1

PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395 rue Wellington
Ottawa (Ontario)
K1A 0N4

Produced by the National Library of Canada

Produit par la Bibliothèque nationale du Canada

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

SPEECH CODING AT LOW TO MEDIUM BIT RATES

by

Wilfrid P. LeBlanc

*A thesis submitted to the faculty
of Graduate Studies and Research in
partial fulfillment of the requirements
for the degree of*

Doctor of Philosophy

***Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada***

September 1st, 1992

© copyright

1992, Wilfrid P. LeBlanc

**National Library
of Canada**

**Acquisitions and
Bibliographic Services Branch**

**395 Wellington Street
Ottawa, Ontario
K1A 0N4**

**Bibliothèque nationale
du Canada**

**Direction des acquisitions et
des services bibliographiques**

**395, rue Wellington
Ottawa (Ontario)
K1A 0N4**

Author's name

Title

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-84012-0

Canada

*The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis.*

**SPEECH CODING AT LOW
TO MEDIUM BIT RATES**

submitted by

Wilfrid P. LeBlanc, B.A.Sc., M. Eng.

*in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

***Chair, Department of Systems and
Computer Engineering***

Thesis Supervisor

External Examiner

Carleton University

November 30th, 1992

Abstract

The research reported herein is concerned with the development of high quality speech coding algorithms at bit rates between 4 and 8 kb/s.

In the literature, analysis by synthesis coding based on a weighted mean squared error distortion criteria has been demonstrated to work well at rates as low as 4-16 kb/s. In this research, improved search techniques coupled with improved codebook design methodologies are proposed to improve the performance of conventional code-excited linear predictive coders.

Improved methods for quantizing the short term filter are developed by applying a tree search algorithm and joint codebook design to multi-stage vector quantization. Joint codebook design procedures are developed to design locally optimal multi-stage codebooks. Weighting during centroid computation is introduced in order to improve the outlier performance of the multi-stage vector quantizer. Multi-stage vector quantization is shown to be both robust against input characteristics and in the presence of channel errors. Spectral distortions of approximately 1 dB are obtained at rates between 22 and 28 bits/frame for various multi-stage configurations.

Structured codebook design procedures for the excitation in code-excited linear predictive coders are compared to general codebook design procedures. It is determined that little is lost in using significant structure in the excitation codebooks, while greatly reducing the search complexity. Sparse multi-stage configurations are proposed as a means of reducing computational complexity and memory size while offering both a variable rate and variable complexity.

Improved search procedures applied to code-excited linear prediction which attempt to jointly optimize the short term filter, the adaptive codebook, and the excitation are proposed. Although the computational complexity is rather large, segmental signal to noise ratio improvements between 1 and 2 dB are realized in practice.

Acknowledgments

I would like to thank Professor S.A. Mahmoud for all his guidance and assistance during the course of this thesis. His suggestions, patience and comments were always greatly appreciated. I would also like to thank Professor Vladimir Cuperman from Simon Fraser University for his comments and suggestions during the course of this thesis.

I am also deeply indebted to Peter Leach and the Telecommunication Institute of Ontario (TRIO) for making this thesis financially possible.

Contents

<i>Abstract</i>	iii
<i>Acknowledgments</i>	iv
<i>List of Figures</i>	ix
<i>List of Tables</i>	xiv
<i>List of Symbols</i>	xvii
<i>List of Acronyms</i>	xvii
<i>Chapter 1</i>	<i>Introduction</i> 1
<i>Section 1.1</i>	<i>Introduction to Speech Coding</i> 1
<i>Section 1.2</i>	<i>Thesis Objectives and Contributions</i> 3
<i>Section 1.3</i>	<i>Thesis Organization</i> 3
<i>Chapter 2</i>	<i>Literature Review</i> 5
<i>Section 2.1</i>	<i>Rate-Distortion Theory</i> 5
<i>Section 2.1.1</i>	<i>Introduction to Rate-Distortion Theory</i> 6
<i>Section 2.1.2</i>	<i>Blahut's Algorithm for Computing $R(D)$</i> 10
<i>Section 2.1.3</i>	<i>Finamore-Pearlman Algorithm for Continuous Sources</i> 13
<i>Section 2.2</i>	<i>Lloyd-Max Scalar Quantization</i> 16
<i>Section 2.3</i>	<i>Vector Quantization</i> 19
<i>Section 2.3.1</i>	<i>The Generalized Lloyd Algorithm</i> 21
<i>Section 2.3.1.1</i>	<i>GLA With Known PDF</i> 22
<i>Section 2.3.1.2</i>	<i>GLA Based on a Training Sequence</i> 23
<i>Section 2.3.2</i>	<i>Structured VQ</i> 26
<i>Section 2.4</i>	<i>Linear Prediction</i> 26
<i>Section 2.4.1</i>	<i>Application of Linear Prediction to Speech Coding</i> 26
<i>Section 2.4.2</i>	<i>Block Algorithms- Basic Formulation and Solution</i> 33
<i>Section 2.4.3</i>	<i>The Likelihood Ratio</i> 36
<i>Section 2.4.4</i>	<i>Spectral Distortion</i> 37
<i>Section 2.4.5</i>	<i>Long Term Model Filter</i> 39
<i>Section 2.4.6</i>	<i>Speech Production Model</i> 41

<i>Section 2.5</i>	<i>Codebook Excited Linear Prediction (CELP)</i>	<i>41</i>
<i>Section 2.5.1</i>	<i>CELP Analysis/Synthesis Model</i>	<i>41</i>
<i>Section 2.5.2</i>	<i>Reduced Complexity Search Techniques</i>	<i>45</i>
<i>Section 2.5.2.1</i>	<i>Adaptive Codebook Search Strategies</i>	<i>52</i>
<i>Section 2.5.2.2</i>	<i>Stochastic Codebook Search Strategies</i>	<i>56</i>
<i>Section 2.6</i>	<i>Multi-Pulse Excited Linear Predictive Coding (MPE)</i>	<i>66</i>
<i>Section 2.6.1</i>	<i>The Multi-pulse Excitation Model</i>	<i>66</i>
<i>Section 2.6.2</i>	<i>Multi-pulse Sequential Search Procedure</i>	<i>67</i>
<i>Section 2.6.3</i>	<i>Multi-pulse Tree Searched Procedure</i>	<i>69</i>
<i>Section 2.6.4</i>	<i>Multi-pulse Simulated Annealing Algorithm</i>	<i>69</i>
<i>Section 2.6.5</i>	<i>Constraining the Multi-pulse Excitation</i>	<i>72</i>
<i>Chapter 3</i>	<i>Multi-stage Vector Quantization of the Short Term Filter</i>	<i>74</i>
<i>Section 3.1</i>	<i>Search Procedure</i>	<i>79</i>
<i>Section 3.2</i>	<i>Distortion Measures and Search Complexity</i>	<i>85</i>
<i>Section 3.3</i>	<i>Codebook Design</i>	<i>89</i>
<i>Section 3.3.1</i>	<i>Sequential and Iterative Sequential Codebook Design</i>	<i>90</i>
<i>Section 3.3.2</i>	<i>Simultaneous Joint Codebook Design</i>	<i>92</i>
<i>Section 3.3.3</i>	<i>Convergence Properties</i>	<i>99</i>
<i>Section 3.4</i>	<i>Multi-stage VQ of Gaussian Sources</i>	<i>100</i>
<i>Section 3.5</i>	<i>LPC Performance of Codebook Design Methodologies</i>	<i>106</i>
<i>Section 3.6</i>	<i>Outlier Weighting</i>	<i>109</i>
<i>Section 3.7</i>	<i>Comparison of Different Parameter Sets and Distortion Measures</i>	<i>112</i>
<i>Section 3.8</i>	<i>Rate, Performance, Complexity Trade-offs</i>	<i>118</i>
<i>Section 3.9</i>	<i>Robustness Issues</i>	<i>124</i>
<i>Section 3.9.1</i>	<i>Performance Across Languages and Input Spectral Shapes</i>	<i>124</i>
<i>Section 3.9.2</i>	<i>Performance in the Presence of Channel Errors</i>	<i>125</i>

Section 3.10	Subjective Evaluation	129
Section 3.11	Summary	130
Chapter 4	Excitation Codebook Design	133
Section 4.1	Rate-Distortion Bounds For Gaussian Excitation	133
Section 4.1.1	Multi-stage VQ (MSVQ)	133
Section 4.1.2	Ternary Codes	134
Section 4.2	The Design of General Codebooks	136
Section 4.3	The Design of Structured Codebooks	140
Section 4.4	The Design of Sparse Codebooks	143
Section 4.5	Codebook Search Complexities	143
Section 4.6	Performance of Codebook Design Methodologies	145
Section 4.6.1	Performance of Random Codebooks	146
Section 4.6.2	Performance of Designed Codebooks	152
Section 4.7	Summary	155
Chapter 5	Joint Search Procedures Applied to CELP	157
Section 5.1	Optimal CELP Search Procedure	157
Section 5.2	Iterative Closed Loop Optimization of the Short Term Predictor	162
Section 5.2.1	Unquantized Case	162
Section 5.2.2	Quantized Case	166
Section 5.3	Excitation Re-optimization	168
Section 5.4	Open Loop/ Closed Loop CELP Optimization	171
Section 5.5	Objective Results	173
Section 5.6	Subjective Results	181
Section 5.7	Summary	187
Chapter 6	Conclusions	189
Section 6.1	Contributions to Knowledge	190
Section 6.2	Suggestions for Future Work	191
Appendix A	Characteristics of the Downsampling Filter	193
Appendix B	A-B Preference Tests	194

Appendix C *Mean Opinion Score (MOS) Evaluations* 196

References 198

List of Figures

Figure 1	A typical rate-distortion function. $R(D)$ is not defined for $D \leq D_{\min}$. For $D \geq D_{\max}$, $R(D) = 0$	11
Figure 2	Source coding with pre-quantization. The expected distortion in the first stage (pre-quantization) is D_Q and the expected distortion in the second stage ($R(D)$ for finite alphabet source) is D_f	15
Figure 3	Performance of Lloyd-Max scalar quantization with and without optimal entropy coding for $L=3, 5, 9, 17, 33, 65, 129$, and 257	20
Figure 4	Convergence both inside and outside the training sequence. For a low number of training vectors, the performance on the training data (inside) is very good although the performance outside the training sequence is relatively poor. As the number of training vectors increases, the difference in performance outside and inside the training sequence approaches zero.	25
Figure 5	Scalar quantization in conjunction with linear prediction. The optimal sequence of indices (l_n) is chosen to minimize the distortion between x_n and $\hat{x}_n^{(l_n)}$	29
Figure 6	Delayed decision quantization in conjunction with a finite order linear predictor. The suboptimal finite length sequence of indices at time T $\{l_T, l_{T-1}, l_{T-P}\}$ is chosen to minimize the average distortion between x_n and $\hat{x}_n^{(l_n)}$ for $n = T - P$ to $n = T$. At time T the $T - P$ -th index is released.	31
Figure 7	Speech production model. $v_{n,k}$ is the k -th sample of the excitation from the n -th frame. G_n is the gain for the n -th frame. $e_{n,k}$ is the k -th sample of the (scaled) excitation at subframe n and $d_{n,k}$ is the k -th sample of the excitation after the long term model filter at subframe n . Finally, $s_{n,k}$ is the k -th sample of the speech at subframe n	42
Figure 8	Analysis-synthesis model. $e_{n,k}$ ($\hat{e}_{n,k}$) is the k -th sample of the (synthetic) excitation at subframe n and $\hat{d}_{n,k}$ ($d_{n,k}$) is the k -th sample of the (synthetic) excitation after the long term model filter at subframe n . Finally, $s_{n,k}$ ($\hat{s}_{n,k}$) is the k -th sample of the (synthetic) speech at subframe n	43

Figure 9	The analysis by synthesis nature of CELP. (a) Synthesis: A vector (selected by an index) is chosen from the VQ codebook, scaled, and filtered through a long and short term model filter. (b) Analysis: For each possible index from the VQ codebook, the synthetic speech is computed and compared based on a weighted mean squared error distortion. Thus the term: analysis by synthesis.	44
Figure 10	Frequency response of $\alpha_p/A(e^{j\omega})$, $\alpha_p/W(e^{j\omega})$, and $S(e^{j\omega})$. The weighting filter tends to boost the noise energy in regions where $S(e^{j\omega})$ is large. The conventional weighting filter has a global structure, but very little spectrally fine structure.	46
Figure 11	CELP structure rearranged. (1)– move location of $W_n(z)$ and (2)– separate the zero state and zero input responses.	47
Figure 12	Analysis by synthesis transform coder. $F(z)$ is the filterbank representation of F . Y_n , U_n , $\hat{Y}_n^{(m)}$, and $\hat{Y}_n^{(l)}$, are the filterbank outputs at time $K_c - 1$	59
Figure 13	Multi-stage VQ of the CELP excitation (MS-CELP) with T_c stages. It is assumed that there are a fixed number of codevectors (L_c) in each stage.	62
Figure 14	MPE tree search procedure. After the first stage, the two minimum WMSE pulse positions are determined (p_1 and p_2). In the second stage, the (p_1, p_2) combination is selected in branch 1-1 and need not be considered in branch 2-2. After the second stage, the $M_m (= 2)$ minimum WMSE sets of pulse positions are kept for subsequent stages. The M_m candidates after the second stage are chosen to minimize $\epsilon^{(p_1, p_2)}$	70
Figure 15	Multi-stage vector quantization (synthesis). The codebook indices (t_j) select vectors from the codebooks (C_j). Prediction of future vectors is based on only a single past quantized vector in this case.	76
Figure 16	M-L (tree) search procedure. The best paths are indicated by heavy lines.	84
Figure 17	MSVQ sequential design algorithm.	93
Figure 18	MSVQ iterative sequential design algorithm.	94
Figure 19	The design of initial random codebooks.	95
Figure 20	Re-ordering procedure used in simultaneous joint design.	101

Figure 21	<i>MSVQ simultaneous joint design algorithm</i>	102
Figure 22	<i>Performance of full search single stage VQ and multi-stage sequentially searched (10 dimensional) VQ. R(D) and Lloyd-Max scalar quantizers shown for reference purposes. Error bars correspond to 95% confidence limits. (i) Lloyd-Max scalar quantization. (ii) R(D). (iii) Unstructured VQ. (iv) Multi stage VQ.</i>	103
Figure 23	<i>Performance of tree searched multi-stage VQ, full searched multi-stage VQ, and (5-5) split VQ (64-2 code). Both 2048 level and 4096 level unstructured full searched codes are included for comparison purposes. (i) F-2048. (ii) F-4096. (iii) S-64-2. (iv) 64-2 designed sequentially with sequential search (v) 64-2 designed iterative sequentially with tree search during design the same as that for testing.</i>	105
Figure 24	<i>TS-MSVQ performance with jointly designed codes designed using various values of M_a during codebook design (i) $M_a = 1$, (ii) $M_a = 2$, (iii) $M_a = 8$. (iv) Full Search. The performance is plotted versus the value of M_a used in the codebook testing.</i>	108
Figure 25	<i>Convergence (a) inside the training sequence of (i) iterative sequential codebook design and (ii) Joint codebook design, and (b) outside the training sequence of (i) iterative sequential codebook design and (ii) Joint codebook design.</i>	110
Figure 26	<i>Performance vs complexity of the predictor parameters with the likelihood ratio for three codes: (i) 4096-2 (ii) 64-4 (iii) C-64-4. In (iii), the chosen second, third, and fourth codebooks are conditional on the index from the first codebook. Thus, there are 64 second, 64 third, and 64 fourth codebooks which are chosen based on the value of the first index.</i>	114
Figure 27	<i>Performance vs. complexity of LSPs with a weighted MSE distortion for (i) 4096-2 (ii) 64-4 (iii) SD-64-4. In curve (iii) the final selection is made based on SD rather than on the WMSE.</i>	115
Figure 28	<i>Performance vs. complexity of LARs with a MSE distortion for (i) 4096-2 (ii) LR-4096-2 (iii) 64-4 (iv) LR-64-4. In curves (ii) and (iv), the final stage select the optimal candidate (out of M_a) based on the likelihood ratio (rather than on MSE).</i>	116

Figure 29	TS-MSVQ performance versus complexity for 24 bits/frame (weighted LSP) (i) 16-6 (ii) 64-4 (iii) 256-3 (iv) 4096-2. The solid point is S-4096-2 (split VQ).	119
Figure 30	TS-MSVQ performance versus complexity for rates of 22-30 bits/frame. (i) 22 bits, 2048-2 (ii) 24 bits, 64-4 (iii) 26 bits, 256-64-3 (iv) 28 bits, 16-7 (v) 30 bits, 4-15. The solid point is S-4096-2.	121
Figure 31	TS-MSVQ performance versus complexity for rates of 26-36 bits/frame using the same A 2-36 code. (i) 26 bits, (ii) 28 bits, (iii) 30 bits, (iv) 32 bits, (v) 34 bits, (vi) 36 bits.	123
Figure 32	TS-MSVQ performance for different languages and input spectral shapings for (i) 64-4 and (ii) 2-27.	126
Figure 33	Preference for the unquantized case (A). The short term filter for stimulus B was quantized using TS-MSVQ (based on LSPs and one of the following structures: (i) 16-5 (ii) 16-6 (iii) 16-7.	131
Figure 34	Multi-stage quantization.	135
Figure 35	Rate-distortion function for white gaussian noise and performance bound on ternary codes. (129 level Lloyd-Max pre-quantizer).	137
Figure 36	$R(D)$ as a function of the optimal output probabilities ($P(y_0)$). At zero rate, the probability of the zero output is (of course) unity.	138
Figure 37	Codebook design for unstructured codebooks.	139
Figure 38	Codebook design for structured codebooks.	144
Figure 39	Performance (SNR_{SEG} and $WSNR_{SEG}$) of random codebooks for various codebook sizes (or rates) for the nominal 4 kb/s codec. Three codebook structures are considered: (i) Unstructured (ii) Overlapping (iii) Basis vector. The error bars are 95% confident intervals.	150
Figure 40	Performance (SNR_{SEG} and $WSNR_{SEG}$) of random codebooks for various codebook sizes (or rates) for the nominal 8 kb/s codec. Three codebook structures are considered: (i) Unstructured (ii) Overlapping (iii) Basis vector. The error bars are 95% confidence intervals.	151

Figure 41	Performance of M-L search MSVQ structured codebooks versus complexity for the reference 4 kb/s codec. (i) 16-3 ($\Delta = 10$, $N_s = 4$) (ii) 64-2 ($\Delta = 10$, $N_s = 4$). The points on the curves (from left to right) correspond to $M_c = 1, 2, 4, 8, 16, 64$	156
Figure 42	The CELP search procedure.	158
Figure 43	Closed loop optimization for the target excitation. $\epsilon_{w,c}$ is the closed loop model filter distortion measure.	163
Figure 44	Equation error optimization for a given synthetic excitation.	165
Figure 45	Performance of optimized (closed loop) system with unquantized short term filter ($A(z)$). (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents no closed loop re-optimization of $B(z)$ (from Table 27a) and curve (ii) represents closed loop optimization of $B(z)$ (from Table 27b). (0 iterations represents the conventional CELP codec. The error bars are computed with respect to the improvement over the conventional CELP codec).	176
Figure 46	Performance comparisons with optimized (closed loop), quantized $A(z)$ and unquantized $A(z)$ (from Tables 27b and 28a). (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents the unquantized case and curve (ii) represents the quantized case.	178
Figure 47	Performance with quantized, interpolated short term filter ($A(z)$). (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents $A(z)$ quantized (and interpolated open loop) (Table 29a) and curve (ii) represents $A(z)$ quantized (and interpolated) closed loop (Table 29b).	180
Figure 48	Performance of OC-CELP as a function of M_o and M_b . Quantized, non-interpolated $A(z)$. (a) 4 kb/s (from Table 30a). (b) 8 kb/s (from Table 30b). (i) $M_b = 1$ (ii) $M_b = 2$ (iii) $M_b = 4$ (iv) $M_b = 16$	183
Figure 49	Preference for VSEL P vs. OC-CELP at various rates.	185
Figure 50	MOS performance of various codecs.	188
Figure 51	Frequency response of the downsampling filter.	193
Figure 52	Directions for the A-B preference tests.	195
Figure 53	Directions for the MOS (Mean Opinion Score) tests.	197

List of Tables

Table 1	<i>Parameters used for SD computation.</i>	37
Table 2	<i>Correspondence between the parameters of the general distortion measure and the parameters of the MSE, WMSE and LR.</i>	90
Table 3	<i>Performance of full search unstructured single stage VQ (F-), and multi-stage sequentially searched VQ.</i>	104
Table 4	<i>Performance of (M-L) tree searched multi-stage VQ (a) designed sequentially with a sequential search and (b) designed iteratively sequentially with a tree search during design (with M_a the same as during test). Full searched multi-stage VQ, and (5-5) split VQ (64-2 code) are shown for comparison purposes.</i>	104
Table 5	<i>Spectral distortion and outliers for various weightings (both inside and outside the training sequence). See (250) and (251).</i>	112
Table 6	<i>The performance (both inside and outside of the training sequence) of full search unstructured 4096 level codes based on the three parameter sets and distortion measures. Performance is measured in terms of average SD, percentage of outliers between 2 and 4 dB, and percentage of outliers above 4 dB.</i>	113
Table 7	<i>Different TS-MSVQ codes to obtain approximately equivalent average SD performance. The rate is given in bits/frame.</i>	122
Table 8	<i>Spectral distortion performance on different language and input spectral shapings. (a) German (b) Italian (c) Norwegian (d) Noisy speech database (e) TIMIT-test.</i>	124
Table 9	<i>Outlier performance (2-4 dB) on different language and input spectral shapings. (a) German (b) Italian (c) Norwegian (d) Noisy speech database (e) TIMIT-test.</i>	125
Table 10	<i>Average spectral distortion for various error rates and for various codes.</i>	127
Table 11	<i>Percentage of outliers within 2-4 dB and greater than 4 dB for various error rates and for various codes.</i>	128

Table 12	MOS results for the unquantized case, three codes at constant complexity (M45-8-6, M45-8-8, and M45-8-10), and three codes at constant rate/frame (M8-512-2, M45-8-6, and M25-2-18).	129
Table 13	Preference for the unquantized case (A, which obtained an SNR_{SEG} of 7.87 dB). The short term filter for stimulus B was quantized using TS-MSVQ based on LSPs and one of the following structures: (i) 16-5 (ii) 16-6 (iii) 16-7. The error in the preference is within 7 percentage points (with 95 % confidence).	130
Table 14	Parameters of reference 4 kb/s codec.	147
Table 15	Parameters of reference 8 kb/s codec.	147
Table 16	Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of an unstructured random code at 7 bits/vector, and overlapping random codes with various degrees of overlap.	148
Table 17	The objective performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of random full search multi-stage codes at rates of 8 and 12 bits per subframe for the reference 4 kb/s codec.	149
Table 18	The objective performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of random full search multi-stage codes at rates of 8 and 12 bits per subframe for the reference 8 kb/s codec.	149
Table 19	Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of unstructured codebooks designed based on the GLA for various codebook sizes (for the reference 4 kb/s codec).	153
Table 20	Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of unstructured codebooks designed based on the GLA for various codebook sizes (for the reference 8 kb/s codec).	153
Table 21	Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of full search structured MSVQ codebooks designed based on the GLA for various codebook sizes and structures for the reference 4 kb/s codec.	154
Table 22	Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of full search structured MSVQ codebooks designed based on the GLA for various codebook sizes and structures for the reference 8 kb/s codec.	154
Table 23	Values of the parameters of the simulated annealing algorithm.	168

Table 24	<i>Parameters of the reference 4 and 8 kb/s CELP Codec. At 4 kb/s, $N = 280$ and $K_c = 70$, and at 8 kb/s $N = 140$ and $K_c = 35$. $\gamma_1 = 1$ and $\gamma_2 = 0.9$ at both 4 and 8 kb/s.</i>	<i>169</i>
Table 25	<i>Parameters of the reference 4 and 8 kb/s OC-CELP Codec. At 4 kb/s, $N = 280$ and $K_c = 70$, and at 8 kb/s $N = 140$ and $K_c = 35$. $\gamma_1 = 1$ and $\gamma_2 = 0.9$ at both 4 and 8 kb/s.</i>	<i>172</i>
Table 26	<i>Performance of reference system. Three reference systems were considered, (R1): No quantization of $A(z)$, (R2): Quantization of $A(z)$, and (R3): Quantization and interpolation of $A(z)$.</i>	<i>174</i>
Table 27	<i>Performance of optimized (closed loop) system with unquantized short term filter ($A(z)$). (a): No closed loop re-optimization of $B(z)$. (b) Closed loop re-optimization of $B(z)$.</i>	<i>175</i>
Table 28	<i>Performance of optimized (closed loop) system with quantized short term filter ($A(z)$). (a): Re-optimization of subframe parameters without knowledge of the future excitation information. (b) Re-optimization of subframe parameters with knowledge of the future excitation information.</i>	<i>177</i>
Table 29	<i>Performance with quantized, interpolated short term filter ($A(z)$). Re-optimization of subframe parameters without knowledge of the future excitation information. (a): $A(z)$ quantized open loop. In this case no re-optimization of $A(z)$ is performed. (b): $A(z)$ quantized closed loop.</i>	<i>179</i>
Table 30	<i>Performance of OC-CELP as a function of M_o and M_b. MSVQ with ($M_a = 16$), non-interpolated $A(z)$. (a) 4 kb/s. (b) 8 kb/s.</i>	<i>182</i>
Table 31	<i>Parameters of the 5, 6, 7, and 8 kb/s OC-CELP Codec.</i>	<i>184</i>
Table 32	<i>Preference for VSELP vs. OC-CELP at various rates. The error is within 7 % with 95 % confidence.</i>	<i>184</i>
Table 33	<i>MOS performance of various codecs.</i>	<i>187</i>

List of Symbols

a	Bold font implies a matrix or vector quantity. All vectors are assumed to be column vectors unless explicitly stated otherwise. The first element is a_0 unless noted otherwise.
$\mathbf{a}^T$	Superscript T implies matrix or vector transpose.
$\mathbf{a}^H$	Superscript H implies complex conjugate transpose.
$\mathbf{a}^r$	Index reversal. For vectors (dimension M), this implies reversing the order of the elements, element j becomes element $M - 1 - j$. For matrices (M by N), rows are reversed (row i becomes row $M - 1 - i$), then columns are reversed (column j becomes column $N - 1 - j$).
$T_s[\mathbf{a}]$	Toeplitz symmetric matrix operator. Symmetric Toeplitz matrix with $\mathbf{a}$ as first row and column.
$T_l[\mathbf{a}]$	Toeplitz lower triangular matrix operator. Lower triangular Toeplitz matrix with $\mathbf{a}$ as first column.
$\mathcal{T}[\mathbf{a}, \mathbf{r}]$	Toeplitz matrix operator. Toeplitz matrix with $\mathbf{a}$ as first column, $\mathbf{r}$ as first row.
α	MSE for a particular predictor, (the model gain).
$\alpha_\infty = \lim_{p \rightarrow \infty} \alpha_p$	Lower limit on prediction gain, (the one step prediction error).
$A_{n,p}(z)$	Order p optimal (minimum MSE) inverse filter. The time index (n) may be dropped if no confusion would result.
$A_n(z)$	Sub-optimal (or quantized) inverse filter. The time index (n) may be dropped if no confusion would result.
$B_n(z)$	Long term predictor/adaptive codebook. The time index (n) may be dropped if no confusion would result.

$\cup$	Convex cup.
$D_p \leftarrow \alpha_p$	Order p , minimum MSE, (the optimal model gain).
$D(R)$	Distortion rate function.
$d_{LR}(\mathbf{a}, \mathbf{R}_p)$	Likelihood ratio distortion measure.
$d_{SD}(A_1, A_2)$	Root Mean Square Log Spectral Distance (SD) between two model filters.
$d_{n,k}$	Open loop residual signal at time k , subframe n . $s_{n,k}$ filtered through $A(z)$.
$\epsilon(l_n, m_n)$	CELP distance measure (weighted MSE) over a single subframe for excitation index l_n and adaptive codebook index m_n . The time index n may be dropped if no confusion would result.
$\epsilon(l_n, l_{n+1}, m_n, m_{n+1})$	CELP distance measure (weighted MSE) over two subframes.
$\epsilon_{n,k}$	Open loop pitch residual at time k , subframe n . $d_{n,k}$ filtered through $B(z)$.
G_n	Excitation gain.
γ_x^2	Spectral flatness measure of the source x .
$\mathbf{h}, \mathbf{H}$	Impulse response, Impulse response matrix ($\mathbf{H} = \mathcal{T}_l[\mathbf{h}]$).
H	Source entropy.
$I(X, Y)$	Mutual information between the random variables X and Y .
K_c	Excitation vector dimension or subframe dimension.
K_o	Number of samples of overlap.

K_r	Impulse response length.
$L_{s,k}$	Number of levels in stage k for the short term filter MSVQ.
$L_{c,k}$	Number of levels in stage k for the excitation codebook. For full search codebooks, or non-MSVQ codebooks, L_s is used.
l_n	A particular excitation index at time n . The optimal or chosen value is denoted by $\hat{l}_n$.
M_a	Number of candidates in the MSVQ of the short term filter.
M_c	Number of candidates in the excitation codebook search procedure.
M_m	Number of candidates in the multi-pulse excitation search procedure.
M_o	Number of candidates in the open loop/closed loop CELP (OC-CELP) optimization procedure.
$M_n/\hat{m}_n$	Chosen pitch period/ adaptive codebook index at time n (m denotes a particular value, perhaps not the optimal or chosen value). The time index (n) may be dropped if no confusion would result.
$\mathbf{P}$	Channel transition matrix.
$p(X)$	Probability of occurrence of the random variable X .
$p(X Y)$	Probability of occurrence of the random variable X given that the event Y has occurred.
$\phi_{xx}(k)$	Autocorrelation of the random variable x ($= E(x_n x_{n+k})$).
$\Phi_{xx}(\omega)$	Z-transform of $\phi_{xx}(k)$ evaluated on the unit circle ($\Phi_{xx}(z) _{z=e^{j\omega}}$).
$Q[\cdot]$	Quantizer ($Q^*[\cdot]$ is the optimal quantizer).

$R(D)$	Rate distortion function.
R_k	Voronoi region, decision region, (or cell).
$\mathbf{R}_p$	Autocorrelation matrix (order p , dimension $p + 1$ by $p + 1$).
$\mathbf{r}$	Autocorrelation vector (order p , dimension $p + 1$).
σ^2	Quantized model gain.
s	Slope of the $R(D)$ curve.
σ_x^2	Variance of x .
$s_{n,k}$	Speech signal at time k , subframe n . (The index n may be dropped if unnecessary).
$\hat{s}_{n,k}, \hat{d}_{n,k}, \hat{y}_{n,k}$	Synthetic speech, excitation, and weighted synthetic speech at time k , subframe n .
t_j	A particular short term filter codebook index (for codebook j). The optimal or chosen value is denoted by $\tilde{t}_j$. The set of codebook indices (whether scalar quantization or MSVQ quantization is used) is denoted by $\tilde{\mathbf{t}}$.
T_a	Number of stages in the short term filter MSVQ codebook. The short term filter codebook structure is typically denoted by L_s - T_a . For example, if $T_a = 4$, and $L_{s,j} = 64$ (all j) the codebook would be referred to as 64-4.
T_e	Number of stages in the MSVQ excitation codebook. The same terminology (as with short term filter MSVQ) is used when referring to the codebook structure.
$u_{n,k}$	Zero input response of weighted synthesis filter ($\tilde{W}(z)/A(z)$).

$v_{n,k}^{(l)}$	Synthetic excitation.
$W(z)$	Perceptual weighting filter.
x_n	The source x at time n .
$y_{n,k}$	Weighted synthetic speech at time k , subframe n . $s_{n,k}$ filtered through $W(z)$.

List of Acronyms

ACF	Autocorrelation function.
ADPCM	Adaptive differential pulse code modulation (CCITT standard G.721).
CCITT	International Telegraph and Telephone Consultative Committee.
CELP	Codebook Excited Linear Prediction.
DFT, FFT	Discrete Fourier Transform, Fast (Discrete) Fourier Transform.
DOD-CELP	US Department of Defence standard 4.8 kb/s CELP algorithm. US Federal Standard 1016 (FS-1016).
D*PCM, DPCM	Differential Pulse Code Modulation.
DSP	Digital Signal Processor/ Processing.
DTMF	Dual Tone Multiple Frequency (telephone signalling tones).
FIR	Finite Impulse Response (filter).
FRLS	Fast Recursive Least Squares.
FS-1016	U.S. Federal Standard 1016 (DOD-CELP).
FTF	Fast Transversal Filter.
GLA	The Generalized Lloyd Algorithm. Equivalent to the LBG (Linde-Buzo-Gray) algorithm.
GSM	Groupe Spécial Mobile.
iff	If and only if.
IIR	Infinite Impulse Response (filter).

IS-CELP	Iterative Sequential CELP.
IS-54	Telecommunication Industries Associations Digital Cellular Standard (see VSELP).
JND	Just Noticeable Distortion/Difference.
kb/s	kilo (1000) bits per second.
LAR	Log Area Ratio.
LBG	Linde-Buzo-Gray Algorithm for VQ design (see GLA).
LD-CELP	Low delay CELP (CCITT standard G.728).
LP, LPC	Linear Prediction, Linear Predictive Coding.
LMS	Least Mean Square.
LR	Likelihood Ratio.
LSP, LSF	Line Spectrum Pair, Line Spectrum Frequency.
LTP	Long Term Predictor.
MIPS	Millions of Instructions per second.
MLOPS	Millions of Floating Point Operations per second.
MNRU	Modulated Noise Reference Unit (see [1]).
MOS	Mean Opinion Score.
MPE	Multi-pulse Excitation (Coding).
MS-CELP	CELP with multi-stage excitation codebooks.
MSE	Mean Squared Error.

msec	Millisecond, 1/1000-th of a second.
MSVQ	Multi-stage VQ
OC-CELP	Open Loop/Closed Loop CELP.
PCM	Pulse Code Modulation (CCITT standard G.711).
PDF	Probability Distribution Function.
RLS	Recursive Least Squares.
RMS	Root Mean Square.
RPE	Regular Pulse Excitation (Coding).
SD	Spectral Distortion.
SG	Stochastic Gradient, or Least Mean Square (LMS) algorithm.
SNR	Signal to Noise Ratio.
SNR_{SEG}	Segmental Signal to Noise Ratio.
SQ	Scalar Quantization/ Scalar Quantizer.
TIA	Telecommunications Industries Association.
TIMIT	Texas Instruments/ Massachusetts Institute of Technology Speech Database.
VSELP	Vector Sum Excited Linear Prediction (TIA IS-54 standard).
VQ	Vector Quantization/ Vector Quantizer.
WMSE	Weighted Mean Squared Error.
WSNR	Weighted Signal to Noise Ratio.

