfied by practical data, and for which the aforementioned matrix is approximately diagonal.

It is possible to characterize practical cases in which \( (C_{yy})^{-1}_{\tilde{C},\tilde{C}} \) is nearly diagonal by expressing that matrix in terms of the NAPC representation of \( y_C(t) \). To simplify the notation let us treat the case where \( \tilde{C} \) contains all the indexes. Recall that \( y(t) = x(t) + n(t) \), where \( x(t) \) is the signal vector and \( n(t) \) is the typical noise vector. The signal and the noise are assumed to be uncorrelated. As discussed in Section 3.3.3, the NAPC representation decomposes \( y(t) \) into an uncorrelated sequence of random variables with maximal signal-to-noise ratio. Using the NAPC representation, the inverse covariance matrix of \( y(t) \) can be written as

\[
C^{-1}_{yy} = \left( C^{-1/2}_{nn} \right) P (\Gamma + I)^{-1} P^T C^{-1/2}_{nn},
\]

where \( \Gamma \) is the diagonal matrix of signal-to-noise ratios of the NAPC, \( P \) is an orthonormal matrix, and \( C^{-1/2}_{nn} \) is a (square) square-root of the noise covariance matrix \( C_{nn} \). If the signal \( x(t) \) is highly correlated so that very few NAPC have high signal-to-noise ratios and most of them have signal-to-noise ratios close to zero, as is the case for many practical signals, then

\[
C^{-1}_{yy} \approx \left( C^{-1/2}_{nn} \right) P \Gamma P^T C^{-1/2}_{nn},
\]

\[
= C^{-1}_{nn}.
\]

Thus, \( C^{-1}_{yy} \) is nearly diagonal if, in addition to the previous assumption about the signal, the noise is uncorrelated, or the noise energy is mostly evenly distributed in the \( N \)-dimensional space so that most of the noise eigenvalues have similar values. We expect our detection algorithm to be most effective when either of these two conditions is met.

6.4.3 The Estimation Step

The estimation step involves computing the linear MMSE estimate of the signal \( x(t) \) using only those elements of the measurement vector \( z(t) \) which are thought not to be corrupted by impulsive noise.

To derive the signal estimate we assume that all the elements of the measurement vector
corrupted by impulsive noise have been correctly identified. For the current measurement vector \( \mathbf{z}(t) \) let \( K \) be the number of iterations performed during the detection step, let \( C^{(K)} \) be the set of indexes of those elements that were labeled as corrupted, and let \( \tilde{C}^{(K)} \) be the complement of \( C^{(K)} \). If the impulsive noises were all detected correctly, then \( \mathbf{z}_{\tilde{C}^{(K)}}(t) = \mathbf{y}_{\tilde{C}^{(K)}}(t) \). Under that assumption, the linear MMSE estimate of \( \mathbf{x}(t) \) given \( \mathbf{z}_{\tilde{C}^{(K)}}(t) \) is given by

\[
\hat{\mathbf{x}}^{*}(\mathbf{z}_{\tilde{C}^{(K)}}(t)) = \Lambda (\mathbf{C}_{xx})_{\tilde{C}^{(K)}} ((\mathbf{C}_{yy})_{\tilde{C}^{(K)}})_{\tilde{C}^{(K)}} (\mathbf{z}_{\tilde{C}^{(K)}}(t) - (\mathbf{m}_y)_{\tilde{C}^{(K)}}) + \mathbf{m}_x,
\]

(6.43)

where, as before, \( \mathbf{m}_x \) and \( \mathbf{C}_{xx} \) are the mean and covariance of \( \mathbf{x}(t) \), and \( \mathbf{m}_y \) and \( \mathbf{C}_{yy} \) are the mean and covariance of \( \mathbf{y}(t) \).

It is possible to reduce the number of operations required to compute the signal estimate by incorporating the prediction errors computed during the detection step into equation 6.43. To effect this rewrite equation 6.43 as

\[
\hat{\mathbf{x}}^{*}(\mathbf{z}_{\tilde{C}^{(K)}}(t)) = \sum_{j=1}^{\lvert \tilde{C}^{(K)} \rvert} (\mathbf{C}_{xx})_{\tilde{C}^{(K)}/(j)} ((\mathbf{C}_{yy})_{\tilde{C}^{(K)},\tilde{C}^{(K)}})_{(j)}^{\scriptscriptstyle -1} (\mathbf{z}_{\tilde{C}^{(K)}}(t) - (\mathbf{m}_y)_{\tilde{C}^{(K)}}) + \mathbf{m}_x.
\]

(6.44)

Then substituting equations 6.32 and 6.37 with \( k = K \) in the expression above gives

\[
\hat{\mathbf{x}}^{*}(\mathbf{z}_{\tilde{C}^{(K)}}(t)) = \sum_{j=1}^{\lvert \tilde{C}^{(K)} \rvert} (\mathbf{C}_{xx})_{\tilde{C}^{(K)}/(j)} ((\mathbf{C}_{yy})_{\tilde{C}^{(K)}/(j)}^{\scriptscriptstyle -2}) (\mathbf{e}_{\tilde{C}^{(K)}/(j)})(\mathbf{z}(t),\tilde{C}^{(K)}) + \mathbf{m}_x.
\]

(6.45)

\[
\hat{\mathbf{x}}^{*}(\mathbf{z}_{\tilde{C}^{(K)}}(t)) = \mathbf{C}_{xx} \text{diag}\left(\sigma_{1}^{(K)} - 2, \ldots, \sigma_{N}^{(K)} - 2\right) \mathbf{e}(\mathbf{z}(t),\tilde{C}^{(K)}) + \mathbf{m}_x.
\]

(6.46)

The number of operations in the last equation is about \( 2N^2 \).

Deriving a general analytical expression for the error covariance matrix of the estimator described above is difficult because that matrix very much depends on the performance of the detection step. Deriving an analytical expression for the expected value of any distortion metric which is a quadratic function of the estimation error would be equally difficult. In practice the error covariance or the expected distortion could be evaluated by Monte Carlo simulations. Nonetheless, it is possible to derive useful formulas for the error covariance under additional assumptions. Below we shall derive some formulas which will be used in the evaluation of the estimator using AIRS data in Section 6.5.
First let us derive an expression for the error covariance *assuming* all the corrupted elements of the measured vector are successfully identified. Specifically, we look for an expression involving the error covariance of the optimal linear signal estimate \( \hat{x}^* (y(t)) \) given the non-corrupted measurement \( y(t) \). The error covariance of the latter estimate is given by

\[
E\left\{ (x(t) - \hat{x}^*(y(t))) (x(t) - \hat{x}^*(y(t)))^T \right\} = C_{xx} - C_{xx}C_{yy}^{-1}C_{xx}.
\]  
(6.47)

Similarly, the error covariance of \( \hat{x}^* (y_{\tilde{C}(K)}(t)) \) is given by

\[
E\left\{ (x(t) - \hat{x}^*(y_{\tilde{C}(K)}(t))) (x(t) - \hat{x}^*(y_{\tilde{C}(K)}(t)))^T \right\} =
C_{xx} - (C_{xx})_{*,\tilde{C}(K)} (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1} (C_{xx})_{\tilde{C}(K),*}.
\]  
(6.48)

We want to find an expression relating the two covariances in equations 6.47 and 6.48. To isolate the contribution of \( y_{\tilde{C}(t)} \) to the first covariance, rewrite the second term in the right-hand side of equation 6.47 as follows

\[
C_{xx}C_{yy}^{-1}C_{xx} = \left[ (C_{xx})_{\tilde{C}(K),*} \right]^T \left[ \begin{array}{cc} (C_{yy}^{-1})_{\tilde{C}(K),\tilde{C}(K)} & (C_{yy}^{-1})_{\tilde{C}(K),*} \\ (C_{yy})_{\tilde{C}(K),\tilde{C}(K)} & (C_{yy})_{\tilde{C}(K),*} \end{array} \right] \left[ \begin{array}{c} (C_{xx})_{\tilde{C}(K),*} \\ (C_{xx})_{\tilde{C}(K),*} \end{array} \right].
\]  
(6.49)

Using equation 6.19 with \( C = C_{yy} \), \( A = \tilde{C}(K) \), and \( B = C(K) \) one obtains

\[
(C_{yy}^{-1})_{\tilde{C}(K),\tilde{C}(K)} = (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1} + (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1} (C_{yy})_{\tilde{C}(K),\tilde{C}(K)} (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1}.
\]  
(6.50)

Substituting this expression for \( (C_{yy}^{-1})_{\tilde{C}(K),\tilde{C}(K)} \) in equation 6.49 and rearranging terms gives

\[
C_{xx}C_{yy}^{-1}C_{xx} = (C_{xx})_{*,\tilde{C}(K)} (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1} (C_{xx})_{\tilde{C}(K),*} +
C_{xx} (C_{yy})_{*,\tilde{C}(K)} (C_{yy})_{\tilde{C}(K),\tilde{C}(K)}^{-1} (C_{yy})_{\tilde{C}(K),*} C_{xx}.
\]  
(6.51)

Combining this result with equations 6.47 and 6.48 gives the desired expression

\[
E\left\{ (x(t) - \hat{x}^*(y_{\tilde{C}(K)}(t))) (x(t) - \hat{x}^*(y_{\tilde{C}(K)}(t)))^T \right\} =
\]
\[ E \left\{ (x(t) - \hat{x}^* (y(t))) (x(t) - \hat{x}^* (y(t)))^T \right\} + C_{xx} (C_{yy}^{-1})_{*,C(k)} \left( C_{yy}^{-1} \right)^{-1}_{C(k),C(k)} \left( C_{yy}^{-1} \right)_{C(k),*} C_{xx}. \] (6.52)

The second term in the right-hand side above, which is positive-semidefinite, is the increase in estimation error covariance caused by excluding the "corrupted" elements of the measurement when estimating the signal. Equation 6.52 quantifies the accuracy of the signal estimate when all corrupted elements are detected.