WSNR_{SEG}	Segmental Weighted Signal to Noise Ratio.
ZIR	Zero Input Response.
ZSR	Zero State Response.

Chapter 1.

Introduction

1.1. Introduction to Speech Coding

The emergence of several communications services and applications requiring integrated voice and data transmission have generated a great demand for a high quality digital voice coder able to utilize the bandwidth available in common analog circuits.

The half rate digital cellular standard and the half rate GSM standard are two key examples of the requirements for improved high quality low bit rate speech coding. The research reported herein is concerned with the investigation of high quality speech coding algorithms at bit rates between 4 and 8 kb/s.

Speech coding is concerned with the transformation of a continuous time, continuous amplitude speech signal into a discrete amplitude, discrete time (digital) signal. The transformation entails a loss of information, and possibly a loss of perceived audio quality. The goal is to maximize the audio quality (perceptual entropy) for a particular bit rate, or to minimize the bit rate for a particular perceptual entropy. The problem is unwieldy due to the variation of the perceptual entropy across individual listeners and individual speakers, and the intractable nature of the distortion measure. Desirable properties of a speech coder include:

1. high audio quality;
2. low bit rate;
3. robustness across different speakers/ languages;
4. robustness in the presence of channel errors;
5. good performance on non-speech (i.e. modem, DTMF) signals;
6. low memory size and low computational complexity; and
7. low end to end delay.

Analysis by Synthesis coding based on a weighted mean squared error distortion criteria has been demonstrated to work well at rates as low as 4-16 kb/s. In this research, improved search techniques coupled with improved codebook design methodologies are proposed to improve the performance of conventional codebook excited linear predictive (CELP) coders.

Efficient, robust quantization of the short term predictor is required to arrive at a high quality, robust codec. Multi-stage vector quantization (MSVQ) of the short term filter is proposed. Typically, a sequential design procedure and sequential search procedure is used with multi-stage VQ. In this research, joint codebook design techniques coupled with improved search procedures are used to obtain an efficient, high quality quantizer.

Structured codebook design procedures for independent gaussian random variables are compared to general codebook design procedures. It is determined- both theoretically and in practice- that little is lost in using significant structure in the codebooks, while greatly reducing the search complexity. Excitation codebook design procedures for a wide variety of codebook structures are presented and it is empirically verified that little is lost in adding significant structure to the excitation codebook.

Improved search procedures which attempt to jointly optimize the short

term filter, the adaptive codebook, and the excitation are proposed. Although the computational complexity is rather large, the improvements obtained are significant.

1.2. Thesis Objectives and Contributions

The goal of the thesis as mentioned previously is to develop high quality speech coding algorithms at bit rates between 4 and 8 kb/s.

The main contributions of the research are:

1. The development of a reduced search, reduced complexity (open loop/closed loop) adaptive codebook search procedure [2].
2. The design of low bit rate, low complexity, high quality, coding techniques for the short term model filter [3, 4, 5].
3. The development of novel codebook design procedures for the structured excitation codebook in CELP coders [6]. Particular emphasis is again placed on the rate, performance and complexity trade-offs.
4. Improved search procedures which attempt to jointly optimize the various parameters of the CELP codec. The search procedures offer crude bounds as to the ultimate limits of the CELP algorithm [7, 8, 9].

1.3. Thesis Organization

In Chapter 2 important concepts with respect to the research reported in this thesis are reviewed. Rate distortion theory, Linear Prediction, Vector Quantization, CELP and Multi-pulse— all extremely important concepts in this thesis— are reviewed. Linear Prediction and Vector Quantization are used extensively in the thesis. Multi-pulse is reviewed for historical reasons and because multi-pulse techniques are used to design sparse excitation codebooks in Chapter 4. Rate distortion theory, Blahut's algorithm, and the Finamore-Pearlman algorithm

are reviewed and also used in Chapter 4 to provide bounds on the performance of sparse excitation codebooks. The knowledgeable reader may skip Chapter 2 entirely.

Efficient, robust quantization of the short term predictor is required to arrive at a high quality, robust codec. In Chapter 3, multi-stage vector quantization of the short term filter is proposed. Joint codebook design and various codebook search techniques are investigated to arrive at an efficient solution (in terms of performance, computational complexity, and memory size).

The determination of the excitation in CELP coders is a relatively complex operation. In practice, considerable structure is typically imposed on the codebook in order to facilitate a fast search of the excitation codebook. In Chapter 4 excitation codebook design is reviewed, and novel approaches to the design of structured excitation codebooks are presented. The impact of adding structure to the excitation codebook is analyzed in both a theoretical and practical manner. Various codebook structures are considered. It is shown that a multi-stage configuration can again provide a nice performance versus complexity trade-off.

In Chapter 6 both iterative sequential (IS-CELP) and open loop/closed loop (OC-CELP) optimization of the parameters of a CELP codec are used in order to improve the objective performance and the subjective quality of the CELP algorithm. The goal is fully joint optimization of the parameters of a CELP codec. However, due to the nature of the algorithm, a suboptimal local minimum is typically found.

The final chapter is a list of suggestions for future work and a short conclusion.

Chapter 2.

Literature Review

In this chapter important concepts with respect to the research reported in this thesis are reviewed. The fundamental theorems of data compression are briefly reviewed in Section 2.1. Sections 2.2 and 2.3 review aspects of quantization and vector quantization respectively. In Section 2.4 linear prediction is reviewed and some justification for CELP is given. CELP is reviewed in Section 2.5. In Section 2.6 the multi-pulse model and different search techniques for suboptimal (or asymptotically optimal) pulse placement algorithms are reviewed.

2.1. Rate-Distortion Theory

This section is concerned with the review of rate-distortion theory and quantization. Rate-distortion theory concerns the information content of sources subject to a fidelity criterion; or the minimum rate required to obtain a particular fidelity or distortion. As Shannon so aptly put it [10]:

“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.”

Quantization concerns the mapping from the possibly continuous amplitude source ensemble to a discrete ensemble. The mapping function maps according

to some distortion criterion or metric and the rate-distortion function is a bound on the performance of quantization. A simple example of a quantizer is an analog to digital converter. Quantization results, in general, in a loss of information.

This section begins with a review of Shannon's 1959 paper entitled "*Coding Theorems For A Discrete Source with a Fidelity Criterion*" [11]. Although Shannon's more well known paper (in book form) "*The Mathematical Theory of Communication*" introduces the concept of the rate-distortion function, the latter paper more formally introduces the notion and introduces the major theorems involved in rate-distortion theory.

Following this review, numerical techniques for computing the rate-distortion function and for computing tight bounds on the rate-distortion function are discussed. Blahut's algorithm for computing the rate-distortion function for discrete input/output alphabets is presented in Section 2.1.2, as well as an algorithm which tightly bounds (from above) the rate-distortion function of continuous input/output sources based on pre-quantization (Finamore-Pearlman Algorithm, Section 2.1.3).

2.1.1. Introduction to Rate-Distortion Theory

Although Shannon's paper (in book form) entitled "*The Mathematical Theory of Communication*" [10] introduced the concept of the rate-distortion function and calculated it for independent gaussian random variables, his next major paper "*Coding Theorems For A Discrete Source with a Fidelity Criterion*" [11] formally introduced the concept, and derived many of the major theorems and properties of rate-distortion theory. The discussion below follows that of Shannon [11]. The discrete source is of primary interest in this presentation, although the results can be generalized to the continuous case. It is assumed the reader is familiar with the concepts of entropy and mutual information [12, 13].

The main theorem, or result, is that for a particular stationary source and distortion measure it is impossible to achieve fidelity D , while transmitting information at a rate less than or equal to $R(D)$, the rate-distortion function. However, it is possible to approach (within arbitrary limits) the rate-distortion function, $R(D)$, while achieving fidelity D .

Assume a discrete source of information $\mathbf{x}$ with alphabet size n , and a reproduction alphabet $\mathbf{y}$ of size l . Each element in the source alphabet or reproduction alphabet is called a letter.

Consider a sequence of K random variables each chosen from the $\mathbf{x}$ ensemble, call this random variable $\mathbf{X}$ and call the reproduction alphabet $\mathbf{Y}$. Each element X_j (or Y_k) of the new source alphabet $\mathbf{X}$ (or reproduction alphabet $\mathbf{Y}$) is called a word, and each letter in the word is from the $\mathbf{x}$ ensemble (or $\mathbf{y}$ ensemble). There are $N = n^K$ elements in the $\mathbf{X}$ source ensemble and $L = l^K$ elements in the reproduction alphabet. To reiterate

$$\mathbf{x} \in \{x_0, x_1, \dots, x_{n-1}\} \quad (1)$$

and

$$\mathbf{y} \in \{y_0, y_1, \dots, y_{l-1}\}. \quad (2)$$

The letters of $\mathbf{X}$, are from the $\mathbf{x}$ ensemble, and the letters of the reproduction words $\mathbf{Y}$ are from the $\mathbf{y}$ ensemble as follows

$$\mathbf{X} \in \{X_0, X_1, \dots, X_{N-1}\}, \quad (3)$$

$$\mathbf{Y} \in \{Y_0, Y_1, \dots, Y_{L-1}\}. \quad (4)$$

Each X_j is a sequence of length K , and X_{ji} is the i -th letter in the sequence.

A *single letter* distortion measure is defined by a matrix with elements d_{jk} ($d_{jk} > 0$) where d_{jk} represents the cost or distortion in representing a single letter x_j by y_k . The distortion, if a particular word X_j is reproduced as word Y_k is

$$d_{jk} = d(X_j, Y_k) = \frac{1}{K} \sum_{i=0}^{K-1} d(X_{ji}, Y_{ki}) \quad (5)$$

where X_{ji} is the i -th letter of X_j and Y_{ki} is the i -th letter of Y_k .

A particular word X_j has probability of occurrence $p(X_j)$, and if X_j is reproduced as Y_k with probability $p(Y_k|X_j)$, then the overall expected (average) per letter distortion is

$$D = \sum_j \sum_k p(X_j) p(Y_k|X_j) d(X_j, Y_k). \quad (6)$$

The mutual information between the source-user pair $\mathbf{X}$ and $\mathbf{Y}$ is

$$I(\mathbf{X}, \mathbf{Y}) = \sum_j \sum_k p(X_j) p(Y_k|X_j) \log \frac{p(Y_k|X_j)}{p(Y_k)} \quad (7)$$

where

$$p(Y_k) = \sum_j p(Y_k|X_j) p(X_j). \quad (8)$$

For a particular source $\mathbf{X}$, $I(\mathbf{X}, \mathbf{Y})$ is only a function of the channel transition matrix $\mathbf{P}$ assuming, in general, there is no control over the input probability distribution where

$$\mathbf{P} = \{p(Y_k|X_j)\}. \quad (9)$$

The rate-distortion function, $R(D)$ is defined as the minimum of $I(\mathbf{X}, \mathbf{Y}) - I(\mathbf{P})$ over the set of all channel transition matrices $\mathcal{P}_D$ which achieve fidelity less than or equal to D , (subject to the limitations imposed on the channel transition matrices).

Thus, the rate-distortion function is

$$R(D) = \min_{\mathbf{P} \in \mathcal{P}_D} I(\mathbf{P}) \quad (10)$$

where

$$\mathcal{P}_D = \left\{ p(Y_k|X_j) : \sum_j \sum_k p(X_j) p(Y_k|X_j) d(X_j, Y_k) \leq D \right\} \quad (11)$$

is the set of all channel transition matrices achieving fidelity D or less and

$$I(\mathbf{P}) = \sum_j \sum_k p(X_j) p(Y_k|X_j) \log \frac{p(Y_k|X_j)}{\sum_i p(Y_k|X_i) p(X_i)}. \quad (12)$$

Mutual information is a convex cup ($\cup$) function with respect to the convex set of channel transition matrices and is monotonically decreasing (since increasing the information rate can only decrease the distortion). The minimum value of D , D_{min} , is found by assigning a probability of 1 for the transition from X_j to Y_k with the lowest $d(X_j, Y_k)$. Therefore

$$D_{min} = \sum_j p(X_j) \min_k d(X_j, Y_k) \quad (13)$$

and $R(D_{min}) = H$ (the source entropy, assumed finite) if $D_{min} = 0$. For the given rate, $R = 0$, $R(D_{max}) = 0$, $p(Y_k|X_j) = p(Y_k)$ (since $\mathbf{X}$ and $\mathbf{Y}$ are independent) and

$$\begin{aligned} D_{max} &= \min_{\mathcal{P}_D} \sum_j \sum_k p(X_j) p(Y_k|X_j) d(X_j, Y_k) \\ &= \min_{\mathcal{P}_D} \sum_j p(X_j) \sum_k p(Y_k) d(X_j, Y_k) \\ &= \sum_j p(X_j) d(X_j, Y^*), \end{aligned} \quad (14)$$

where Y^* is the minimum average distortion reconstruction value. For $D \geq D_{max}$

$$R(D) = R(D_{max}) = 0. \quad (15)$$

The rate-distortion function is defined for $D \geq D_{min}$, is monotonically decreasing, and $R(D) = R(D_{max}) = 0$ for $D \geq D_{max}$. A typical rate-distortion function is shown in Fig. 1.

In the following sections algorithms are presented for computing the rate-distortion function for discrete input, discrete output alphabets based on an algorithm developed by Blahut [14], and an asymptotically convergent algorithm for continuous input, continuous output alphabets based on a generalization of Blahut's algorithm [15].

2.1.2. Blahut's Algorithm for Computing $R(D)$

As shown previously, for the case of discrete input, discrete output alphabets, the rate-distortion function can be written as

$$R(D) = \min_{\mathbf{P} \in \mathcal{P}_D} I(\mathbf{P}) \quad (16)$$

where

$$\mathcal{P}_D = \left\{ p(Y_k|X_j) : \sum_j \sum_k p(X_j) p(Y_k|X_j) d(X_j, Y_k) \leq D \right\} \quad (17)$$

is the set of all channel transition matrices achieving fidelity D or less and

$$I(\mathbf{P}) = \sum_j \sum_k p(X_j) p(Y_k|X_j) \log \frac{p(Y_k|X_j)}{\sum_i p(Y_k|X_i) p(X_i)}. \quad (18)$$

Let $\mathbf{P}^*$ be the optimal transition matrix. Thus, (18) is minimized given (the constraint)

$$D - \sum_j \sum_k p(X_j) p^*(Y_k|X_j) d(X_j, Y_k) = 0. \quad (19)$$

To minimize, the technique of Lagrange Multipliers is used where

$$R(D) = \min_{\mathbf{P}_D} \left[\sum_j \sum_k p(X_j) p(Y_k|X_j) \log \frac{p(Y_k|X_j)}{\sum_i p(Y_k|X_i) p(X_i)} + \right]$$

Figure 1: A typical rate-distortion function. $R(D)$ is not defined for $D \leq D_{\min}$. For $D \geq D_{\max}$, $R(D) = 0$.

$$\begin{aligned}
& s \left(D - \sum_j \sum_k p(X_j) p(Y_k | X_j) d(X_j, Y_k) \right) \Bigg] \\
& = sD + \min_{P_D} \left[\sum_j \sum_k p(X_j) p(Y_k | X_j) \log \frac{p(Y_k | X_j)}{\sum_i p(Y_k | X_i) p(X_i)} \right. \\
& \quad \left. s \sum_j \sum_k p(X_j) p(Y_k | X_j) d(X_j, Y_k) \right] \tag{20}
\end{aligned}$$

and where D is given by (19). Note that the slope of $R(D)$ is simply s , since the second term (and thus, the minimization) in the above equation does not depend on D .

The iterative convergent (see [14]) algorithm for computing $R(D)$ is based on the following two criteria:

1. For a fixed set of transition probabilities, $R(D)$ is minimized by choosing [14]

$$p(Y_k) = \sum_j p(X_j) p(Y_k | X_j). \tag{21}$$

2. For fixed $\{p(Y_k)\}$, (for all k), $R(D)$ is minimized by choosing

$$p(Y_k | X_j) = \frac{p(Y_k) e^{s d(X_j, Y_k)}}{\sum_i p(Y_i) e^{s d(X_j, Y_i)}}. \tag{22}$$

The first iteration of the algorithm picks a starting distribution

$$p^{(0)} = \{p(Y_k), 0 \leq k < L\} \tag{23}$$

and then computes the optimal transition matrix (for $p^{(0)}$) according to (22). The next iteration then computes the optimal output probabilities ($p^{(1)}$) for the new

transition matrix. Substitution of (22) into (21) and defining

$$c^{(n)}(Y_k) = \sum_j \frac{p(X_j) \epsilon^{sd(X_j, Y_k)}}{\sum_i p^{(n)}(Y_i) \epsilon^{sd(X_j, Y_i)}} \quad (24)$$

yields

$$p^{(n+1)}(Y_k) = p^{(n)}(Y_k) c^{(n)}(Y_k) \quad (25)$$

where the superscript n indicates the iteration number. The algorithm is terminated when

$$\sum_k p(Y_k) \log c^{(n)}(Y_k) - \max_k \log c^{(n)}(Y_k) < \epsilon \quad (26)$$

where ϵ is an upper bound on the error in the estimate of $R(D)$ [14]. When the condition is met (at iteration N , say), the algorithm is terminated, and

$$R(D) = sD - \sum_j p(X_j) \log \left[\sum_k \epsilon^{sd(X_j, Y_k)} p^{(N)}(Y_k) \right] - \sum_k p^{(N)}(Y_k) \log [c^{(N)}(Y_k)] \quad (27)$$

at

$$D = \sum_j \sum_k p(X_j) p^{(N)}(Y_k | X_j) d(X_j, Y_k) \quad (28)$$

where $p^{(N)}(Y_k | X_j)$ is given by (22).

2.1.3. Finamore-Pearlman Algorithm for Continuous Sources

An algorithm to compute very tight upper bounds on the rate-distortion function assuming continuous input, continuous output alphabets was presented by Finamore and Pearlman in [15].

The algorithm is a generalization of Blahut's algorithm and changes the output alphabet's levels (or distortion) in a recursive manner. The goal of the algorithm

is to approach the rate distortion function for the continuous input - continuous output case with only a finite input alphabet and a finite output alphabet.

The algorithm is based on the notion of pre-quantization as shown in Fig. 2. The distortion D_Q is the average distortion introduced by the pre-quantizer and D_f is the average distortion in mapping $\{X_j\}$ to $\{Y_k\}$.

Assuming adequately fine pre-quantization, $(Q[\cdot])$, the algorithm adjusts the transition matrix and the L output probabilities and output values in an attempt to approach the rate distortion bound. For (expected) distortions satisfying the triangle inequality

$$D \leq D_Q + D_f \quad (29)$$

the rate-distortion function is upper bounded by $R_{N,L}(D)$ where $R_{N,L}(D)$ is the discrete rate-distortion function (between X_j and Y_k) at $D = D_Q + D_f$.

As an example, consider the rate-distortion function for unit variance gaussian random variables and a mean squared error distortion criterion. With pre-quantization, the average pre-quantization distortion is

$$D_Q = \sum_{j=0}^{N-1} \int_{R_j} d(X, X_j) p(X) dX \quad (30)$$

where

$$d(X, X_j) = (X - X_j)^2, \quad (31)$$

and $p(X)$ is the probability density function of X . R_j is the region in which X is closest (in a mean squared error sense) to centroid (or reproduction level) X_j .

Assuming, for a moment, $N = L$, the rate-distortion function for the discrete case can be computed using Blahut's algorithm. An upper bound on the rate distortion function can be constructed by using the bound on the total distortion

Figure 2: Source coding with pre-quantization. The expected distortion in the first stage (pre-quantization) is D_Q and the expected distortion in the second stage ($R(D)$ for finite alphabet source) is D_f .

as given by (29). This bound on the distortion is simply the sum of the pre-quantization distortion D_Q and the distortion from the discrete rate-distortion function D_f . For the mean squared error (MSE)

$$\begin{aligned}
 D &= E[(X - Y_k)^2] \\
 &= E[(X - X_j)^2 + (X_j - Y_k)^2 + 2(X - X_j)(X_j - Y_k)] \\
 &= E[(X - X_j)^2] + E[(X_j - Y_k)^2] \\
 &= D_Q + D_f,
 \end{aligned} \tag{32}$$

and the triangle inequality is satisfied (with equality).

In [15], an algorithm is presented to adjust the output values assuming $N \neq L$. The interested reader is referred to [15].

The rate-distortion bound for the continuous input, continuous output case can only be reached with an infinite number of output values. However, as shown in [15] only a small number of output values are required in order to very tightly upper bound the rate distortion function. Thus, quantization appears to be a suitable method to implement systems which can perform close to the rate-distortion bound. Quantization is the subject of the next section.

2.2. Lloyd-Max Scalar Quantization

This section considers methods to design optimal L -level one dimensional (scalar) quantizers, based on Lloyd's algorithm (Method I) [16]. Another optimal algorithm developed independently by Lloyd (Method II) and Max [17] will not be considered in this thesis, but can also be used to design optimal L -level quantizers. For this reason, optimal L -level quantizers are known as Lloyd-Max (scalar) quantizers.

The algorithm is optimal for the design of L -level one dimensional quantizers. That is, no attempt is made to minimize output entropy, but only to optimize the

L -level output alphabet. In [18, 19] it was determined that for sufficiently large L , uniform quantization with entropy coding can perform better (lower output entropy) than optimal entropy coded Lloyd-Max (non-uniform) quantization. If entropy coding after quantization is not used (or as an L -level pre-quantizer), Lloyd-Max scalar quantization is optimal, otherwise, better quantization levels exist.

An L -level quantizer maps the input X into one of L quantized values or centroids according to the minimum distortion search

$$Q[X] = X_{k^*} \quad (33)$$

where k^* is the optimum quantizer index according to

$$d(X_{k^*}, X) \leq d(X_k, X) \quad \forall \quad k \neq k^*. \quad (34)$$

It is assumed that the distortion is non negative. The set of optimal centroids is that which minimizes the average or expected distortion. The average or expected distortion given by

$$D = E[d(Q[X], X)]. \quad (35)$$

The optimal quantizer Q^* minimizes the expected distortion over all L -level quantizers. The quantizer Q^* is optimal iff

$$E[d(Q^*[X], X)] \leq E[d(Q[X], X)] \quad (36)$$

for all quantizers Q .

The decision regions (cells, Voronoi regions), R_k are the set of points with nearest centroid X_k . That is

$$X \in R_{k^*} \Rightarrow d(X_{k^*}, X) \leq d(X_k, X) \quad \forall \quad k \neq k^*. \quad (37)$$

For a given decision region, the centroid, or center of mass is that which minimizes the expected distortion over that region. The centroid X_k of region R_k obeys

$$\int_{R_k} d(X_k, X) p(X) dX \leq \int_{R_k} d(\hat{X}_k, X) p(X) dX \quad \forall X_k \neq \hat{X}_k. \quad (38)$$

For the mean squared error distortion measure

$$\begin{aligned} \min_{X_k} \int_{R_k} (X - X_k)^2 p(X) dX &\Rightarrow \frac{d}{dX_k} \left(\int_{R_k} (X - X_k)^2 p(X) dX \right) = 0 \\ &\Rightarrow \int_{R_k} X p(X) dX = \int_{R_k} X_k p(X) dX \end{aligned} \quad (39)$$

and thus the centroid is given by

$$X_k = \frac{\int_{R_k} X p(X) dX}{\int_{R_k} p(X) dX}. \quad (40)$$

Lloyd used the two criteria– minimum distortion search ((33) and (34)) and centroid computation ((40))– to design optimal L level quantizers using a recursive algorithm with two main steps:

1. **Optimize Partition:** For a given set of centroids, compute the optimal decision regions.
2. **Optimize Centroids:** Given the decision regions, compute the optimal centroids.

The algorithm will converge, for any source with a known continuous PDF and a computable centroid and distortion measure. For discrete PDFs it can be shown that the algorithm may not converge to a local minimum. Although Lloyd was concerned only with the mean squared error distortion and one dimensional

quantizers, the algorithm was extended to non-MSE distortion measures, multi-dimensional quantizers, and designs based only on a training sequence of data (unknown PDF). For more information see [20] and/or [21].

An example of the performance of Lloyd-Max scalar quantization with and without optimal entropy coding is shown in Fig. 3. In general, scalar quantization can not attain the $R(D)$ bound. In Section 2.3 reasons for this suboptimal performance are given, and an asymptotically optimal encoding algorithm is discussed which can approach to within arbitrary limits the $R(D)$ bound.

2.3. Vector Quantization

Vector quantization (VQ) concerns the mapping in multi-dimensional space from a (possibly continuous amplitude) source ensemble to a discrete ensemble. The mapping function maps according to some distortion criterion or metric and the rate-distortion function is a bound on the performance of vector quantization—at least for stationary sources for which a rate-distortion function exists. The degenerate example of a vector quantizer is a scalar quantizer.

Vector quantization is the joint quantization of a block of parameters and takes advantage of four interrelated properties [20]:

1. PDF shape;
2. linear dependency (correlation);
3. non-linear dependency; and
4. vector dimensionality (or cell shape).

Property 4 implies that more efficient quantizers are possible by simply increasing the vector dimension. In fact, even for independent data, the rate distortion bound may only be approached as the vector dimension approaches infinity due to the less restricted cell shapes which are possible in higher dimensions.

Figure 3: Performance of Lloyd-Max scalar quantization with and without optimal entropy coding for $L=3, 5, 9, 17, 33, 65, 129$, and 257 .

Other important properties of VQ include:

1. Higher memory size and computational complexity than scalar quantization.
2. Relatively straightforward (locally optimal) codebook design algorithms exist for a wide variety of distortion measures.
3. VQ can operate successfully at rates as low (and lower) than 1 bit per sample— rates at which scalar quantization tends to perform poorly.

2.3.1. The Generalized Lloyd Algorithm

This section considers methods to design locally optimal L -level K -dimensional vector quantizers, based on Lloyd's algorithm (Method I) [16]. The algorithm is known as the Generalized Lloyd Algorithm (GLA), or the Linde-Buzo-Gray (LBG) algorithm [21].

The algorithm is locally optimal for the design of L -level vector quantizers. Again, no attempt is made to minimize output entropy, but only to optimize the L -level output alphabet.

An L -level vector quantizer consists of a codebook

$$\mathcal{C} = \{\mathbf{X}_k, 0 \leq k < L\} \quad (41)$$

and mapping function which maps the input vector $\mathbf{X}$ into one of L quantized values (often referred to as centroids) according to the minimum distortion search

$$\mathbf{Y} = \hat{\mathbf{X}} = Q[\mathbf{X}] = \mathbf{X}_k. \quad (42)$$

iff

$$d(\mathbf{X}_{k^*}, \mathbf{X}) \leq d(\mathbf{X}_k, \mathbf{X}) \quad \forall \quad k \neq k^*. \quad (43)$$

The set of optimal centroids is that which minimizes the average distortion. The average or expected distortion is

$$D = E[d(Q[\mathbf{X}], \mathbf{X})]. \quad (44)$$

An L -level vector quantizer is optimal if it minimizes the expected distortion. Thus, the quantizer Q^* is optimal iff

$$E[d(Q^*[X], X)] \leq E[d(Q[X], X)] \quad (45)$$

for all quantizers Q . The quantizer Q^* minimizes the expected distortion over all L -level quantizers.

There are two important algorithms based on Lloyd's Method I for VQ design. The choice of algorithm depends on whether or not the multi-dimensional source probability distribution function is known. If the PDF is unknown, the design is based on a sufficiently rich training sequence.

2.3.1.1 GLA With Known PDF The decision regions (or cells), R_k are the set of points with nearest centroid X_k . That is

$$X \in R_{k^*} \Rightarrow d(X_{k^*}, X) \leq d(X_k, X) \quad \forall k \neq k^*. \quad (46)$$

For a given cell, the centroid, or center of mass is that which minimizes the expected distortion over that region. The centroid X_k of region R_k obeys

$$\int_{R_k} d(X_k, X) p(X) dX \leq \int_{R_k} d(\tilde{X}_k, X) p(X) dX \quad \forall X_k \neq \tilde{X}_k. \quad (47)$$

For the mean squared error distortion measure

$$\min_{X_k} \int_{R_k} (X - X_k)^2 p(X) dX \Rightarrow \int_{R_k} X p(X) dX = \int_{R_k} X_k p(X) dX, \quad (48)$$

and thus the centroid is given by

$$X_k = E[X|X \in R_k] = \frac{\int_{R_k} X p(X) dX}{\int_{R_k} p(X) dX}. \quad (49)$$

The centroid is only a function of the distortion measure, since the source distribution is given. Thus, for a given multi-dimensional PDF the optimal L -level, K -dimensional vector quantizer only depends on the chosen distortion measure. However, for the GLA, the centroid must be computable from the distortion and the PDF, otherwise other algorithms for VQ design must be used.

2.3.1.2 GLA Based on a Training Sequence Assuming a sufficiently rich training sequence (T) of representative vectors ($\mathbf{X}^{(j)}$)

$$T = \{ \mathbf{X}^{(j)}, 0 \leq j < N_t \}, \quad (50)$$

the training sequence is partitioned into L cells or regions according to the minimum distortion search

$$\mathbf{Y}^{(j)} = \hat{\mathbf{X}}^{(j)} = Q[\mathbf{X}^{(j)}] = \mathbf{X}_{k^*}, \quad (51)$$

iff

$$d(\mathbf{X}_{k^*}, \mathbf{X}^{(j)}) \leq d(\mathbf{X}_k, \mathbf{X}^{(j)}) \quad \forall \quad k \neq k^*. \quad (52)$$

That is, the j -th training vector belongs to cell k^* if and only if the distortion between centroid k^* is less than or equal to the distortion between the j -th training vector and all other centroids.

The decision regions (or cells), R_{k^*} are the set of points with nearest centroid $\mathbf{X}_{k^*}$. That is

$$\mathbf{X} \in R_{k^*} \Rightarrow d(\mathbf{X}_{k^*}, \mathbf{X}) \leq d(\mathbf{X}_k, \mathbf{X}) \quad \forall \quad k \neq k^*. \quad (53)$$

For a given cell, the centroid, or center of mass is that which minimizes the expected distortion over that region. The centroid $\mathbf{X}_k$ of region R_k obeys

$$\sum_{R_k} d(\mathbf{X}_k, \mathbf{X}^{(j)}) \leq \sum_{R_k} d(\tilde{\mathbf{X}}_k, \mathbf{X}^{(j)}) \quad \forall \quad \mathbf{X}_k \neq \tilde{\mathbf{X}}_k. \quad (54)$$

For the mean squared error distortion measure the centroid is given by

$$\mathbf{X}_k = \frac{\sum_{R_k} \mathbf{X}^{(j)}}{N_k} \quad (55)$$

where N_k is the number of training vectors in cell k .

Essentially, both algorithms are very similar. The algorithm based on a training sequence attempts to compute the true centroid, but suffers if there is insufficient data in the training sequence. The algorithm typically converges inside and outside the training sequence as shown in Fig. 4.

Vector quantizer design based on the GLA uses the two criteria— minimum distortion search ((42) and (43)) and centroid computation ((49) and (55))— to design locally optimal L level quantizers using a recursive algorithm with two *main* steps:

1. Set $n = 0$. Choose an initial quantizer $Q^{(0)}$. Anything will do, but avoid $p(\mathbf{X}_k) = 0$, and $\mathbf{X}_k = \mathbf{X}_j$, $k \neq j$. Compute the average distortion $D^{(0)}$.
2. For the given quantizer ($Q^{(n)}$), compute the decision regions using the minimum distortion search criterion. Calculate optimal centroids for the newly computed decision regions ($Q^{(n+1)}$).
3. Compute the average distortion $D^{(n+1)}$. If $D^{(n)} - D^{(n+1)} < \epsilon$ then terminate. Otherwise set $n = n + 1$, and go to 2.

The algorithm will converge under certain conditions to a local minimum for any reasonable source with a known continuous PDF or a sufficiently rich training sequence and a computable centroid and distortion measure. For more information see [20] and/or [21].

Figure 4: Convergence both inside and outside the training sequence. For a low number of training vectors, the performance on the training data (inside) is very good although the performance outside the training sequence is relatively poor. As the number of training vectors increases, the difference in performance outside and inside the training sequence approaches zero.

2.3.2. Structured VQ

Structured VQ is vector quantization whereby structure is added to the codebook in order to reduce the computational complexity and/or memory size. For example, scalar quantization can be thought of as a higher dimensional VQ with structure enabling the simplification of the codebook search procedure. Other techniques for adding structure to simplify the search complexity and or memory size include:

1. tree structured VQ [20, 22];
2. gain shape VQ [20, 23];
3. product code (split) VQ [20, 24, 25];
4. basis vector VQ [26, 27, 28];
5. multi-stage VQ [20, 29];
6. overlapping (shift symmetric) VQ [30, 31]; and
7. sparse VQ [32, 33].

2.4. Linear Prediction

In this section linear prediction is briefly reviewed. The review is quite brief and the interested reader is encouraged to consult [34, 35, 36].

2.4.1. Application of Linear Prediction to Speech Coding

For a stationary, non-white gaussian stochastic process x , the distortion-rate function (the minimum distortion for the given rate) is given by [13, 12]

$$D_x(R) = 2^{-2R} \sigma_x^2 \gamma_x^2 \quad (56)$$

where σ_x^2 is the variance of x and γ_x^2 is the spectral flatness measure, given by

$$\gamma_x^2 = \frac{\exp\left(\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln(\Phi_{xx}(\omega)) d\omega\right)}{\frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{xx}(\omega) d\omega}, \quad (57)$$

where $\Phi_{xx}(\omega)$ is the Z -transform of $\phi_{xx}(k) (= E(x_n x_{n+k}))$ evaluated on the unit circle. For white gaussian sources where $\Phi_{xx}(\omega)$ is a constant, the gain (in dB) in quantizing the non-white source over the white source with equal variance is

$$-10 \log_2 (\gamma_x^2) \geq 0. \quad (58)$$

Whitening of the source can be accomplished by pre-filtering by the optimal (infinite order) predictor where

$$H(z) = 1 + \sum_{k=1}^{\infty} h_k z^{-k}. \quad (59)$$

By Wold's Decomposition [37], there exists a filter $H(z)$, such that y_n is a sequence of independent identically distributed gaussian random variables (white gaussian). The spectral flatness measure of y is therefore unity. The variance of y is

$$\sigma_y^2 = \sigma_x^2 \gamma_x^2 \quad (60)$$

with distortion-rate function

$$\begin{aligned} D_y(R) &= 2^{-2R} \sigma_y^2 \gamma_y^2 \\ &= 2^{-2R} \sigma_x^2 \gamma_x^2 \gamma_y^2 \\ &= 2^{-2R} \sigma_x^2 \gamma_x^2 \\ &= D_x(R). \end{aligned} \quad (61)$$

Fundamentally, nothing is gained by using an infinite order pre-whitening filter.

Another justification for the use of linear prediction is required. The optimal rate R coding scheme for the stochastic process x can be constructed as follows:

1. Construct a table of length L and dimension K randomly. L and K are chosen such that the rate is

$$R = \frac{-\sum_i P_i \log(P_i)}{K} \quad (62)$$

where P_i is the i -th codeword probability. Let both L and K approach infinity, while keeping the rate constant.

2. To quantize x , choose an entry in the table to minimize the distortion between x and $\hat{x}$ (the minimum distortion entry in the table).
3. Over all possible randomly selected infinite length tables, there is at least one which approaches to within infinitesimal limits the distortion-rate bound.

Clearly, this scheme is not at all practical. Simple scalar quantization is practical, although the performance of scalar quantization is *at least* $-10 \log_{10}(\gamma_x^2)$ dB above the rate-distortion bound, since there is no exploitation of the spectral characteristics of x . A more efficient suboptimal approach is shown in Fig. 5. In this case, the optimal set of indices (from $n = -\infty$ to $n = \infty$) which results in minimal distortion is chosen, at a net decrease in distortion (over scalar quantization) of $-10 \log_{10}(\gamma_x^2)$ dB. Due to stationarity, $1/H(z)$ is known at both the encoder and decoder and thus only the set of scalar indices is required to reconstruct $\hat{x}$. This system, although suboptimal, is still impractical since the delay is infinite due to the infinite response and infinite order of $1/H(z)$ and the requirement of searching over all possible sequences of scalar quantizer indices (from $n = -\infty$ to $n = \infty$). To obtain finite delay, suboptimal indices must

Figure 5: Scalar quantization in conjunction with linear prediction. The optimal sequence of indices (l_n) is chosen to minimize the distortion between x_n and $\hat{x}_n^{(l_n)}$.

be released. A finite order p (although an infinite impulse response) filter is used to obtain the first practical system encountered thus far— Delayed Decision Differential Quantization— as shown in Fig. 6. The inverse quantizer Q^{-1} need not be a scalar quantizer, although it does have a finite dimension and a finite number of possible output symbols. The scheme in Fig. 6 is known as *analysis by synthesis* since the goal is to determine the minimum distortion *synthesized* signal $\hat{x}_n$. The scheme is suboptimal for two *very* important reasons:

1. the encoder structure is suboptimal since the finite dimensional quantizer imposes limits on the structure of the encoder; and
2. the search for the optimal indices is suboptimal since symbols are released before considering their full impact ($1/H(z)$ is IIR).

The decision depth (P) should be sufficiently large to ensure that the contribution of past samples (via the model filter $1/H(z)$) are taken into account. For $P = 0$, ordinary DPCM results, which uses no information about future symbols in order to determine the current symbol.

For real sampled speech signals which are neither gaussian nor stationary, a rate distortion function does not exist, and even if it did would be of marginal importance unless the distortion function had very accurate knowledge of how the particular (or at least average) far end user perceived the distortion. What is required is a *Perceptual Entropy* function— a function of time which indicates the minimum rate required to obtain no noticeable distortion. This so called Perceptual Entropy function (or Just Noticeable Distortion (JND)) is time varying and, in general, the channel is fixed rate, which further complicates matters.

Since speech is non-stationary- the sounds produced depend on the state of the vocal tract- a fixed predictor performs poorly when compared to adaptive prediction.

Figure 6: Delayed decision quantization in conjunction with a finite order linear predictor. The suboptimal finite length sequence of indices at time T $\{l_T, l_{T-1}, l_{T-P}\}$ is chosen to minimize the average distortion between x_n and $\hat{x}_n^{(l_n)}$ for $n = T - P$ to $n = T$. At time T the $T - P$ -th index is released.

Two main adaptive prediction methods exist:

1. block adaptation algorithms; and
2. recursive adaptation algorithms;

and two main adaptation techniques:

1. forward adaptation; and
2. backward adaptation.

Forward adaptation uses information at the encoder which is not available at the decoder, and backward adaptation does not. For this reason, the coefficients of $H(z)$ must be sent as side information in the case of forward adaptation. Recursive algorithms update the parameters of $H(z)$ on a sample by sample basis using a (typically) recursive window whereas block algorithms use a finite length window of samples and update $H(z)$ periodically every 10 to 30 msec. Typically- although not essentially- block algorithms are used in conjunction with forward adaptation and recursive algorithms are used in conjunction with backward adaptation. Recursive procedures applied to forward adaptation are typically quite complex and there is no requirement to have a new model filter for each input sample. Thus, recursive procedures are not used extensively in this research, and are not reviewed. The primary recursive algorithms are:

1. the recursive least squares algorithms (recursive least squares (RLS), Fast RLS (FRLS), fast transversal filter (FTF)); and
2. the stochastic gradient (SG) or least mean square (LMS) algorithm.

The interested reader is referred to [37] for an informative discussion of the recursive algorithms and their computationally efficient equivalents.

2.4.2. Block Algorithms- Basic Formulation and Solution

Linear prediction (LP), as the name implies, is concerned with the prediction of future samples of a deterministic (but unknown) or stochastic process.

The one step order p linear prediction filter, or the inverse filter

$$A_p(z) = 1 + \sum_{k=1}^p a_k z^{-k} \quad (63)$$

is typically designed to minimize the mean squared error between the process (s_n) and the estimated values of the process ($\hat{s}_n$), based on p previous values of s_n where

$$\hat{s}_n = - \sum_{k=1}^p a_k s_{n-k}. \quad (64)$$

The error (d_n) at time n can be written as

$$\begin{aligned} d_n &= s_n - \hat{s}_n \\ &= s_n + \sum_{k=1}^p a_k s_{n-k} \\ &= \mathbf{s}_n^T \mathbf{a}, \end{aligned} \quad (65)$$

where

$$\mathbf{s}_n^T = [s_n \quad s_{n-1} \quad s_{n-2} \quad \dots \quad s_{n-p}], \quad (66)$$

and

$$\mathbf{a}^T = [1 \quad a_1 \quad a_2 \quad \dots \quad a_p]. \quad (67)$$

The average mean squared error (over N samples) for a particular set of predictor coefficients $\mathbf{a}$ is given by

$$\begin{aligned} D_p &= \frac{1}{N} \sum_n d_n^2 \\ &= \frac{1}{N} \mathbf{d}^T \mathbf{d} \\ &= \frac{1}{N} \mathbf{a}^T \mathbf{S}^T \mathbf{S} \mathbf{a} \\ &= \mathbf{a}^T \mathbf{R}_p \mathbf{a}, \end{aligned} \quad (68)$$

where $\mathbf{d}$ is a column vector (dimension N)

$$\mathbf{d}^T = [\dots \quad d_{j-1} \quad d_j \quad d_{j+1} \quad \dots], \quad (69)$$

and $\mathbf{S}$ is a matrix (N by p)

$$\mathbf{S}^T = [\dots \quad s_{j-1} \quad s_j \quad s_{j+1} \quad \dots]. \quad (70)$$

As N approaches infinity, the matrix $\mathbf{R}_p = \mathbf{S}^T \mathbf{S} / N$ converges to a matrix with entries

$$R_{ij} = r_{i-j} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=-N/2}^{(N-1)/2} s_k s_{k-|i-j|}. \quad (71)$$

The solution for $\mathbf{a}$ (given $a_0 = 1$) is obtained by minimizing (68) with respect to $\mathbf{a}$, a straightforward linear system with p equations in p unknowns. An elegant solution can be arrived at using the technique of Lagrange Multipliers, with objective function

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^T \mathbf{R}_p \mathbf{a} + \lambda a_0. \quad (72)$$

Differentiation with respect to $\mathbf{a}$ and solving for λ leads to

$$\lambda = \mathbf{r}_p^T \mathbf{a} \quad (73)$$

where $\mathbf{r}_p$ is the first column of $\mathbf{R}_p$. By substituting the value for λ back into (72), and differentiating with respect to $\mathbf{a}$, it can be shown that the minimum MSE predictor is given by solving

$$\mathbf{R}_p \mathbf{a} = \mathbf{I}_{p,1} \mathbf{a}^T \mathbf{r}_p, \quad (74)$$

where $\mathbf{I}_{p,1}$ is a $p + 1$ by 1 column vector with a 1 in the first position, and zeros elsewhere. This set of equations are known as the Yule-Walker Equations or the Normal Equations. Substitution of (74) into (68) leads to the following expression for the minimum mean squared error for the *optimal* predictor ($\mathbf{a}_p$)

$$\begin{aligned} D_p &= \mathbf{a}_p^T \mathbf{I}_{p,1} \mathbf{a}_p^T \mathbf{r}_p \\ &= \mathbf{a}_p^T \mathbf{r}_p. \end{aligned} \quad (75)$$

For a stochastic process, the first $p + 1$ autocorrelation coefficients are required to obtain the optimal (linear) predictor of order p . For a non-stationary process or a stochastic process in which the autocorrelations must be estimated over a finite set of data, two major techniques are most often used; the Autocorrelation Method or the Covariance Method, and basically differ only in how the data is segmented into a finite length record.

The Covariance Method assumes that N rows of $\mathbf{S}$ are known, where $\mathbf{S}$ can be written as (the Toeplitz matrix)

$$\mathbf{S} = \mathcal{T} \left([s_0 \ s_1 \ \cdots \ s_{N-1}]^T, [s_0 \ s_{-1} \ \cdots \ s_{-p}]^T \right). \quad (76)$$

The Autocorrelation method, first windows the assumed infinite length record into a finite length record. Assuming the windowing is done such that

$$s_n = 0, \ 0 > n > N - 1 \quad (77)$$

the matrix $\mathbf{S}$ ($N + p + 1$ by $p + 1$) is given by

$$\mathbf{S} = \mathcal{T}_l([s_0 \ s_1 \ \cdots \ s_{N-1} \ 0 \ \cdots \ 0]). \quad (78)$$

In both the Autocorrelation method and the Covariance method, $\mathbf{S}$ is Toeplitz. However, the Autocorrelation method has the additional property that $\mathbf{R}_p = \mathbf{S}^T \mathbf{S}$ is also Toeplitz, which leads to a more computationally efficient solution for $\mathbf{a}$.

2.4.3. The Likelihood Ratio

The likelihood ratio (LR) is given by

$$\begin{aligned} d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p) &= \frac{\mathbf{a}^T \mathbf{R}_p \mathbf{a}}{\alpha_p} - 1 \\ &= \frac{\alpha}{\alpha_p} - 1, \end{aligned} \quad (79)$$

where

$$\alpha_p = \min_{\mathbf{a}} D_p = \min_{\mathbf{a}} \alpha \quad (80)$$

is the minimum mean squared error for the predictor computable from $\mathbf{R}_p$ and $\alpha = D_p$ is the mean squared error for a particular predictor $\mathbf{a}$ of order p . The optimal order p predictor is denoted by $\mathbf{a}_p$, thus

$$d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p) = \frac{\mathbf{a}^T \mathbf{R}_p \mathbf{a}}{\mathbf{a}_p^T \mathbf{r}_p} - 1. \quad (81)$$

The likelihood ratio has many of the properties of a distortion measure:

1. the LR is non-negative,

$$d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p) \geq 0; \quad (82)$$

2. at the minimum

$$\min_{\mathbf{a}} d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p) = d_{\text{LR}}(\mathbf{a}_p, \mathbf{R}_p) = 0, \quad (83)$$

and $\mathbf{a} = \mathbf{a}_p$.

Minimizing the LR is equivalent to finding the stationary point where $\mathbf{a} = \mathbf{a}_p$.

The likelihood ratio is not a distance, since the likelihood ratio is not symmetric. For two model filters $A_{1,p}(z)$ and $A_{2,p}(z)$ resulting from autocorrelations $R_{1,p}$ and $R_{2,p}$ respectively, in general

$$d_{\text{LR}}(\mathbf{a}_2, \mathbf{R}_{1,p}) \neq d_{\text{LR}}(\mathbf{a}_1, \mathbf{R}_{2,p}). \quad (84)$$

2.4.4. Spectral Distortion

The root mean square log spectral distortion (SD) between two model filters is given by

$$d_{SD}(A_1(z), A_2(z)) = \sqrt{\frac{1}{n_1 - n_0} \sum_{n=n_0}^{n_1-1} \left[10 \log_{10} \frac{|A_1(e^{j2\pi n/N})|^2}{|A_2(e^{j2\pi n/N})|^2} \right]^2}. \quad (85)$$

Typically [38, 39], n_1 and n_0 correspond to about 3.1 kHz and 125 Hz respectively. An N -point FFT can be used to compute $A_1(e^{j2\pi n/N})$ and $A_2(e^{j2\pi n/N})$ for all n . The values shown in Table 1 were used throughout this thesis.

Assuming that the autocorrelation method is used with, of course, a finite length window (of length N samples), then the Z -transform of s_n is

$$S(z) = \sum_{n=0}^{N-1} s_n z^{-n}. \quad (86)$$

Evaluating $S(z)$ on the unit circle is equivalent to computing

$$S(e^{j\omega}) = \sum_{n=0}^{N-1} s_n e^{-j\omega n}. \quad (87)$$

Similarly, the predictor response on the unit circle is

$$A_p(e^{j\omega}) = \sum_{n=0}^p a_n e^{-j\omega n} \quad (88)$$

By utilizing the properties of the Z -transform, the Z -transform of d (the prediction error) is

$$\begin{aligned} D(e^{j\omega}) &= \sum_{n=0}^{N-1} d_n e^{-j\omega n} \\ &= S(e^{j\omega}) A_p(e^{j\omega}). \end{aligned} \quad (89)$$

Table 1: Parameters used for SD computation.

N	n_0	n_1
256	4	100

Furthermore, from Parseval's Relations

$$\begin{aligned}
 \alpha &= \sum_n d_n^2 \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} D(e^{j\omega}) D^*(e^{j\omega}) d\omega \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |S(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega.
 \end{aligned} \tag{90}$$

As the predictor order tends to infinity, the shape of $1/A(e^{j\omega})$ can exactly match $S(e^{j\omega})$ (by Wold's Decomposition [37]). Thus

$$S(e^{j\omega}) S^*(e^{j\omega}) A(e^{j\omega}) A^*(e^{j\omega}) \tag{91}$$

is a constant, and

$$\begin{aligned}
 \ln \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} |S(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega \right) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln (|S(e^{j\omega})|^2 |A(e^{j\omega})|^2) d\omega \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln (|S(e^{j\omega})|^2) d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln (|A(e^{j\omega})|^2) d\omega \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln (|S(e^{j\omega})|^2) d\omega \\
 &= \ln (\alpha_{\infty}).
 \end{aligned} \tag{92}$$

Substitution of (79) into (90) leads to

$$\begin{aligned}
 d_{\mathbf{LR}}(\mathbf{a}, \mathbf{R}_p) &= \frac{\alpha}{\alpha_p} - 1 \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{|S(e^{j\omega})|^2 |A(e^{j\omega})|^2}{\alpha_p} d\omega - 1 \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{|A(e^{j\omega})|^2}{|A_p(e^{j\omega})|^2} d\omega - 1,
 \end{aligned} \tag{93}$$

since the autocorrelation of the impulse response of $1/A_p(z)$ matches the signal autocorrelation (r_p) at $p + 1$ points. Furthermore, given that

$$\begin{aligned} u &= \exp(\ln(u)) \\ &\approx 1 + \ln(u) + \frac{1}{2}(\ln(u))^2 \end{aligned} \quad (94)$$

(for $u \approx 1$) and

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |A(e^{j\omega})|^2 d\omega = 0, \quad (95)$$

then

$$\begin{aligned} d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p) &\approx \frac{1}{4\pi} \int_{-\pi}^{\pi} \left(\ln |A(e^{j\omega})|^2 - \ln |A_p(e^{j\omega})|^2 \right)^2 d\omega \\ &= \frac{\int_{-\pi}^{\pi} \left(10 \log_{10} |A(e^{j\omega})|^2 - 10 \log_{10} |A_p(e^{j\omega})|^2 \right)^2 d\omega}{400\pi \log_{10}^2(\epsilon)}, \end{aligned} \quad (96)$$

or

$$\begin{aligned} 10 \log_{10}(\epsilon) \sqrt{2d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p)} &\approx \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\ln |A(e^{j\omega})|^2 - \ln |A_p(e^{j\omega})|^2 \right)^2 d\omega} \\ 6.142 \sqrt{d_{\text{LR}}(\mathbf{a}, \mathbf{R}_p)} &\approx d_{\text{SD}}(A(z), A_p(z)). \end{aligned} \quad (97)$$

Thus, for small distortions the square root of the likelihood ratio is (approximately) proportional to the root mean square log spectral distortion.

2.4.5. Long Term Model Filter

Voiced speech is characterized by a quasi-periodic speech signal and a periodic residual (or prediction error). By exploiting the long term correlation, or equivalently, the spectral fine structure, further prediction gains can be realized. The long term inverse filter is typically represented by

$$B(z) = 1 - \sum_{k=-q}^q b_k z^{-(m+k)} \quad (98)$$

where m is a particular pitch period and the b_k s are the long term predictor parameters. The optimal or chosen pitch period is denoted by M . The order of the long term predictor (LTP) is quite large ($m+q$) but many of the coefficients are zero. The filtering of $\mathbf{d}$ (dimension N) through the LTP can be represented by

$$\mathbf{e} = \mathbf{d} - \mathbf{D}\mathbf{b} \quad (99)$$

where

$$\mathbf{d} = [d_0 \quad d_1 \quad d_2 \quad \dots \quad d_{N-1}]^T, \quad (100)$$

$$\mathbf{b} = [b_{-q} \quad \dots \quad b_0 \quad b_1 \quad \dots \quad b_q]^T, \quad (101)$$

and $\mathbf{D}$ is a $2q+1$ by N dimensional Toeplitz matrix with

$$[d_{q-m} \quad d_{q+1-m} \quad d_{2+q-m} \quad \dots \quad d_{N+q-m-1}]^T \quad (102)$$

in the first column and

$$[d_{q-m} \quad d_{q-1-m} \quad \dots \quad d_{-m} \quad \dots \quad d_{-q-m}] \quad (103)$$

in the first row. Minimizing the open loop prediction error or minimizing the energy in $\mathbf{e}$ is equivalent to minimizing

$$\mathbf{e}^T \mathbf{e} = \mathbf{d}^T \mathbf{d} - 2\mathbf{d}^T \mathbf{D}\mathbf{b} + \mathbf{b}^T \mathbf{D}^T \mathbf{D}\mathbf{b} \quad (104)$$

which has the straightforward solution

$$\mathbf{b} = [\mathbf{D}^T \mathbf{D}]^{-1} \mathbf{D}^T \mathbf{d}, \quad (105)$$

which can be solved using Choleski Decomposition.

The minimizing solution over all values of m (typically $20 \leq m \leq 147$), is non linear although solvable by minimization of (104) over all values of m .

2.4.6. Speech Production Model

The speech production model used in this work and in common use in the speech coding community is shown in Fig. 7. The model is consistent with the analysis model developed in the previous sections. The synthetic speech is modelled by a relatively white source driving two all pole time varying linear filters. The long term model filter has a relatively large number of taps, although many of the tap coefficients are zero, and attempts to model the spectral fine structure present in voiced speech. The short term filter models the spectrally smooth structure or formants. An analysis-synthesis scheme is shown in Fig. 8.

2.5. Codebook Excited Linear Prediction (CELP)

2.5.1. CELP Analysis/Synthesis Model

Codebook Excited Linear Prediction (CELP) [40] is a popular analysis by synthesis technique whereby the optimal excitation vector is selected from a codebook to minimize a perceptual weighted mean squared error distortion measure. The analysis by synthesis model used in CELP is shown in Fig. 9.

The original speech is partitioned into frames, and the frames are further partitioned into subframes. The k -th speech sample in the n -th subframe is denoted by $s_{n,k}$, or where no confusion would result it is denoted by simply s_k . Similarly, the k -th synthetic speech sample, codevector, codebook excitation, and synthetic excitation for the l -th codevector is denoted by $\hat{s}_{n,k}^{(l)}$, $v_{n,k}^{(l)}$, $\hat{e}_{n,k}^{(l)}$, and $\hat{d}_{n,k}^{(l)}$ respectively. $G_n^{(l)}$ is the optimal gain for the l -th codevector, $1/B_n(z)$ is the long term model filter at subframe n and $1/A_n(z)$ is the quantized short term model filter at subframe n . Again, the time index n is occasionally dropped if no confusion would result. The distortion measure commonly used is weighted mean squared error (WMSE)- the difference between the original and synthetic

Figure 7: Speech production model. $v_{n,k}$ is the k -th sample of the excitation from the n -th frame. G_n is the gain for the n -th frame. $e_{n,k}$ is the k -th sample of the (scaled) excitation at subframe n and $d_{n,k}$ is the k -th sample of the excitation after the long term model filter at subframe n . Finally, $s_{n,k}$ is the k -th sample of the speech at subframe n .

Figure 8: Analysis-synthesis model. $e_{n,k}$ ($\hat{e}_{n,k}$) is the k -th sample of the (synthetic) excitation at subframe n and $\hat{d}_{n,k}$ ($d_{n,k}$) is the k -th sample of the (synthetic) excitation after the long term model filter at subframe n . Finally, $e_{n,k}$ ($\hat{s}_{n,k}$) is the k -th sample of the (synthetic) speech at subframe n .

Figure 9: The analysis by synthesis nature of CELP. (a) Synthesis: A vector (selected by an index) is chosen from the VQ codebook, scaled, and filtered through a long and short term model filter. (b) Analysis: For each possible index from the VQ codebook, the synthetic speech is computed and compared based on a weighted mean squared error distortion. Thus the term: analysis by synthesis.

speech weighted by

$$W_n(z) = \frac{A_{n,p}(z/\gamma_1)}{A_{n,p}(z/\gamma_2)} = \frac{1 + \sum_{k=1}^p a_{1,k} z^{-k}}{1 + \sum_{k=1}^p a_{2,k} z^{-k}}, \quad (106)$$

where $A_{n,p}(z)$ is the order p unquantized optimal short term model filter at subframe n .

The function of the weighting filter is to exploit the masking properties of the average human auditory system. The weighting filter attempts to obtain a more uniform signal to noise ratio in each particular frequency band. An example of a frequency response of a weighting filter is shown in Fig. 10.

The inclusion of a weighting filter in the CELP structure assumes that the quantization noise is white, which it is not, especially at low rates. Thus, at lower rates the usefulness of the weighting filter is somewhat suspect. More general weighting filters are also possible as discussed in [41, 42, 43]. The more general weighting filters typically include spectral fine structure to ensure adequate periodicity during voiced segments.

2.5.2. Reduced Complexity Search Techniques

The CELP structure is usually rearranged into the form shown in Fig. 11.

The optimal minimum weighted mean squared error (WMSE) encoder for the given CELP structure, and the given codebooks with the specific bit allocations minimizes over all time indices, and over all codebook indices

$$\sum_n \epsilon^{(l_n, m_n)} = \sum_n \left| y_n - u_n - \hat{y}_n^{(m_n)} - \hat{y}_n^{(l_n)} \right|^2. \quad (107)$$

Strictly speaking, the short term weighting filter zero input response u_n is a function of l_{n-k} and m_{n-k} for all $k > 0$. Each term is also a function of the

Figure 10: Frequency response of $\alpha_p/A(e^{j\omega})$, $\alpha_p/W(e^{j\omega})$, and $S(e^{j\omega})$. The weighting filter tends to boost the noise energy in regions where $S(e^{j\omega})$ is large. The conventional weighting filter has a global structure, but very little spectrally fine structure.

Figure 11: CELP structure rearranged. (1)– move location of $W_n(z)$ and (2)– separate the zero state and zero input responses.

short term filter codebook indices. The optimal (minimum WMSE) solution is prohibitively complex and has infinite delay.

Typically, the short term predictor is computed based solely on the original speech (forward adaptive) and not in order to minimize WMSE. Furthermore, to reduce computational complexity and delay, at subframe n , decisions are made on both l_n and m_n . Thus, given $A_n(z)$, and assuming decisions on l_n and m_n must be made at subframe n , the suboptimal solution minimizes

$$\epsilon^{(l,m)} = \left| \mathbf{y} - \mathbf{u} - \hat{\mathbf{y}}^{(m)} - \hat{\mathbf{y}}^{(l)} \right|^2, \quad (108)$$

where the time indices (n) have now been dropped.

Since the influence of samples in the current frame may have a large influence in the next frame, especially for samples near the end of a subframe, subframe overlap can be used to consider the influence of samples in the current subframe on future subframes. Overlap can be implemented by letting the weighted synthesis filters ring past the end of the current subframe. The number of samples of overlap is denoted by K_o .

The zero input responses (ZIR) $\hat{\mathbf{y}}^{(m)}$ and $\hat{\mathbf{y}}^{(l)}$ can be written as a matrix vector product

$$\hat{\mathbf{y}}^{(m)} = \mathbf{H} \hat{\mathbf{d}}^{(m)} \quad (109)$$

and

$$\hat{\mathbf{y}}^{(l)} = \mathbf{H} \hat{\mathbf{d}}^{(l)}, \quad (110)$$

where $\mathbf{H}$ is the impulse response matrix of $W(z)/A(z)$, a $K = K_c + K_o$ by K_c lower triangular Toeplitz matrix with the impulse response $\mathbf{h}$ in the first column. Although $W(z)/A(z)$ is IIR, the impulse response is typically truncated after $K_r (< K_c + K_o)$ samples (for complexity reasons). Thus, the WMSE over a

single subframe is

$$\epsilon^{(l,m)} = \left| \mathbf{y} - \mathbf{u} - \mathbf{H}\hat{\mathbf{d}}^{(m)} - \mathbf{H}\hat{\mathbf{d}}^{(l)} \right|^2. \quad (111)$$

The long term filter $1/B(z)$ is given by

$$\frac{1}{B(z)} = \frac{1}{1 - \sum_{k=-q}^q b_k z^{-(M+k)}}. \quad (112)$$

$B(z)$ is usually determined either closed loop in order to minimize weighted mean squared error before determining the codebook excitation parameters, or open loop as discussed in Section 2.4.5. Closed loop methods are, in general, superior since final weighted mean squared error rather than open loop mean squared error is minimized. In this section only the closed loop methods are discussed.

Consider for a moment the all pole long term filter. The output for a particular value of m can be written as

$$\tilde{d}_k^{(m)} = G^{(l)} v_k^{(l)} + \sum_{j=-q}^q b_j \hat{d}_{k-(m+j)}. \quad (113)$$

If $m \geq K_c + q$ (113) can be rewritten in matrix vector form over a single subframe as

$$\begin{aligned} \hat{\mathbf{d}}^{(m)} &= G^{(l)} \mathbf{v}^{(l)} + \sum_{j=-q}^q b_j^{(m)} \hat{\mathbf{d}}_{-(m+j)} \\ &= G^{(l)} \mathbf{v}^{(l)} + \mathbf{D}^{(m)} \mathbf{b}^{(m)}, \end{aligned} \quad (114)$$

where

$$\mathbf{b}^{(m)} = [b_{-q}^{(m)} \quad \dots \quad b_0^{(m)} \quad \dots \quad b_{+q}^{(m)}]^T, \quad (115)$$

$$\hat{\mathbf{d}}_{-(m+j)} = [\hat{d}_{-(m+j)} \quad \hat{d}_{1-(m+j)} \quad \dots \quad \hat{d}_{K_c-1-(m+j)}]^T, \quad (116)$$

and $\mathbf{D}^{(m)}$ is a K_c by $2q + 1$ dimensional matrix

$$\mathbf{D}^{(m)} = [\hat{\mathbf{d}}_{q-m} \quad \hat{\mathbf{d}}_{q-1-m} \quad \dots \quad \hat{\mathbf{d}}_{-m-q}]. \quad (117)$$

Note that $\mathbf{D}^{(m)}$ does not depend on any values of $\hat{d}_k^{(m)}$ for $k \geq 0$ and (114) can only be written in this form if the constraint ($m \geq K_c + q$) holds. The vector of predictor parameters $\mathbf{b}^{(m)}$ is now written as a function of m since $\mathbf{b}$ will be optimized closed loop for each candidate value of m .

Joint optimization of (111) over m and l is prohibitively complex [44]. Rather than solving the joint problem, a suboptimal sequential approach is usually used whereby the optimal values of m and $\mathbf{b}^{(m)}$ are determined by minimizing (111) assuming $G^{(l)}$ is zero and then the optimal values of l and $G^{(l)}$ are determined based on the optimum values of m and $\mathbf{b}^{(m)}$ as follows:

$$\begin{aligned} \epsilon^{(m)} &= \left| (\mathbf{y} - \mathbf{u}) - \mathbf{H}\mathbf{D}^{(m)}\mathbf{b}^{(m)} \right|^2 \\ &= \mathbf{y}_1^T \mathbf{y}_1 - 2\mathbf{y}_1^T \mathbf{H}\mathbf{D}^{(m)}\mathbf{b}^{(m)} + \mathbf{b}^{(m)T} \mathbf{D}^{(m)T} \mathbf{H}^T \mathbf{H} \mathbf{D}^{(m)} \mathbf{b}^{(m)}, \end{aligned} \quad (118)$$

and

$$\begin{aligned} \epsilon^{(l)} &= \left| \left(\mathbf{y} - \mathbf{u} - \mathbf{H}\mathbf{D}^{(M)}\mathbf{b}^{(M)} \right) - G^{(l)} \mathbf{H}\mathbf{v}^{(l)} \right|^2 \\ &= \epsilon^{(m)} - 2G^{(l)} \mathbf{y}_2^T \mathbf{H}\mathbf{v}^{(l)} + G^{(l)2} \mathbf{v}^{(l)T} \mathbf{H}^T \mathbf{H} \mathbf{v}^{(l)}, \end{aligned} \quad (119)$$

where

$$\mathbf{y}_1 = \mathbf{y} - \mathbf{u} \quad (120)$$

and

$$\mathbf{y}_2 = \mathbf{y} - \mathbf{u} - \mathbf{H}\mathbf{D}^{(M)}\mathbf{b}^{(M)}. \quad (121)$$

The terms $\mathbf{c}_1 = \mathbf{H}^T \mathbf{y}_1$ and $\mathbf{R} = \mathbf{H}^T \mathbf{H}$ do not depend on m or l , and thus may be pre-computed. Both $\mathbf{c}_1 = \mathbf{H}^T \mathbf{y}_1$ and $\mathbf{c}_2 = \mathbf{H}^T \mathbf{y}_2$ can be computed by so-called *backward filtering* [45, 46] which simply takes advantage of the fact that $\mathbf{H}$ is a (Toeplitz) impulse response matrix and thus

$$\mathbf{H}^T \mathbf{y}_2 = \mathbf{H}^T \mathbf{y}_2 = (\mathbf{H} \mathbf{y}_2^r)^r. \quad (122)$$

Thus, $\mathbf{H}^T \mathbf{y}_2$ can be computed by reversing $\mathbf{y}_2$, filtering it through $W(z)/A(z)$ and reversing the result. Equation (118) and (119) can now be rewritten as

$$\epsilon^{(m)} = \mathbf{y}_1^T \mathbf{y}_1 - 2\mathbf{c}_1^T \mathbf{D}^{(m)} \mathbf{b}^{(m)} + \mathbf{b}^{(m)T} \mathbf{D}^{(m)T} \mathbf{R} \mathbf{D}^{(m)} \mathbf{b}^{(m)} \quad (123)$$

and

$$\epsilon^{(l)} = \epsilon^{(m)} - 2G^{(l)} \mathbf{c}_2^T \mathbf{v}^{(l)} + G^{(l)2} \mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)}. \quad (124)$$

To reiterate, l is optimized for the given values of $m = M$ and $\mathbf{b}^{(M)}$. The optimal value of $G^{(l)}$ for a particular index l , determined by minimization of (124) with respect to $G^{(l)}$ is given by

$$G^{(l)} = \frac{\mathbf{c}_2^T \mathbf{v}^{(l)}}{\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)}}. \quad (125)$$

Substitution of (125) into (124) yields

$$\epsilon^{(l)} = \epsilon^{(m)} - \frac{(\mathbf{c}_2^T \mathbf{v}^{(l)})^2}{\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)}}. \quad (126)$$

Similarly, the optimal value of $\mathbf{b}^{(m)}$ can be shown to be

$$\mathbf{b}^{(m)} = \left(\mathbf{D}^{(m)T} \mathbf{R} \mathbf{D}^{(m)} \right)^{-1} \mathbf{D}^{(m)T} \mathbf{c}_1 \quad (127)$$

and

$$\epsilon^{(m)} = \mathbf{y}_1^T \mathbf{y}_1 - 2\mathbf{c}_1^T \mathbf{D}^{(m)} \left(\mathbf{D}^{(m)T} \mathbf{R} \mathbf{D}^{(m)} \right)^{-1} \mathbf{D}^{(m)T} \mathbf{c}_1. \quad (128)$$

Equation (128) is prohibitively complex for $q > 0$ since m is usually one of 128 or 256 values. For $q = 0$, $\mathbf{b}_m$ is a scalar, and a reasonable complexity results. Typically, if a multiple tap long term model filter is used, the search for M assumes $q = 0$, and then the optimal set of quantized parameters is determined by minimizing (123) (with $q \neq 0$).