Another useful expression is one that compares the error covariance in equation 6.52 to that of the optimal linear signal estimate computed ignoring the presence of impulsive noise. The latter estimate is given by

\[ \hat{x}^* (z(t)) = C_{xx} C_{yy}^{-1} (z(t) - m_y) + m_x, \] (6.53)
\[ = C_{xx} C_{yy}^{-1} (y(t) - m_y) + m_x + C_{xx} C_{yy}^{-1} u(t), \] (6.54)
\[ = \hat{x}^* (y(t)) + C_{xx} C_{yy}^{-1} u(t). \] (6.55)

Using the fact that \( \hat{x}^* (y(t)) \) is unbiased, the error covariance of \( \hat{x}^* (z(t)) \) conditioned on \( u(t) \) is given by

\[ E \left\{ (x(t) - \hat{x}^* (z(t))) (x(t) - \hat{x}^* (z(t)))^T / u(t) \right\} = E \left\{ (x(t) - \hat{x}^* (y(t))) (x(t) - \hat{x}^* (y(t)))^T \right\} + C_{xx} C_{yy}^{-1} u(t) u(t)^T C_{yy}^{-1} C_{xx}. \] (6.56)

Combining equations 6.52 and 6.56 one obtains

\[ E \left\{ (x(t) - \hat{x}^* (y_{C(k)}(t))) (x(t) - \hat{x}^* (y_{C(k)}(t)))^T \right\} =
E \left\{ (x(t) - \hat{x}^* (z(t))) (x(t) - \hat{x}^* (z(t)))^T / u(t) \right\} -
C_{xx} \left( C_{yy}^{-1} u(t) u(t)^T C_{yy}^{-1} - \left( C_{yy}^{-1} \right)_{*,C(k)} \left( C_{yy}^{-1} \right)^{-1}_{C(k),C(k)} \left( C_{yy}^{-1} \right)_{C(k),*} C_{xx}. \right. \] (6.57)

To show how the expressions above may be used to evaluate or approximate the performance of the estimator, let us consider further the simple case of singly-corrupted measurements, i.e. when only one element per vector is corrupted by impulsive noise. Suppose the impulse occurs in the \( i \)th element and has amplitude \( \mu \). First let us compute the noise-
adjusted distortion assuming the impulse is detected and removed. The distortion in that case is given by

\[
D(i) \triangleq E\left\{ (x(t) - \hat{x}(y_i(t)))^T C_{nn}^{-1} (x(t) - \hat{x}(y_i(t))) \right\}, \tag{6.58}
\]

\[
= \text{tr} \left[ E\left\{ (x(t) - \hat{x}(y_i(t))) (x(t) - \hat{x}(y_i(t)))^T \right\} C_{nn}^{-1} \right], \tag{6.59}
\]

where \( \tilde{i} \) denotes the set of all channels except the \( i^{th} \) one, and \( \text{tr} \left[ \cdot \right] \) computes the trace of a matrix. Using equation 6.52 with \( C^{(K)} = i \) one obtains

\[
D(i) = D^* + \left( C_{yy}^{-1} \right)^{-1}_{i,i} \left( C_{yy}^{-1} \right)^{(-1)}_{i,*} C_{xx} C_{nn}^{-1} C_{xx} \left( C_{yy}^{-1} \right)^{(-1)}_{*,i}, \tag{6.60}
\]

where \( D^* \) is the distortion when the impulsive noise is zero;

\[
D^* = \text{tr} \left[ E\left\{ (x(t) - \hat{x}(y(t))) (x(t) - \hat{x}(y(t)))^T \right\} C_{nn}^{-1} \right]. \tag{6.61}
\]

Similarly, using equation 6.56 one can compute the distortion assuming the impulse is not detected. In that case the distortion is given by

\[
\tilde{D}(i, \mu) \triangleq D^* + \mu^2 \left( C_{yy}^{-1} \right)^{-1}_{i,i} \left( C_{yy}^{-1} \right)^{(-1)}_{i,*} C_{xx} C_{nn}^{-1} C_{xx} \left( C_{yy}^{-1} \right)^{(-1)}_{*,i}. \tag{6.62}
\]

Now suppose that the detection threshold \( \eta \) is selected such that probability of a false alarm, i.e. an erroneous detection, is practically zero. Then there are only two possible outcomes for the detection step. Either the impulse in the \( i^{th} \) element is detected, or no impulses are detected at all. Let \( P_D(i, \mu) \) be the probability of detecting the impulse for the chosen value of the threshold \( \eta \). Then the expected distortion \( D(i, \mu) \) is approximately

\[
D(i, \mu) \approx D^* + \left( P_D(i, \mu) \left( C_{yy}^{-1} \right)^{-1}_{i,i} + (1 - P_D(i, \mu)) \mu^2 \right) \left( C_{yy}^{-1} \right)^{-1}_{i,*} C_{xx} C_{nn}^{-1} C_{xx} \left( C_{yy}^{-1} \right)^{(-1)}_{*,i}. \tag{6.63}
\]

Thus, for high probability of detection the distortion approaches the optimal distortion that can be obtained ignoring the channel corrupted by the impulse.
6.5 Evaluation of the Two-step Estimator using AIRS Data

This section documents the performance of the suboptimal estimator described in Section 6.4 when it is applied to a set of simulated AIRS spectra. The subsequent analysis by no means constitutes a complete characterization of the estimator. The objective is to give an indication of its performance in plausible cases. The performance of the estimator is evaluated numerically. First the performance of the detection step is evaluated separately in terms of the probability of false alarm and the probability of detection. These probabilities are evaluated for singly-corrupted spectra, i.e. one corrupted channel per spectrum, as a function of the detection threshold $\eta$ and the impulsive noise amplitude. The performance of the detection and estimation steps together is evaluated in terms of the mean noise-adjusted distortion, where the noise-adjustment is done with respect to the typical noise component. The distortion is also evaluated for singly-corrupted spectra. The performance of the estimator in the case of multiply-corrupted spectra is illustrated qualitatively by means of examples.

The results reported below are based on the assumption that the second-order statistics of the noiseless radiance and the typical noise are known. The statistics of the noiseless radiance were estimated from a set of 7496 simulated noiseless AIRS spectra; the same data set used in Section 5.5 to test the coding algorithm for data corrupted by Gaussian noise only. However, only 600 of those spectra were used to numerically evaluate the performance of the detection step. The typical noise component was simulated by adding pseudo-random, zero-mean Gaussian noise to the set of 600 noiseless spectra. Most of the results reported below are for the case where the covariance matrix of the typical noise equals the nominal (diagonal) AIRS noise covariance matrix.

Let us start the evaluation by examining the correlation between the prediction errors for AIRS data. It was argued in Section 6.4 that our detection algorithm would be most effective in cases where the cross-correlation coefficients of the prediction errors are close to zero. Consider first the case in which the typical noise component has the nominal (diagonal) AIRS noise covariance matrix. The degree of correlation between the prediction errors in that case is bounded as shown in Figure 6.2(a), which shows the off-diagonal entry of largest amplitude in each column of the matrix of correlation coefficients. It is clear that in this case the prediction errors are weakly correlated. The maximum correlation coefficient
Figure 6-2: Maximum amplitude of the cross-correlation coefficient of the prediction errors for all the spectral channels of AIRS. For each channel the maximum amplitude of the correlation coefficient between its prediction error and the prediction error of any of the other channels is shown. Each graph corresponds to a different choice of the covariance matrix of the typical noise. In graph (a) the typical noise is assumed to have the nominal (diagonal) noise covariance matrix. In graph (b) the noise covariance matrix is block-diagonal, with the noise on any pair of channels located in the same detector array having a correlation coefficient of 0.3. The noise variances take the nominal values. The covariance matrix of the noiseless radiance was estimated from the set of 7,496 simulated noiseless AIRS spectra.

amplitude exceeds a value of 0.1 in sixty-three out of 2371 channels. That indicates that an impulsive noise burst of significant amplitude on any channel should stand out clearly in the corresponding prediction error. The rms values of the prediction errors in all channels are shown in Figure 6-3 in units of rms typical noise. In all channels the rms prediction error exceeds the rms noise by less than twenty percent, and in ninety-six percent of the channels by less than five percent. The fact that the rms prediction errors in all channels are so close to the rms noises is a clear indication of the high level of redundancy in AIRS spectra. For instance, these results imply that if any single channel is sporadically missing, the signal-to-noise ratio (in terms of variances) when it is missing would be more than seventy percent of the signal-to-noise ratio when it is actually measured! This would be assuming that the particular channel is measured frequently enough that one could accurately estimate its cross-correlation with other channels, so that it can be predicted from them. With regard
Figure 6-3: Root-mean-square prediction errors in the absence of impulsive noise for two different typical noise covariance matrices. In one case the noises in any two channels are uncorrelated. In the other case the noises in any two channels located in the same detector array have a correlation coefficient of 0.3. The noise variances take the nominal values in both cases.

to the detection of impulsive noise, Figure 6-3 suggests that for these data one should be able to detect impulsive noise bursts with amplitudes of a few times the rms typical noise of the corresponding channel.

For comparison let us examine how the correlation of the prediction errors changes when the typical noise is weakly correlated between spectral channels located on the same detector array. The spectral coverage of each of the seventeen detector arrays of AIRS is listed in Table 2.1. Specifically, let the correlation coefficient of the typical noises on any two channels located in the same array be equal to 0.3. In addition, let the noise variance take its nominal value for each channel. Figure 6-2(b) shows the maximum correlation coefficient amplitudes for this case. The results are practically the same as for the case of uncorrelated noise. One significant difference is that the rms prediction errors are noticeably lower than in the case of uncorrelated noise, as shown in Figure 6-3. This is explained by noting that now a component of the noise in each channel can be predicted from the noise in the other.
channels in the same detector array. Thus, if the typical noise is uniformly correlated to some extent within a detector array it should be possible to detect impulsive noise bursts of smaller amplitude in comparison with the case where the noise is uncorrelated. During the rest of this section we focus on the latter case.

To evaluate the performance of the detection step when applied to AIRS data, the probabilities of detection and false alarm were evaluated numerically for singly-corrupted spectra, i.e. when only one channel was corrupted by impulsive noise. Suppose that an impulsive noise burst of value \( \mu \) occurs on the \( i^{th} \) channel. Accordingly, we define the hypothesis

\[
H(i, \mu) : \quad u_j(t) = \begin{cases} 
\mu, & j = i; \\
0, & 1 \leq j \leq N, j \neq i;
\end{cases}
\]  

(6.64)

where \( N \) is the number of spectral channels. As before, let \( K \) denote the number of iterations of the detection algorithm that were performed for the observed spectrum \( z(t) \), and let \( C^{(K)} \) denote the set of channels that were labeled as “corrupted.” The probability of false alarm is defined as

\[
P_{FA}(i, \mu) = \Pr \{ j \in C^{(K)} \text{ for any } j \neq i / H(i, \mu) \},
\]  

(6.65)

and the probability of the detection as

\[
P_{D}(i, \mu) = \Pr \{ i \in C^{(K)} / H(i, \mu) \}.
\]  

(6.66)

where \( \Pr \{ A/B \} \) denotes the conditional probability of event \( A \) given event \( B \) occurred. Although not indicated explicitly, \( P_{FA}(i, \mu), P_{D}(i, \mu) \) depend on the value of the threshold \( \eta \) applied to the magnitudes of the normalized prediction errors.