The search for the optimal long term predictor coefficients and the search for the excitation codebook parameters are very similar (for the case $q = 0$, and $m \geq K_c$). For this reason, the long term model filter is often called an adaptive codebook. However, when a candidate pitch period m is less than the subframe size, (113) contains terms in b_j^2 and cross products and (118) would contain terms in b_j^4 . A solution for the optimal $\mathbf{b}$ is non-linear and rather complex. The problem is usually handled by redefining the operation of the long term filter as

$$\tilde{d}_k^{(m)} = G^{(l)} v_k^{(l)} + \sum_{j=-q}^q b_j \begin{cases} \hat{d}_{k-(m+j)} & k-j < m \\ \hat{d}_{k-(2m+j)} & m \leq k-j < 2m \\ \hat{d}_{k-(3m+j)} & 2m \leq k-j < 3m \\ \dots & \dots \end{cases} \quad (129)$$

or in vector form

$$\tilde{\mathbf{d}}^{(m)} = G^{(l)} \mathbf{v}_k^{(l)} + \mathbf{D}^{(m)} \mathbf{b}^{(m)}, \quad (130)$$

where $\mathbf{D}^{(m)}$ is defined by the last term in (129). This reformulation is equivalent to the definition of the long term filter in (113) for $m \geq K_c + q$. This definition is typically called an adaptive codebook, and the configuration in (113) is called a long term model filter or long term predictor. Note that the long term predictor and adaptive codebook strategies are identical for $m \geq K_c + q$ but differ for $m < K_c + q$.

It has been observed that the adaptive codebook procedure and the long term predictor procedure are subjectively very similar. That is, in terms of subjective quality, nothing is lost, or gained, in using the modified procedure.

2.5.2.1 Adaptive Codebook Search Strategies Consider for a moment the case $q = 0$ in which case $\mathbf{D}^{(m)}$ is a vector (K_c by 1) and $\mathbf{b}^{(m)}$ is a scalar. The value of m which results in minimum weighted MSE can be determined by

minimizing

$$\epsilon^{(m)} = \mathbf{y}_1^T \mathbf{y}_1^T - \frac{(\mathbf{c}_1^T \mathbf{D}^{(m)})^2}{|\mathbf{H} \mathbf{D}^{(m)}|^2} \quad (131)$$

where (for $m \geq K_c$)

$$\mathbf{D}^{(m)} = [\hat{d}_{-m} \quad \hat{d}_{-(m-1)} \quad \dots \quad \hat{d}_{-(m-K_c+2)} \quad \hat{d}_{-(m-K_c+1)}]^T. \quad (132)$$

The denominator term in (131) dominates the computation of $\epsilon^{(m)}$ over all m . To reduce the complexity, the recursive nature of the adaptive codebook is exploited by noting that

$$\begin{aligned} \mathbf{D}^{(m-1)} &= [\hat{d}_{-(m-1)} \quad \hat{d}_{-(m-2)} \quad \dots \quad \hat{d}_{-(m-K_c+1)} \quad \hat{d}_{-(m-K_c)}]^T \\ &= \mathbf{S}_1 \mathbf{D}^{(m)} + \mathbf{I}_{K_c-1, K_c-1} \mathbf{D}^{(m-1)}, \end{aligned} \quad (133)$$

where $\mathbf{S}_j$ is a upper triangular Toeplitz shifting matrix with a single one in the first row in column j and $\mathbf{I}_{j,k}$ is a diagonal masking matrix with a ones on the main diagonal beginning in row j and ending in row k . The filtering operation in the denominator term may be written as

$$\begin{aligned} \hat{\mathbf{y}}^{(m-1)} &= \mathbf{H} \mathbf{D}^{(m-1)} \\ &= \mathbf{H} \mathbf{S}_1 \mathbf{D}^{(m)} + \mathbf{H} \mathbf{I}_{K_c-1, K_c-1} \mathbf{D}^{(m-1)}. \end{aligned} \quad (134)$$

Since $\mathbf{H} \mathbf{S}_1 = \mathbf{S}_1 \mathbf{H} + \mathbf{I}_{K-1, K-1} \mathbf{H} \mathbf{S}_1 - \mathbf{S}_1 \mathbf{H} \mathbf{I}_{0,0}$ it follows that

$$\mathbf{y}^{(m-1)} = \mathbf{S}_1 \hat{\mathbf{y}}^{(m)} + \mathbf{I}_{K-1, K-1} \mathbf{H} \mathbf{S}_1 \mathbf{D}^{(m)} - \mathbf{S}_1 \mathbf{h} \hat{d}_{-m} + \mathbf{S}_{K_c-1}^T \mathbf{h} \hat{d}_{K_c-m}. \quad (135)$$

The first term requires no operations since it is just a shifting operation, and the second term requires $K_r - K_o$ operations. The third term requires K_r operations and the final (fourth) term requires K_o operations. Computing magnitude squared requires K operations and thus the total operation count for computing the denominator of (131) is $K + 2K_r$. This technique is referred to as *end correction* [47, 30, 48].

A more efficient solution (known as *energy recursion* [30, 48]) is arrived at by noting that if $\mathbf{R} = \mathbf{H}^T \mathbf{H}$ is Toeplitz, then $\mathbf{x}^T \mathbf{R} \mathbf{x}$ can be computed efficiently using

$$\mathbf{x}^T \mathbf{R} \mathbf{x} = 2\mathbf{r}_r^T \mathbf{r}, \quad (136)$$

where $\mathbf{r}$ is the first column of $\mathbf{R}$ and $\mathbf{r}_r$ is the modified autocorrelation with elements given by

$$r_{r,k} = \begin{cases} \frac{1}{2} \sum_{i=0}^{K_r-1} x_i^2 & k=0 \\ \sum_{i=0}^{K_r-1-k} x_i x_{i+k} & 0 < k < K_r. \end{cases} \quad (137)$$

Typically, $\mathbf{H}^T \mathbf{H}$ is approximated by $\mathbf{R}$ where the elements of $\mathbf{R}$ are given by

$$R_{i,i+k} = r_k = \sum_{t=0}^{K_r-1} h_t h_{t+k}. \quad (138)$$

Note that

$$\mathbf{r} = (\mathbf{H} \mathbf{h}^T)^T, \quad (139)$$

where

$$\mathbf{r}^T = [\mathbf{r}_0 \quad \mathbf{r}_1 \quad \dots \quad \mathbf{r}_{K_r-1}] \quad (140)$$

and

$$\mathbf{h}^T = [\mathbf{h}_0 \quad \mathbf{h}_1 \quad \dots \quad \mathbf{h}_{K_r-1}]. \quad (141)$$

A recursive solution is again used by noting that the modified autocorrelation of $\mathbf{D}^{(m-1)}$ can be computed based on the modified autocorrelation of $\mathbf{D}^{(m)}$ (for

$k \neq 0$)

$$\begin{aligned}
 r_{d,k}^{(m-1)} &= \sum_{i=0}^{K_c-1-k} \hat{d}_{i+1-m} \hat{d}_{k+i+1-m} \\
 &= \sum_{i=1}^{K_c-k} \hat{d}_{i-m} \hat{d}_{k+i-m} \\
 &= \hat{d}_{K_c-k-m} \hat{d}_{K_c-m} - \hat{d}_{-m} \hat{d}_{k-m} + \sum_{i=0}^{K_c-k-1} \hat{d}_{i-m} \hat{d}_{k+i-m} \\
 &= \hat{d}_{K_c-k-m} \hat{d}_{K_c-m} - \hat{d}_{-m} \hat{d}_{k-m} + r_{d,k}^{(m)},
 \end{aligned} \tag{142}$$

and for $k = 0$

$$\begin{aligned}
 r_{d,0}^{(m-1)} &= \frac{1}{2} \sum_{i=0}^{K_c-1} \hat{d}_{i+1-m}^2 \\
 &= \frac{1}{2} \sum_{i=1}^{K_c} \hat{d}_{i-m}^2 \\
 &= \frac{1}{2} \hat{d}_{K_c-m}^2 - \frac{1}{2} \hat{d}_{-m}^2 + \frac{1}{2} \sum_{i=0}^{K_c-1} \hat{d}_{i-m}^2 \\
 &= \frac{1}{2} \hat{d}_{K_c-m}^2 - \frac{1}{2} \hat{d}_{-m}^2 + r_{d,0}^{(m)}.
 \end{aligned} \tag{143}$$

Thus

$$\begin{aligned}
 \mathbf{D}^{(m-1)T} \mathbf{R} \mathbf{D}^{(m-1)} &= 2 \sum_{k=0}^{K_c-1} r_k r_{d,k}^{(m-1)} \\
 &= \mathbf{D}^{(m)T} \mathbf{R} \mathbf{D}^{(m)} + r_0 \left(\hat{d}_{K_c-m}^2 - \hat{d}_{-m}^2 \right) + \\
 &\quad 2 \sum_{k=1}^{K_c-1} r_k \left(\hat{d}_{K_c-k-m} \hat{d}_{K_c-m} - \hat{d}_{k-m} \hat{d}_{-m} \right)
 \end{aligned} \tag{144}$$

which requires only $2K_r$ operations for each value of m .

The case $m < K_c$ can be treated similarly. For $q \neq 0$, similar complexity reduction techniques can be arrived at. However, the resulting equations (which entail a $2q + 1$ by $2q + 1$ matrix inversion) are too complex to be considered

seriously. Rather, a suboptimal solution whereby the optimal adaptive codebook lag (M) is determined assuming $q = 0$, and then the multi-tap coefficients are determined to minimize

$$\epsilon^{(M)} = \mathbf{y}_1^T \mathbf{y}_1 - 2\mathbf{c}_1^T \mathbf{D}^{(M)} \mathbf{b}^{(M)} + \mathbf{b}^{(M)T} \mathbf{D}^{(M)T} \mathbf{H}^T \mathbf{H} \mathbf{D}^{(M)} \mathbf{b}^{(M)}. \quad (145)$$

This equation can be efficiently computed by taking advantage of the Toeplitz nature of both $\mathbf{H}$, $\mathbf{D}^{(m)}$ and the forced Toeplitz structure of $\mathbf{R} = \mathbf{H}^T \mathbf{H}$.

To reduce both the complexity and rate required for transmitting the adaptive codebook index, an open loop/ closed loop procedure may be used for the adaptive codebook search [2]. Essentially, a low complexity technique is used to determine the pitch period *open loop* on a frame by frame basis, and then the adaptive codebook index and gain are optimized *closed loop* on a subframe by subframe basis with the added constraint that the adaptive codebook index must be near to the open loop pitch period estimate. In [2] the open loop pitch period was computed over a 20 msec frame, and the closed loop pitch period was constrained to be within $+1, 0, 1$ or 2 of the open loop value. This constraint reduces both the bandwidth required for sending the pitch information and the complexity in calculating this information. The algorithm was subsequently studied in [49] and termed *restrictive pitch deviation coding*.

2.5.2.2 Stochastic Codebook Search Strategies The search for the optimal codebook index from the stochastic codebook minimizes (126), (rewritten for ease of reading purposes)

$$\epsilon^{(l)} = \epsilon^{(M)} - \frac{(\mathbf{c}_2^T \mathbf{v}^{(l)})^2}{\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)}} \quad (146)$$

which is equivalent to maximizing the second term since $\epsilon^{(M)}$ is fixed, assuming the adaptive codebook parameters are already computed. The computational

complexity of both the numerator and denominator terms (C_N and C_D respectively) of the second term of (146) is given by

$$\begin{aligned} C_D &\approx K_c^2 \\ C_N &= K_c. \end{aligned} \quad (147)$$

This straightforward method of computation, assuming or forcing no structure on $\mathbf{R}$ (other than the symmetric nature of $\mathbf{R}$) is sometimes termed the covariance approach.

Complexity reduction can be accomplished by making simplifying assumptions on the Toeplitz nature of $\mathbf{R} = \mathbf{H}^T \mathbf{H}$, and/or by adding sufficient structure to the codebook in order to facilitate fast search algorithms. If $\mathbf{R}$ is given by (138), then

$$\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)} = 2\mathbf{r}^T \mathbf{r}^{(l)} \quad (148)$$

where

$$r_k^{(l)} = \begin{cases} \frac{1}{2} \sum_{i=0}^{K_c-1} \left(v_i^{(l)} \right)^2 & k = 0 \\ \sum_{i=0}^{K_c-1-k} v_i^{(l)} v_{i+k}^{(l)} & k \neq 0 \end{cases} \quad (149)$$

and where $r_k = R_{1,1 \pm k}$. By storing both the codebook, and the modified autocorrelation $\mathbf{r}^{(l)}$ is the denominator computational complexity (per codevector) is

$$C_D = K_c. \quad (150)$$

This simplification, is usually termed the Autocorrelation Approach after [50]. Including the pre-computation of $\mathbf{R}$ and $\mathbf{c}_2$, the total computational complexity is

$$C = 2K_c(L_c + p). \quad (151)$$

The CELP search procedure (defined by (119)) can be modified into an equivalent transform domain approach by transforming the weighted error by an orthonormal transformation matrix $\mathbf{F}$ (where $\mathbf{F}^T \mathbf{F} = \mathbf{I}$). The weighted error is

$$\begin{aligned}\epsilon^{(l)} &= \left| \mathbf{F} \left(\mathbf{y} - \mathbf{u} - \mathbf{H} \mathbf{D}^{(M)} \mathbf{b}^{(M)} \right) - G^{(l)} \mathbf{F} \mathbf{H} \mathbf{v}^{(l)} \right|^2 \\ &= \epsilon^{(m)} - 2G^{(l)} \mathbf{y}_2^T \mathbf{F}^T \mathbf{F} \mathbf{H} \mathbf{v}^{(l)} + G^{(l)2} \mathbf{v}^{(l)T} \mathbf{H}^T \mathbf{F}^T \mathbf{F} \mathbf{H} \mathbf{v}^{(l)}.\end{aligned}\quad (152)$$

If the entries of $\mathbf{F}$ are given by

$$F_{ik} = \frac{\exp(-j2\pi ik/K_c)}{\sqrt{K_c}}, \quad (153)$$

(i.e. a simple scaled FFT), then the orthonormal property holds and a frequency domain approach [50] results as shown in Fig. 12. By using this approach, the filtering operations can be represented by a multiplication operation rather than a convolution operation and good complexity reduction results. The codewords are stored in both a time domain and transform domain form.

Alternatively, the codevectors may be represented by a simple transformation, such as

$$\mathbf{v}^{(l)} = \mathbf{C} \mathbf{b}^{(l)} \quad (154a)$$

where $\mathbf{b}^{(l)}$ is a low dimensional (possibly binary) L_c -level code, and $\mathbf{C}$ is a matrix of basis vectors [26, 27, 28, 43, 51]. Essentially the codebook is comprised of the sum of basis vectors, with the scaling provided by the code. For example, $\mathbf{b}^{(l)}$ may be from the L_c -level straight binary Gray code (with ± 1 valued entries), with $B_c = \log_2(L_c)$ bits per vector. Thus, $\mathbf{C}$ has dimension K_c by B_c , and $\mathbf{b}^{(l)}$ has dimension B_c by 1. Equation (119) can be rewritten as

$$\begin{aligned}\epsilon^{(l)} &= \epsilon^{(m)} - 2G^{(l)} \mathbf{y}_2^T \mathbf{H} \mathbf{C} \mathbf{b}^{(l)} + G^{(l)2} \mathbf{b}^{(l)T} \mathbf{C}^T \mathbf{H}^T \mathbf{H} \mathbf{C} \mathbf{b}^{(l)} \\ &= \epsilon^{(m)} - 2G^{(l)} \mathbf{c}_v^T \mathbf{b}^{(l)} + G^{(l)2} \mathbf{b}^{(l)T} \mathbf{R}_v \mathbf{b}^{(l)},\end{aligned}\quad (155)$$

Figure 12: Analysis by synthesis transform coder. $F(z)$ is the filterbank representation of $\mathbf{F}$. Y_n , U_n , $\hat{Y}_n^{(m)}$, and $\hat{Y}_n^{(l)}$ are the filterbank outputs at time $K_c - 1$.

and assuming the optimal gain is chosen

$$\epsilon^{(l)} = \epsilon^{(m)} - \frac{(\mathbf{c}_v^T \mathbf{b}^{(l)})^2}{\mathbf{b}^{(l)T} \mathbf{R}_v \mathbf{b}^{(l)}}. \quad (156)$$

Since the codewords are defined by a Gray code (i.e. only one entry at position k in codeword $\mathbf{b}^{(l)}$ is changed from codeword $\mathbf{b}^{(l-1)}$), the recursive computation of the numerator term $\mathbf{c}_v^T \mathbf{b}^{(l)}$ requires only a single addition and the denominator term can likewise be efficiently computed by noting that

$$\begin{aligned} \frac{1}{4} \mathbf{b}^{(l)T} \mathbf{R}_v \mathbf{b}^{(l)} &= \frac{1}{4} \sum_{i=0}^{B_c-1} \sum_{j=0}^{B_c-1} b_i^{(l)} b_j^{(l)} R_{v,i,j} \\ &= \frac{1}{4} \mathbf{b}^{(l-1)T} \mathbf{R}_v \mathbf{b}^{(l-1)} + b_k^{(l)} \sum_{j=0, j \neq k}^{B_c-1} b_j^{(l)} R_{v,k,j}. \end{aligned} \quad (157)$$

The per codevector computational complexity of these *basis vector* structured codes, used in VSELP (Vector Sum Excited Linear Prediction) is approximately

$$\begin{aligned} C_D &\approx B_c \\ C_N &= 1 \end{aligned} \quad (158)$$

and the memory size is also greatly reduced since only the basis vectors are stored. The pre-computation of $\mathbf{c}_v$ and $\mathbf{R}_v$ requires $K_c p + K_c B_c$ and $K_c B_c (B_c + 1)/2 + K_c B_c p$ operations respectively for a total complexity of

$$C_v = (B_c + 1)L_c + K_c p + K_c B_c \left(1 + \frac{1}{2} (B_c + 1) + p \right). \quad (159)$$

In the previous section, it was shown that since the adaptive codebook has a shift (or overlapping) symmetry, the computation of the denominator term is simplified. The stochastic codebook can also be written as an overlapping codebook with the l -th codevector given by

$$\mathbf{v}^{(l)} = \mathbf{B}^{(l)} \mathbf{c}, \quad (160)$$

where $\mathbf{c}$ is the overlapping codebook (a $L_c + K_c - 1$ dimensional vector) and $\mathbf{B}^{(l)}$ is a K_c by $L_c + K_c - 1$ dimensional upper triangular Toeplitz matrix with the first row having a one in position l and zeros elsewhere. It was determined previously that assuming $\mathbf{R}$ was Toeplitz and symmetric that

$$\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)} = 2 \mathbf{r}^T \mathbf{r}^{(l)} \quad (161)$$

where

$$r_k^{(l)} = \begin{cases} \frac{1}{2} \sum_{i=0}^{K_c-1} \left(v_i^{(l)} \right)^2 & k = 0 \\ \sum_{i=0}^{K_c-1-k} v_i^{(l)} v_{i+k}^{(l)} & k \neq 0. \end{cases} \quad (162)$$

With the definition in (160)

$$\begin{aligned} r_k^{(l)} &= \begin{cases} \frac{1}{2} \sum_{i=0}^{K_c-1} (c_{i+l})^2 & k = 0 \\ \sum_{i=0}^{K_c-1-k} c_{i+l} c_{i+k+l} & k \neq 0 \end{cases} \\ &= \begin{cases} r_0^{(l-1)} - \frac{1}{2} (c_{l-1})^2 + \frac{1}{2} (c_{l+K_c-1})^2 & k = 0 \\ r_k^{(l-1)} - c_{l-1} c_{l+k-1} + c_{l+K_c-k-1} c_{l+K_c-1} & k \neq 0 \end{cases} \end{aligned} \quad (163)$$

and thus

$$\begin{aligned} \mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)} &= \mathbf{v}^{(l-1)T} \mathbf{R} \mathbf{v}^{(l-1)} - c_{l-1} \left(\frac{1}{2} r_0 c_{l-1} + \sum_{k=1}^{K_c} r_k c_{l+k-1} \right) \\ &\quad + c_{l+K_c-1} \left(\frac{1}{2} r_0 c_{l+K_c-1} + \sum_{k=1}^{K_c} r_k c_{l+K_c-k-1} \right). \end{aligned} \quad (164)$$

The recursive nature of computing $\mathbf{r}^{(l)}$ offers no distinct advantages unless more structure is imposed on the codebook (since by storing the codevector autocorrelations only pre-computation of $\mathbf{r}^{(l)}$ is necessary). By forcing the codevector ($\mathbf{c}$) to have many zero elements the recursive computation of (163) can be greatly simplified. That is, if both c_{l-1} and c_{l+K_c-1} are zero then $\mathbf{v}^{(l)T} \mathbf{R} \mathbf{v}^{(l)} = \mathbf{v}^{(l-1)T} \mathbf{R} \mathbf{v}^{(l-1)}$.

An alternative low complexity technique first applied to CELP in [52] is arrived at by structuring the excitation codebook into a multi-stage codebook

Figure 13: Multi-stage VQ of the CELP excitation (MS-CELP) with T stages. It is assumed that there are a fixed number of codevectors (L_i) in each stage.

configuration as shown in Fig. 13. Note that the configuration of Fig. 13 is slightly different (less constrained) than that of [52] in which separate gains were applied to each excitation codebook. Furthermore, in [52] a suboptimal sequential search procedure was used which may severely limit the performance. In this research, the scheme in Fig. 13 was used in conjunction with an M - L search procedure for the following reasons:

1. Only a single gain term is used which is believed to be superior to the multiple gain configuration of [52] since less structure is imposed on the codebooks.
2. The performance penalty of MSVQ with a sequential search procedure can be severe. In this research an M - L search procedure is applied to the MS-CELP structure.
3. The adaptive codebook search may be included in the M - L search procedure.

The excitation codevector is constructed by choosing a codevector from each excitation codebook as shown in Fig. 13

$$\mathbf{v}^{(l_0, l_1, \dots, l_{T_c-1})} = \mathbf{v}_0^{(l_0)} + \mathbf{v}_1^{(l_1)} + \dots + \mathbf{v}_{T_c-1}^{(l_{T_c-1})}. \quad (165)$$

The sequential search procedure first determines the first index to minimize the CELP distortion. The first index is held fixed and the second index is optimized for the known excitation in the first stage. The remaining indices are optimized in a likewise manner. Clearly, this is a suboptimal search procedure since the ensemble of possible reconstruction vectors should be investigated. Another suboptimal approach, the M - L search procedure, maintains M_c candidate indices from the first stage and optimizes the second stage index over all M_c possibilities from the first stage. After the second stage, M_c index pairs are maintained and the third

index is optimized while considering all M_c previous candidate pairs. Subsequent stages are added in a likewise manner. Clearly, the sequential search procedure is a degenerate case of the M - L procedure whereby $M_c = 1$.

The issue of gain quantization is rather more complex. In the first stage, the optimal gain is computed for the assumed first index. During the second stage, the gain is fixed during the search procedure and optimized *after* the second index is determined (for complexity reasons). Note that the M - L search procedure is equivalent to the full search procedure if $M_c = L_c^T$.

In [53] a rather different (two stage) search was used whereby a set of candidate indices was first determined from each codebook independently of the other codebook, and then the optimal pair over the subset of indices was chosen to minimize the distortion. For a small number of candidates, this procedure is expected to perform poorly since the choice of the second candidate indices does not consider the first set of candidate indices (as with M - L). However, at low rates, whereby the error between the target and synthetic vector is quite large, this heuristic approach may also work quite well. The techniques developed in [53] are not considered further.

The search for the first index (l_0) minimizes (from (146))

$$\epsilon^{(l_0)} = \epsilon^{(m)} - \frac{\left(\mathbf{c}_2^T \mathbf{v}_0^{(l_0)}\right)^2}{\mathbf{v}_0^{(l_0)T} \mathbf{R} \mathbf{v}_0^{(l_0)}}, \quad (166)$$

where the gain is computed (from (125)) as

$$G^{(l_0)} = \frac{\mathbf{c}_2^T \mathbf{v}_0^{(l_0)}}{\mathbf{v}_0^{(l_0)T} \mathbf{R} \mathbf{v}_0^{(l_0)}}. \quad (167)$$

A set of the best M_c indices (the M_c indices resulting in minimum WMSE from (166)) are maintained. The second stage keeps the gain fixed from the first stage,

and minimizes (from (124))

$$\begin{aligned}
 \epsilon^{(l_0, l_1)} &= \epsilon^{(m)} - 2G^{(l_0)} \mathbf{c}_2^T \left(\mathbf{v}_0^{(l_0)} + \mathbf{v}_1^{(l_1)} \right) + G^{(l_0)2} \left(\mathbf{v}_0^{(l_0)} + \mathbf{v}_1^{(l_1)} \right)^T \mathbf{R} \left(\mathbf{v}_0^{(l_0)} + \mathbf{v}_1^{(l_1)} \right) \\
 &= \epsilon^{(m)} - 2G^{(l_0)} \mathbf{c}_2^T \mathbf{v}_0^{(l_0)} + G^{(l_0)2} \mathbf{v}_0^{(l_0)T} \mathbf{R} \mathbf{v}_0^{(l_0)} - \\
 &\quad 2G^{(l_0)} \mathbf{v}_1^{(l_1)T} \left(\mathbf{c}_2 - G^{(l_0)} \mathbf{R} \mathbf{v}_0^{(l_0)} \right) + G^{(l_0)2} \mathbf{v}_1^{(l_1)T} \mathbf{R} \mathbf{v}_1^{(l_1)} \\
 &= \epsilon^{(l_0)} - 2G^{(l_0)} \mathbf{v}_1^{(l_1)T} \mathbf{c}_{2,1} + G^{(l_0)2} \mathbf{v}_1^{(l_1)T} \mathbf{R} \mathbf{v}_1^{(l_1)},
 \end{aligned} \tag{168}$$

where

$$\epsilon^{(l_0)} = \epsilon^{(m)} - 2G^{(l_0)} \mathbf{c}_2^T \mathbf{v}_0^{(l_0)} + G^{(l_0)2} \mathbf{v}_0^{(l_0)T} \mathbf{R} \mathbf{v}_0^{(l_0)} \tag{169}$$

and

$$\begin{aligned}
 \mathbf{c}_{2,1} &= \mathbf{c}_2 - G^{(l_0)} \mathbf{R} \mathbf{v}_0^{(l_0)} \\
 &= \mathbf{H}^T \left(\mathbf{y}_2 - G^{(l_0)} \mathbf{H} \mathbf{v}_0^{(l_0)} \right).
 \end{aligned} \tag{170}$$

The term $\epsilon^{(l_0)}$ is a by-product of the search through the first codebook. The complexity of pre-computation (of $\mathbf{c}_2$ and $\mathbf{R}$) is, as before, equal to $2K_c p$ (assuming an Autocorrelation Approach is used). The complexity of the search through the first codebook (numerator and denominator of (167)) is

$$C_N = K_c \tag{171}$$

$$C_D = K_c. \tag{172}$$

The complexity of pre-computation before the second stage search (computation of (170)) is (for each candidate index)

$$C = 2K_c p \tag{173}$$

and the complexity of searching through the second codebook is again

$$C_N = K_c \tag{174}$$

$$C_D = K_c \quad (175)$$

(for each candidate index). The total complexity, for T_c stages is thus

$$C_M = 2K_c(p + L_c + (T_c - 1)M_c(p + L_c)). \quad (176)$$

Note that each stage codebook may in turn have significant structure to enable even faster search procedures.

2.6. Multi-Pulse Excited Linear Predictive Coding (MPE)

Multi-pulse LPC was one of the first techniques to apply analysis by synthesis coding to the excitation in block based LP coding techniques. Both MPE and CELP algorithms have typically outperformed delayed decision tree coding systems, which generally cannot delay an appropriate length of time before a decision is made due to the exponential growth in memory size and computational complexity with increasing delay.

2.6.1. The Multi-pulse Excitation Model

Multi-pulse is essentially a CELP algorithm whereby the multi-pulse excitation codevector is a sparse vector with J non-zero values in each K_c dimensional subframe. The k -th sample from the excitation codevector is given by

$$\hat{e}_k^{(l)} = \sum_{j=0}^{J-1} c_j^{(l)} b_{k-l_j} \quad (177)$$

or in vector notation as

$$\begin{aligned} \hat{\mathbf{e}}^{(l)} &= \sum_{j=0}^{J-1} c_j^{(l)} \mathbf{v}^{(l_j)} \\ &= \mathbf{V}^{(l)} \mathbf{G}^{(l)} \end{aligned} \quad (178)$$

where δ_n is the dirac-delta function

$$\delta_n = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (179)$$

$\mathbf{l}_J$ is a vector of pulse positions

$$\mathbf{l}_J = [l_0, l_1, \dots, l_{J-1}]^T \quad (180)$$

and $G_j^{(l_j)}$ is the gain of the j -th non-zero pulse. The K_c dimensional vector $\mathbf{v}^{(l_j)}$ is sparse with a single non zero (unity) sample in location l_j . The K_c by J matrix $\mathbf{V}^{(l_j)}$ has $\mathbf{v}^{(l_j)}$ in its j -th column and the J dimensional column vector $\mathbf{G}^{(l_j)}$ has the j -th gain $G_j^{(l_j)}$ in the j -th row.

Both the positions ($\mathbf{l}_J$) and the gain vector $\mathbf{G}^{(l_j)}$ are required to reconstruct the excitation. For high quality coding, typically 10 percent non-zero pulses are necessary. Thus, a relatively high rate is required for the multi-pulse algorithm.

The solution for the optimal pulse positions and amplitudes is a non-linear combinatorial optimization problem whereby the optimal solution may only be found, in general, by trying all possible combinations of pulse positions. Once the optimal positions, or a candidate set of positions is determined, the computation of the pulse amplitude vector is relatively straightforward.

2.6.2. Multi-pulse Sequential Search Procedure

The multi-pulse sequential search procedure (MPE_{SEQ}) begins by determining a single pulse position ($\mathbf{l}_1$) and a single amplitude to minimize (from (124))

$$\epsilon^{(l_1)} = \epsilon^{(m)} - 2\mathbf{c}^{(0)T}\hat{\mathbf{e}}^{(l_1)} + \hat{\mathbf{e}}^{(l_1)T}\mathbf{R}\hat{\mathbf{e}}^{(l_1)} \quad (181a)$$

$$= \epsilon^{(m)} - 2G_0^{(l_0)}\mathbf{c}^{(0)T}\mathbf{v}^{(l_0)} + G_0^{(l_0)2}\mathbf{v}^{(l_0)T}\mathbf{R}\mathbf{v}^{(l_0)} \quad (181b)$$

$$= \epsilon^{(m)} - 2G_0^{(l_0)}c_{l_0}^{(0)} + G_0^{(l_0)2}R_{l_0,l_0}. \quad (181c)$$

Minimizing (181c) (a scalar equation) for the unknown $G_0^{(l_0)}$ leads to

$$G_0^{(l_0)} = \frac{c_{l_0}^{(0)}}{R_{l_0,l_0}} \quad (182)$$

and

$$\epsilon^{(1)} = \epsilon^{(m)} - \frac{\left(c_{l_0}^{(0)}\right)^2}{R_{l_0, l_0}}. \quad (183)$$

The first stage tries all values of l_0 to find the minimum of (183).

Subsequent stages are added recursively, based on previous stages. From (178) the excitation for the $i + 1$ -st stage is

$$\begin{aligned} \hat{\mathbf{e}}^{(1, i)} &= \sum_{j=0}^i G_j^{(l_i)} \mathbf{v}^{(l_i)} \\ &= G_i^{(l_i)} \mathbf{v}^{(l_i)} + \sum_{j=0}^{i-1} G_j^{(l_i)} \mathbf{v}^{(l_i)} \\ &= G_i^{(l_i)} \mathbf{v}^{(l_i)} + \hat{\mathbf{e}}^{(1, i)}, \end{aligned} \quad (184)$$

and thus

$$\begin{aligned} \epsilon^{(1, i)} &= \epsilon^{(m)} - 2\mathbf{c}^{(0)T} \hat{\mathbf{e}}^{(1, i)} + \hat{\mathbf{e}}^{(1, i)T} \mathbf{R} \hat{\mathbf{e}}^{(1, i)} \\ &= \epsilon^{(m)} - 2\mathbf{c}^{(0)T} \left(\hat{\mathbf{e}}^{(1, i)} + G_i^{(l_i)} \mathbf{v}^{(l_i)} \right) + \\ &\quad \left(\hat{\mathbf{e}}^{(1, i)} + G_i^{(l_i)} \mathbf{v}^{(l_i)} \right)^T \mathbf{R} \left(\hat{\mathbf{e}}^{(1, i)} + G_i^{(l_i)} \mathbf{v}^{(l_i)} \right) \\ &= \epsilon^{(m)} - 2\mathbf{c}^{(0)T} \hat{\mathbf{e}}^{(1, i)} - 2G_i^{(l_i)} \mathbf{v}^{(l_i)T} \left(\mathbf{c}^{(0)} - \mathbf{R} \hat{\mathbf{e}}^{(1, i)} \right) + \hat{\mathbf{e}}^{(1, i)T} \mathbf{R} \hat{\mathbf{e}}^{(1, i)} \\ &\quad + G_i^{(l_i)2} \mathbf{v}^{(l_i)T} \mathbf{R} \mathbf{v}^{(l_i)} \\ &= \epsilon^{(1, i)} - 2G_i^{(l_i)} \mathbf{c}^{(i)T} \mathbf{v}^{(l_i)} + G_i^{(l_i)2} \mathbf{v}^{(l_i)T} \mathbf{R} \mathbf{v}^{(l_i)} \\ &= \epsilon^{(1, i)} - \frac{\left(c_{l_i}^{(i)}\right)^2}{R_{l_i, l_i}} \end{aligned} \quad (185)$$

where

$$\mathbf{c}^{(i)} = \mathbf{c}^{(0)} - \mathbf{R} \hat{\mathbf{e}}^{(1, i)}.$$

Once the $i + 1$ -st pulse position is found, all amplitudes ($\mathbf{G}^{(1, i+1)}$) can be optimized by solving

$$\begin{aligned} \epsilon^{(1, i+1)} &= \epsilon^{(m)} - 2\mathbf{c}^{(0)T} \hat{\mathbf{e}}^{(1, i+1)} + \hat{\mathbf{e}}^{(1, i+1)T} \mathbf{R} \hat{\mathbf{e}}^{(1, i+1)} \\ &= \epsilon^{(m)} - 2\mathbf{c}^{(0)T} \mathbf{V}^{(1, i+1)} \mathbf{G}^{(1, i+1)} + \mathbf{G}^{(1, i+1)T} \mathbf{V}^{(1, i+1)T} \mathbf{R} \mathbf{V}^{(1, i+1)} \mathbf{G}^{(1, i+1)} \end{aligned} \quad (187)$$

which is a relatively straightforward matrix equation.

2.6.3. Multi-pulse Tree Searched Procedure

The multi-pulse tree search procedure (MPE_{TREE}) is similar to the sequential approach but rather than keeping only a single set of candidates positions from one stage to the next, M_m candidates are maintained. The multi-pulse tree searched procedure with M_m candidates is denoted by ($\text{MPE}_{\text{TREE}}^{(M_m)}$).

The procedure must ensure that different branches do not arrive at the same pulse positions. For example, if the optimal pulse position in stage 1 is p_1 and the next to optimal position is p_2 , and if in the second stage (branch 1) p_2 is chosen as the second pulse position, then branch two cannot choose p_1 since the pulse positions of the two branches will be identical. This situation is shown in Fig. 14 (with $M_m = 2$). Likewise, in branch 2-1-2 or 2-1-1, p_1 could not be chosen if p_2 was chosen in branch 1-2-1 or in 1-2-2.

The MPE_{TREE} procedure is not guaranteed to perform as well as the multi-pulse sequential procedure (for a particular subframe) since the tree searched multi-pulse procedure doesn't necessarily branch out along the same single path as the multi-pulse sequential procedure. However, it is expected that in practice the multi-pulse tree searched procedure should be superior since it investigates more branches than the sequential procedure.

2.6.4. Multi-pulse Simulated Annealing Algorithm

Given a starting configuration of pulse positions ($\mathbf{l}_J$), chosen randomly or chosen using a sequential multi-pulse algorithm, simulated annealing [54, 55, 56] can be used to find (theoretically) the global optimum of the placement of the non-zero pulse positions. In practice, a relatively good suboptimal solution is typically found.

Three key requirements are necessary for a simulated annealing algorithm [55]:

Figure 14: MPE tree search procedure. After the first stage, the two minimum WMSE pulse positions are determined (p_1 and p_2). In the second stage, the (p_1, p_2) combination is selected in branch 1-1 and need not be considered in branch 2-2. After the second stage, the $M_m (= 2)$ minimum WMSE sets of pulse positions are kept for subsequent stages. The M_m candidates after the second stage are chosen to minimize $\epsilon^{(p_1, p_2)}$

2

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

1. a random generator of moves from one set of parameters to another (probabilistically modify each pulse position from one position to another unoccupied position);
2. an objective function ($\epsilon^{(l_j)}$); and
3. an annealing schedule for the effective temperature (constant N_f iterations, exponentially decreasing for N_f iterations).

An analogy is made with the behavior of systems in thermal equilibrium at a finite temperature. As the temperature is decreased, the magnitude of the random perturbations is decreased and the system *freezes* into a solid. If the temperature is reduced too rapidly, the solid may have no crystalline structure. A near optimal annealing schedule, (a temperature function) is determined such that with a very high degree of probability the liquid when cooled on this schedule will form into a crystalline structure.

In the MPE search procedure, a random generator of moves from one state of non-zero pulse positions to a new state of pulse positions introduces a perturbation into the system. The objective function determines the increase or decrease in the cost of this perturbation. If the perturbation results in a decrease in the objective function the change is accepted, otherwise the change is probabilistically accepted, depending on the increase in the cost function and the effective temperature.

The random generator of moves was chosen such that each pulse position ($l_j, 0 \leq j < J$) was changed randomly and independently with a probability equal to

$$P_j(i) = \begin{cases} \alpha & i < N_f \\ \alpha \exp\left(\frac{N_f - i}{\kappa_p}\right) & i \geq N_f \end{cases} \quad (188)$$

where i is the iteration index and κ_p is a constant which determines the rate at which the magnitude of the random perturbations are decreased with decreasing

temperature (or increasing time). The change is only accepted if the new pulse position is unoccupied.

The annealing schedule is defined by

$$t = \begin{cases} t_0 & i < N_f \\ t_0 \exp\left(\frac{N_f - i}{\kappa_t}\right) & i \geq N_f \end{cases} \quad (189)$$

where t_0 is the initial equilibrium temperature and κ_t is a constant which determines the rate at which the temperature is decreased.

At iteration i a random perturbation is made, and the quantity

$$q = \exp\left(\frac{\epsilon(l_j^{(i-1)}) - \epsilon(l_j^{(i)})}{\epsilon(l_j^{(i-1)})t}\right) \quad (190)$$

is computed. If $q \geq 1$ (i.e. $\epsilon(l_j^{(i)}) < \epsilon(l_j^{(i-1)})$) the perturbation is accepted, otherwise the perturbation is accepted with probability q .

Both the number of changes in the pulse positions and the probability of accepting a particular change in pulse positions is controlled by the effective temperature. At very high temperatures, the perturbation is typically quite large and accepted quite readily since q is very near unity. The equilibrium temperature is set such that if the global minimum is found, all changes will be unacceptable henceforth. In practice, this requirement is difficult to uphold since typically very little is known about the nature of the error surface. Furthermore, the best state found is also maintained and thus if a global minimum is entered the algorithm will release the optimal position (and not the final stopping state).

2.6.5. Constraining the Multi-pulse Excitation

Further constraints on the multi-pulse excitation can be made in order to reduce the data rate required for transmitting the pulse position and pulse gain information. Examples of these constraints include:

1. Equal amplitude multi-pulse, or Binary Multi-pulse Excitation (BMPE) whereby

$$\hat{e}_k^{(l_J)} = G^{(l_J)} \sum_{j=0}^{J-1} s_j^{(l_J)} \delta_{k-l_j}. \quad (191)$$

A single equal gain is applied to all non-zero pulse positions although the sign of each pulse position ($s_j^{(l_J)}$) is optimized. As with the conventional multi-pulse procedure the excitation in (191) can be defined recursively and a similar multi-pulse algorithm to that defined above results.

2. Δ -spaced Multi-pulse, whereby the minimum non-zero pulse spacing is constrained to be greater than or equal to Δ . For $\Delta > 1$, fewer bits are required to specify the positions of the non-zero pulse positions. The conventional multi-pulse algorithm is easily adapted to handle the Δ -spaced constraint.
3. Equal-Spaced Multi-pulse (or regular pulse excitation, RPE) whereby equal spacing between the non-zero pulses is required. Once the first pulse position is specified, all pulse positions are known. This algorithm is quite different from the conventional MPE algorithm, and the interested reader is referred to [57] for details.

Chapter 3.

Multi-stage Vector Quantization of the Short Term Filter

In this chapter efficient methods for quantizing the short term model parameters are presented and compared to conventional techniques. The primary technique, multi-stage vector quantization (MSVQ), is compared against split VQ of the line spectrum pairs [39]. The techniques are compared using the root mean square average log spectral distance (d_{SD} , see Section 2.4.4), the percentage of outliers with a spectral distortion between 2 and 4 dB, and the percentage of outliers with a spectral distortion greater than 4 dB.

The goal is to determine suitable moderate complexity methods for quantizing the short term predictor. Three popular parameterizations of the short term filter and distortion measures are investigated:

1. the predictor parameters and the likelihood ratio;
2. the log area ratios (LARs) and the mean squared error distortion (MSE); and
3. the line spectrum pairs (LSPs) and a weighted mean squared error (WMSE).

It has been shown that virtually transparent quantization of $A(z)$ is possible if the number of outliers greater than 2 dB (d_{SD}) is less than 2 percent, with no outliers greater than 4 dB, at an average d_{SD} less than 1 dB [39]. Transparent quantization implies that the synthetic speech quality is identical (in A-B tests) whether or not quantization is used. Transparent quantization might be a reasonable goal for a high quality 8 kb/s speech codec. However, the trade-offs between the various bit allocations for the various parameters in no way indicates that transparent quality is required at rates near 4 kb/s.

Full search (unstructured) VQ requires both a large computational complexity and a large memory size to obtain a spectral distortion below 1 dB. Thus, techniques to reduce both the computational complexity and memory size of the VQ while maintaining good performance is paramount to the realization of a low bit rate codec.

Multi-stage Vector Quantization [29] (MSVQ) is a multiple step vector quantization technique whereby the reconstruction vector ($\hat{u}_n$) is comprised of the addition (possibly with rotations) of the outputs of a number of codebooks as shown in Fig. 15.

Conventionally, a sequential search is used with MSVQ as outlined below. The target vector u is first predicted based on previous quantized vectors. The predictor matrix F may be fixed, or chosen from a codebook. The multi-stage VQ then typically chooses the index into the first codebook to minimize (over all t_0) the distortion between u_n and $F\hat{u}_{n-1} + H_0 y^{(t_0)}$ where $y^{(t_0)}$ is an element (indexed by t_0) of the first codebook. The conditioning matrix H_0 is, in general, a function of F . The second index is determined to minimize (over all t_1) the distortion between u_n and $F\hat{u}_{n-1} + H_0 y^{(t_0)} + H_1 y^{(t_1)}$ assuming the first index is fixed. Subsequent indices are determined in a likewise manner. The matrix H_1 is a *rotation* (or cell conditioning matrix) which attempts to condition each

Figure 15: Multi-stage vector quantization (synthesis). The codebook indices (t_j) select vectors from the codebooks (C_j). Prediction of future vectors is based on only a single past quantized vector in this case.

cell such that the pdf of $\mathbf{u}_n - (\mathbf{F}\hat{\mathbf{u}}_{n-1} + \mathbf{H}_0\mathbf{y}^{(t_1)})$ is the same as $\mathbf{H}_1\mathbf{y}^{(t_1)}$ for all cells. Strictly speaking, $\mathbf{H}_0$ is a function of $\mathbf{F}$, $\mathbf{H}_1$ is a function of t_0 and $\mathbf{F}$, $\mathbf{H}_2$ is a function of t_1 , t_0 , and $\mathbf{F}$ and so on. Cell conditioning can cause the memory requirements of MSVQ to increase quite dramatically, and for some distortion measures (notably the likelihood ratio) cell conditioning may remove structure which is required for complexity reduction techniques. For these reasons, cell conditioning is not considered in this treatise. Prediction is also not considered further, although the codebook search and design techniques used in this chapter are easily extended to encompass prediction as well.

In split VQ, the target vector $\mathbf{u}$ is first split into a number (often 2) of sub-vectors and each is independently quantized in a full search VQ. Clearly, split VQ is nothing more than MSVQ whereby certain codevector elements are forced to zero. Since split VQ is MSVQ with added structure it is fundamentally inferior to MSVQ. However, the performance/rate/complexity trade-offs between the two techniques are not at all clear.

Recently, spectral distortions near 1 dB have been obtained [39] using Line Spectral Pairs (LSPs) and a weighted distortion measure in a 24 bits/frame split VQ system having two 4096 level codebooks. The split codebook system was reported to have performed better than a sequential two-stage system with codebooks of the same size. However, since split codebook VQ is equivalent to two-stage VQ where some vector components are forced to zero, the low reported performance of two-stage VQ must be related to sub-optimality traditionally used in the design and the search of multi-stage VQ codebooks. MSVQ was previously studied in [58, 3]. In [58] results were presented for a 5 stage codebook with 64 codevectors per stage. In [3] the M - L algorithm was investigated in greater detail, and 1 dB spectral distortion was obtained for codes operating at rates as low as 22 bits/frame.

In Section 4.1.1 it is shown that under certain assumptions and as the vector dimension approaches infinity optimal performance can still be maintained by using a multi-stage approach. However, in practice with finite dimensional vectors and correlated sources, conventional Multi-stage VQ is suboptimal for 3 important reasons:

1. the codebooks are forced into a certain structure which imposes limits on the ultimate performance;
2. the search procedure used is (typically) suboptimal; and
3. the codebook design procedures are suboptimal.

Although the structural constraint cannot be removed without increasing significantly the computational complexity and/or memory size, the search technique and the codebook training can be significantly improved. For split VQ, the sequential search is optimal (for the given codebook) and equivalent to a full search of the codebooks. However, the additional structural constraint in a split codebook may result in significant degradation.

The sequential search procedure assumes at each stage that the vectors from all subsequent stages are zero. For the given constrained codebook structure, an optimal (full) search requires considering all the combinations of vectors drawn from different stages and has the same complexity as full search VQ albeit at a greatly reduced memory size. In this chapter the M - L search is used as a trade-off between the complexity of a full search and the performance degradation of a sequential search. It is shown that in practice the M - L search achieves performance close to that of the optimal search for a relatively low complexity.

Stage by stage (sequential) training optimizes the current stage codebook under the assumption that all subsequent stages are populated by zero vectors only. Both the search and centroid computations are suboptimal. In this chapter a

training procedure (based on the generalized Lloyd Algorithm, GLA [21]) which optimizes the reproduction vectors of the multi-stage VQ is introduced. For convergence to a local minimum, a full search is required, although convergence is observed in practice with a suboptimal search combined with an optimal centroid calculation. A training procedure which reduces outliers at the expense of a slight average performance degradation is also introduced.

Experimental results are presented which show that the tree searched multi-stage VQ achieves significantly better performance/complexity trade-offs than the split codebook approach.

In this chapter, fully and partially joint codebook design procedures are developed to jointly optimize the multi-stage codebooks. Design procedures for split VQ algorithms based on the multi-stage VQ schemes are also provided for completeness.

3.1. Search Procedure

The multi-stage codebooks, or *codes* in this chapter are referred to by:

1. the search technique used, identified by M_a (M in the M - L algorithm);
2. the distortion measure- whether weighted (W) or not;
3. the parameter set and distortion measure used- predictor parameters and the likelihood ratio (ALR), line spectrum pairs and a weighted mean squared error (WLSP), or log area ratios and a mean squared error (LAR); and
4. the dimension of each codebook, $L_{s,0}:L_{s,1}:\dots:L_{s,T_a-1}$ (when each codebook has the same number of levels, the code is referred to as $L_{s,0}-T_a$, or when N stages in succession are of the same dimension, $L_{s,k}-N$ is used rather than $L_{s,k}:L_{s,k+1}:\dots:L_{s,k+N-1}$).

For example, an $M_a = 4$, LSP code based on a weighted MSE distortion with 4 stages of 64 levels each is referred to as $M4\text{-WLSP-64-4}$. An $M_a = 8$ LAR code based on a MSE distortion with the first stage having 32 levels, the second through fifth having 16 levels, and the final stage having 8 levels would be referred to as $M8\text{-LAR-32:16-4:8}$. Some of the identifiers are dropped when no confusion would result. The LSP split VQ code of [39] is referred to as LSP-S-4096-2 (or just S-4096-2). In the final stage of the $M\text{-}L$ search, rather than using MSE or WMSE, spectral distortion or the likelihood ratio may be used to select from the M_a final candidates. In this case LR or SD is added to the code descriptor to indicate that the final selection is based on the likelihood ratio or spectral distortion. For example, an $M4\text{-LAR-64-4}$ with final selection by the LR would be referred to as $M4\text{-LR-LAR-64-4}$ or $\text{LR-}M4\text{-LAR-64-4}$.

Given a p dimensional vector of predictor parameters, line spectrum pairs, reflection coefficients, or log area ratios, (or some other parameter set) $\mathbf{u}_n$, the reconstructed vector of parameters $\hat{\mathbf{u}}_n$ is given by

$$\hat{\mathbf{u}}_n^{(\mathbf{t})} = \sum_{k=0}^{T_a-1} \mathbf{y}_k^{(t_k)} \quad (192)$$

where t_k is the index into the k -th codebook of error vectors ($\mathbf{y}_k^{(t_k)}$) and $\mathbf{t}$ is the set of T_a codebook indices (one for each stage). Equation (192) can be rewritten as

$$\begin{aligned} \hat{\mathbf{u}}_n^{(\mathbf{t})} &= \sum_{k=0}^{T_a-1} \mathbf{B}_n^{(t_k)} \mathbf{c}_k \\ &= \mathbf{B}_n^{(\mathbf{t})} \mathbf{c}, \end{aligned} \quad (193)$$

where $\mathbf{c}_k$ contains the stacked codevectors

$$\mathbf{c}_k^T = [\mathbf{y}_k^{(0)T} \quad \mathbf{y}_k^{(1)T} \quad \mathbf{y}_k^{(2)T} \quad \dots \quad \mathbf{y}_k^{(L_{a,k}-1)T}] \quad (194)$$

and $\mathbf{B}_n^{(t_k)}$ selects the t_k -th codevector from $\mathbf{c}_k$. The number of levels in the k -th codebook is $L_{a,k}$. The matrix $\mathbf{B}_n^{(\mathbf{t})}$ is comprised of all the $\mathbf{B}_n^{(t_k)}$ matrices

$$\mathbf{B}_n^{(\mathbf{t})} = [\mathbf{B}_n^{(t_0)} \quad \mathbf{B}_n^{(t_1)} \quad \mathbf{B}_n^{(t_2)} \quad \dots \quad \mathbf{B}_n^{(t_{T_a-1})}], \quad (195)$$

and $\mathbf{c}$ is a column vector of the stacked $\mathbf{c}_k$ vectors

$$\mathbf{c} = [\mathbf{c}_0^T \quad \mathbf{c}_1^T \quad \mathbf{c}_2^T \quad \dots \quad \mathbf{c}_{T_s-1}^T]^T. \quad (196)$$

The optimal (full search procedure) evaluates $\hat{\mathbf{u}}_n^{(t)}$ over the ensemble of possible t and chooses the set of indices which minimizes the distortion. Thus, the minimum distortion set of indices ($\tilde{t}$) satisfies

$$\tilde{t} : d(\mathbf{u}_n, \hat{\mathbf{u}}_n^{(\tilde{t})}) \leq d(\mathbf{u}_n, \hat{\mathbf{u}}_n^{(t)}) \quad \forall \quad t \neq \tilde{t} \quad (197)$$

where $d()$ is the chosen distortion measure. This is equivalent minimizing the distortion over the entire product codebook (the ensemble of possible reconstruction values).

The determination of the optimal set of indices is a complex combinatorial optimization problem which can only be solved in general by trying all sets of indices (as above). Suboptimal sequential solutions are typically used in practice. However, to reduce the complexity while maintaining good performance, other suboptimal search procedures may be used. In this research, three search procedures are investigated:

1. sequential;
2. iterative sequential; and
3. M - L tree search.

For split VQ, all three search procedures are equivalent by virtue of the codebook design algorithm which partitions the space such that a sequential search is optimal. For MSVQ, the three search procedures are very different.

The sequential search procedure determines t in a sequential manner. The first index, $\tilde{t}_0$ is chosen such that

$$\tilde{t}_0 : d(\mathbf{u}_n, \mathbf{y}_0^{(\tilde{t}_0)}) \leq d(\mathbf{u}_n, \mathbf{y}_0^{(k)}) \quad \forall \quad k \neq \tilde{t}_0. \quad (198)$$

The second index is chosen to minimize the distortion assuming the first index is fixed (at $\tilde{i}_0$)

$$\tilde{i}_1 : d\left(\mathbf{u}_n, \mathbf{y}_0^{(\tilde{i}_1)} + \mathbf{y}_1^{(\tilde{i}_1)}\right) \leq d\left(\mathbf{u}_n, \mathbf{y}_0^{(\tilde{i}_1)} + \mathbf{y}_1^{(k)}\right) \quad \forall k \neq \tilde{i}_1. \quad (199)$$

Likewise, the j -th index is determined assuming all previous indices are fixed

$$\tilde{i}_j : d\left(\mathbf{u}_n, \sum_{i=0}^{j-1} \mathbf{y}_i^{(\tilde{i}_i)} + \mathbf{y}_j^{(\tilde{i}_j)}\right) \leq d\left(\mathbf{u}_n, \sum_{i=0}^{j-1} \mathbf{y}_i^{(\tilde{i}_i)} + \mathbf{y}_j^{(k)}\right) \quad \forall k \neq \tilde{i}_j. \quad (200)$$

The iterative sequential search procedure begins with the indices provided by the sequential search, and re-optimizes each index assuming all other indices are fixed. Index j is determined to satisfy

$$\tilde{i}_j : d\left(\mathbf{u}_n, \sum_{i=0}^{T_a-1} \mathbf{y}_i^{(\tilde{i}_i)}\right) \leq d\left(\mathbf{u}_n, \sum_{i=0, i \neq j}^{T_a-1} \mathbf{y}_i^{(\tilde{i}_i)} + \mathbf{y}_j^{(k)}\right) \quad \forall k \neq \tilde{i}_j. \quad (201)$$

A single iteration re-optimizes $\tilde{i}_j$ over all j , from $j = 0$ to $j = T_a - 1$. The iterative sequential search procedure is guaranteed to have better (or at worst the same) performance as the sequential search procedure. However, it was determined that the iterative sequential procedure performed only marginally better than the sequential search procedure, and for this reason, the iterative sequential search procedure was not considered further.

The M - L [59] procedure is a generalization of the sequential search procedure in which more than a single set of indices is passed on from one stage to the next. Rather than keeping only a single set of indices, M_a sets of indices are maintained. In the first stage, the set of M_a minimum distortion indices are determined. Denote this set by $\{\mathcal{T}_0\}$. Thus, $\{\mathcal{T}_0\}$ satisfies

$$l \in \{\mathcal{T}_0\} \text{ iff : } d\left(\mathbf{u}_n, \mathbf{y}_0^{(l)}\right) \leq d\left(\mathbf{u}_n, \mathbf{y}_0^{(k)}\right) \quad \forall k \notin \{\mathcal{T}_0\}. \quad (202)$$

In the second stage, a joint search is performed with the added constraint that the first index is from the set $\{T_0\}$. The M_a minimum distortion sets of indices are maintained from this search, and this set of (pairs of) indices is denoted by $\{T_1\}$ which satisfies

$$(l_0, l_1) \in \{T_1\} \text{ iff: } d(\mathbf{u}_n, \mathbf{y}_0^{(l_0)} + \mathbf{y}_1^{(l_1)}) \leq d(\mathbf{u}_n, \mathbf{y}_0^{(l_0)} + \mathbf{y}_0^{(k)}) \quad (203)$$

$$\forall l_0 \in \{T_0\}, (l_0, k) \notin \{T_1\}.$$

Similarly, at stage j , the set of indices $\{T_j\}$ is defined by

$$(l_0, l_1, \dots, l_j) \in \{T_j\} \text{ iff:}$$

$$d\left(\mathbf{u}_n, \sum_{i=0}^{j-1} \mathbf{y}_i^{(l_i)} + \mathbf{y}_j^{(l_j)}\right) \leq d\left(\mathbf{u}_n, \sum_{i=0}^{j-1} \mathbf{y}_i^{(l_i)} + \mathbf{y}_j^{(k)}\right) \quad (204)$$

$$\forall (l_0, l_1, \dots, l_{j-1}) \in \{T_{j-1}\}, (l_0, l_1, \dots, l_{j-1}, k) \notin \{T_j\}.$$

At the final stage, the minimum distortion set of indices from the set $\{T_{T_a-1}\}$ is chosen as the best. The search can be drawn as a tree as shown in Fig. 16. Note that a particular index in any stage may be the root of many paths, and that only M_a paths are kept after each stage. To improve performance, in the final stage, the spectral distortion or likelihood ratio may be used to select the optimal set of indices over the set $\{T_{T_a-1}\}$ [3].

In the example of Fig. 16, the first codebook is searched, and the $(M_a - 1) 3$ minimum distortion indices for the first stage are determined. In this example, the minimum distortion indices are

$$\{T_0\} = \{0, 1, 3\}. \quad (205)$$

The second stage tests the following pairs of indices

$$\begin{array}{lll} (0, 0) & (0, 1) & (0, 2) \\ (1, 0) & (1, 1) & (1, 2) \\ (3, 0) & (3, 1) & (3, 2) \end{array} \quad (206)$$

Figure 16: M-L (tree) search procedure. The best paths are indicated by heavy lines.

and maintains the best M_d pairs of indices for the final stage. In this example

$$\{\mathcal{T}_1\} = \{(1, 0), (1, 2), (3, 0)\}. \quad (207)$$

The final stage tests the following triplets of indices

$$\begin{array}{ll} (1, 0, 0) & (1, 0, 1) \\ (1, 2, 0) & (1, 2, 1) \\ (3, 0, 0) & (3, 0, 1). \end{array} \quad (208)$$

Again, in this example

$$\{\mathcal{T}_2\} = \{(1, 0, 0), (1, 2, 0), (3, 0, 1)\}. \quad (209)$$

The minimum distortion set of indices from $\{\mathcal{T}_2\}$ is then chosen.

The M - L (for $M_d > 1$) search procedure is not guaranteed to obtain lower distortion than the sequential search procedure. In our example above, assume index 0 was the minimum distortion index for the first stage. The M - L search pruned this branch after the second stage since it was not one of the three best paths, although branch 0 could eventually lead to the minimum distortion set of indices (or a lower distortion than the M - L search). However, in general the M - L procedure tends to obtain a lower distortion than the sequential search procedure since more paths are considered.

3.2. Distortion Measures and Search Complexity

In this section a number of parameter sets and distortion measures are considered for MSVQ. As mentioned previously, three parameter sets and distortion measures are considered:

1. the predictor parameters and the likelihood ratio;
2. the line spectrum pairs and a weighted mean squared error distortion measure; and
3. the log area ratios and a mean squared error distortion measure.

The weighted mean squared error between the target vector $\mathbf{u}_n$ and the reconstruction vector $\hat{\mathbf{u}}_n^{(t)}$ is

$$d_{\text{wmse}}(\mathbf{u}_n, \hat{\mathbf{u}}_n^{(t)}) = (\mathbf{u}_n - \hat{\mathbf{u}}_n^{(t)})^T \mathbf{W}_n (\mathbf{u}_n - \hat{\mathbf{u}}_n^{(t)}), \quad (210)$$

where $\mathbf{W}_n$ is a symmetric weighting matrix (which may depend on the unquantized parameters). The mean squared error is the degenerate case where $\mathbf{W}_n = \mathbf{I}$.

Various weighting matrices applicable to LSPs were evaluated. The weightings introduced in [39], [58], and [60, 61] were all tested, and it was found that the latter method performed slightly better (slightly lower spectral distortion and lower percentage of outliers) than that of [39] and considerably better than that of [58]. The weighting matrix entries are given by [61]

$$W_{n,i,j} = \begin{cases} 0 & i \neq j \\ u_i \sqrt{D_i/D_{\max}} & D_{\text{crit}} \leq D_i \leq D_{\max} \\ u_i D_i / \sqrt{1.375 D_{\max}} & D_i < D_{\text{crit}}, \end{cases} \quad (211)$$

where

$$u_i = \begin{cases} 1 & x_i < f_{\text{crit}} \\ 1 - \frac{1}{6,000} (x_i - f_{\text{crit}}) & f_{\text{crit}} \leq x_i \leq f_s/2, \end{cases} \quad (212)$$

and where D_i is the group delay of the ratio filter at a frequency corresponding to the i -th LSP, $f_{\text{crit}} = 1000$ Hz, $D_{\text{crit}} = 1.375$ msec, $D_{\max} = 20$ msec and the sampling rate is $f_s = 8000$ Hz.

The likelihood ratio (LR) distortion measure between the target set of predictor parameters (with autocorrelation matrix $\mathbf{R}_p$, autocorrelation vector $\mathbf{r}_p$ and minimum residual energy α_p) and reconstruction vector of predictor parameters $\hat{\mathbf{a}}^{(t)}$ is given by

$$d_{\text{LR}}(\hat{\mathbf{a}}^{(t)}, \mathbf{R}_p, \mathbf{r}_p) = \frac{r_{p,0} + 2\hat{\mathbf{a}}^{(t)T} \mathbf{r}_p + \hat{\mathbf{a}}^{(t)T} \mathbf{R}_p \hat{\mathbf{a}}^{(t)}}{\alpha_p} - 1 \quad (213)$$

where

$$R_{p,i,j} = r_{p,|i-j|}, \quad 0 \leq i < p, \quad 0 \leq j < p, \quad (214)$$

$$\mathbf{r}_p = [r_{p,1} \quad r_{p,2} \quad \dots \quad r_{p,n}]^T. \quad (215)$$

$$\hat{\mathbf{a}}^{(t)} = [\hat{a}_1^{(t)} \quad \hat{a}_2^{(t)} \quad \dots \quad \hat{a}_p^{(t)}]^T, \quad (216)$$

and

$$\alpha_p = r_0 + \mathbf{r}_p^T \mathbf{R}_p^{-1} \mathbf{r}_p. \quad (217)$$

Note that $\hat{\mathbf{a}}^{(t)}$ is constructed as with $\hat{\mathbf{u}}^{(t)}$ (a different symbol is used only for clarity). Note that the notational definitions of $\mathbf{R}_p$, $\mathbf{a}$ and $\mathbf{r}_p$ are somewhat different than the definitions presented in Section 2.4.3 although the distortion measure is identical. In [62] a different approach was used whereby a cascaded analysis filter was used. The approach presented here coupled with the M - L search procedure leads to better performance than that obtained in [62].

In this thesis, the search complexity is defined as the number of arithmetic operations required to obtain the quantized parameter and is typically presented on a logarithmic scale.

With the M - L search procedure, the distortion computation at different nodes in the tree has many elements in common and if the tree has not fully fanned out, the number of best paths may be significantly less than M_a . The number of paths at stage j ($j \neq 0$) is just

$$\hat{M}_j = \min \left(M_a, \prod_{k=0}^{j-1} L_{s,k} \right). \quad (218)$$

To evaluate the computational complexity for the (weighted or unweighted) MSE case, denote the m -th candidate reconstruction vector up to (but not including stage j) as $\hat{\mathbf{u}}^{(m,j)}$ and the error between $\hat{\mathbf{u}}^{(m,j)}$ and $\mathbf{u}$ as

$$\mathbf{e}^{(m,j)} = \mathbf{u} - \hat{\mathbf{u}}^{(m,j)}. \quad (219)$$

The weighted error at the j -th stage is

$$d(\mathbf{u}, \hat{\mathbf{u}}^{(m,j)} + \mathbf{y}_j^{(t,j)}) = \mathbf{e}^{(m,j)T} \mathbf{W} \mathbf{e}^{(m,j)} - 2\mathbf{e}^{(m,j)T} \mathbf{W} \mathbf{y}_j^{(t,j)} + \mathbf{y}_j^{(t,j)T} \mathbf{W} \mathbf{y}_j^{(t,j)}. \quad (220)$$

The first term of (220) is the minimum distortion from the parent node and is known, the second term requires $\min(L_{s,j}, \hat{M}_j)p + \hat{M}_j p L_{s,j}$ multiply adds, and the final term requires $2pL_{s,j}$ multiply adds. The total complexity is therefore

$$C_{\text{WMSE}} = \log_2 \left(3pL_{s,0} + p \sum_{j=1}^{T_s-1} \min(L_{s,j}, \hat{M}_j) + (\hat{M}_j + 2)L_{s,j} \right). \quad (221)$$

For unweighted distortion measures ($\mathbf{W} = \mathbf{I}$), and assuming $\mathbf{y}_j^{(t,j)T} \mathbf{y}_j^{(t,j)}$ is pre-computed and stored as part of the codebook, the complexity is just

$$C_{\text{MSE}} = \log_2 \left(pL_{s,0} + p \sum_{j=1}^{T_s-1} \hat{M}_j L_{s,j} \right). \quad (222)$$

For the likelihood ratio, denote the m -th candidate reconstruction vector up to (but not including stage j) as $\hat{\mathbf{a}}^{(m,j)}$ and thus

$$\begin{aligned} d_{\text{LR}}(\hat{\mathbf{a}}_n^{(m,j)} + \mathbf{y}_j^{(t,j)}, \mathbf{R}_n, \mathbf{r}_n) &= \frac{r_{n,0}}{\alpha_{n,p}} + \frac{2}{\alpha_{n,p}} (\hat{\mathbf{a}}_n^{(m,j)} + \mathbf{y}_j^{(t,j)})^T \mathbf{r}_n + \\ &\quad \frac{1}{\alpha_{n,p}} (\hat{\mathbf{a}}_n^{(m,j)} + \mathbf{y}_j^{(t,j)})^T \mathbf{R}_n (\hat{\mathbf{a}}_n^{(m,j)} + \mathbf{y}_j^{(t,j)}) - 1, \\ &= d_{\text{LR}}(\hat{\mathbf{a}}_n^{(m,j)}, \mathbf{R}_n, \mathbf{r}_n) + \frac{2}{\alpha_{n,p}} \mathbf{y}_j^{(t,j)T} (\mathbf{r}_n + \mathbf{R}_n \hat{\mathbf{a}}_n^{(m,j)}) + \\ &\quad \frac{1}{\alpha_{n,p}} \mathbf{y}_j^{(t,j)T} \mathbf{R}_n \mathbf{y}_j^{(t,j)}. \end{aligned} \quad (223)$$

Since $\mathbf{R}_n$ is Toeplitz, the last term of (223) can be computed in $L_{s,j}p$ multiply adds (assuming storage of the codebook correlations) as described in Section 2.5.2.2. The first term is known from the parent node, and the second term requires $\hat{M}_j L_{s,j}p + \min(\hat{M}_j p, L_{s,j}(p+1))p$ multiply adds. The total complexity is

$$C_{\text{LR}} = \log_2 \left(2pL_{s,0} + p \sum_{j=1}^{T_s-1} (\hat{M}_j + 1)L_{s,j} + \min(\hat{M}_j p, L_{s,j}(p+1)) \right). \quad (224)$$

Indexing operations required for the M - L algorithm, and adding the various scalar terms (additions and subtractions in (220)) together is not considered in this complexity measure. Ultimate complexity in terms of MIPS is system and/or processor dependent.