Figure 6-4 shows graphs of probability of false alarm vs. probability of detection as a function of the threshold \( \eta \) for three different impulsive noise amplitudes. Each graph is a so-called operating characteristic. The probabilities were evaluated by adding an impulse of the specified magnitude to a single channel in each of 600 spectra and then counting the number of detections and false alarms obtained when the detection algorithm was applied to those data. This exercise was performed for six different spectral channels: 4.3596 \( \mu m \), 7.0778 \( \mu m \), 9.0352 \( \mu m \), 10.7676 \( \mu m \), 12.9343 \( \mu m \), and 15.3951 \( \mu m \). For each channel the amplitude of the added impulse was separately set to three, four, and six times the rms prediction error. Each point in the figure shows the average probabilities over the set of six
Figure 6-4: Operating characteristic of the detection step for singly-corrupted spectra. Each curve shows the probabilities of detection and false alarm evaluated at four different values of the detection threshold for the impulse amplitude indicated in the legend. The value of the threshold is indicated alongside each point. The probabilities were averaged over the occurrence of the impulse noise burst on six different spectral channels: 4.3596 μm, 7.0778 μm, 9.0352 μm, 10.7676 μm, 12.9343 μm, and 15.3951 μm. The center of each “error-box” corresponds to the average probability, and the edges correspond the minimum and maximum values over the data set. Six hundred spectra were used to generate the curves.

channels, and the “error-boxes” around each point indicate the maximum and minimum values over the same set of channels. Because the amplitude of the impulsive noise added to each channel was chosen relative to its rms prediction error, the probabilities of detection and false alarm are roughly the same in each case. As expected, both probabilities increase as the value of the threshold is decreased.

The probability of false alarm is not affected appreciably by the amplitude of the added impulse. This is so because the number of channels is large, and only the prediction error in the one channel is affected by the impulse. A good approximation of the probability of false alarm can be obtained by assuming: 1) there are no impulses, and 2) the prediction errors are mutually independent Gaussian random variables. Then the probability of false
alarm can be approximated as

\[ P_{FA}(i, \mu) \approx 1 - \prod_{j=1}^{N} \Pr \left\{ \left| e_j^{(1)} \right| \leq \eta \sigma_j^{(1)} \right\} , \]  

(6.67)

\[ = 1 - (2\Phi(\eta) - 1)^N , \]  

(6.68)

where \( e_j^{(1)} \) is the vector of prediction errors in the first iteration of the algorithm, \( \sigma_j^{(1)} \) is the rms value of \( e_j^{(1)} \), and \( \Phi(\cdot) \) is the cumulative distribution of a zero-mean unit-variance Gaussian random variable. For example, for \( \eta \) equal to 3.5, 4, and 4.5 this approximation gives probabilities of false alarm of 0.668, 0.140, and 0.016, respectively. These values agree closely with those in Figure 6-4.

The results in Figure 6-4 show that by applying our detection algorithm to these data it is possible to detect impulses of amplitude equal to or greater than six times the rms prediction error in the corresponding channel with very high probability of detection (0.931 for \( \eta = 4.5 \)) and very low probability of false alarm (0.016 for \( \eta = 4.5 \)). This is achieved using a threshold \( \eta = 4.5 \).

Figure 6-5 illustrates the accuracy of the two-step estimator for the case of singly-corrupted spectra. The amplitude of the impulse was set to six times the rms prediction error of the channel where it occurred. The figure shows the fractional increment in noise-adjusted distortion, relative to the case of zero impulsive noise, when the impulse occurs in each of the spectral channels. The distortion was computed using equation 6.63 with a probability of detection of 0.931 for \( \eta = 4.5 \) and zero for \( \eta = \infty \). The covariance matrix \( C_{xx} \) of the noiseless spectrum was estimated from the set of 7496 simulated AIRS spectra. The nominal (diagonal) AIRS noise covariance matrix was used. Less than a one percent increase in distortion is observed for most channels when the detection threshold \( \eta \) is set to 4.5. A increase of up to ten percent can occur if the impulses are not detected \( (\eta = \infty) \) and removed before estimating the signal. Note that for \( \eta = \infty \) the distortion would increase in proportion to the squared amplitude of the impulse (see equation 6.63). On the other hand, for \( \eta = 4.5 \) one would not expect the distortion to increase much more if larger impulses occur because they would be detected with even higher probability.

Figures 6-6 and 6-7 show two examples of how the two-step estimator performs when multiple impulses are present in a single spectrum. Note that in these examples it is difficult to identify the corrupted channels by looking directly at the spectrum (Figures 6-
Figure 6-5: Fractional increment in noise-adjusted distortion (all channels) caused by an impulse of amplitude equal to six times the rms prediction error. For each spectral channel the graph shows the fractional increment which would be caused by the presence of an impulse of the specified magnitude on that channel.

6(b) and 6-7(b)) but they are easy to identify in the vector prediction errors. The errors in the final estimate of the spectrum (Figures 6-6(e) and 6-7(e)) are virtually identical to the ones obtained in the absence of any impulses. These examples show that, at least when the number of corrupted channels is small compared to the total number of channels per spectrum, multiple impulses can be detected and removed accurately.

6.6 The Coding Algorithm

The coding algorithm for data corrupted by additive Gaussian noise and impulsive noise combines the two-step estimator developed in Section 6.4 with the NAPC-based codebook discussed in Section 3.3.3. Although this combination does not yield an optimal codebook for the case at hand, the codebook is optimal when no impulses are present in the data provided that no false alarms occur, i.e. if no impulses are incorrectly detected. Below we discuss the implementation of the coding algorithm.
Figure 6-6: Example of estimator performance for a multiply-corrupted spectrum. The detection threshold $\eta$ was set to 4.5. Graph (a) shows impulses in five different channels, their amplitudes given in units of rms prediction error. Graph (b) shows the corrupted spectrum. Graph (c) shows the errors, in units of rms noise, in the estimate of the spectrum before the impulses are detected. The normalized (unit-variance) prediction errors in the first iteration of the detection step are shown in graph (d). The errors in the final estimate of the spectrum are shown in graph (e) in units of rms noise. They are practically identical to the errors obtained in the absence of impulsive noise. All these results assume that the typical noise has the nominal AIRS noise covariance matrix.
Figure 6-7: Example of estimator performance for a multiply-corrupted spectrum. The detection threshold $\eta$ was set to 4.5. Graph (a) shows a pulse spanning around nine channels, with the pulse amplitudes given in units of rms prediction error. Only a few channels in the neighborhood of the pulse are shown. Graph (b) shows the corrupted spectrum. Graph (c) shows the errors, in units of rms noise, in the estimate of the spectrum before the impulses are detected. The normalized (unit-variance) prediction errors in the first iteration of the detection step are shown in graph (d). Only a few channels in the neighborhood of the pulse are shown. The errors in the final estimate of the spectrum is shown in graph (e) in units of rms noise. They are practically identical to the errors obtained in the absence of impulsive noise. All these results assume that the typical noise has the nominal AIRS noise covariance matrix.
Figure 6-8: Block diagram of the coding algorithm for data corrupted by additive Gaussian noise and impulsive noise.

The coding algorithm transforms the signal estimator given by equation 6.46 to NAPC space and separately quantizes the NAPC coefficients. The NAPC transformation is done with respect to the typical noise component. Recall that using the NAPC decomposition of the signal-plus-typical-noise $y(t)$ one can express the covariance matrix $C_{xx}$ of the noiseless signal $x(t)$ as

$$C_{xx} = C_{nn}^{1/2} U \Gamma U^T \left(C_{nn}^{1/2}\right)^T,$$  \hspace{1cm} (6.69)

where $\Gamma$ is the diagonal matrix of signal-to-noise ratios of the NAPC, $U$ is an orthogonal matrix, and $C_{nn}^{1/2}$ is a square-root of the covariance matrix of the typical noise. To transform the signal estimator in equation 6.46 to the NAPC domain one premultiplies the estimator by $U^T C_{nn}^{-1/2}$ after subtracting the mean $m_x$. Thus, the transformed estimator is given by

$$U^T C_{nn}^{-1/2} \left(\tilde{x}^* - E(z(t)) - m_x\right) = \Gamma U^T \left(C_{nn}^{1/2}\right)^T \text{diag}\left(\left(\sigma_1^{(K)}\right)^{-2}, \ldots, \left(\sigma_N^{(K)}\right)^{-2}\right) e\left(z(t), \bar{C}^{(K)}\right).$$  \hspace{1cm} (6.70)

The elements of the transformed estimator are then quantized using Lloyd-Max quantizers as before. After quantization, the signal estimate is premultiplied by $C_{nn}^{1/2} U$ to transform it back to regular space. Figure 6-8 shows a partial diagram of the algorithm. Although not shown explicitly in the diagram, the strategy described in Section 5.6 for coding individual channels with large noise estimates would be included as part of the algorithm.
6.7 Summary and Additional Comments

In this chapter we developed a compression algorithm for multispectral data corrupted by both additive Gaussian noise and impulsive noise. The Gaussian noise component serves as a model for both instrument noise and residual calibration errors. Impulsive noise may occur, for example, due to the radiation environment in orbit in the case of a spaceborne sensor. The compression algorithm uses the same NAPC-based codebook as the algorithm developed in Chapter 5 for data free of impulsive noise, except that the signal estimator was modified to minimize the impact of impulsive noise.

The new signal estimator consists of a detection step followed by an estimation step. First any channels in the measured spectrum which likely have been corrupted by impulsive noise are detected, and then the signal (noiseless spectrum) is estimated from the non-corrupted channels. The detection step is implemented through an iterative procedure which involves computing the prediction of each channel based on all other channels. Channels for which the prediction error exceeds a specified threshold are declared corrupted and ignored in subsequent iterations. The process stops when none of the prediction errors exceed the threshold. Subsequently the signal is estimated in a linear minimum mean-square error sense assuming all channels corrupted by impulsive noise were correctly identified during the detection step. Two advantages of this estimator are the following:

1. It does not make use of any statistics of the impulsive noise. This is convenient in cases where it is difficult to estimate those statistics because the impulsive noise occurs infrequently.