The complexity of split VQ (sequential search) is

$$C_S = \log_2 \left(3n_0 L_{s,0} + 3 \sum_{j=1}^{T_s-1} L_{s,j} n_j \right) \quad (225)$$

where n_j is the dimension of the j -th codebook.

The memory requirements of MSVQ with a (weighted or unweighted) MSE distortion (number of codebook elements) is just

$$M_{WMSE} = p \sum_{j=0}^{T_s-1} L_{s,j}, \quad (226)$$

and for split VQ is

$$M_S = \sum_{j=0}^{T_s-1} L_{s,j} n_j. \quad (227)$$

The memory size of MSVQ with a likelihood ratio distortion measure is

$$M_{LR} = 2p \sum_{j=0}^{T_s-1} L_{s,j}. \quad (228)$$

3.3. Codebook Design

In this section, straightforward application of the Generalized Lloyd Algorithm (GLA) to VQ design of MSVQ codebooks is performed. The GLA is reviewed in Section 2.3.1. The GLA requires only the definition of a distortion measure and a corresponding centroid.

The subjective performance of VQ is often hindered by the outliers which may have relatively little effect on the average distortion but which may have a rather large (annoying) subjective effect. By appropriately weighting the distortion measure during the centroid calculation an improved outlier performance is realized at a small increase in average distortion.

Three distortion measures are considered:

1. the likelihood ratio;
2. the weighted mean squared error; and
3. the mean squared error.

Each distortion measure can be written as

$$d(\mathbf{u}_n, \hat{\mathbf{u}}_n^{(t)}) = \epsilon_n - 2\mathbf{f}_n^T \hat{\mathbf{u}}_n^{(t)} + \hat{\mathbf{u}}_n^{(t)T} \mathbf{F}_n \hat{\mathbf{u}}_n^{(t)}. \quad (229)$$

The corresponding scalars, vectors, and matrices for the various distortion measures are shown in Table 2.

3.3.1. Sequential and Iterative Sequential Codebook Design

At iteration r , the distortion over the training sequence is

$$d_r = \sum_n d(\mathbf{u}_n, \hat{\mathbf{u}}_{n,r}^{(t)}) \quad (230a)$$

$$= \sum_n \left(\epsilon_n - 2\mathbf{f}_n^T \hat{\mathbf{u}}_{n,r}^{(t)} + \hat{\mathbf{u}}_{n,r}^{(t)T} \mathbf{F}_n \hat{\mathbf{u}}_{n,r}^{(t)} \right), \quad (230b)$$

Table 2: Correspondence between the parameters of the general distortion measure and the parameters of the MSE, WMSE and LR.

Parameter	Distortion		
	MSE	WMSE	LR
ϵ_n	$\mathbf{u}_n^T \mathbf{u}_n$	$\mathbf{u}_n^T \mathbf{W}_n \mathbf{u}_n$	$\mathbf{r}_{n,0}$
$\mathbf{f}_n$	$\mathbf{u}_n$	$\mathbf{W}_n \mathbf{u}_n$	$-\mathbf{r}_n$
$\mathbf{F}_n$	$\mathbf{I}$	$\mathbf{W}_n$	$\mathbf{R}_n$

where $\hat{\mathbf{u}}_{n,r}^{(t)}$ is (of course) a function of r .

To design stage j , for both the sequential and iterative sequential algorithms, the average distortion over the training sequence is minimized assuming all other stages are fixed. For the sequential algorithm, the synthetic vector including all stages up to (but not including) stage j is

$$\mathbf{u}_{n,j}^{(t)} = \hat{\mathbf{u}}_{n,r}^{(t)} - \mathbf{c}_{j,r} \mathbf{B}_{n,j}^{(t,r)} = \sum_{k=0}^{j-1} \mathbf{B}_{n,k}^{(t_k)} \mathbf{c}_k \quad (231)$$

and is not a function of the stage j codebook at iteration (r). In this case, referred to as *sequential optimization*, the j -th stage is redesigned given a j stage codebook. Typically, stage 0 is designed until convergence, then stage 1, stage 2 and so on.

For the iterative sequential algorithm the reconstruction vector including all stages except stage j is

$$\mathbf{u}_{n,j}^{(t)} = \hat{\mathbf{u}}_{n,r}^{(t)} - \mathbf{B}_{n,j}^{(t,r)} \mathbf{c}_{j,r} = \sum_{k=0, k \neq j}^{T_s-1} \mathbf{B}_{n,k}^{(t_k)} \mathbf{c}_k \quad (232)$$

and again is not a function of the stage j codebook at iteration (r). In this case, referred to as *iterative sequential optimization* all other stages are assumed fixed and known, and the requirement is simple re-optimization of stage j . Once an initial set of codebooks are obtained, each stage can be optimized given the other stages. Thus, the final weighted mean squared error is minimized rather than the weighted mean squared error after each stage. Substitution of (231) or (232) into (230) leads to

$$d_r = \sigma_j^2 - 2\psi_j^T \mathbf{c}_{j,r} + \mathbf{c}_{j,r}^T \Psi_j \mathbf{c}_{j,r} \quad (233)$$

where

$$\sigma_j^2 = \sum_n d(\mathbf{u}_n, \mathbf{u}_{n,j}^{(t)}), \quad (234)$$

$$\psi_j = \sum_n \mathbf{B}_{n,j}^{(t_j)T} (\mathbf{f}_n - \mathbf{F}_n \mathbf{a}_{n,j}^{(t_j)}), \quad (235)$$

and

$$\Psi_j = \sum_n \mathbf{B}_{n,j}^{(t_j)T} \mathbf{F}_n \mathbf{B}_{n,j}^{(t_j)}. \quad (236)$$

Minimizing (233) with respect to $\mathbf{c}_{j,r}$ leads to the familiar least squares solution

$$\mathbf{c}_{j,r} = \Psi_j^{-1} \psi_j. \quad (237)$$

Typically, the iterative sequential design algorithm begins with an initial codebook trained possibly on a portion of the training sequence. The retraining procedure retrains the stages $j = 0, 1, \dots, T_a - 1$, one at a time repetitively [3, 4, 5]. The procedure is similar to the joint codebook design of [63, 64], and in this thesis the technique is referred to as iterative sequential optimization. The sequential design algorithm is shown in Fig. 17 and the iterative sequential design algorithm is shown in Fig. 18.

The design of random or initial codebooks is a rather important problem. In this research the method shown in Fig. 19 was used to design the initial codebook required for sequential, iterative sequential, and joint codebook optimization.

For codebooks with many stages the convergence of the sequential design procedures can be painstakingly slow since only a single codebook is optimized after each pass over the training sequence. A significant improvement in convergence speed and slightly better performance can be obtained by the simultaneous joint codebook design introduced in the following section.

3.3.2. Simultaneous Joint Codebook Design

The goal of simultaneous joint codebook design is to jointly optimize all codevectors over all stages after each iteration, and therefore joint design can

1. *Initialize.* Set $j = 0$. Set $r = 1$. Set $d^{(0)} = \infty$.
2. *Random Codebook.* Create random codebook for stage j .
3. *Partition.* Set all elements of Ψ_j and ψ_j to zero, then for each vector in the training sequence, determine the codebook indices (sequentially) which minimize the distortion. Compute both $\mathbf{B}_{n,j}^{(t)T} \mathbf{F}_n \mathbf{B}_{n,j}^{(t)}$ and $\mathbf{B}_{n,j}^{(t)T} (\mathbf{f}_n - \mathbf{F}_n \mathbf{u}_{n,j}^{(t)})$ and update the running sum Ψ_j , ψ_j and d_r . ($\mathbf{u}_{n,j}^{(t)}$ is given by (231)).
4. *Compute new codebook.* Compute $c_{j,r}$ according to (237).
5. *Convergence test.* If $(d_{r-1} - d_r)/d_r > \epsilon$, set $r = r + 1$ and goto 3.
6. *New Stage.* If $j < T_s - 1$ set $j = j + 1$ and goto 2.
7. *Terminate.*

Figure 17: MSVQ sequential design algorithm.

1. **Initialize.** Set $r = 1$. Set $j = 0$. Create T_s stage random codebook.
2. **Partition.** Set all elements of Ψ_j and ψ_j to zero, then for each vector in the training sequence, determine the codebook indices (sequential, tree or full search) which minimize the distortion. Compute both $\mathbf{B}_{n,j}^{(t)T} \mathbf{F}_n \mathbf{B}_{n,j}^{(t)}$ and $\mathbf{B}_{n,j}^{(t)T} (\mathbf{f}_n - \mathbf{F}_n \mathbf{u}_{n,j}^{(t)})$ and update the running sum Ψ_j , ψ_j and d_r . ($\mathbf{u}_{n,j}^{(t)}$ is given by (232)).
3. **Compute new codebook.** Compute $\mathbf{c}_{j,r}$ according to (237).
4. **Convergence test.** If $|d_{r-1} - d_r|/d_r > \epsilon$, set $r = r+1$, $j = (j + 1) \bmod T_s$ and goto 2.
5. **Terminate.**

Figure 16: MSVQ iterative sequential design algorithm.

1. *Use a very small subset of the full training sequence. Typically, the size of the initial training sequence (N_1) was chosen such that*

$$N_1 = 10 \max(L_{s,0}, L_{s,1}, \dots, L_{s,T_s-1}). \quad (238)$$

This small training sequence is used (below) to design the initial codebook.

2. *Design a 1 level 1 stage (optimal) codebook. (The weighted mean value of the training sequence).*
3. *Use codebook splitting and redesigning until the first stage has the desired number of levels. If sequential design, exit.*
4. *Add another stage (with the desired number of levels) randomly (small random gaussian numbers), and optimize using a sequential design algorithm.*
5. *Repeat step 4 until the desired number of stages have been created and exit.*

Figure 19: The design of initial random codebooks.

converge far quicker, and to possibly a lower final distortion. The concept is to optimize the ensemble of possible reconstruction vectors, rather than minimizing the MSE or WMSE after each stage.

Generally, in the GLA algorithm the objective is iterative optimization of decision boundaries and reconstruction values (or centroids). A particular code-vector from a particular stage codebook affects many of the final reconstruction vectors. For this reason, it is more convenient to consider the multi-stage codebook as a single entity $\mathbf{c}$ with the structure of the code fully contained in the rulebase for constructing $\mathbf{B}$ as in (193) and (195). Additional structure may be added by changing the rulebase for constructing $\mathbf{B}$.

Monotonic convergence to a local minimum is guaranteed if a full search procedure is used. In the following presentation it is assumed that a full search is used. It is then shown experimentally that a full search is not required to obtain performance very near to the full search algorithm.

The training sequence is first partitioned according to the minimum distortion (full search) algorithm. For the given partitioning, the codebooks are re-optimized. With the new codebooks, the training sequence is then repartitioned, and the process repeated.

To jointly optimize each codebook after each pass over the training sequence the average distortion d_r (from (230)) is minimized where $\hat{\mathbf{u}}_{n,r}^{(t)} = \mathbf{B}_{n,r}^{(t)} \mathbf{c}_r$ as in (193), and thus

$$d_r = \sum_n d(\mathbf{u}_n, \hat{\mathbf{u}}_{n,r}^{(t)}) \quad (239a)$$

$$= \sum_n \left(\epsilon_n - 2 \mathbf{f}_n^T \mathbf{B}_{n,r}^{(t)} \mathbf{c}_r + \mathbf{c}_r^T \mathbf{B}_{n,r}^{(t)T} \mathbf{F}_n \mathbf{B}_{n,r}^{(t)} \mathbf{c}_r \right) \quad (239b)$$

$$= \left(\sum_n \epsilon_n \right) - 2 \left(\sum_n \mathbf{f}_n^T \mathbf{B}_{n,r}^{(t)} \right) \mathbf{c}_r + \mathbf{c}_r^T \left(\sum_n \mathbf{B}_{n,r}^{(t)T} \mathbf{F}_n \mathbf{B}_{n,r}^{(t)} \right) \mathbf{c}_r. \quad (239c)$$

$$= d_{0,r} - 2 \mathbf{Y}_r^T \mathbf{c}_r + \mathbf{c}_r^T \mathbf{Q}_r \mathbf{c}_r \quad (239d)$$

As with the GLA, $\mathbf{B}_{n,r}^{(t)}$ (for all n) is determined given $\mathbf{c}_{r-1}$, and $\mathbf{c}_r$ is then computed for the given $\mathbf{B}_{n,r}^{(t)}$. The minimizing solution should satisfy $\mathbf{Q}\mathbf{c} = \mathbf{Y}$ (the iteration number r will henceforth be dropped to simplify the notation). It is tempting to write the codebook $\mathbf{c}$ which minimizes (239) in the form

$$\mathbf{c} = \mathbf{Q}^{-1}\mathbf{Y}. \quad (240)$$

However, in general, the inverse does not exist, since $\mathbf{Q} = \sum_n \mathbf{B}_n^{(t)T} \mathbf{F}_n \mathbf{B}_n^{(t)}$ is not full rank. An infinite number of solutions for the joint codebook exist since:

1. adding a constant vector $\mathbf{v}$ to all the vectors of any one stage, while subtracting the same constant vector from all the vectors of any other stage leads to the same ensemble of possible reconstruction vectors $\hat{\mathbf{u}}$ for any $\mathbf{v}$; and
2. the ordering of the codebooks is unimportant since the ordering does not change the ensemble of reconstruction values.

All minimizing solutions of (239) are equivalent. A number of techniques such as Newton's method, steepest descent, or conjugate gradient can be used to determine a solution which minimizes (239) and thus satisfies $\mathbf{Q}\mathbf{c} = \mathbf{Y}$.

A projection method (to minimize (239)) is used for simplicity reasons. The stage codebooks ($\mathbf{c}_j$) are optimized sequentially during the same iteration r , by holding all $\mathbf{c}_k$, $k \neq j$ fixed in (239), for $j = 0, 1, \dots, T_a - 1$. The codebook $\mathbf{c}$ is written as a function of the stage codebooks

$$\mathbf{c}^T = [\mathbf{c}_0^T \quad \mathbf{c}_1^T \quad \dots \quad \mathbf{c}_{j-1}^T \quad \mathbf{c}_j^T \quad \mathbf{c}_{j+1}^T \quad \dots \quad \mathbf{c}_{T_a-1}^T], \quad (241)$$

$$\mathbf{c} = \bar{\mathbf{c}}_j + \mathbf{S}_j \mathbf{c}_j, \quad (242)$$

where $\mathbf{S}_j$ is a simple shifting matrix and

$$\bar{\mathbf{c}}_j = [\mathbf{c}_0^T \quad \mathbf{c}_1^T \quad \dots \quad \mathbf{c}_{j-1}^T \quad \mathbf{0}_{pL_{a,j}}^T \quad \mathbf{c}_{j+1}^T \quad \dots \quad \mathbf{c}_{T_a-1}^T] \quad (243)$$

where $\mathbf{0}_{pL_{\cdot,j}}$ is an all zero $pL_{\cdot,j}$ dimensional vector. Substitution of (242) into (239) leads to

$$d_r = d_{j,r} - 2\mathbf{c}_j^T \mathbf{Y}_j + \mathbf{c}_j^T \mathbf{Q}_{j,j} \mathbf{c}_j, \quad (244)$$

where

$$d_{j,r} = d_{0,r} - 2\bar{\mathbf{c}}_j^T \mathbf{Y} + \bar{\mathbf{c}}_j^T \mathbf{Q} \bar{\mathbf{c}}_j, \quad (245)$$

$$\mathbf{Q}_{j,j} = \mathbf{S}_j^T \mathbf{Q} \mathbf{S}_j, \quad (246)$$

and

$$\mathbf{Y}_j = \mathbf{S}_j^T (\mathbf{Y} - \mathbf{Q} \bar{\mathbf{c}}_j). \quad (247)$$

Thus, the j -th stage codebook is given by

$$\mathbf{c}_j = \mathbf{Q}_{j,j}^{-1} \mathbf{Y}_j. \quad (248)$$

Conveniently, $\mathbf{Q}_{j,j}$ is diagonal (if $\mathbf{F}$ is), which leads to a relatively simple evaluation of $\mathbf{c}_j$.

The vector $\mathbf{Y}_j$ is a function of $\mathbf{c}$, $\mathbf{Q}$ and $\mathbf{Y}$. Thus, the full vector $\mathbf{Y}$ and matrix $\mathbf{Q}$ must be stored during the design procedure. Clearly, $\mathbf{c}_j$ in (248) depends on all the other stage codebooks. For a given partition of the training sequence, the algorithm repeatedly re-designs the codebooks $\mathbf{c}_j$, $j = 0, 1, \dots, T_a - 1$ until convergence. This inner loop provides a re-optimized codebook $\mathbf{c}$.

Once $\mathbf{c}$ has been re-optimized, the training sequence is again repartitioned given the new codebooks. This method is referred to as simultaneous joint design since all codebooks are re-optimized simultaneous and jointly after each pass over the training sequence.

3.3.3. Convergence Properties

The GLA converges to a local minimum under the two fundamental assumptions of the algorithm [65]:

1. the minimum distortion partition is determined (encoder search is optimal); and
2. the minimum distortion centroid or reconstruction value is used (decoder is optimal).

In the MSVQ context, 1 implies that a full search procedure is used during the design process. For the sequential design algorithm, the codebooks are designed sequentially, and thus the minimum distortion centroid or reconstruction value is not determined. Convergence to a local minimum is not guaranteed even if a full search procedure is used.

For the iterative sequential algorithm (assuming a full search procedure is used), 2 is not satisfied, although the centroids are iteratively optimized in order to converge to the minimum distortion centroid. The codebooks are alternately optimized and the algorithm may get mired on a stationary point (saddle point). However, this is rather unlikely and, in fact, convergence was always observed in practice. For simultaneous joint design, items 1 and 2 above are satisfied assuming the solution to $\mathbf{Qc} = \mathbf{Y}$ is always found, which is not necessarily guaranteed for the algorithm presented. However, in practice, a solution to $\mathbf{Qc} = \mathbf{Y}$ was always found by the projection method described above.

If a sequential search or a tree search are used, the performance of the code may degrade. Theoretically, joint design combined with sequential search may even result in worse performance than sequentially designed codebooks.

However, as it will be shown later in this chapter, the tree search approaches the full search performance for a moderate value of M_a . Hence, jointly designed

codebooks are expected to perform well for tree search with large M_d . Moreover, a relatively simple empirical procedure when applied to the joint design algorithm was found to result in *robust* codebooks which have good performance for sequential and tree search while not affecting the ensemble of reconstruction values. This empirical procedure is outlined in Fig. 20 and the joint design procedure is presented in Fig. 21.

The performance of the jointly designed codebooks in conjunction with the tree search (used for training and testing the codebooks) was investigated experimentally and found to justify the use of the simultaneous joint design algorithm.

3.4. Multi-stage VQ of Gaussian Sources

The distortion performance of single stage and sequentially searched multi-stage (10 dimensional) VQ of gaussian sources for various rates is shown in Fig. 22 and in Table 3. The distortion rate bound and the performance of non-entropy coded Lloyd-Max quantizers is also shown for reference purposes. The Multi-stage VQs (shown in Table 3) were designed sequentially using the sequential search procedure. That is, the previous stage codebooks were not re-optimized during the design process, and a sequential search was used to determine the codebook indices for both the design and testing phase.

The performance of various tree searches on the sequentially designed 64-2 code is shown in Table 4 and in Fig. 23. It is interesting to note that the split VQ code performs better than the sequentially searched code, even though the split VQ code search can be structured as a multi-stage search (with the first stage codevectors having all zero entries in locations five to nine and the second stage having all zero entries in locations zero to four). Clearly, the sequential design procedure is not determining the optimal codebook. This should not be too

- a. *Ensure that the sum of all codevectors of each codebook (except the first codebook) is the all-zero vector*

$$\sum_{k=0}^{L_{s,j}-1} \mathbf{y}_j^{(k)} = \mathbf{0}_p$$

where $\mathbf{0}_p$ is an all zero p dimensional vector. This may be accomplished without affecting the ensemble of reconstruction values, or the value of (239).

- b. *Modify the order of the codebooks such that the energy in $\mathbf{c}_j$ is less than the energy in $\mathbf{c}_k$, for all $j > k$*

$$\mathbf{c}_j^T \mathbf{c}_j \leq \mathbf{c}_k^T \mathbf{c}_k \quad \forall \quad j > k.$$

The energy of the first codebook is computed after the mean is subtracted. Although the ordering of the codebooks does not influence the ensemble of possible reconstruction values, (239) is affected since the chosen codebook indices are evident in both $\mathbf{Y}$ and $\mathbf{Q}$. Thus, reordering should only be done when the re-optimization of $\mathbf{c}$ is complete and just prior to repartitioning of the training sequence.

- c. *Begin with an initial codebook sequentially designed. The initialization by a sequentially designed codebook enables the energy ordering described in step b.*

Figure 20: Re-ordering procedure used in simultaneous joint design.

1. *Initialize.* Set $r = 1$. Create T_a stage random codebook.
2. *Partition.* Set all elements of $\mathbf{Y}$ and $\mathbf{Q}$ to zero, then for each vector in the training sequence, determine the codebook indices which minimize the WMSE. Compute both $\mathbf{B}_n^{(t)T} \mathbf{F}_n \mathbf{B}_n^{(t)}$ and $\mathbf{f}_n^T \mathbf{B}_n^{(t)}$ and update the running sum $\mathbf{Q}$, $\mathbf{Y}$ and d_r .
3. *Compute new codebook.* For $j = 0$ to $T_a - 1$ compute $\mathbf{c}_j = \mathbf{Q}_{j,j}^{-1} \mathbf{Y}_j$, where $\mathbf{Q}_{j,j} = \mathbf{S}_j^T \mathbf{Q} \mathbf{S}_j$, and $\mathbf{Y}_j = \mathbf{S}_j^T (\mathbf{Y} - \mathbf{Q} \bar{\mathbf{c}}_j)$.
4. *Convergence of $\mathbf{c}$.* Repeat step 3 until convergence.
5. *Reorder.* Reorder according to Fig. 20.
6. *Convergence test.* If $|d_{r-1} - d_r|/d_r > \epsilon$, set $r = r + 1$, and goto 2.
7. *Terminate.*

Figure 21: MSVQ simultaneous joint design algorithm.

Figure 22: Performance of full search single stage VQ and multi-stage sequentially searched (10 dimensional) VQ. $R(D)$ and Lloyd-Max scalar quantizers shown for reference purposes. Error bars correspond to 95% confidence limits. (i) Lloyd-Max scalar quantization. (ii) $R(D)$. (iii) Unstructured VQ. (iv) Multi-stage VQ.

Table 3: Performance of full search unstructured single stage VQ (F-), and multi-stage sequentially searched VQ.

Code	MSE (dB)	Quantizer	MSE (dB)
F-8	-1.14	M1-4-2	-1.44
F-64	-2.77	M1-32-2	-4.46
F-256	-3.86	M1-128-2	-6.62
F-1024	-4.96	M1-512-2	-8.80
F-2048	-5.51	M1-2048-2	-10.98
F-4096	-6.08	M1-4096-2	-12.06

surprising since the sequential MSVQ design and search procedure are suboptimal. However, when an optimal (full search) procedure is used in both the design and test phase the 64-2 multi-stage code significantly outperforms the split VQ code and in fact approaches the performance of the unstructured VQ code.

Table 4: Performance of (M-L) tree searched multi-stage VQ (a) designed sequentially with a sequential search and (b) designed iterative sequentially with a tree search during design (with M_a the same as during test). Full searched multi-stage VQ, and (5-5) split VQ (64-2 code) are shown for comparison purposes.

Code	Sequential Design	Iterative Seq Design
	MSE (dB)	MSE (dB)
M1-64-2	-5.528	-5.563
M2-64-2	-5.729	-5.768
M4-64-2	-5.806	-5.880
M8-64-2	-5.845	-5.960
M64-64-2	-5.847	-5.998
F-64-2	-5.847	-5.998
S-64-2	-5.774	-

Figure 23: Performance of tree searched multi-stage VQ, full searched multi-stage VQ, and (5-5) split VQ (64-2 code). Both 2048 level and 4096 level unstructured full searched codes are included for comparison purposes. (i) F-2048. (ii) F-4096. (iii) S-64-2. (iv) 64-2 designed sequentially with sequential search (v) 64-2 designed iteratively with tree search during design the same as that for testing.

3.5. LPC Performance of Codebook Design Methodologies

Computer simulations were conducted over a relatively large training speech database (TIMIT SX phonetically compact sentences) consisting of 462 speakers (136 females, 326 males) from 8 different (U.S.) dialect regions (approximately 113 minutes of speech). The test sequence consisted of 168 speakers (56 females, 112 males) from the same dialect regions (approximately 40 minutes of speech).

The 16 kHz TIMIT database was down sampled to 8 kHz using the downsampling filter specified in Appendix A. Autocorrelations were computed over a 192 point Hamming window. Levinson recursion was used to find an initial set of predictor parameters. High frequency correction was applied to the autocorrelation vector based on the residual energy as follows [66]

$$\begin{aligned} r_0 &= r_0 + \alpha_p \mu_0 \\ r_1 &= r_1 + \alpha_p \mu_1 \\ r_2 &= r_2 + \alpha_p \mu_2 \end{aligned} \tag{249}$$

where $\mu_0 = 3/160$, $\mu_1 = -1/80$, and $\mu_2 = 1/320$. The autocorrelation are again transformed into predictor parameters and 10 Hz of bandwidth expansion was applied to soften the possibly large spectral peaks. This set of predictor parameters was then transformed back to the autocorrelation domain by utilizing the autocorrelation of the impulse response of the synthesis filter [35]. The training database consisted of 374,317 vectors and the TIMIT-test database consisted of 121,200 vectors. Very low energy (silence) autocorrelation vectors were not included in the testing or training database. Approximately 15 percent of the vectors were classed as silence. Thus, even for large codebooks of dimension 4096, there were approximately 80 training vectors/codevector. In some noted cases (see below) a smaller database was used to shorten the length of time required for the simulations.

As an example, consider the design of a three stage, 12 bit LSP codebook, (WLSP-16-3) over a relatively small training database (34,467 vectors) and a small test sequence (24,240 vectors). The performance of joint design whereby a tree search with different values of M_a is used during the design procedure is displayed in Fig. 24. For comparison purposes, the performance of full search unstructured codes with 512, 1024, 2048 and 4096 levels was 2.78, 2.66, 2.47 and 2.35 dB respectively. Full search/joint design is marginally superior for large M_a , although the performance suffers for small M_a . The full search/joint design algorithm obtains performance comparable to the unstructured 2048 level code. It is apparent that approximately one bit has been lost in adding the significant structure to the code. The sequentially tree searched/jointly designed codebook with $M_a = 8$ also obtained performance very close to the 11 bit (2048 level) code implying an approximate 1 bit loss while reducing the complexity by a factor of 10. With these observations, the following conclusions can be made:

1. sequential search is inferior (to tree search) for codebook design;
2. asymptotically (large M_a), sequential tree search procedures coupled with joint design procedures are almost as good as full search/joint design procedures (and often better for small M_a);
3. relatively small values of M_a are required to obtain performance very close to full search; and
4. little is lost (approximately 1 bit) in adding significant structure to the codebook while greatly reducing memory size and computational complexity.

Convergence properties of the iterative sequential design procedure and simultaneous joint design procedure for a WLSP-16-6 code are displayed in Fig. 25a as a function of the number of iterations over the training sequence. After each iteration, the code was tested on a database outside the training sequence with the

Figure 24: TS-MSVQ performance with jointly designed codes designed using various values of M_a during codebook design (i) $M_a = 1$, (ii) $M_a = 2$, (iii) $M_a = 8$. (iv) Full Search. The performance is plotted versus the value of M_a used in the codebook testing.

results shown in Fig. 25b. Clearly, convergence speed is superior for the joint design procedure. It is expected that with a sufficient number of iterations the iterative sequential design procedure should converge to (on average) the same spectral distortion as the joint procedure. However, as mentioned previously, convergence properties are rather hard to predict due to possible convergence to one of the multiple local minima. In most cases with large T_a , iterative design is impractical, and joint design is required.

3.6. Outlier Weighting

One of the problems of concern in VQ of the LPC parameters are the so-called outliers – input vectors which are poorly represented in the training sequence and are quantized with a spectral distortion much larger than the average. The outlier performance of a TS-MSVQ quantizer can be significantly improved by appropriately weighting the distortion measure during training. With appropriate weighting, the robustness of the code can also be improved. To define a weighting for reducing outliers, the training sequence is (as before) partitioned according to the nearest neighbour criterion and then once $\hat{u}_n^{(t)}$ is determined, rather than minimizing (at iteration r) d_r in the centroid calculation (as in (230) or (239)) a weighted error is minimized according to $d_r = \sum_n f(SD)d_n$ where $f(SD)$ is some function of the codebook at iteration r and the spectral distortion between the target vector and the optimum quantized vector (in dB). The determination of the optimum quantized vector (encoder) is independent of the spectral distortion at time n , although the spectral distortion is used in the new codebook (decoder) computation. A number of functions were investigated:

1.

$$f(SD) = SD^p. \quad (250)$$

Figure 25: Convergence (a) inside the training sequence of (i) iterative sequential codebook design and (ii) Joint codebook design, and (b) outside the training sequence of (i) iterative sequential codebook design and (ii) Joint codebook design.

2.

$$f(SD) = \begin{cases} 1 & SD \leq 1 \\ 1 + \lceil \eta(SD - 1) \rceil & SD > 1 \end{cases} \quad (251)$$

where $\lceil \cdot \rceil$ is the ceiling function. This measure ($\eta > 0$) appends *virtual copies* of training vectors to the training sequence dependent on the spectral distortion.

The optimal centroid according to the distortion measure is not computed and thus the convergence of the algorithm is not guaranteed. Experimentally (for relatively small values of ρ or η), it was found that these approaches do not affect significantly the average SD, nor the training convergence.

The properties (spectral distortion and outliers between 2 and 4 dB) of a 16-6 code utilizing $M_a = 4$ and joint design (24 iterations) for various values of ρ and $\eta > 0$ are shown in Table 5 for performance inside and outside the training sequence. The various values of ρ or η provide a trade-off between outlier performance and spectral distortion performance.

For example, inside the training sequence, by using $\rho = 3$ increased average SD by less than 0.02 dB compared to no weighting ($\rho = 0$) while significantly decreasing the number of outliers from approximately 2 percent to under 1 percent. The same decrease (1 percent) in percentage of outliers and approximately the same small increase in average SD was observed outside of the training sequence. Suitable weighting ($\rho = 2$) was used during the design procedure for the codes described in the rest of the thesis.

Table 5: Spectral distortion and outliers for various weightings (both inside and outside the training sequence). See (250) and (251).

Weighing	SD (dB)		% Outliers, 2-4 dB	
	Inside	Outside	Inside	Outside
$\rho = 0$	1.11	1.13	2.04	2.32
$\rho = 1$	1.12	1.13	1.46	1.84
$\rho = 2$	1.12	1.14	1.18	1.42
$\rho = 3$	1.14	1.15	0.941	1.30
$\rho = 5$	1.17	1.19	0.775	1.20
$\eta = 2$	1.11	1.13	1.39	1.75
$\eta = 4$	1.12	1.13	1.20	1.65
$\eta = 16$	1.13	1.15	1.02	1.40
$\eta = 100$	1.15	1.16	0.917	1.26

3.7. Comparison of Different Parameter Sets and Distortion Measures

In this section, various parameter sets and distortion measures are investigated in order to determine a parameter set which can be efficiently used in the MSVQ context.

As mentioned previously, three parameter sets and accompanying distortion measures are investigated:

1. the predictor parameters and the likelihood ratio;
2. the log area ratios (LARs) and the mean squared error distortion (MSE);
3. the line spectrum pairs (LSPs) and a weighted mean squared error (WMSE).

The performance of full search unstructured 4096 level codes based on these three distortion measures is displayed in Table 6. It is apparent that for full search unstructured VQ, LSPs with a weighted distortion measure and the predictor parameters with the likelihood ratio offer very similar performance. The performance of LARs is clearly inferior. Even though a very large training set was used (374,317 vectors), spectral distortion performance outside the training sequence is clearly inferior in all cases considered (by approximately 0.1 dB).

The performance (versus complexity) of two code structures 4096-2, and 64-4 for the three parameter sets and distortion measures is shown in Figs. 26, 27, and 28 for the predictor parameters and the likelihood ratio, the LSPs and a WMSE distortion, and the LARs and a MSE distortion respectively. The consecutive points correspond to $M_a = 1, 2, 4, 8, 16$ and so on.

In curves (i) and (ii) of Fig. 26, the performance of an ALR-4096-2 and an ALR-64-4 code is displayed. In curves (i) and (ii) of Fig. 27 the performance of the same code structures for LSPs and a weighted distortion measure is shown. The performance of the ALR-64-4 and ALR-4096-2 codes are significantly worse than the performance of the WLSP-64-4 and WLSP-4096-2 codes implying that

Table 6: The performance (both inside and outside of the training sequence) of full search unstructured 4096 level codes based on the three parameter sets and distortion measures. Performance is measured in terms of average SD, percentage of outliers between 2 and 4 dB, and percentage of outliers above 4 dB.

Code	Inside			Outside		
	SD (dB)	2-4 dB	> 4 dB	SD (dB)	2-4 dB	> 4 dB
LR	1.81	32.9	0.095	1.91	40.8	0.203
LSP	1.80	32.7	0.046	1.90	40.3	0.132
LAR	2.14	50.5	2.42	2.25	55.6	3.30

Figure 26: Performance vs complexity of the predictor parameters with the likelihood ratio for three codes: (i) 4096-2 (ii) 64-4 (iii) C-64-4. In (iii), the chosen second, third, and fourth codebooks are conditional on the index from the first codebook. Thus, there are 64 second, 64 third, and 64 fourth codebooks which are chosen based on the value of the first index.