2. In contrast to simpler filtering techniques like median filtering, it does not require the underlying signal to be locally smooth. This is important since high-spectral-resolution remote sensing data are generally not smooth along the spectral dimension.

An efficient implementation of the coding algorithm was given. When the number of corrupted channels per spectrum is small compared to the total number of channels $N$, the computational cost of the signal estimator is approximately $N^2$ operations. The rest of the coding algorithm has a computational cost of about $2Nm$, where $m$ is the number of NAPC which are assigned non-zero bit allocations.

The signal estimator was evaluated using simulated AIRS data. For those data it was found that impulses of amplitude equal to or greater than six or seven times the rms
Gaussian noise, depending on the channel, can be detected with very high probability of detection and very low probability of false alarm. This applies when the Gaussian noise is uncorrelated between channels. If that noise is correlated between channels, impulses of smaller amplitude can be detected with the same efficiency. For spectra corrupted by single impulses of amplitude equal to six times the rms Gaussian noise it was found that the noise-adjusted distortion in the signal estimate was increased by less than one percent. If those impulses had not been removed prior to estimating the signal the increment in distortion would have been about ten times higher. Further examples suggested that the estimator is also effective when multiple channels per spectrum are corrupted.
Chapter 7

Conclusion

7.1 Summary of the Thesis

The objective of this thesis was to develop efficient and robust non-reversible coding algorithms for multispectral remote sensing data. Even though many efficient non-reversible or lossy coding algorithms have been proposed for such data, their application is often limited due to the risk of excessively degrading the quality of the data if changes in operating conditions occur. Specifically, a coding algorithm for remote sensing data should be robust to changes in instrument characteristics and atmospheric/surface statistics, and data outliers. For their inherent ability to reconstruct the data exactly under all circumstances, reversible or lossless compression algorithms are preferred by most users of remote sensing data. However, reversible compression algorithms are known to yield very limited compression when applied to data from most modern remote sensing platforms. Since the need for data compression is likely to be greater for future systems, the formulation of coding algorithms which combine the robustness of lossless techniques with the efficiency of lossy techniques is of much interest to the remote sensing community.

The coding algorithms developed in this work achieve high data compression by treating the problem of compressing remote sensing data as a noisy source coding problem. One algorithm assumes an independent additive Gaussian model, which is used to model both sensor noise and residual data calibration errors. A second algorithm is applicable when the data are also corrupted by impulsive noise. Both algorithms use codebooks with an estimator-coder structure, which is suggested by information theory and quantization theory for coding noisy data. In both cases the codebook computes an estimate of the signal—
in our case the noiseless radiance spectrum—and then codes that estimate efficiently. As a result, the original data may be compressed essentially by discarding as much noise as possible and retaining only information about the signal. From that point of view the compression provided by our algorithms can be considered to be near-lossless.

Both coding algorithms were made robust to slow variations in receiver gain and offset, and to data outliers. Robustness with respect to slow instrument drift is achieved by calibrating the data before it is coded, so that receiver gain and offset variations are attenuated through the use of calibration data. Individual outlier channels in each spectrum are preserved by separately coding the signal estimate and the noise estimate for those channels. This allows the user to reconstruct the outlier channels at the same resolution of the original data. In addition, in the case of data free of impulsive noise the coding algorithm adapts to changes in the second-order statistics of the data by estimating those statistics from each block of data to be coded.

Part of the motivation for using an estimation-coding approach was provided by studies conducted by others, which explained the ineffectiveness of reversible compression of multispectral data in terms of the entropy of the noise in those data [12]. In spite of those studies, one finds that the estimation-coding paradigm has been rarely applied to multispectral data. Most of the data compression techniques used in that application are either lossless or lossy in the traditional sense, i.e. without distinguishing between signal and noise. Although not without precedent [6], this thesis gives evidence of the benefit of using an estimation-coding approach for compression of high-spectral-resolution multispectral data. In our view, this is a natural approach for applications which require near-lossless compression.

Because our interest was mainly in vector measurements of high dimensionality, we chose to implement the estimation-coding paradigm with codebooks based on linear matrix operators and scalar quantization as opposed to full vector quantization. The motivation for this choice was simplicity of implementation. In addition, our algorithms rely on the use of data-dependent statistical transforms rather than sinusoidal transforms which, although computationally efficient, are generally not appropriate for exploiting the correlation structure of arbitrary vector data, and, in particular, of multispectral data. Malvar was the first, to our knowledge, to obtain the optimal codebook structure upon which our work built [17]. Others have studied the optimal structure and design of vector quantizers for noisy data [16, 19].
7.1.1 Main Contributions

Even though the primary objective of the thesis was compression of remote sensing data, some of its results have wider application.

One generic contribution of the thesis relates to empirical design of transform codebooks for noisy Gaussian data. We considered a scenario in which the codebook must be designed from a finite number of data samples, and a priori knowledge of the statistics of the noise. Under those conditions it is not possible to design a codebook by direct optimization with respect to mean-weighted-square error because the problem is not well-defined. We pursued a design approach consisting of estimating the statistics of the signal and the noisy data, and implementing the optimal codebook for those estimates. Two interesting results we obtained are the following:

- It was shown that codebooks designed as above are asymptotically optimal whenever the estimates of the statistics are consistent. This result gives validity to the proposed design approach. A similar result is known for vector quantizers [19] under different assumptions. We are not aware of a previous proof applying specifically to the transform codebooks considered in this thesis. One limitation of our result is that it assumes both the signal and the independent additive noise have multivariate Gaussian distributions.

- The second result has to do with the specific choice of estimates of the statistics used to design the codebook. It was observed through numerical simulations that codebooks designed using Maximum-Likelihood (ML) estimates (for Gaussian signal and noise vectors) often have poor rate-distortion characteristics even when the number of data samples used to design the codebook is several times the vector dimension. In a simulated example with low signal-to-noise ratio this occurred even when the number of samples was eight times that dimension. We proposed an ad hoc modification of the ML estimates that improved the rate-distortion characteristics of the codebook significantly. The modified estimates incorporate information about the number of data samples used to obtain the ML estimates, and lead to a better estimate of the linear MMSE matrix. For this reason the modified estimates can be beneficial for noise filtering of vector data whenever the statistics of the signal are not known a priori and a limited amount of noisy data is available for estimating those statistics.
Another contribution of general character is the two-step estimator developed for signals in an additive mixture of Gaussian noise and impulsive noise. Our estimator relies on the second-order statistics of the signal and the non-impulsive noise to detect data samples corrupted by impulsive noise and remove them before estimating the signal. The estimator may be applied to any kind of stationary data that admits block processing. Also, it may be possible to obtain a sliding-window version of the estimator for scalar time-series data, for example, sampled speech data, although that possibility was not explored in this work. One practical advantage of the estimator is that it does not depend on the distribution of the impulsive noise. Another advantage is that, contrary to simpler filtering techniques like median filtering, it does not require the underlying signal to be low-pass in nature.

The modified log-variance bit allocation rule derived in Section 5.4, which takes into account codebook overhead, may be used in other data-dependent transform coding algorithms similar to the ones developed in this work.

The two coding algorithms developed in the thesis are contributions mainly within the application of remote sensing. Similar algorithms have been applied previously to other kinds of data, for example, speech and monochromatic images. The work of Malvar establishes a precedent in this respect [17]. Nonetheless, there are some novel features in the algorithms.

One novel aspect is the use of an optimal estimator-coder structure as a basis for near-lossless compression. We exploit the fact that at high bit-rates the rate-distortion characteristic of such a codebook approaches an asymptote corresponding to the distortion of the “best” signal estimate obtained from the original data. In our algorithms “best” holds in a linear minimum mean-square error sense. Because the distortion asymptote is finite, it is possible to operate within a desired fraction of that optimal distortion at a finite bit-rate. This approach provides a very good basis for near-lossless compression since the resulting distortion can be chosen to be practically indistinguishable from the optimal distortion. The same is not possible within the confines of the traditional lossy coding approach because the desired target is the original noisy data, so the optimal distortion is zero.

Another feature of the coding algorithms which is of practical importance in remote sensing applications is their capability for preserving individual outlier channels in each spectrum. This was implemented by identifying channels in which the estimated noise (noisy data minus signal estimate) was unusually large. For those channels the noise estimates were
quantized in addition to the signal estimate so that the original data could be reconstructed approximately and inspected by the user if desired.

Finally, the results on compression of simulated AIRS data are another contribution of the thesis. First let us summarize the results of simulations without impulsive noise. For the case in which the rms additive Gaussian noise equals the nominal AIRS noise-equivalent radiance for a 250 K target, with no correlation between channels, the main results are the following:

- Near-lossless compression ratios of up to 16:1 (0.8 bits/channel) for blocks of 2400 spectra, and 32:1 (0.4 bits/channel) for blocks of 4800 and 7200 spectra were obtained with negligible distortion due to coding. These figures include the codebook overhead. A maximum lossless compression ratio of 4.7:1 was predicted for the same data assuming only the noise is coded.

- The average noise variance reduction at high bit-rates varied from 12 dB to 14 dB, with higher noise reduction for the longer blocks.

- At low bit-rates (below 0.8 bits/channel for blocks of 2400 spectra, and 0.4 bits/channel for blocks of 4800 and 7200 spectra) deviations were observed between the actual and predicted distortion of the codebook. These deviations arise because the distributions of some of the transform coefficients (NAPC) depart from a Gaussian shape, and the quantizers were designed for Gaussian inputs. The quantizer mismatch becomes irrelevant at higher bit-rates.

Additional simulations showed that if the noise is correlated (with correlation coefficient of 0.3) between any two channels located in the same detector array, the deviations between the predicted and actual distortions are larger. Again, this occurs due to deviations from the Gaussian assumption.

For AIRS data corrupted by impulsive noise it was shown that impulses of amplitude greater than or equal to six or seven times the rms Gaussian noise, depending on the spectral channel, can be detected with high probability of detection and very low probability of false alarm using the two-step estimator. This applies when the Gaussian noise is uncorrelated between channels. If that noise is correlated between channels, impulses of smaller amplitude can be detected with the same efficiency. For spectra corrupted by single impulses of amplitude equal to six times the rms Gaussian noise it was found that the noise-adjusted
distortion of the signal estimate increased by less than one percent. If those impulses had not been removed prior to estimating the signal, the increment in distortion would have been about ten times higher. Further examples suggested that the estimator is also effective when multiple channels per spectrum are corrupted.

7.2 Suggestions for Further Work

In our treatment of empirical design of transform codebooks for noisy Gaussian data we established that consistency of the estimates of the statistics is sufficient to guarantee asymptotic optimality of the codebook. We did not, however, attempt to quantify the rate of convergence to the optimum as the number of data samples used to design the codebook increases. One possible extension would be to quantify the rate of convergence in terms of a bound on the performance as a function of the number of data samples.

A related problem that could be investigated further is the choice of estimates of the statistics used to design the codebook. We considered using Maximum-Likelihood (ML) estimates of the signal statistics for given noise statistics and found the rate-distortion performance of the resulting codebooks to be poor in general. We were able to improve the performance of the empirical codebooks by performing an \textit{ad hoc} modification of the ML estimates. One alternative would be to use an expectation-maximization approach as in [54] to recursively estimate the signal and noise statistics simultaneously from the noisy data, starting with \textit{a priori} estimates of those statistics derived from past data. The final estimates of the statistics obtained that way would be more consistent with the observed data in the sense that they would increase the likelihood of the data when compared to the initial ones. One practical limitation of that approach may be a higher computational burden.

Yet another related research opportunity is to formulate empirical codebook design procedures which are more appropriate for adaptive applications. To clarify, this is always in relation to codebooks based on data-dependent statistical transforms. We used a design procedure in which the codebook for each data block is designed using only the data in that block. In particular, other data adjacent in time were not used at all. It seems that it would be advantageous to use other data adjacent in time to design the codebook if the statistics are known to vary “slowly.” Modeling changes in statistics with respect to time
would appear to be beneficial to undertake such an approach.

Finally, many improvements can be made to the coding algorithms developed in this work. Possible directions for improvement are the following:

- Extend the coding algorithms to take advantage of spatial correlations as well as inter-channel correlations. Higher compression ratios may be obtained by block-coding spatially adjacent pixels using, for example, a block transform like the DCT. The potential for higher compression will depend on the spatial homogeneity of the particular data set. In the case of AIRS, radiances in channels which are relatively insensitive to clouds should be fairly homogeneous across many pixels, especially after the noise has been filtered out. The high spatial redundancy of those data may allow many times higher compression than that obtained by coding each pixel separately. However, AIRS also has hundreds of window (≈ 700) and humidity sounding (≈ 400) channels that are sensitive to clouds. Radiiances observed in those channels over cloudy skies would be less homogeneous across adjacent pixels, and the potential for additional compression would be lower than for radiances observed in clear skies.

- Offer various choices of Lloyd-Max quantizers optimized for input distributions other than Gaussian. In particular, for AIRS data some of the NAPC coefficients have skewed distributions. It is desirable to identify more appropriate models for those distributions.

- Formulate better strategies for preserving data outliers. In particular, find ways of incorporating this requirement formally in the design of the codebook.

- Extend the coding algorithm for data corrupted by impulsive noise so that it can be used in adaptive mode, without requiring the statistics of the signal to be specified a priori.
Appendix A

Proof of Proposition 3.1

Proposition Let \( z \) and \( v \) be zero-mean jointly Gaussian random variables with \( \sigma_v^2 \triangleq E\{v^2\} > 0 \), \( \sigma_{vz} \triangleq E\{vz\} \), and \( \sigma_z^2 \triangleq E\{z^2\} \). The \( L \)-level quantizer \( q(\cdot) \) that minimizes \( E\{(z - q(v))^2\} \) is given by \( q^*(v) = \tilde{q}^* \left( \frac{\sigma_z}{\sigma_v} v \right) \), where \( \tilde{q}^*(\cdot) \) is the \( L \)-level Lloyd-Max quantizer for a zero-mean Gaussian random variable with variance \( \sigma_{zu}^2 / \sigma_v^2 \). The optimal distortion is given by

\[
E\{(z - q^*(v))^2\} = \sigma_z^2 - \frac{\sigma_{zu}^2}{\sigma_v^2} (1 - d(L)),
\]

where \( d(L) \) is the distortion of the \( L \)-level Lloyd-Max quantizer for a unit-variance Gaussian random variable.

Proof: First note that one can write

\[
E\{(z - q(v))^2\} = E\{(z - z_B(v))^2\} + E\{(z_B(v) - q(v))^2\} \tag{A.1}
\]

where \( z_B(v) = \frac{\sigma_{zu}}{\sigma_z} v \) is the Bayesian least-squares estimate of \( z \) given \( v \) [29]. First consider the case when \( \sigma_{zu} \neq 0 \). From equation A.1 the optimal quantizer must minimize

\[
E\{(z_B(v) - q(v))^2\} = E\left\{ \left( z_B(v) - q \left( \frac{\sigma_z^2}{\sigma_{zu}} z_B(v) \right) \right)^2 \right\} \tag{A.2}
\]

\[
= E\left\{ (z_B(v) - \tilde{q}(z_B(v)))^2 \right\} \tag{A.3}
\]

where \( \tilde{q}(z_B(v)) = q(v) \). Thus, the optimal \( \tilde{q} \) is by definition the Lloyd-Max quantizer for a zero-mean Gaussian random variable with variance \( \sigma_{z_B}^2 = \sigma_{zu}^2 / \sigma_z^2 \) [39, 40]. It is easy to show that the optimal quantizer for a zero-mean unit-variance Gaussian random variable
$z_o$ is $q_o^*(z_o) = \frac{1}{\sigma_{z_B}} \tilde{q}^* (\sigma_{z_B} z_o)$. Thus, the optimal distortion is given by

$$E\{ (z - q^*(v))^2 \} = E\{ (z - z_B(v))^2 \} + E\{ (z_B - \tilde{q}^* (z_B))^2 \}$$

(A.4)

$$= \sigma_z^2 - \frac{\sigma_{z_B}^2}{\sigma_q^2} + E\{ (z_B - \tilde{q}^* (z_B))^2 \}$$

(A.5)

$$= \sigma_z^2 - \frac{\sigma_{z_B}^2}{\sigma_q^2} + \sigma_{z_B}^2 E\left\{ \left( \frac{z_B}{\sigma_{z_B}} - \frac{1}{\sigma_{z_B}} \tilde{q}^* (\sigma_{z_B} (z_B/\sigma_{z_B})) \right)^2 \right\}$$

(A.6)

$$= \sigma_z^2 - \frac{\sigma_{z_B}^2}{\sigma_q^2} + \sigma_{z_B}^2 E\{ (z_a - \tilde{q}_o^* (z_o))^2 \}$$

(A.7)

$$= \sigma_z^2 - \frac{\sigma_{z_B}^2}{\sigma_q^2} + \sigma_{z_B}^2 d(L)$$

(A.8)

$$= \sigma_z^2 - \frac{\sigma_{z_B}^2}{\sigma_q^2} + \frac{\sigma_{z_B}^2}{\sigma_q^2} d(L)$$

(A.9)

where, $d(L)$ is the distortion of the $L$-level Lloyd-Max quantizer for a unit-variance Gaussian random variable. Equation A.4 follows from equation A.3, and equation A.5 follows from the properties of the Bayesian least-squares estimate. If $\sigma_{z_B} = 0$, from equation A.1 it is clear that the optimal quantizer is simply $q(v) = 0$ and the optimal distortion equals $\sigma_z^2$.

Q.E.D.
Appendix B

Maximum-Likelihood Estimates of Gaussian Statistics from Noisy Observations

Here we derive Maximum-Likelihood (ML) estimates of the mean and covariance of a Gaussian signal corrupted by independent additive Gaussian noise. Let \( \mathcal{N}(m, C) \) denote the Gaussian distribution with mean \( m \) and covariance \( C \). The signal \( x \) and the noise \( n \) are distributed \( \mathcal{N}(m_x, C_{xx}) \) and \( \mathcal{N}(0, C_{nn}) \), respectively. Both \( x \) and \( n \) are \( N \)-dimensional vectors. Let \( Y^{(k)} = \{y(1), y(2), \ldots, y(k)\} \) denote a sequence of \( k \) samples of the noisy observation \( y = x + n \). We want to maximize the log-likelihood of \( Y^{(k)} \) as a function of \( m_x \) and \( C_{xx} \):

\[
\mathcal{L}(m_x, C_{xx}; Y^{(k)}) \triangleq -\frac{Nk}{2} \log 2\pi - \frac{k}{2} \log |C_{xx} + C_{nn}| \\
- \frac{1}{2} \text{tr} \left[ (C_{xx} + C_{nn})^{-1} \sum_{t=1}^{k} (y(t) - m_x)(y(t) - m_x)^T \right]. \tag{B.1}
\]

The \( \text{tr} \{ \cdot \} \) operator computes the trace of a matrix. The maximization of this particular log-likelihood is easier to carry out in a coordinate system where the noise covariance equals the identity matrix. For any invertible \( C_{nn} \) this can be effected by finding a square matrix \( C_{nn}^{1/2} \) satisfying \( C_{nn}^{1/2} \left( C_{nn}^{1/2} \right)^T = C_{nn} \). In the new coordinate system the log-likelihood is

221
given by
\[
\mathcal{L} \left( \mathbf{m}_{\tilde{x}}, \mathbf{C}_{\tilde{x}\tilde{x}}; \tilde{Y}^{(k)} \right) = -\frac{Nk}{2} \log 2\pi - \frac{k}{2} \log |\mathbf{C}_{nn}| - \frac{k}{2} |\mathbf{C}_{\tilde{x}\tilde{x}} + I| \\
-\frac{1}{2} \text{tr} \left[ (\mathbf{C}_{\tilde{x}\tilde{x}} + I)^{-1} \sum_{t=1}^{k} (\tilde{y}(t) - \mathbf{m}_{\tilde{x}})(\tilde{y}(t) - \mathbf{m}_{\tilde{x}})^T \right],
\]
where tildes are used to denote the transformed variables; \( \tilde{y} = \mathbf{C}_{nn}^{-1/2} y, \tilde{x} = \mathbf{C}_{nn}^{-1/2} x. \)

We first find the ML estimates of the parameters in the transformed space and then transform them to the original space. Since the last term on the right hand side of equation B.2 is quadratic negative-definite with respect to \( \mathbf{m}_{\tilde{x}} \), the ML estimate of the mean is obtained simply by setting to zero the gradient of this term with respect to \( \mathbf{m}_{\tilde{x}} \).