Figure 27: Performance vs. complexity of LSPs with a weighted MSE distortion for (i) 4096-2 (ii) 64-4 (iii) SD-64-4. In curve (iii) the final selection is made based on SD rather than on the WMSE.

Figure 28: Performance vs. complexity of LARs with a MSE distortion for (i) 4096-2 (ii) LR-4096-2 (iii) 64-4 (iv) LR-64-4. In curves (ii) and (iv), the final stage select the optimal candidate (out of M_a) based on the likelihood ratio (rather than on MSE).

the likelihood ratio and the predictor parameters may not be suitable in a MSVQ framework. It is apparent that for the ALR codes the error vectors associated with different cells from the first stage have very different probability distribution functions. To mitigate this affect with no increase in computational complexity (while incurring a large increase in the memory requirements) the codebook at the second, third (and so on) stages can be conditional upon the chosen first index. Thus, there are $L_{s,0}$ second codebooks, and the codebook selected depends on the first index. The code is basically a $L_{s,0}:L_{s,0}L_{s,1}:L_{s,0}L_{s,2}, \dots$ code whereby the allowed indices in the future stages are constrained. The codebook design is done by simply treating the code in this manner. Curve (iii) of Fig. 26 displays the performance of this conditional (or *fanned out* [63, 64]) C-64-4 code. The performance is only slightly inferior to the 64-4 LSP code. The program or implementation complexity of the ALR-C-64-4 code is much less than that of the LSP or LAR based codes since no Levinson Durbin recursion or parameter transformations are required, although the memory size is much larger. For a dedicated VLSI implementation, there is a considerable advantage to this approach.

When the spectral distortion is used for the selection process in the final stage, the performance of the WLSP-64-4 code is significantly improved (as shown in curve (iii) of Fig. 27). For LSPs, selection based on the LR offered no improvement over WMSE.

As displayed in Fig. 28, the performance of the LAR codes based solely on MSE is clearly inferior to the LSP codes. However, once the likelihood ratio is used for selection in the final stage (curves (ii) and (iv) of Fig. 28), the performance of the LAR and LSP codes is very similar.

Since the codes based on the LSPs appear to be as good or better than any distortion measures and parameter sets considered, they will be solely considered

in the rest of this chapter and the remainder of this thesis.

3.8. Rate, Performance, Complexity Trade-offs

In the traditional sequentially searched MSVQ, the design has historically been oriented toward the largest implementable codebooks and the smallest number of stages. For example, a quantizer using 24 bits per frame would typically be implemented using two codebooks of size 4096. Increasing the number of stages for sequentially searched, sequentially designed MSVQ leads to a quick degradation in performance. The introduction of tree search for multi-stage VQ leads to a significant improvement in the performance, particularly for configurations having a relatively large number of small codebooks.

Figure 29 shows spectral distortion (SD) versus complexity for four TS-MSVQ (LSP) codes and one split VQ code, all at a rate of 24 bits/frame. For optimal performance, the code with only two stages is indeed superior, albeit at quite a high computational complexity. One of the best codes in terms of the trade-off between complexity and performance in Fig. 29 is 64-4 which achieves a spectral distortion of about 1 dB at a complexity more than 8 times lower than the 4096-2 split VQ code. Moreover, 64-4 requires storing only 2,560 codewords as compared to 40,960 codewords required by the 4096-2 split VQ code. In all the codes of Fig. 29 (with M_a chosen such that $SD \leq 1$ dB), there are no outliers greater than 4 dB and the percentage of outliers greater than 2 dB is under 1%. The WLSP-S-4096-2 code (split VQ) used the same partitioning of the LSP vector as in [39] and obtained virtually equivalent performance. With the iterative sequential design procedure the WLSP-4096-2 multi-stage code obtains better performance than the split VQ code albeit at double the complexity (for $M_a = 1$). However, the 16-6, 64-4, and 256-3 codes all obtain superior performance to split VQ at a lower computational complexity and a much lower memory size. It is interesting

Figure 29: TS-MSVQ performance versus complexity for 24 bits/frame (weighted LSP) (i) 16-6 (ii) 64-4 (iii) 256-3 (iv) 4096-2. The solid point is S-4096-2 (split VQ).

to note that using 6 stages, with only 16 codewords in each stage (92 vectors total), imposes a structural constraint which degrades the performance less than partitioning the vector into two sub-vectors and coding each with a 4096 level full search code.

Figure 30 shows spectral distortion versus complexity at rates of 22-30 bits/frame. The 2048-2 code with $M_a = 2$ obtains virtually equivalent performance, complexity, and memory requirements as the split VQ code at a rate of only 22 bits/frame. By using a slightly higher rate, complexity can be reduced considerable while maintaining the spectral distortion performance. Note that the 28 bits/frame system has a very low complexity at a spectral distortion of 1 dB and a memory size of only 112 codevectors. From Fig. 30 the following key observations can be made:

1. as the number of stages increase and the number of codewords per stage decrease, more is gained from the M - L algorithm as M_a is increased;
2. the multi-stage codes have excellent rate, performance, and complexity trade-offs; and
3. the 22 bit 2048-2 code at $M_a = 2$ obtains performance virtually identical (in terms of computational complexity and memory size) to the 24 bit split VQ.

Figure 31 shows the performance of a 2-36 antipodal code (A-2-36), a code whereby each codebook is comprised of only 2 vectors, each being the negative of the other (except the first codebook which consists of two unconstrained vectors) by virtue of the re-ordering procedure described in Section 3.3 (Fig. 20). The performance using only the first 26, 28, 30, 32 and 34 stages from this code is also shown in Fig. 31. Although this code is extremely structured, it still obtains a SD very near 1 dB at a rate of only 26 bits/frame with a very low percentage

Figure 30: TS-MSVQ performance versus complexity for rates of 22-30 bits/frame. (i) 22 bits, 2048-2 (ii) 24 bits, 64-4 (iii) 26 bits, 256-64-3 (iv) 28 bits, 16-7 (v) 30 bits, 4-15. The solid point is S-4096-2.

of outliers (less than 0.7% outliers with a spectral distortion within 2 and 4 dB). This code is very compact (only 36 vectors are required), and supports various rates at various complexities. At 34 bits, the code obtains a spectral distortion of 1 dB at a very low complexity, whereas at 26 bits only 2 additional bits are required over the more complex 16-6, 64-4 or 256-3 (24 bit) codes.

The complexity and rate required to obtain near 1 dB average SD for various codes is displayed in Table 7. A spectral distortion of 1 dB can be obtained by multi-stage VQ with only two antipodal codevectors per stage and 28 stages. The 16-7 code offers relatively low memory size and computational complexity with the possibility of obtaining lower SD by using a larger value of M_a (future expandability). Virtually every MSVQ code considered has a lower memory requirement and a lower computational complexity than split VQ.

Table 7: Different TS-MSVQ codes to obtain approximately equivalent average SD performance. The rate is given in bits/frame.

Code	M_a	C	Rate	SD (dB)	% Outliers	
					2-4 dB	> 4 dB
S-4096-2	-	16.9	24	1.04	0.53	0.00
2048-2	2	17.1	22	1.04	0.67	0.00
64-4	4	13.7	24	1.04	0.47	0.00
16-6	16	13.9	24	1.04	0.59	0.00
4-13	8	12.4	26	1.03	1.49	0.01
16-7	2	12.1	28	1.05	0.80	0.00
2-28	8	11.5	28	1.00	0.48	0.00

Figure 31: TS-MSVQ performance versus complexity for rates of 26-36 bits/frame using the same A-2-36 code. (i) 26 bits, (ii) 28 bits, (iii) 30 bits, (iv) 32 bits, (v) 34 bits, (vi) 36 bits.

Table 8: Spectral distortion performance on different language and input spectral shapings. (a) German (b) Italian (c) Norwegian (d) Noisy speech database (e) TIMIT-test.

Code	M_n	(a)	(b)	(c)	(d)	(e)	C
64-4	32	1.04	1.02	1.02	1.16	0.936	16.1
256:64-3	8	0.951	0.939	0.911	1.05	0.846	14.7
S-4096-2	-	1.13	1.08	1.10	1.20	1.04	16.9
8-9	8	1.00	0.98	0.937	1.03	0.955	12.8
A-2-27	16	1.03	1.00	0.965	1.00	1.02	12.2

3.9. Robustness issues

3.9.1. Performance Across Languages and Input Spectral Shapes

Vector Quantization has often suffered from robustness problems whereby the performance of the VQ may be quite poor on data not represented in the training sequence. In this section the robustness across different languages and input spectral shapings is studied.

Tables 8 and 9 shows spectral distortion and outlier performance of a number of TS-MSVQ codes for (a) German, (b) Italian, and (c) Norwegian speech data bases. The foreign language database includes IRS weighted speech which was used for testing codecs in the CCITT 16 kb/s low-delay competition. Note the good robustness across languages for both TS-MSVQ systems tested.

In the same Table, (e) displays the performance on the TIMIT-test database, while (d) shows the performance on an English test database consisting of speech recorded through a high quality microphone. The IRS weighted databases and the TIMIT databases have similar average spectral characteristics (spectral roll off of approximately 2 dB/octave) whereas the English database (d) has a somewhat

Table 9: Outlier performance (2-4 dB) on different language and input spectral shapings. (a) German (b) Italian (c) Norwegian (d) Noisy speech database (e) TIMIT-test.

Code	M_n	(a)	(b)	(c)	(d)	(e)	C
64-4	32	1.14	1.07	1.28	4.35	0.291	16.1
256:64-3	8	0.262	0.472	0.780	2.51	0.106	14.7
S-4096-2	-	1.40	0.558	1.70	2.69	0.530	16.9
8-9	8	0.349	0.429	0.851	1.52	0.228	12.8
A-2-27	16	0.873	0.429	1.63	1.29	0.612	12.2

high spectral roll off (approximately 5 dB/octave). For these cases, the albeit higher rate small codebook systems show significantly better robustness than the lower rate larger codebook 4096-2 systems (including split VQ). The spectral distortion as a function of complexity is shown for the 2-27 and 64-4 code in Fig. 32. Although similar performance was observed both inside the training sequence and on the TIMIT-test sequence, on foreign language databases and on databases with different spectral shapings the higher rate antipodal (binary) code is seen to be very robust, at a very low complexity. Note that the 8-9 code was also quite robust.

It is apparent that robust VQ can be accomplished by adding suitable structure to the code while impairing average performance only slightly.

3.9.2. Performance in the Presence of Channel Errors

Good performance in the presence of channel errors is critical for a robust codec. In this section the performance of MSVQ with pseudo-Gray coding is studied in the presence of channel errors. The scalar quantizer was sorted and encoded according to a Gray code while each stage of the MSVQ was pseudo-Gray coded (based on MSE) using the algorithm of [67].

Figure 32: TS-MSVQ performance for different languages and input spectral shapings for (i) 64-4 and (ii) 2-27.

Table 10: Average spectral distortion for various error rates and for various codes.

Code	$P_e = 0$	10^{-4}	10^{-3}	10^{-2}
SQ-1016	1.48	1.48	1.50	1.62
S-4096-2	1.04	1.04	1.07	1.35
M1-4096-2	0.954	0.958	0.990	1.32
M16-A-2-27	1.02	1.02	1.05	1.30
M8-8-9	0.955	0.958	0.985	1.27
WLSP-4096-1024	0.998	1.00	1.04	1.34
M8-64-4	0.970	0.974	1.00	1.26
M8-64-4 (unsorted)	0.970	0.975	1.00	1.31
M16-2-32	0.679	0.687	0.735	0.956
CO-M16-2-32	0.856	0.861	0.893	1.032

The performance (average SD, percentage of outliers within 2 and 4 dB, and percentage of outliers above 4 dB) in the presence of channel errors is shown in Tables 10, and 11. The increase in the number of outliers is a much better indication of degradation in performance in the presence of channel errors than average SD since errors occur relatively infrequently but may cause a very large (and very audible) spectral error. The results for the split VQ 4096-2 code is comparable (although slightly better in terms of outlier performance) to that obtained in [68], although the scalar quantizer performance (especially at an error rate of 10^{-2}) was much superior to that obtained in [68]. It is quite possible that the scalar quantizer indices were not encoded with a Gray code in [68].

Tables 10 and 11 show that while VQ based systems have lower average SD and lower 2-4 dB outliers even with transmission errors, SQ-1016 (the scalar quantizer used in FS-1016 [69]) may lead to lower 4 dB outliers particularly at high error rates.

Performance was only marginally better for the 64-4 pseudo-Gray coded

Table 11: Percentage of outliers within 2-4 dB and greater than 4 dB for various error rates and for various codes.

Code	$P_e = 0$		$P_e = 10^{-4}$		$P_e = 10^{-3}$		$P_e = 10^{-2}$	
	2-4	> 4	2-4	> 4	2-4	> 4	2-4	> 4
SQ-1016	11.4	0.01	11.4	0.02	11.8	0.19	15.3	1.7
S-4096-2	0.53	0.00	0.57	0.04	1.00	0.40	4.79	4.1
M1-4096-2	0.39	0.00	0.45	0.05	0.97	0.41	5.73	4.2
M16-A-2-27	0.58	0.00	0.63	0.04	1.09	0.34	5.58	3.3
M8-8-9	0.23	0.00	0.28	0.03	0.82	0.33	5.86	3.6
M16-4096-1024	0.53	0.00	0.59	0.045	1.00	0.45	5.03	4.2
M8-64-4	0.35	0.00	0.41	0.036	1.00	0.30	6.09	3.2
M8-64-4 (unsorted)	0.35	0.00	0.40	0.050	0.92	0.37	6.06	4.0
M16-2-32	0.11	0.01	0.21	0.089	0.96	0.56	4.20	2.93
CO-M16-2-32	1.01	0.013	1.18	0.035	2.16	0.23	6.94	1.06

code compared to the 64-4 unsorted code. Although the first stage had inherent robustness since it was initialized using a *splitting* procedure (see [70]) the latter stages were designed randomly and thus had no such structure. The channel optimized 2-32 code (CO-M16-2-32) was optimized to minimize the noisy channel criteria [70] at a random error rate of 10^{-2} . For this code, the channel optimized VQ design is trivial since the index assignment optimization is not required. Although average SD is often higher for the noisy channel optimized code than the clean channel optimized code, the percentage of outliers above 4 dB is significantly less for the noisy channel optimized code. By judicious choice of the weighting factor ρ (again $\rho = 2$ was used in Tables 10 and 11), codes with fewer or more outliers can be obtained. Tests were conducted that indicated that the design algorithm may diverge in the noisy channel case for large ρ (i.e. $\rho = 5$ or more). Note that the channel optimized 2-32 code obtains performance superior to (SQ-1016) scalar quantization in almost every case considered.

3.10. Subjective Evaluation

Subjective evaluation of the WLSP codes based on a 4 kb/s CELP coder is shown in Table 12. The nominal rate is 4 kb/s, although when testing the various codes the remaining parameters were fixed, and thus the rate was often slightly more or less than 4 kb/s (24 bits were reserved for the LPC parameters). In this way only the effect of changing the quantization of $A(z)$ was considered.

The MOS tests were conducted (at Simon Fraser University [71]) using 23 listeners and 8 speakers (4 male and 4 female). The MOS scores are considered accurate to within ± 0.16 MOS points with 95% probability.

A-B pairwise preference tests (see Appendix B and/or [72]) were also conducted to compare the performance of a 4 kb/s codec with quantization of the short term filter based on MSVQ and the performance based on no quantization of the short term filter. The codec was designed assuming 24 bits/frame for short

Table 12: MOS results for the unquantized case, three codes at constant complexity (M45-8-6, M45-8-8, and M45-8-10), and three codes at constant rate/frame (M8-512-2, M45-8-6, and M25-2-18).

Code	Rate	Complexity	MOS	SD	Outliers (%)	
					2-4 dB	> 4 dB
Unquantized	-	-	3.23	0	0	0
SQ-1016	34	-	2.89	1.64	19.7	0.44
M45-8-6	18	14	2.76	1.51	18.1	0.15
M29-8-8	24	14	2.84	1.17	2.12	0.15
M21-8-10	30	14	3.07	0.85	0.29	0.07
M8-512-2	18	16	2.77	1.58	13.8	0.15
M45-8-6	18	14	2.76	1.61	18.1	0.15
M25-2-18	18	12	2.58	1.78	28.2	0.15

term filter quantization. Thus, the rate may be slightly higher or lower depending on the MSVQ code in question. The preference for the unquantized system is displayed in Table 13 and in Fig. 33.

A number of observations can be made based on Tables 12 and 13:

1. A spectral distortion very near 1 dB is required for *truly transparent* quantization, even at 4 kb/s.
2. 24 bits/frame MSVQ obtains MOS performance equivalent to the 34 bits/frame FS-1016 LSP quantizer.
3. At a constant rate (18 bits per frame), the 8-6 code obtains an MOS equivalent to the 512-2 code at a much lower complexity. The 2-18 code is only slightly inferior.

3.11. Summary

In this chapter computational and memory efficient multi-stage vector quantization was applied to quantization of log area ratios based on a MSE distortion, line spectrum pairs based on a weighted MSE distortion, and the predictor pa-

Table 13: Preference for the unquantized case (A, which obtained an SNR_{SEG} of 7.87 dB). The short term filter for stimulus B was quantized using TS-MSVQ based on LSPs and one of the following structures: (i) 16-5 (ii) 16-6 (iii) 16-7. The error in the preference is within 7 percentage points (with 95 % confidence).

B	Preference	SD	SNR_{SEG}
16-5	67 %	1.42	7.30
16-6	58 %	1.13	7.54
16-7	52 %	0.88	7.64

Figure 33: Preference for the unquantized case (A). The short term filter for stimulus B was quantized using TS-MSVQ (based on LSPs and one of the following structures: (i) 16-5 (ii) 16-6 (iii) 16-7.

rameters based on the likelihood ratio. From the results presented herein, the following conclusions may be made:

1. MSVQ is superior in virtually all criteria considered to split VQ.
2. The M - L algorithm converges quite rapidly to the full search solution. Using a larger M_a and more stages is often superior from a complexity-performance-rate trade-off.
3. A relatively large value of M should be used during the design procedure. If a sequential ($M_a = 1$) search is used during the design procedure, inferior performance may result.
4. Joint design procedures are superior in terms of convergence speed, and often in practice, convergence depth;
5. By increasing the number of stages more robustness against input conditions can be realized.
6. By appropriately weighting the distortion measure during the centroid calculation, improved performance on outliers (and thus improved robustness) can be obtained.
7. LARs and LSPs are almost equivalent if the likelihood ratio is used as a selection criterion in the final stage. This result tends to imply that LARs are as suitable as LSPs in an MSVQ configuration.
8. Performance when using the predictor parameters and the likelihood ratio distortion was almost as good as the line spectrum pairs with a weighted mean squared error criterion. However, the memory size was quite large in the former case.
9. Very near 1 dB (or less) is required for transparent quantization at 8 and 4 kb/s.

Chapter 4.

Excitation Codebook Design

In this chapter the design of general and structured CELP codebooks is considered. As discussed in Section 2.5, structure is often required in order to reduce the large memory size and/or computational requirements of the CELP search procedure.

In Section 4.1 the impact of adding structure on the rate required for a given distortion for gaussian excitation is analyzed using both multi-stage VQ arguments and the Blahut Algorithm in conjunction with the Pearlman-Finamore algorithm. It is determined that theoretically very little is lost in the low rate region by adding significant structure.

In Section 4.2 the design of structured and unstructured CELP codebooks is considered. A variation on the (closed-loop) Generalized Lloyd Algorithm previously studied in [33] is used. The treatise in this chapter follows closely that of [6].

4.1. Rate-Distortion Bounds For Gaussian Excitation

4.1.1. Multi-stage VQ (MSVQ)

Multi-stage VQ can be shown to be asymptotically optimal under certain

conditions. For an uncorrelated gaussian source x_0 , the Distortion Rate function (assuming an MSE distortion and R_0 bits per sample) is

$$D_0(R_0) = 2^{-2R_0} \sigma_0^2. \quad (252)$$

Optimal quantization as shown in Fig. 34a results in an average mean squared error (between x_0 and y) of $D_0(R) = \sigma_1^2$. By defining $x_1 = x_0 - y$, the distortion rate function of x_1 (R_1 bits per sample) is upper bounded by the rate distortion function of a gaussian source with equal variance

$$\begin{aligned} D_1(R_1) &\leq 2^{-2R_1} \sigma_1^2 \\ &\leq 2^{-2(R_0+R_1)} \sigma_0^2. \end{aligned} \quad (253)$$

Quantization of x_0 using $R = R_0 + R_1$ bits cannot obtain performance less than the rate distortion bound ($2^{-2R} \sigma_0^2$), and therefore x_1 is gaussian, and

$$\begin{aligned} D_1(R_1) &= 2^{-2R_1} \sigma_1^2 \\ &= 2^{-2R_1} D_0(R_0) \\ &= 2^{-2(R_0+R_1)} \sigma_0^2. \end{aligned} \quad (254)$$

Theoretically, minimum MSE quantization of uncorrelated gaussian samples can be carried out in two (or more) stages, with no loss of optimality.

4.1.2. Ternary Codes

In this section, it is shown that by using ternary codes asymptotic performance very near the rate distortion bound in the low rate region can be obtained.

The Pearlman-Finamore algorithm, discussed in Section 2.1.3, is based on a pre-quantizer in conjunction with the Blahut algorithm to compute the transitional probabilities for the discrete source. In this research, the input levels are designed by using a relatively fine Lloyd-Max scalar pre-quantizer, and only three output

(a)

(b)

Figure 34: Multi-stage quantization.

levels are chosen from the same set of levels as the Lloyd-Max pre-quantizer. Due to symmetry, the center output level is forced to be zero ($y_0 = 0$), and the minimum and maximum output levels are forced to be equal ($y_1 = y_{-1}$). The search is constrained to a search for y_1 over all possible output values of the front end Lloyd-Max pre-quantizer. Due to the constrained search, an exhaustive search may be used.

In Fig. 35 the performance bound of the ternary code and the rate distortion function is shown. The function $R_3(D)$ is an upper bound on the attainable performance of quantizers with only 3 (ternary) output levels. It is confirmed that virtually nothing is lost in using a sparse ternary code at rates below 0.5 bits per sample. In Fig. 36 the rates for the various optimal output probabilities are shown. At rates below 0.5 bits/sample more than 70% of the output values should be zero.

Thus, in this short section it has been theoretically shown that the output levels in the low rate region can be severely constrained without adversely affecting the theoretical performance.

4.2. The Design of General Codebooks

The Generalized Lloyd Algorithm (GLA) requires a sequence of training vectors and a suitable distortion and centroid measure. The training sequence (size N_t) consists of the weighted synthesis filter and the weighted error vector before the contribution of the excitation is determined (y_2 in (121)). The GLA, applied to CELP consists of the steps shown in Fig. 37 [33]. As observed in [33], the training sequence vectors ($y_{2,n}$) are a function of the codevectors due to feedback through the adaptive codebook or long term model filter. Thus, convergence is not guaranteed although in practice it is observed. Strictly speaking, therefore, the algorithm is not a GLA since the minimum distortion partition (over all time) is not determined.

Figure 35: Rate-distortion function for white gaussian noise and performance bound on ternary codes. (129 level Lloyd-Max pre-quantizer).

*Figure 36: $R(D)$ as a function of the optimal output probabilities ($P(y_0)$).
At zero rate, the probability of the zero output is (or course) unity.*

1. Set $j = 0$ and choose an initial codebook.
2. Partition each vector (for $n = 0$ to $N_t - 1$) from the training sequence into one of L_c cells according to the minimum distortion (nearest neighbour) search criterion (from (124))

$$\epsilon^{(i_n)} = \min_{l_n} \epsilon^{(l_n)} = \min_{l_n} \left(\epsilon^{(M_n)} - 2(x_n^{(l_n)} \mathbf{c}_{2,n}^T \mathbf{v}^{(l_n)} + (x_n^{(l_n)})^2 \mathbf{v}^{(l_n)T} \mathbf{R}_n \mathbf{v}^{(l_n)} \right). \quad (255)$$

3. For each cell, compute a new centroid which minimizes the distortion over that cell. For the i -th cell, compute

$$\sum_{c_i} \epsilon^{(i_n)} = \sum_{c_i} \left(\epsilon^{(M_n)} - 2(x_n^{(i_n)} \mathbf{c}_{2,n}^T \mathbf{v}^{(i_n)} + (x_n^{(i_n)})^2 \mathbf{v}^{(i_n)T} \mathbf{R}_n \mathbf{v}^{(i_n)} \right). \quad (256)$$

For the i -th cell, $\tilde{l}_n = i$ (a constant for a particular cell) and thus

$$\sum_{c_i} \epsilon^{(i)} = \sum_{c_i} \epsilon^{(M_n)} - 2 \left(\sum_{c_i} (x_n^{(i)} \mathbf{c}_{2,n}^T) \right) \mathbf{v}^{(i)} + \mathbf{v}^{(i)T} \left(\sum_{c_i} (x_n^{(i)})^2 \mathbf{R}_n \right) \mathbf{v}^{(i)}. \quad (257)$$

The optimal (new centroid) for the given cell is

$$\mathbf{v}^{(i)} = \left(\sum_{c_i} (x_n^{(i)})^2 \mathbf{R}_n \right)^{-1} \left(\sum_{c_i} (x_n^{(i)} \mathbf{c}_{2,n}^T) \right). \quad (258)$$

4. Set $j = j + 1$ and go to step 2.

Figure 37: Codebook design for unstructured codebooks.

For structured codebooks (such as basis vector codebooks, overlapping codebooks, or algebraic codebooks), distinct codevectors may have many elements in common and thus partitioning the training sequence into L_c distinct cells and independent design over each cell is not appropriate. In the following section, design of general codebooks is approached by reformulating the problem whereby explicitly partitioning into separate cells is not performed. The codebook is designed (in essence) in a single step and not by partitioning into cells and designing each codevector independently.

Utilizing long term averaged weighted mean squared error as a distortion measure tends to weight high energy segments much higher than low energy segments. To compensate a segmental approach is suggested, whereby a normalized weighted mean squared error is used and very low energy segments are given zero weight.

4.3. The Design of Structured Codebooks

From (124), the weighted mean squared error over subframe n for a particular codebook index is given by

$$\epsilon^{(l_n)} = \epsilon^{(m_n)} - 2(G_n^{(l_n)} \mathbf{c}_{2,n}^T \mathbf{v}^{(l_n)} + G_n^{(l_n)2} \mathbf{v}^{(l_n)T} \mathbf{R}_{n1} \mathbf{v}^{(l_n)}), \quad (259)$$

In this section, two VQ optimization criteria are considered:

1. average weighted mean squared error

$$\epsilon = \sum_{n=0}^{N_f-1} \epsilon^{(l_n)}; \quad (260)$$

and

2. segmental average weighted mean squared error

$$\epsilon_{\text{seg}} = \sum_{n=0}^{N_f-1} \epsilon^{(l_n)} / \sigma_{s,n}^2 \quad (261)$$

where $\sigma_{s,n}^2 = \sum_{k=0}^{K_s-1} s_{n,k}^2$ is the average speech energy over the n -th subframe. Subframes of low energy ($\sigma_{s,n}^2 < \sigma_{min}^2$) are discarded, and not included in the summation or in the minimization process below.

The first criterion places more emphasis on high energy frames whereas the second criteria attempts to weight all subframes evenly except for very low energy (silent) frames which are given no weight. It was determined that designing the codebooks based on segmental quantities resulted in slightly higher SNR_{SEG} and WSNR_{SEG} than designs based on non-segmental quantities. Perceptual quality was also informally judged to be slightly superior. For these reasons, only segmental design procedures were used for generating the results in this section and the remaining chapters and sections of this thesis.

The algorithm is suitable for codebook structures where the excitation can be written as

$$\mathbf{v}^{(l_n)} = \mathbf{W}_n \mathbf{B}^{(l_n)} \mathbf{C} \quad (262)$$

where $\mathbf{W}_n$ is a symmetric weighting or whitening matrix. The vector $\mathbf{C}$ is a vector of codebook elements, and $\mathbf{B}^{(l_n)}$ is a matrix which represents the rules for constructing $\mathbf{v}^{(l_n)}$ from $\mathbf{C}$.

For example, for a general, unstructured codebook, $\mathbf{W}_n = \mathbf{I}$ (K_c by K_c), $\mathbf{C}$ is a $L_c K_c$ dimensional column vector comprised of the stacked codevectors and $\mathbf{B}^{(l_n)}$ is an upper triangular K_c by $L_c K_c$ Toeplitz matrix with the first row having a single one in position $l_n K_c$ and zeros elsewhere.

For an overlapping or shift symmetric codebook (with t non-overlapping samples) $\mathbf{C}$ is a $K_c + t(L_c - 1)$ dimensional vector and $\mathbf{B}^{(l_n)}$ is an upper triangular K_c by $K_c + t(L_c - 1)$ dimensional matrix with the first row having a single one in position $l_n t$ and zeros elsewhere.

For basis vector structured codebooks the $K_c \log_2(L_c)$ dimensional vector $\mathbf{C}$ is comprised of the stacked basis vectors and $\mathbf{B}^{(l_n)}$ is an upper triangular Toeplitz matrix with the sign of the j -th basis vector in column $K_c j$.

Multiple stage codebooks can also be written in a similar manner by stacking the codebooks for each stage. For stage j and associated index $l_{n,j}$ the excitation for stage j is

$$\mathbf{v}_j^{(l_{n,j})} = \mathbf{W}_n \mathbf{B}_j^{(l_{n,j})} \mathbf{C}_j, \quad (263)$$

and the composite (or total) excitation is

$$\begin{aligned} \mathbf{v}^{(l_n)} &= \mathbf{W}_n \sum_{j=0}^{T_c-1} \mathbf{B}_j^{(l_{n,j})} \mathbf{C}_j \\ &= \mathbf{W}_n \mathbf{B}^{(l_n)} \mathbf{C}. \end{aligned} \quad (264)$$

Assuming the same number of levels (L_c) for each of T_c stages then $\mathbf{C}$ is a $T_c L_c K_c$ dimensional vector and $\mathbf{B}^{(l_n)}$ is Toeplitz K_c by $T_c L_c K_c$ dimensional matrix with sub-matrices $(\mathbf{B}_j^{(l_{n,j})})$ constructed as defined for the general, unstructured codebooks.

Many codebook structures can be written in this manner, such as algebraic codes [73], BPE (binary pulse excitation) [74] and so on.

Substitution of (262) and (259) into (261) leads to

$$\begin{aligned} \epsilon &= \sum_n \frac{\epsilon^{(l_n)}}{\sigma_{s,n}^2}, \\ &= \sum_n \frac{1}{\sigma_{s,n}^2} \left(\epsilon^{(M_n)} - 2 \zeta_n^{(l_n)} \mathbf{c}_{2,n}^T \mathbf{W}_n \mathbf{B}^{(l_n)} \mathbf{C} + \right. \\ &\quad \left. \zeta_n^{(l_n)^2} \mathbf{C}^T \mathbf{B}^{(l_n)T} \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \mathbf{B}^{(l_n)} \mathbf{C} \right), \\ &= \left(\sum_n \frac{\epsilon^{(M_n)}}{\sigma_{s,n}^2} \right) - 2 \left(\sum_n \frac{\zeta_n^{(l_n)}}{\sigma_{s,n}^2} \mathbf{c}_{2,n}^T \mathbf{W}_n \mathbf{B}^{(l_n)} \right) \mathbf{C} + \\ &\quad \mathbf{C}^T \left(\sum_n \frac{\zeta_n^{(l_n)^2}}{\sigma_{s,n}^2} \mathbf{B}^{(l_n)T} \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \mathbf{B}^{(l_n)} \right) \mathbf{C}. \end{aligned} \quad (265)$$

The solution satisfies

$$\left(\sum_n \frac{G_n^{(i_n)^2}}{\sigma_{s,n}^2} \mathbf{B}^{(i_n)T} \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \mathbf{B}^{(i_n)} \right) \mathbf{C} = \sum_n \frac{G_n^{(i_n)}}{\sigma_{s,n}^2} \mathbf{B}^{(i_n)T} \mathbf{W}_n \mathbf{c}_{s,n} \quad (266)$$

and is computable using Choleski Decomposition if the matrix inside the first set of parenthesis is positive definite.

The entire codebook (for one iteration) is designed in a single step, and not codevector by codevector as in the conventional GLA algorithm. However, for general unstructured CELP codebooks, (266) can be split into L_c distinct equations as in the conventional algorithm. The structured codebook design algorithm is outlined in Fig. 38.

4.4. The Design of Sparse Codebooks

Sparse and/or discrete amplitude excitation codebooks can be designed by utilizing an optimization procedure similar to that in Multi-pulse Linear Predictive coding. The goal is to determine $\mathbf{C}$ to minimize (265) while placing constraints on the elements of $\mathbf{C}$. For sparse excitation codebooks, a straightforward multi-pulse procedure can be used, while for sparse discrete amplitude codebooks the multi-pulse procedure must be modified (slightly) to compensate for the discrete amplitude nature of $\mathbf{C}$ as discussed in Section 2.6.5.

Constraints may also be placed on the location and or minimum distance between pulses to take advantage of reduced complexity search techniques as described in [74].