\[
(\mathbf{C}_{\tilde{x}\tilde{x}} + I)^{-1} \sum_{t=1}^{k} (\tilde{y}(t) - \hat{\mathbf{m}}_{\tilde{x}}^{ML}) = 0 \tag{B.3}
\]
\[
\sum_{t=1}^{k} (\tilde{y}(t) - \hat{\mathbf{m}}_{\tilde{x}}^{ML}) = 0 \tag{B.4}
\]
\[
\mathbf{C}_{nn}^{1/2} \sum_{t=1}^{k} (\tilde{y}(t) - \hat{\mathbf{m}}_{\tilde{x}}^{ML}) = 0 \tag{B.5}
\]
\[
\sum_{t=1}^{k} (y(t) - \hat{\mathbf{m}}_{x}^{ML}) = 0 \tag{B.6}
\]
Thus, the ML estimate of the mean is the sample mean of the noisy data.

\[
\hat{\mathbf{m}}_{x}^{ML} = \bar{y} = \frac{1}{k} \sum_{t=1}^{k} y(t) \tag{B.7}
\]

The estimate of \( \mathbf{C}_{\tilde{x}\tilde{x}} \) must be constrained to be positive semi-definite in order for it to be a valid covariance matrix. This constraint is equivalent to requiring \( \mathbf{C}_{\tilde{x}\tilde{x}} = \mathbf{P}\Gamma\mathbf{P}^T \), where \( \mathbf{P} \) is an orthogonal matrix and \( \Gamma \) is a diagonal matrix with non-negative entries. One can obtain the positive semi-definite ML estimate of \( \mathbf{C}_{\tilde{x}\tilde{x}} \) by finding the ML estimates of the factors \( \Gamma \) and \( \mathbf{P} \). First rewrite equation B.2 in terms of \( \Gamma \) and \( \mathbf{P} \);

\[
\mathcal{L} \left( \mathbf{C}_{nn}^{-1/2} \tilde{y}, \Gamma, \mathbf{P}; \tilde{Y}^{(k)} \right) = \text{constant} - \frac{k}{2} \sum_{t=1}^{N} \left( \log |\gamma_t + 1| + \frac{\mathbf{P}_{*,i}^T \hat{\mathbf{C}}_{\tilde{y}\tilde{y}} \mathbf{P}_{*,i}}{\gamma_t + 1} \right) \tag{B.8}
\]
where \( \mathbf{P}_{*,i} \) is the \( i \)th column of \( \mathbf{P} \), \( \gamma_t \) is the \( i \)th diagonal entry in \( \Gamma \), and \( \hat{\mathbf{C}}_{\tilde{y}\tilde{y}} \) is the estimate

222
of the covariance of $\tilde{y}$ given by

$$
\hat{C}_{\tilde{y}\tilde{y}} = \frac{1}{k} \sum_{t=1}^{k} \left( \tilde{y}(t) - C_{n}^{-1/2}\tilde{y} \right) \left( \tilde{y}(t) - C_{n}^{-1/2}\tilde{y} \right)^T.
$$

(B.9)

The Kuhn-Tucker conditions applied to equation B.8 give necessary conditions for an optimal non-negative $\gamma_i$:

$$
\gamma_i > 0 \quad \text{iff} \quad \partial L / \partial \gamma_i = 0, \quad \gamma_i = 0 \quad \text{iff} \quad \partial L / \partial \gamma_i \leq 0;
$$

(B.10)

where

$$
\frac{\partial L}{\partial \gamma_i} = \frac{k}{2} \left[ P_{*,i}^T \hat{C}_{\tilde{y}\tilde{y}} P_{*,i} - \frac{1}{(\gamma_i + 1)^2} - \frac{1}{\gamma_i + 1} \right].
$$

(B.11)

Since the partial derivative is zero only at one point, the optimal $\gamma_i$, if it exists, is given by

$$
\hat{\gamma}_i = \begin{cases} 
0, & P_{*,i}^T \hat{C}_{\tilde{y}\tilde{y}} P_{*,i} \leq 1; \\
\frac{1}{\gamma_i + 1} P_{*,i}^T \hat{C}_{\tilde{y}\tilde{y}} P_{*,i} - 1, & \text{otherwise}.
\end{cases}
$$

(B.12)

That this is indeed the optimal solution can be verified by checking that $\hat{\gamma}_i > 0$ implies $\partial^2 L / \partial \gamma_i^2 |_{\gamma_i = \hat{\gamma}_i} < 0$, and $\hat{\gamma}_i = 0$ implies $\partial L / \partial \gamma_i < 0$ for $\gamma_i > 0$.

It remains to obtain the orthogonal matrix $P$ that maximizes the likelihood. From equation B.8 the matrix $P$ must solve the following problem:

$$
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} \frac{1}{\gamma_i + 1} P_{*,i}^T \hat{C}_{\tilde{y}\tilde{y}} P_{*,i} \\
\text{subject to} & \quad P_{*,i}^T P_{*,j} = \begin{cases} 
1, & i = j; \\
0, & \text{else}.
\end{cases}
\end{align*}
$$

(B.13)

It is well known that the solution to this problem is an orthogonal matrix of eigenvectors of $\hat{C}_{\tilde{y}\tilde{y}}$ with the columns (the eigenvectors) ordered such that the $i^{th}$ largest eigenvalue is matched with the $i^{th}$ smallest of the coefficients $1 / (\gamma_j + 1)$, $1 \leq j \leq N$. Let $\lambda_i$ be the eigenvalue corresponding to $P_{*,i}$, then

$$
P_{*,i}^T \hat{C}_{\tilde{y}\tilde{y}} P_{*,i} = \lambda_i.
$$

(B.14)

The combination of equations B.12 and B.14 gives $\hat{\gamma}_i = \max (\lambda_i - 1, 0)$. This completes the specification of the ML estimates of $\Gamma$ and $P$. 

223
Finally, the maximum likelihood estimate of $\mathbf{C}_{xx}$ is given by

$$\hat{\mathbf{C}}_{xx}^{ML} = \mathbf{C}_{nn}^{1/2} \mathbf{P} \max (\Lambda - \mathbf{I}, 0) \mathbf{P}^T \left( \mathbf{C}_{nn}^{1/2} \right)^T,$$

where $\mathbf{P}$ and $\Lambda$ are the matrices of eigenvectors and eigenvalues, respectively, of $\hat{\mathbf{C}}_{yy}$. Thus, the ML estimate of the signal covariance can be obtained by “noise-adjusting” the data (using $\mathbf{C}_{nn}^{1/2}$), computing the covariance estimate given by equation B.9, subtracting unity from the eigenvalues, truncating any negative eigenvalues to zero, and transforming the resulting matrix to the original space (using $\mathbf{C}_{nn}^{1/2}$).
Appendix C

Proof of Proposition 3.2

Proposition Let $y$ be the noisy source in Figure 3-1, where the signal $x$ and the noise $n$ are independent Gaussian random vectors. Let $m_x$ and $C_{xx}$ denote the mean and covariance of $x$, and $m_y$ and $C_{yy}$ denote the mean and covariance of $y$. Also let $D_W$ denote the expected distortion of the optimal transform codebook for a given weight matrix $W$, and a given quantizer level allocation $\{L_i\}_{i=1}^N$. Finally, for $k \in \{1, 2, \ldots\}$ let $\hat{m}_x^{(k)}$, $\hat{m}_y^{(k)}$, $\hat{C}_{xx}^{(k)}$, and $\hat{C}_{yy}^{(k)}$ be estimates of the mean vectors and covariance matrices of the signal and the noisy source, respectively, and denote by $D_W^{(k)}$ the expected distortion of the optimal codebook designed for the same quantizer level allocation $\{L_i\}_{i=1}^N$, and assuming the estimates above are the true parameters. If the estimates above are consistent, then $D_W^{(k)}$ converges to $D_W$ with probability one.

Proof: The proof is not complicated but it is rather lengthy. During the proof we shall work with a sequence of estimates of $\hat{m}_x^{(k)}$, $\hat{m}_y^{(k)}$, $\hat{C}_{xx}^{(k)}$, and $\hat{C}_{yy}^{(k)}$ which converges to the true parameters. We shall prove that for such a sequence of estimates the distortion of the corresponding sequence of empirical codebooks converges to the distortion of the optimal codebook. Since a convergent sequence of estimates occurs with probability one, by assumption, the distortion also converges with probability one. The proof assumes that $\hat{C}_{xx}^{(k)}$ and $\hat{C}_{yy}^{(k)}$ are positive semi-definite, and $C_{yy}$ is positive definite.

Let us begin by reproducing here, for clarity, the definitions of the optimal and empirical codebooks. The optimal codebook is described by the MMSEE matrix $L_{xy}$, an encoder matrix $H$, a decoder matrix $G$, and $N$ Lloyd-Max quantizers denoted as a vector function $q(\cdot)$. For a given weight matrix $W$ and a quantizer level allocation $\{L_i\}_{i=1}^N$ these components
are obtained as follows:

a) \( L_{xy} \) solves \( L_{xy}C_{yy} = C_{xx} \);

b) \( G = W^{-1}U \) and \( H = U^TWL_{xy} \), where \( U \) is a matrix of eigenvectors of \( WL_{xy}C_{yy}L_{xy}^TW^T \);

c) \( q_i(\cdot) \) is the \( L_i \)-level Lloyd-Max quantizer for a zero-mean Gaussian distribution with variance \( \lambda_i \), where \( \lambda_i \) is the \( i^{th} \) largest eigenvalue of \( WL_{xy}C_{yy}L_{xy}^TW^T \).

The output \( \tilde{x}^* \) of the optimal codebook is given by

\[
\tilde{x}^* = Gq(H(y - m_y)) + m_x, \quad (C.1)
\]

and in Section 3.3.1 the optimal distortion was shown to be given by

\[
D_W^* = E\left\{\|W(x - \tilde{x}^*)\|^2 \right\} + \sum_{i=1}^{N} \lambda_i d(L_i), \quad (C.2)
\]

where

\[
\tilde{x}^* = L_{xy}(y - m_y) + m_x \quad (C.3)
\]

is the MMSEE of \( x \) given \( y \), and \( d(\cdot) \) is the distortion of the Lloyd-Max quantizer for a unit-variance Gaussian input. Similarly, the empirical codebook based on the estimates \( \hat{m}_x^{(k)}, \hat{m}_y^{(k)}, \hat{C}_{xx}^{(k)}, \) and \( \hat{C}_{yy}^{(k)} \) consists of an analogous set of components \( \hat{L}_{xy}^{(k)}, \hat{H}^{(k)}, \hat{G}^{(k)}, \) and \( \hat{q}_i^{(k)}(\cdot) \) obtained as follows:

a) \( \hat{L}_{xy}^{(k)} \) solves \( \hat{L}_{xy}^{(k)}\hat{C}_{yy}^{(k)} = \hat{C}_{xx}^{(k)} \);

b) \( \hat{G}^{(k)} = W^{-1}U^{(k)} \) and \( \hat{H}^{(k)} = \left(U^{(k)}\right)^TW\hat{L}_{xy}^{(k)} \), where \( U^{(k)} \) is a matrix of eigenvectors of \( W\hat{L}_{xy}^{(k)}\hat{C}_{yy}^{(k)}\left(\hat{L}_{xy}^{(k)}\right)^TW^T \);

c) \( \hat{q}_i^{(k)}(\cdot) \) is the \( L_i^{(k)} \)-level Lloyd-Max quantizer for a zero-mean Gaussian distribution with variance \( \lambda_i^{(k)} \), where \( \lambda_i^{(k)} \) is the \( i^{th} \) largest eigenvalue of \( W\hat{L}_{xy}^{(k)}\hat{C}_{yy}^{(k)}\left(\hat{L}_{xy}^{(k)}\right)^TW^T \).

The output \( \tilde{x}^{(k)} \) of the codebook is given by

\[
\tilde{x}^{(k)} = G^{(k)}\hat{q}_i^{(k)}(H^{(k)}(y - \hat{m}_y^{(k)})) + \hat{m}_x^{(k)}, \quad (C.4)
\]
and the distortion can be written as

\[
D^{(k)}_W = E \left\{ \| W (x - \hat{x}^*) \|^2 \right\} + E \left\{ \| W (\hat{x}^* - \hat{x}^{(k)}) \|^2 \right\}. \tag{C.5}
\]

In addition, to prove the proposition it is useful to denote as \( \hat{x}^{(k)} \) the approximation of \( \hat{x}^* \) obtained from the empirical codebook;

\[
\hat{x}^{(k)} = \hat{L}_{xy}^{(k)} (y - \hat{m}_y^{(k)}) + \hat{m}_x^{(k)}. \tag{C.6}
\]

Comparing \( D_W \) and \( D^{(k)}_W \), the proposition is true if and only if

\[
\tilde{D}^{(k)}_W \overset{\Delta}{=} E \left\{ \| W (\hat{x}^* - \hat{x}^{(k)}) \|^2 \right\} \rightarrow \sum_{i=1}^{N} \lambda_i d(L_i) \tag{C.7}
\]

as \( k \rightarrow \infty \). This is proved below.

For clarity, we divide the proof into small numbered portions. We start by referring to two well-known results.

1. For any vector \( \mathbf{v} \) it is known that

\[
\| W \mathbf{v} \| \leq \| W \| \| \mathbf{v} \|, \tag{C.8}
\]

where \( \| W \| \) equals the square root of the maximum eigenvalue of \( W^T W \) [38].

2. For any two random vectors \( \mathbf{v} \) and \( \mathbf{u} \)

\[
\left| E \{ \mathbf{v}^T W^T W \mathbf{u} \} \right| \leq \sqrt{E \{ \| W \mathbf{v} \|^2 \} \cdot E \{ \| W \mathbf{u} \|^2 \}}, \tag{C.9}
\]

\[
\leq \| W \|^2 \sqrt{E \{ \| \mathbf{v} \|^2 \} \cdot E \{ \| \mathbf{u} \|^2 \}}, \tag{C.10}
\]

where C.9 is the Schwartz inequality, and C.10 follows from C.8.

3. Applying C.10 with \( \mathbf{v} = W (\hat{x}^{(k)} - \hat{x}^{(k)}) \) and \( \mathbf{u} = W (\hat{x}^* - \hat{x}^{(k)}) \) to \( \tilde{D}^{(k)}_W \) in C.7 one
obtains

\[ \tilde{D}_{W}^{(k)} \leq \|W\|^2 E\left\{ \left\| (\hat{x}^* - \hat{x}^{(k)}) \right\|^2 \right\} + \\
2 \|W\|^2 \sqrt{E\left\{ \left\| (\hat{x}^* - \hat{x}^{(k)}) \right\|^2 \right\} E\left\{ \left\| (\hat{x}^{(k)} - \tilde{x}^{(k)}) \right\|^2 \right\}} + \\
E\left\{ \|W\| (\hat{x}^{(k)} - \tilde{x}^{(k)}) \right\|^2 \right\}. \quad (C.11) \]

Next we want to show that the first two terms on the right-hand side of C.11 tend to zero as \( k \to \infty \).

4. First we show \( E\left\{ \left\| (\hat{x}^* - \hat{x}^{(k)}) \right\|^2 \right\} \) tends to zero. Using the definitions of \( \hat{x}^* \) and \( \hat{x}^{(k)} \) one can write

\[ E\left\{ \left\| (\hat{x}^* - \hat{x}^{(k)}) \right\|^2 \right\} = \text{tr} \left[ (L_{xy} - \tilde{L}_{xy}^{(k)}) C_{yy} \left( L_{xy} - \tilde{L}_{xy}^{(k)} \right)^T \right] + \\
\left\| \tilde{L}_{xy}^{(k)} (\tilde{m}_y^{(k)} - m_y) + m_x - \tilde{m}_x^{(k)} \right\|^2, \quad (C.12) \]

where \( \text{tr}[\cdot] \) computes the trace of a matrix. Thus, it is sufficient to show that \( \tilde{L}_{xy}^{(k)} \to L_{xy} \) as \( k \to \infty \). From the definitions of \( \tilde{L}_{xy}^{(k)} \) and \( L_{xy} \) one can write

\[ (L_{xy} - \tilde{L}_{xy}^{(k)}) \hat{C}_{yy}^{(k)} = C_{xx} - \hat{C}_{xx}^{(k)} + L_{xy} \left( \hat{C}_{yy}^{(k)} - C_{yy} \right). \quad (C.13) \]

Since \( \hat{C}_{yy}^{(k)} \) converges to \( C_{yy} \), which by assumption is positive definite, it follows that there exists \( \alpha > 0 \) such that

\[ \alpha \left\| (L_{xy} - \tilde{L}_{xy}^{(k)})_{i,*} \right\| \leq \left\| (L_{xy} - \tilde{L}_{xy}^{(k)})_{i,*} \hat{C}_{yy}^{(k)} \right\| \quad (C.14) \]

for large enough \( k \), where \((\cdot)_{i,*}\) denotes the \( i^{th} \) row of the corresponding matrix. Combining with equation C.13 one obtains

\[ \lim_{k \to \infty} \left\| (L_{xy} - \tilde{L}_{xy}^{(k)})_{i,*} \right\| \leq \lim_{k \to \infty} \frac{1}{\alpha} \left\| (C_{xx} - \hat{C}_{xx}^{(k)})_{i,*} + (L_{xy})_{i,*} \left( \hat{C}_{yy}^{(k)} - C_{yy} \right) \right\| = 0. \quad (C.15) \]

Then from equation C.12 it follows that \( E\left\{ \left\| (\hat{x}^* - \hat{x}^{(k)}) \right\|^2 \right\} \to 0 \) as \( k \to \infty \).
5. From the equation C.11 and the previous result it follows that for any \( \epsilon > 0 \)

\[
\hat{D}_W^{(k)} \leq E\left\{ \left\| W \left( \hat{x}^{(k)} - \tilde{x}^{(k)} \right) \right\|^2 \right\} + \epsilon 
\]

for large enough \( k \) provided that \( E\left\{ \left\| \left( \hat{x}^{(k)} - \tilde{x}^{(k)} \right) \right\|^2 \right\} \) is bounded above. Below we show that \( E\left\{ \left\| W \left( \hat{x}^{(k)} - \tilde{x}^{(k)} \right) \right\|^2 \right\} \) converges, which implies that \( E\left\{ \left\| \left( \hat{x}^{(k)} - \tilde{x}^{(k)} \right) \right\|^2 \right\} \) is bounded above for \( k \) large enough.

6. Putting \( z^{(k)} = H^{(k)} \left( y - \hat{m}_y^{(k)} \right) \) and performing simple linear algebra one can show

\[
E\left\{ \left\| W \left( \hat{x}^{(k)} - \tilde{x}^{(k)} \right) \right\|^2 \right\} = \sum_{i=1}^{N} E\left\{ \left( z_i^{(k)} - \hat{q}_i^{(k)} \left( z_i^{(k)} \right) \right)^2 \right\}. 
\]

(C.17)

To prove the proposition we need to show that

\[
E\left\{ \left( z_i^{(k)} - \hat{q}_i^{(k)} \left( z_i^{(k)} \right) \right)^2 \right\} \to \lambda_i d(L_i)
\]

(C.18)

as \( k \to \infty \). If this is true, then from equation C.16

\[
\hat{D}_W^{(k)} \leq \sum_{i=1}^{N} \lambda_i d(L_i) + \epsilon
\]

(C.19)

for arbitrarily small \( \epsilon > 0 \) and \( k \) large enough. Since \( \hat{D}_W^{(k)} \) could not be less than the optimal coding distortion \( \sum_{i=1}^{N} \lambda_i d(L_i) \) one must conclude that equation C.7 holds, which is what we had to prove.

7. We have to prove C.18. Note that \( z_i^{(k)} \) is Gaussian with mean \( H_i^{(k)} \left( m_y - \hat{m}_y^{(k)} \right) \) and variance \( H_i^{(k)} C_{yy} \left( H_i^{(k)} \right)^T \), while the quantizer \( \hat{q}_i^{(k)} \) is optimal for a zero-mean Gaussian with variance \( \lambda_i^{(k)} \). Let \( q_i^{(k)} \) be the optimal quantizer for \( z_i^{(k)} \). Writing the expectation in C.18 in terms of the errors

\[
z_i^{(k)} - q_i^{(k)} \left( z_i^{(k)} \right) \text{ and } \hat{q}_i^{(k)} \left( z_i^{(k)} \right) - q_i^{(k)} \left( z_i^{(k)} \right),
\]

(C.20)
and applying the Schwartz inequality \[38\] one obtains

\[
E\left\{ \left( z_i^{(k)} - \hat{q}_i^{(k)} (z_i^{(k)}) \right)^2 \right\} \leq E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} + 2 \sqrt{E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} E\left\{ \left( \hat{q}_i^{(k)} (z_i^{(k)}) - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} } + E\left\{ \left( \hat{q}_i^{(k)} (z_i^{(k)}) - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\},
\]  

(C.21)

where

\[
E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} = H_{i,*}^{(k)} C_{yy} \left( H_{i,*}^{(k)} \right)^T d(L_i).
\]  

(C.22)

Since \( q_i^{(k)} \) is optimal for \( z_i^{(k)} \) it must also be true that

\[
E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} \leq E\left\{ \left( z_i^{(k)} - \hat{q}_i^{(k)} (z_i^{(k)}) \right)^2 \right\},
\]  

(C.23)

Thus, if the last term in equation C.21 tends to zero, we have that

\[
E\left\{ \left( z_i^{(k)} - \hat{q}_i^{(k)} (z_i^{(k)}) \right)^2 \right\} - E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} \rightarrow 0
\]  

(C.24)

as \( k \rightarrow \infty \). For now we assume that this is the case.

Recall that \( \lambda_i \) and \( \lambda_i^{(k)} \) are the \( i \)-th largest eigenvalues of \( W L_{xy} C_{yy} (L_{xy})^T W^T \) and \( W L_{xy} C_{yy} (\hat{L}_{xy}^{(k)})^T W^T \), respectively. The second matrix converges to the first one and it follows that \( \lambda_i^{(k)} \rightarrow \lambda_i \) as \( k \rightarrow \infty \). In addition, note that

\[
H_{i,*}^{(k)} C_{yy} \left( H_{i,*}^{(k)} \right)^T - \lambda_i^{(k)} = H_{i,*}^{(k)} \left( C_{yy} - \hat{C}_{yy}^{(k)} \right) \left( H_{i,*}^{(k)} \right)^T \rightarrow 0
\]  

(C.25)

as \( k \rightarrow \infty \). Therefore, from equation C.22

\[
E\left\{ \left( z_i^{(k)} - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} \rightarrow \lambda_i d(L_i)
\]  

(C.26)

as \( k \rightarrow \infty \). This proves the proposition if C.24 holds.

8. To prove C.24 in light of C.21 and C.23, it suffices to show that

\[
E\left\{ \left( \hat{q}_i^{(k)} (z_i^{(k)}) - q_i^{(k)} (z_i^{(k)}) \right)^2 \right\} \rightarrow 0
\]  

(C.27)
as $k \to \infty$. This is proved below.

To simplify the notation let $\mu^{(k)} = H_{i,*}^{(k)} (m_y - \hat{m}_y^{(k)})$ and $\nu^{(k)} = H_{i,*}^{(k)} C_{yy} (H_{i,*}^{(k)})^T$.

Then $z_i^{(k)}$ has distribution $\mathcal{N} (\mu^{(k)}, \nu^{(k)})$. Recall that $q_i^{(k)}$ is the Lloyd-Max quantizer for that distribution, and $\hat{q}_i^{(k)}$ is the Lloyd-Max quantizer for $\mathcal{N} (0, \lambda_i^{(k)})$.

Let $\{r(j)\}_{j=1}^{L_i}$ be the quantization (or reconstruction) levels and $\{l(j)\}_{j=1}^{L_i-1}$ be the decision levels of the Lloyd-Max quantizer for the distribution $\mathcal{N} (0, \lambda_i)$. Also, let $\{r^{(k)}(j)\}_{j=1}^{L_i}$ be the quantization levels and $\{l^{(k)}(j)\}_{j=1}^{L_i-1}$ be the decision levels of $q_i^{(k)}$, and let $\{\hat{r}^{(k)}(j)\}_{j=1}^{L_i}$ be the quantization levels and $\{\hat{l}^{(k)}(j)\}_{j=1}^{L_i-1}$ be the decision levels of $\hat{q}_i^{(k)}$. Because all three quantizers are optimal for the indicated distributions, the following relationships hold:

$$r^{(k)}(j) = \mu^{(k)} + \sqrt{\frac{\nu^{(k)}}{\lambda_i}} r(j), \quad \hat{r}^{(k)}(j) = \sqrt{\frac{\lambda_i^{(k)}}{\lambda_i}} r(j), \quad (C.28)$$

for $1 \leq j \leq L_i$, and

$$l^{(k)}(j) = \mu^{(k)} + \sqrt{\frac{\nu^{(k)}}{\lambda_i}} l(j), \quad \hat{l}^{(k)}(j) = \sqrt{\frac{\lambda_i^{(k)}}{\lambda_i}} l(j), \quad (C.29)$$

for $1 \leq j < L_i$. Since $\mu^{(k)} \to 0$, $\nu^{(k)} \to \lambda_i$, and $\lambda_i^{(k)} \to \lambda_i$ as $k \to \infty$, it is clear that $r^{(k)}(j) \to r(j)$, $\hat{r}^{(k)}(j) \to r(j)$, $l^{(k)}(j) \to l(j)$, and $\hat{l}^{(k)}(j) \to l(j)$.

Assume without loss of generality that that $l(1) < l(2) \cdots < l(L_i - 1)$. Since $\hat{l}^{(k)}(j) \to l(j)$ and $l^{(k)}(j) \to l(j)$ as $k \to \infty$, there exists a positive integer $K$ such that

$$\max \left( \hat{l}^{(k)}(j-1), l^{(k)}(j-1) \right) < \min \left( \hat{l}^{(k)}(j), l^{(k)}(j) \right)$$

for $2 \leq j < L_i$ and all $k > K$. For $k > K$ define the intervals

$$S^{(k)}(j) = \left[ \max \left( \hat{l}^{(k)}(j-1), l^{(k)}(j-1) \right), \min \left( \hat{l}^{(k)}(j), l^{(k)}(j) \right) \right] \quad (C.30)$$

for $2 \leq j < L_i$,

$$S^{(k)}(1) = \left( -\infty, \min \left( \hat{l}^{(k)}(1), l^{(k)}(1) \right) \right], \quad (C.31)$$

for $2 \leq j < L_i$,
\[ S^{(k)}(L_i) = \left[ \max \left( \tilde{\ell}^{(k)}(L_i - 1), \ell^{(k)}(L_i - 1) \right), \infty \right), \]  \hspace{1cm} (C.33) 

and

\[ s^{(k)}(j) = \left[ \min \left( \tilde{\ell}^{(k)}(j), \ell^{(k)}(j) \right), \max \left( \tilde{\ell}^{(k)}(j), \ell^{(k)}(j) \right) \right] \]  \hspace{1cm} (C.34) 

for \( 1 \leq j < L_i \). Note that for \( z_i^{(k)} \in S^{(k)}(j) \)

\[ \hat{q}_{i}^{(k)}(z_i^{(k)}) - q_{i}^{(k)}(z_i^{(k)}) = \tilde{r}^{(k)}(j) - r^{(k)}(j), \]  \hspace{1cm} (C.35) 

and for \( z_i^{(k)} \in s^{(k)}(j) \)

\[ \left| \hat{q}_{i}^{(k)}(z_i^{(k)}) - q_{i}^{(k)}(z_i^{(k)}) \right| 
\leq \max \left( \left| \tilde{r}^{(k)}(j + 1) - r^{(k)}(j) \right|, \left| \tilde{r}^{(k)}(j) - r^{(k)}(j + 1) \right| \right), \]  \hspace{1cm} (C.36) 

\[ \leq 2 \left( r(L_i) - r(1) \right), \]  \hspace{1cm} (C.37) 

for \( k \) large enough. Thus, one can write

\[ E \left\{ \left( \hat{q}_{i}^{(k)}(z_i^{(k)}) - q_{i}^{(k)}(z_i^{(k)}) \right)^2 \right\} \leq \sum_{j=1}^{L_i} \Pr \{ z_i^{(k)} \in S^{(k)}(j) \} \left( \tilde{r}^{(k)}(j) - r^{(k)}(j) \right)^2 + \sum_{j=1}^{L_i-1} \Pr \{ z_i^{(k)} \in s^{(k)}(j) \} 4 \left( r(L_i) - r(1) \right)^2 \]  \hspace{1cm} (C.38) 

for \( k \) large enough.

Finally, for any \( \epsilon > 0 \) there exists \( K' > K \) such that

\[ \left( \tilde{r}^{(k)}(j) - r^{(k)}(j) \right)^2 < \epsilon \]  \hspace{1cm} (C.39) 

and

\[ \Pr \left\{ z_i^{(k)} \in s^{(k)}(j) \right\} < \epsilon L_i^{-1} \left( r(L_i) - r(1) \right)^{-2} \]  \hspace{1cm} (C.40) 

for \( k > K' \), and therefore

\[ E \left\{ \left( \hat{q}_{i}^{(k)}(z_i^{(k)}) - q_{i}^{(k)}(z_i^{(k)}) \right)^2 \right\} \leq 5\epsilon. \]  \hspace{1cm} (C.41) 

232
Appendix D

Examples of Gain and Offset Covariances Estimated from AIRS EM Test Data

The following are examples of estimates of the joint covariance sequence of gain and offset for the AIRS instrument. The estimates were obtained using preliminary test data from the AIRS Engineering Model (EM). The procedure used to obtain the estimates is described in Sections 4.4 and 4.5.2.
Figure D-1: Estimate of the joint covariance sequence of the gain and offset on channel 42. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Figure D-2: Estimate of the joint covariance sequence of the gain and offset on channel 140. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Figure D-3: Estimate of the joint covariance sequence of the gain and offset on channel 815. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Figure D-4: Estimate of the joint covariance sequence of the gain and offset on channel 1042. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Figure D-5: Estimate of the joint covariance sequence of the gain and offset on channel 2089. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Figure D-6: Estimate of the joint covariance sequence of the gain and offset on channel 2253. The graphs on the left show the fit to the sample covariance sequence of the calibration data. The estimates of the joint covariance of the gain and offset shown on the right are obtained by a linear transformation of the fit.
Bibliography


[51] Adam J. Efron, Peter F. Swaszek, and Donald W. Tufts. Insight into detection of
deterministic and gaussian signals in correlated plus impulsive noise environments.

[52] M. Shao and C. L. Nikias. Signal processing with fractional lower order moments:

[53] Jun Shen and Chrysostomos L. Nikias. Estimation of multilevel digital signals in the
presence of arbitrary impulsive interference. IEEE Transactions on Signal Processing,

the EM algorithm. IEEE Transactions on Acoustics, Speech, and Signal Processing,