4.5. Codebook Search Complexities

The total search complexity (including pre-computation of $\mathbf{y}^T \mathbf{H}$ and $\mathbf{H}^T \mathbf{H}$) of unstructured codes, as reviewed in Section 2.5.2.2 (and assuming the Autocor-

1. Set $j = 0$ and choose an initial codebook.
2. Partition each vector (for $n = 0$ to $N_t - 1$) from the training sequence into one of L_c cells according to the minimum distortion (nearest neighbour) search criterion (from (124))

$$\epsilon^{(i_n)} = \min_{l_n} \epsilon^{(i_n)} = \min_{l_n} \epsilon^{(M_n)} - 2G_n^{(i_n)} \mathbf{c}_{2,n}^T \mathbf{v}^{(i_n)} + G_n^{(i_n)2} \mathbf{v}^{(i_n)T} \mathbf{R}_n \mathbf{v}^{(i_n)} \quad (267)$$

where

$$\begin{aligned} \mathbf{v}^{(i_n)} &= \mathbf{W}_n \sum_{j=0}^{T_c-1} \mathbf{B}_j^{(i_n,j)} \mathbf{C}_j \\ &= \mathbf{W}_n \mathbf{B}^{(i_n)} \mathbf{C}. \end{aligned} \quad (268)$$

3. Compute the new centroids which minimizes the average distortion (from (265))

$$\begin{aligned} \epsilon &= \left(\sum_n \frac{\epsilon^{(M_n)}}{\sigma_{s,n}^2} \right) - 2 \left(\sum_n \frac{G_n^{(i_n)}}{\sigma_{s,n}^2} \mathbf{c}_{2,n}^T \mathbf{W}_n \mathbf{B}^{(i_n)} \right) \mathbf{C} + \\ &\quad \mathbf{C}^T \left(\sum_n \frac{G_n^{(i_n)2}}{\sigma_{s,n}^2} \mathbf{B}^{(i_n)T} \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \mathbf{B}^{(i_n)} \right) \mathbf{C}. \end{aligned} \quad (269)$$

The new codebook is therefore the solution to

$$\left(\sum_n \frac{G_n^{(i_n)2}}{\sigma_{s,n}^2} \mathbf{B}^{(i_n)T} \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \mathbf{B}^{(i_n)} \right) \mathbf{C} = \sum_n \frac{G_n^{(i_n)}}{\sigma_{s,n}^2} \mathbf{B}^{(i_n)T} \mathbf{W}_n \mathbf{c}_{2,n}. \quad (270)$$

4. Set $j = j + 1$ and go to step 2.

Figure 38: Codebook design for structured codebooks.

relation Approach is used) is approximately

$$C_A = 2(L_c + p)K_c. \quad (271)$$

The search complexity of the basis vector codes is approximately

$$C_v = (B_c + 1)L_c + K_c p + K_c B \left(1 + \frac{1}{2}(B_c + 1) + p \right). \quad (272)$$

For multi-stage codes, the total complexity is (from Section 2.5.2.2) approximately

$$C_M = 2K_c(p + L_c + (T_c - 1)M_c(p + L_c)). \quad (273)$$

For sparse multi-stage codes, with only K_s non-zero samples in each excitation vector, the complexity of the numerator search is now proportional to K_s (rather than K_c) and the denominator search is proportional to the number of non-zero positions in the autocorrelation of the sparse excitation vector (approximately $1 + K_s(K_s - 1)/2$) for a total complexity of

$$C_M = (2K_c p + K_x L_c + (T_c - 1)M_c(2K_c p + K_x L_c)) \quad (274)$$

where

$$K_x = 1 + K_s(K_s + 1)/2. \quad (275)$$

4.6. Performance of Codebook Design Methodologies

Computer simulations were conducted over a relatively large training speech database (TIMIT SX phonetically compact sentences) consisting of 462 speakers (136 females, 326 males) from 8 different (U.S.) dialect regions (approximately

113 minutes of speech). The test sequence consisted of 168 speakers (56 females, 112 males) from the same dialect regions (approximately 40 minutes of speech).

The 16 kHz TIMIT database was down sampled to 8 kHz using the downsampling filter specified in Appendix A. Due to the relatively large size of the TIMIT database, a small subset (500 second) portion was used for excitation codebook design. Thus, the number of training vectors was

$$500 \times 8000 / K_c \quad (276)$$

which results in more than 50,000 training vectors for codebook dimensions less than 80. Thus, even for large codebooks (say $L_c = 4096$) there are at least 10 training vectors/codevector (on average).

Likewise, in this chapter a small portion (200 seconds) of the TIMIT test database was used for testing purposes.

4.6.1. Performance of Random Codebooks

The performance of various codebook structures whereby each codebook element was chosen randomly from the normal distribution is investigated in this section for codecs operating at (nominal) rates of 4 kb/s and 8 kb/s. The codec parameters are shown in Tables 14 and 15. To avoid any problems attributable to short term filter quantization or gain quantization the short term filter and excitation gains were left unquantized. All remaining parameters were fully quantized. Error bars were computed by testing 5 random codes over the same test database.

The performance of various codebook structures whereby each codebook element was chosen randomly from the normal distribution is shown in Figs. 39 and 40 for the nominal 4 kb/s codec and 8 kb/s codec respectively. Three codebook structures were considered in the figure:

1. Unstructured codebooks (L_c codevectors). The rate is just $\log_2(L_c)$.

2. Overlapping codebooks (one sample of overlap, L_c codevectors). Again the rate is $\log_2(L_c)$.
3. Basis vector structured codebooks. In this case, the rate is defined as the number of basis vectors less one, since the basis vector codebooks

Table 14: Parameters of reference 4 kb/s codec.

Parameter	Value
Frame size (N)	280 samples
Subframe size (K_c)	70 samples
Frame gain (G_f)	4 bits/frame
Short term filter ($A(z)$)	24 bits/frame
Adaptive codebook index (M)	7 bits/subframe
Adaptive codebook gains (b)	5 bits/subframe
Excitation gain (G)	5 bits/subframe
Excitation indices (l)	12 bits/subframe
	144 bits/frame (≈ 4 kb/s)

Table 15: Parameters of reference 8 kb/s codec.

Parameter	Value
Frame size (N)	160 samples
Subframe size (K_c)	40 samples
Frame gain (G_f)	4 bits/frame
Short term filter ($A(z)$)	28 bits/frame
Adaptive codebook index (M)	7 bits/subframe
Adaptive codebook gains (b)	5 bits/subframe
Excitation gain (G)	6 bits/subframe
Excitation indices (l)	14 bits/subframe
	160 bits/frame (≈ 8 kb/s)

have an inherent sign bit whereas the unstructured codebooks do not (and the sign is included as part of the gain term). Thus, the rate is $\log_2(L_c) - 1 = B_c - 1$.

The basis vector codes have relatively poor performance compared with the less structured overlapping codes or unstructured codes. Furthermore, by considering the error bars of Figs. 39 and 40 the well structured random basis vector codes have considerable variance across different random codes, whereas the full search unstructured codes have similar performance across different random codes. Thus, it is evident that either proper choice or design of the basis vector codes is essential.

The overlapping codebooks have performance very close to the unstructured codebooks, particularly at high rates. For the reference 8 kb/s codec, at 12 bits per vector the overlapping and unstructured codes perform almost identically. At lower rates, the difference in performance becomes larger (more than one bit) although the difference can be made smaller by increasing the shift between successive codevectors as shown in Table 16 (and in [30, 48]).

Table 16: Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of an unstructured random code at 7 bits/vector, and overlapping random codes with various degrees of overlap.

Overlap (t)	SNR_{SEG}	$WSNR_{SEG}$
7() (unstructured)	6.35	9.93
1	6.00	9.62
2	6.19	9.82
4	6.29	9.89
8	6.35	9.92

Table 17: The objective performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of random full search multi-stage codes at rates of 8 and 12 bits per subframe for the reference 4 kb/s codec.

Code	Rate/subframe	SNR_{SEG}	$WSNR_{SEG}$
16-2	8	6.42	9.96
4096-1	12	7.80	11.11
64-2	12	7.58	10.91
16-3	12	7.37	10.72
8-4	12	7.15	10.54
4-6	12	6.78	10.22
2-12	12	6.14	9.70

Table 18: The objective performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of random full search multi-stage codes at rates of 8 and 12 bits per subframe for the reference 8 kb/s codec.

Code	Rate/subframe	SNR_{SEG}	$WSNR_{SEG}$
16-2	8	9.62	12.71
4096-1	12	11.34	14.13
64-2	12	11.15	13.95
16-3	12	10.89	13.73
8-4	12	10.54	13.46
4-6	12	10.19	13.11
2-12	12	9.08	12.26

Figure 39: Performance (SNR_{SEG} and $WSNR_{SEG}$) of random codebooks for various codebook sizes (or rates) for the nominal 4 kbit/s codec. Three codebook structures are considered: (i) Unstructured (ii) Overlapping (iii) Basis vector. The error bars are 95% confident intervals.

Figure 40: Performance (SNR_{SEG} and $WSNR_{SEG}$) of random codebooks for various codebook sizes (or rates) for the nominal 5 kbps codec. Three codebook structures are considered: (i) Unstructured (ii) Overlapping (iii) Basis vector. The error bars are 95% confidence intervals.

The performance and complexity of 8 bit and 12 bit random full search multi-stage codes is shown in Tables 17 and 18 for the nominal 4 kb/s codec and 8 kb/s codec respectively. A full search is employed so the impact of the structure imposed by the multi-stage configuration can be assessed. The codes are identified by the numbers of levels in each stage and the number of stages. For example, the 16-3 code has 16 levels (4 bits) per stage and 3 stages (12 bits total) and the 64-2 code has 64 levels per stage and 2 stages (again 12 bits total). The performance of the 64-2 code is slightly inferior (less than 1 bit) to the unstructured 4096 level (12 bit) code. As more structure is added (by increasing the number of stages at a constant rate) the performance penalty becomes more severe. Note that the 2-12 code is very similar to an 11 bit basis vector code.

4.6.2. Performance of Designed Codebooks

In this section the performance of designed structured codebooks is investigated. For multi-stage codes, the full search performance is considered first to determine the performance penalty due to the structure of the codes. The performance of the codes are then investigated when an M - L search procedure is used rather than a full search procedure.

The performance of unstructured codebooks of various sizes designed as described in Section 4.2 is displayed in Tables 19 and 20 for the reference 4 kb/s and 8 kb/s codec respectively. The improvement over random codebooks is approximately 0.5 dB or 2 bits/vector from the SNR_{SEG} values or approximately 0.6-0.8 dB or 3 bits/vector from the WSNR_{SEG} values. The improvement in terms of objective measures is quite significant.

The full search performance of designed multi-stage codebooks (64-2), overlapping multi-stage codebooks (64-2), sparse multi-stage codebooks and sparse overlapping multi-stage codebooks is displayed in Tables 21 and 22 for the refer-

ence 4 and 8 kb/s codecs respectively. The performance with an M - L tree search procedure versus complexity of the sparse 64-2 code ($\Delta = 10$) as in Table 21 is shown in Fig. 41. The performance of a designed sparse ($\Delta = 10$) 16-3 code is also shown in Fig. 41. The complexity of the sparse excitation codebooks was discussed in Section 4.5. From Fig. 41 it is evident that from a complex-

Table 19: Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of unstructured codebooks designed based on the GLA for various codebook sizes (for the reference 4 kb/s codec).

Levels	Rate	Designed		Random	
		SNR_{SEG}	$WSNR_{SEG}$	SNR_{SEG}	$WSNR_{SEG}$
256	8	7.31	10.96	6.67	10.19
512	9	7.54	11.12	6.99	10.44
1024	10	7.79	11.31	7.26	10.67
2048	11	8.01	11.47	7.54	10.89
4096	12	8.21	11.63	7.80	11.11

Table 20: Performance (SNR_{SEG} and $WSNR_{SEG}$ both in dB) of unstructured codebooks designed based on the GLA for various codebook sizes (for the reference 8 kb/s codec).

Levels	Rate	Designed		Random	
		SNR_{SEG}	$WSNR_{SEG}$	SNR_{SEG}	$WSNR_{SEG}$
256	8	10.46	13.60	9.92	12.96
512	9	10.85	13.91	10.30	13.27
1024	10	11.17	14.16	10.66	13.56
2048	11	11.49	14.44	11.01	13.86
4096	12	11.78	14.66	11.34	14.13

Table 21: Performance (SNR_{SEG} and WSNR_{SEG} both in dB) of full search structured MSVQ codebooks designed based on the GLA for various codebook sizes and structures for the reference 4 kb/s codec.

Code	Special Structure	SNR_{SEG}	WSNR_{SEG}
64-2	-	8.04	11.49
64-2	overlapping	7.60	11.06
64-2	sparse ($\Delta = 10$, $N_s = 4$)	7.97	11.35
64-2	sparse overlapping ($\Delta = 10$, $N_s = 6$)	7.37	10.80

Table 22: Performance (SNR_{SEG} and WSNR_{SEG} both in dB) of full search structured MSVQ codebooks designed based on the GLA for various codebook sizes and structures for the reference 8 kb/s codec.

Codec	Special Structure	SNR_{SEG}	WSNR_{SEG}
64-2	-	11.54	14.41
64-2	overlapping	11.20	14.05
64-2	sparse ($\Delta = 8$, $N_s = 2$)	11.38	14.18
64-2	sparse overlapping ($\Delta = 10$, $N_s = 5$)	11.07	13.99

ity/performance point of view, moving to more stages (at a constant rate of 12 bits/frame) offers no distinct advantage.

From Tables 19, 20, 21, and 22 the objective (SNR_{SEG}) performance of the 12 bit sparse MSVQ codebook is approximately equal to the performance of the 12 bit unstructured random codebooks or the 11 bit unstructured designed codebooks. Thus, very little performance penalty has been imposed by using the well structured multi-stage codes.

From Fig. 41, the SNR_{SEG} performance penalty when an M - L search procedure is used with the multi-stage codes is very low for $M_c > 8$. From the same

figure and from Table 19 SNR_{SEG} performance equivalent to the unstructured designed 10 bit codebooks can be obtained for $M_c = 2$. At $M_c = 1$ (sequential search), the performance is only slightly better than the unstructured designed 9 bit codebook. Thus, at 4 kb/s, approximately 1 bit is lost due to the structure imposed on the multi-stage codes, and up to 2 additional bits are lost if a sequential search procedure is used. Furthermore, the 64-2 code (12 bit) with $M_c = 2$ offers approximately equivalent SNR_{SEG} performance to the 4096 level (12 bit) random codebook (from Fig. 41 and Table 19).

For random codebooks (see the previous section), it was quite evident that as the number of stages increased (at a constant rate), considerable degradation in performance was incurred. It is apparent that this is also the case for designed codebooks. Furthermore, as more stages are added, the complexity of pre-computation required in the M - L search procedure is significant, and adding more stages does not decrease the complexity considerably. Considering the complexity performance trade-offs, the sparse 2 stage multi-stage codebooks perform very well.

Informal listening tests also indicated that for a particular rate very little (if any) difference was detectable between the random full search unstructured codes and the sparse multi-stage codes.

4.7. Summary

In this section the design of CELP excitation codebooks was considered. It was observed that good performance complexity trade-offs could be realized with multi-stage codes with a small number of stages (i.e. 2). The multi-stage codes are inherently multi-rate and multi-complexity and thus provide a nice trade-off between rate, complexity and performance. The multi-stage codes can obtain performance as good as random unstructured codes at a much lower complexity

Figure 41: Performance of M-L search MSVQ structured codebooks versus complexity for the reference 4 kb/s codec. (i) 16-3 ($\Delta = 10$, $N_s = 4$) (ii) 64-2 ($\Delta = 10$, $N_s = 4$). The points on the curves (from left to right) correspond to $M_c = 1, 2, 4, 8, 16, 64$.

Chapter 5.

Joint Search Procedures Applied to CELP

In this chapter, both iterative sequential (IS-CELP) and open loop/closed loop (OC-CELP) optimization of the parameters of a CELP codec are used in order to improve the objective performance and (hopefully) the subjective quality of the CELP algorithm. The goal is fully joint optimization of the parameters of a CELP codec. However, due to the nature of the algorithms, at best a suboptimal local minimum is found.

5.1. Optimal CELP Search Procedure

The optimal CELP search procedure must determine the sequence of codebook indices which minimizes the weighted mean squared error. The CELP search procedure is again shown in Fig. 42. The weighted mean squared error (over subframe n) may be written as

$$\begin{aligned}\epsilon_n^{(l,m)} &= \left| \mathbf{y}_n - \mathbf{u}_n - \hat{\mathbf{y}}_n^{(m)} - \hat{\mathbf{y}}_n^{(l)} \right|^2 \\ &= \left| \mathbf{y}_n - \mathbf{u}_n - \mathbf{H}_n \mathbf{D}_n^{(m)} \mathbf{b}_n^{(m)} - G_n^{(l)} \mathbf{H}_n \mathbf{v}^{(l)} \right|^2\end{aligned}\tag{277}$$

where $\mathbf{u}_n$ is the zero input response of the weighted synthesis filter ($W(z)/A(z)$), $\mathbf{y}_n$ is the weighted speech vector, $\mathbf{H}_n$ is the Toeplitz (K_r by K_r) impulse response

Figure 42: The CELP search procedure.

matrix of $W(z)/A(z)$, $\hat{\mathbf{d}}_n^{(m)} = \mathbf{D}_n^{(m)} \mathbf{b}_n^{(m)}$ is the scaled excitation contributed by the adaptive codebook and $\hat{\mathbf{a}}_n^{(l)} = G_n^{(l)} \mathbf{v}^{(l)}$ is the scaled stochastic excitation. In [44], joint optimization over all indices l and m was performed, resulting in an increased objective and (purportedly) subjective quality. The impulse response matrix $\mathbf{H}_n$, and the zero input response $\mathbf{u}_n$ are both functions of the weighted synthesis filter $W(z)/A(z)$ which is typically optimized at the frame rate. Thus, joint optimization over the short term predictor clearly should consider more than a single subframe.

The weighted mean squared error over *two* adjacent subframes (n and $n + 1$) can be written as

$$\begin{aligned} \epsilon_{n,n+1}^{(l_n, l_{n+1}, m_n, m_{n+1})} = & \left\| \mathbf{y}_{n,n+1} - \mathbf{u}_{n,n+1} - \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{D}_n^{(m_n)} \mathbf{b}_n^{(m_n)} \right. \\ & - \mathbf{S} \mathbf{H}_{n+1} \mathbf{D}_{n+1}^{(m_{n+1})} \mathbf{b}_{n+1}^{(m_{n+1})} - (G_n^{(l_n)} \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{v}^{(l_n)}) \\ & \left. - (G_{n+1}^{(l_{n+1})} \mathbf{S} \mathbf{H}_{n+1} \mathbf{v}^{(l_{n+1})}) \right\|^2. \end{aligned} \quad (278)$$

The $2K_c$ dimensional vector $\mathbf{y}_{n,n+1}$ is the concatenation of the weighted speech vector at time n and $n + 1$, and $\mathbf{u}_{n,n+1}$ is the zero input response over two subframes. The matrix $\mathbf{H}_{n,n+1}$ is the $2K_c$ by $2K_c$ dimensional impulse response matrix of $W(z)/A(z)$ and $\mathbf{H}_b$ is the $(2K_c$ by K_c non Toeplitz) impulse response matrix of $1/B(z)$ over subframes n and $n + 1$. The third term in (278) is the weighted synthetic speech in subframe n and $n + 1$ due to the adaptive codebook parameters in subframe n . Likewise, the fourth term in (278) is the weighted synthetic speech in subframe n and $n + 1$ due to the codebook parameters in subframe n . The final two terms are the adaptive codebook and stochastic synthetic speech vectors (subframe $n + 1$) with the shifting matrix $\mathbf{S}$ placing the result in the correct location (only the last K_c samples of $\mathbf{S} \mathbf{H}_{n+1} \mathbf{D}_{n+1}^{(m_{n+1})} \mathbf{b}_{n+1}^{(m_{n+1})}$ and $G_{n+1}^{(l_{n+1})} \mathbf{S} \mathbf{H}_{n+1} \mathbf{v}^{(l_{n+1})}$ are non zero). Assuming for a moment two subframes per frame, the fully joint optimal solution (over a single frame) of (278) over all possible combinations of $l_n, l_{n+1}, m_n, m_{n+1}$ and $1/A(z)$ is prohibitively complex.

However, once the indices l_n , l_{n+1} , m_n , and m_{n+1} are known, computation of the optimal $G_n^{(l_n)}$, $G_{n+1}^{(l_{n+1})}$, $\mathbf{b}_n^{(m_n)}$ and $\mathbf{b}_{n+1}^{(m_{n+1})}$ is relatively straightforward. For typical parameters at 4 kb/s:

1. Excitation codebook dimension, $L_e = 1024$.
2. Gain codebook dimension, $L_G = 32$.
3. Adaptive codebook dimension, $L_m = 128$.
4. Adaptive codebook gain dimension, $L_p = 16$.
5. $A(z)$ quantized with 28 bits.

Assuming, rather simplistically, two subframes per frames, the fully optimal solution (over a single frame) must consider 2^{80} possible combinations of indices, a relatively daunting task.

Rather than searching for the optimal solution, an iterative sequential solution is attempted by first computing l_n and m_n in a sequential fashion, then computing l_{n+1} and m_{n+1} . Once l_{n+1} and m_{n+1} are computed, l_n and m_n can be re-optimized, by noting that the weighted mean squared error (again over two subframes) when l_{n+1} and m_{n+1} are known is given by

$$\epsilon_{n,n+1}^{(l_n, m_n)} = \left| \mathbf{y}_{n,n+1} - \mathbf{w}_{n,n+1} - \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{D}_n^{(m_n)} \mathbf{b}_n^{(m_n)} - G_n^{(l_n)} \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{v}^{(l_n)} \right|^2 \quad (279)$$

where

$$\mathbf{w}_{n,n+1} = \mathbf{u}_{n,n+1} + \mathbf{S} \mathbf{H}_{n+1} \mathbf{D}_{n+1}^{(m_{n+1})} \mathbf{b}_{n+1}^{(m_{n+1})} + G_{n+1}^{(l_{n+1})} \mathbf{S} \mathbf{H}_{n+1} \mathbf{v}^{(l_{n+1})}. \quad (280)$$

Thus, the $2K_e$ dimensional vector $\mathbf{w}_{n,n+1}$ is the zero input response plus the weighted speech contributed by the subframe parameters in the next subframe ($n+1$).

This method was not used across frame boundaries or if short term filter interpolation was used, although in principle it could be used since the time varying IIR filter can still be (non-trivially) represented by a matrix formulation.

A number of solutions exist for the manner, or order, in which the sequential optimizations are performed. In this paper, the following sequential approach, with N_I iterations, is chosen:

1. Compute $A(z)$ open loop (autocorrelation method) and the subframe parameters of the CELP codec over an entire frame (using for example, a conventional CELP codec). Set $j = 0$.
2. Re-optimize $A(z)$ to minimize the weighted mean squared error (over the frame), using the given subframe parameters (from step 1 or step 3). Set $j = j + 1$. If $j = N_I$ exit.
3. Re-optimize the subframe parameters on a subframe by subframe basis using $A(z)$ from step 2. At this point, (277) or (279) can be used, since future tentative excitations are known (from previous iterations, or from step 1).
4. Goto step 2.

Step 1 consists of the steps in a conventional CELP codec. The adaptive codebook parameters are computed assuming no stochastic excitation, and then the stochastic codebook parameters are chosen given the adaptive codebook parameters. This step is termed iteration zero. In step 2 $A(z)$ is re-optimized for the given excitation over the entire frame. In step 3, the excitation parameters for each subframe are sequentially re-optimized.

In the following section, closed loop optimization (step 2) is presented, and excitation re-optimization is again considered in Section 5.3.

5.2. Iterative Closed Loop Optimization of the Short Term Predictor

5.2.1. Unquantized Case

In order to determine an upper bound on the performance in the case of model filter quantization, the unquantized case is first considered.

Assume the target excitation

$$\hat{\mathbf{d}} = [\hat{d}_0, \hat{d}_1, \hat{d}_2, \dots, \hat{d}_{N-1}]^T \quad (281)$$

has been computed (for the entire frame) using, for example, a conventional CELP algorithm. The optimal short term predictor is that which minimizes $\epsilon_{we} = \mathbf{e}^T \mathbf{e}$ (Fig. 43), the average weighted mean squared error over N samples, over all model filters $1/A(z)$, with possibly the additional constraint that $A(z)$ has zeros within the unit circle. With $\hat{d}_k$ known, ϵ_{we} (in Fig. 43) is a function of $A(z)$ only. A gradient based algorithm can be applied to find a local minimum over the p dimensional surface.

Assuming an implementation of $A(z)$ parameterized by the vector $\mathbf{k}$ is used (where $\mathbf{k}$ could represent a set of predictor parameters, reflection coefficients, line spectrum pairs and so on), then

$$\frac{d\epsilon_{we}}{d\mathbf{k}} = \frac{d}{d\mathbf{k}} \left[\sum_{n=0}^{N-1} \epsilon_n^2 \right] = 2 \sum_{n=0}^{N-1} \epsilon_n \frac{d}{d\mathbf{k}} \epsilon_n = -2 \sum_{n=0}^{N-1} \epsilon_n \frac{d}{d\mathbf{k}} \hat{y}_n, \quad (282)$$

which leads to the update equations

$$\mathbf{k}^{(T+1)} = \mathbf{k}^{(T)} - \mu \sum_{n=0}^{N-1} \epsilon_n \frac{d}{d\mathbf{k}} \hat{y}_n, \quad (283)$$

where from the definition of $W(z)$ in (106)

$$\frac{d\hat{y}_n}{d\mathbf{k}} = \frac{d\hat{s}_n}{d\mathbf{k}} + \sum_{k=1}^p a_{1,k} \frac{\hat{s}_{n-k}}{d\mathbf{k}} - \sum_{k=1}^p a_{2,k} \frac{\hat{y}_{n-k}}{d\mathbf{k}} \quad (284)$$

Figure 43: Closed loop optimization for the target excitation.

ϵ_{we} is the closed loop model filter distortion measure.

and where μ is a small stepsize constant.

A lattice IIR implementation is used, due to the mathematical tractability, ease of stability monitoring, and practically equivalent mean squared error performance when compared to direct form implementations [75]. The lattice equations (for $i = p \dots 1$) are

$$\begin{aligned} f_{i-1,n} &= f_{i,n} + k_i b_{i-1,n-1} \\ b_{i,n} &= b_{i-1,n-1} - k_i f_{i-1,n}, \end{aligned} \quad (285)$$

where $f_{p,n} = \hat{d}_n$ and $\hat{s}_n = f_{0,n} = b_{0,n}$. The elements of the sample by sample gradient vector $d\hat{s}_n/dk_j = db_{0,n}/dk_j = df_{0,n}/dk_j$ are computed from the lattice equations using simple differentiation,[76],

$$\begin{aligned} F_{p,j,n} &= \frac{\partial \hat{d}_n}{\partial k_j} = 0 \\ F_{i-1,j,n} &= \begin{cases} F_{i,j,n} + k_i B_{i-1,j,n} & i \neq j \\ F_{i,j,n} + k_i B_{i-1,j,n} + b_{i-1,n-1} & i = j \end{cases} \\ B_{i,j,n} &= \begin{cases} B_{i-1,j,n-1} - k_i F_{i-1,j,n} & i \neq j \\ B_{i-1,j,n-1} - k_i F_{i-1,j,n} - f_{i-1,n} & i = j \end{cases} \end{aligned} \quad (286)$$

where $F_{i,j,n} = \partial f_{i,n} / \partial k_j$ and $B_{i,j,n} = \partial b_{i,n} / \partial k_j$

Once the new filter parameters are computed by this recursion, the excitation can be re-optimized for the new model filter.

Rather than computing the short term model filter to minimize the weighted mean squared error using a gradient adaptive IIR method, a transversal (FIR) filter (or equation error approach [75]) can be used to minimize the mean squared error between the original and synthetic *excitation* as shown in Fig. 44. The Equation Error method, however, does not minimize the desired distortion (weighted mean squared error), but does have a unique global minimum with respect to the distortion (ϵ_{ee}) in Fig. 44. Furthermore, the Equation Error method is much more computationally efficient.

Figure 44: Equation error optimization for a given synthetic excitation.

5.2.2. Quantized Case

In the quantized case the goal is the minimization of the weighted mean squared error over the space defined by the quantized parameters. It is relatively straightforward conceptually to optimize $A(z)$ by trying all possible quantized model filters in the CELP structure of Fig. 43, but this is usually not practical for complexity reasons. The problem fits into a wide class of combinatorial optimization problems (with the travelling salesman problem also belonging to this class).

Again the reflection coefficients are used, and the quantization procedure described in [51] is used to define the quantized parameter space. A Simulated Annealing (SA) procedure [54, 55, 56] was used to find a near optimal choice of quantized parameters for the chosen excitation parameters. Simulated annealing was discussed previously in Section 2.6.4.

In the CELP structure, a random generator of moves from one state of (short term filter) codebook indices to a new state of codebook indices introduces a perturbation into the system. The objective function determines the increase or decrease in the cost of this perturbation. If the perturbation results in a decrease in the objective function the change is accepted, otherwise the change is probabilistically accepted, depending on the increase in the cost function and the effective temperature.

The random generator of moves was chosen such that each codebook index was changed randomly and independently by ± 1 , with a probability equal to

$$P_i(j) = \begin{cases} \alpha & j < N_f \\ \alpha \exp\left(\frac{N_f - j}{\kappa_p}\right) & j \geq N_f \end{cases} \quad (287)$$

where j is the iteration index and κ_p is a constant which determines the rate at which the magnitude of the random perturbations are decreased with decreasing temperature (or increasing time). The sign of the change in a particular codebook

index is equiprobable and the perturbation is limited (clipped) to keep it in the range of allowable codebook indices.

The annealing schedule is defined by

$$t = \begin{cases} t_0 & j < N_f \\ t_0 \exp\left(\frac{N_f - j}{\kappa_p}\right) & j \geq N_f \end{cases} \quad (288)$$

where t_0 is the initial equilibrium temperature and κ_p is a constant which determines the rate at which the temperature is decreased.

At iteration j a random perturbation is made, and the quantity

$$q = \exp\left(\frac{\epsilon_{we,j-1} - \epsilon_{we,j}}{\epsilon_{we,j-1}t}\right) \quad (289)$$

is computed. If $q \geq 1$ (i.e. $\epsilon_{we,j} < \epsilon_{we,j-1}$) the perturbation is accepted, otherwise the perturbation is accepted with probability q .

The values of the various parameters as shown in Table 23 were found to be a good trade-off between complexity and performance (less than 0.2 dB from the optimal unconstrained set of simulated annealing parameters with $N_t = 1000$). It was conjectured that:

1. since the short term filter was most likely near a local minimum (from previous iterations or from open loop analysis), little was gained by using high temperatures; or
2. the error surface is relatively well behaved, and allowing perturbations in the direction of increasing ϵ_{we} is not desirable; or
3. since N_t is relatively small considering the dimension of the problem, it is not advantageous to accept perturbations in the direction of increasing ϵ_{we} .

3

OF/DE

3

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

The discussion thus far has not included intraframe interpolation of $A(z)$. As discussed in [38], interpolation of Line Spectral Frequencies (LSPs), Log Area Ratios, or autocorrelation function (ACF) all perform similarly in terms of the mean opinion score (MOS). In this paper the ACF intraframe interpolation method was used, and (as above) the distortion measure (ϵ_{wre}) in Fig. 43 was used in conjunction with the simulated annealing algorithm to determine a locally optimal interpolated model filter. For a particular $A(z)$ generated from the simulated annealing algorithm, the interpolated values are computed using the ACF interpolation method. For the given excitation and interpolated model filters, the weighted mean squared error can then be computed and used in the simulated annealing procedure as discussed above. Thus, the short term filter codebook indices were optimized such that the weighted mean squared error after interpolation was minimized.

5.3. Excitation Re-optimization

The reference CELP coder at 4 and 8 kb/s has the parameters as shown in Table 24. Four subframes per frame are used, and the frame sizes at 8 and 4 kb/s are 140 samples and 280 sample respectively (35 and 70 sample subframes). The adaptive codebook has three non-zero taps ($q = 1$). The adaptive codebook search first determines the pitch period assuming a single non-zero tap (and no quantization), and then determines the optimal adaptive codebook gains for the chosen pitch period. A sparse shift symmetric (overlapping) stochastic codebook is used, and designed to minimize the weighted mean squared error (of the

Table 23: Values of the parameters of the simulated annealing algorithm.

N_i	N_f	κ_p	t_0	α
200	0	1.5	0	0.5

Table 24: Parameters of the reference 4 and 8 kb/s CELP Codec.

At 4 kb/s, $N = 280$ and $K_r = 70$, and at 8 kb/s $N = 140$ and $K_r = 35$. $\gamma_1 = 1$ and $\gamma_2 = 0.9$ at both 4 and 8 kb/s.

Parameter	Symbol	Value
Short term model filter	$A(z)$	VSELP (Reflection Coefficient) 38 bits
Adaptive Codebook Index	M	$M_{min} = 25, M_{max} = 152$ (7 bits)
Adaptive Codebook Gains	$\mathbf{b}^{(M)}$	5 bits ($q = 1$)
Codebook Index	$\tilde{l}$	9 bits
Codebook Gain and Sign	$G^{(i)}$	6 bits
		140 bits, 4/8 kb/s

reference coder) as described in [6]. The codebooks of adaptive codebook gains and stochastic gains are also designed using VQ techniques, with weighted mean squared error (of the reference coder) as the minimization criterion.

In this section techniques to sequentially re-optimize the stochastic codebook and adaptive codebook excitation are considered. Three cases are investigated, assuming:

1. Adaptive codebook parameters are optimized in the first iteration according to (277) (with $G_n^{(l)} = 0$), and held constant during subsequent iterations. Thus, only the stochastic excitation and the short term model filter are sequentially optimized.
2. Future excitation vectors and adaptive codebook parameters are considered unknown. i.e. the excitation parameters are computed by minimizing (277).

3. Future excitation vectors and adaptive codebook parameters are considered known. i.e. the excitation parameters are computed by minimizing (279).

The simplest form of excitation re-optimization (case 1 above) is to assume that the long term adaptive codebook parameters (for each subframe) are not re-optimized which leads to

$$\epsilon_n^{(l)} = \left| \left(y_n - u_n - H_n D_n^{(M)} b_n^{(M)} \right) - G_n^{(l)} H_n v^{(l)} \right|^2 \quad (290)$$

where H_n is the impulse response matrix of the *optimized* short term model filter in cascade with the weighting filter ($W(z)/A(z)$). The optimal codebook indices for each subframe n can then be computed by minimizing (290) over all l .

Alternatively, and slightly more complex, the adaptive codebook excitation can be computed by first assuming the stochastic excitation is known, and then the stochastic excitation is computed assuming the adaptive codebook excitation is known. That is

$$\epsilon_n^{(m)} = \left| \left(y_n - u_n - G_n^{(i)} H_n v^{(i)} \right) - H_n D_n^{(m)} b_n^{(m)} \right|^2 \quad (291a)$$

$$\epsilon_n^{(l)} = \left| \left(y_n - u_n - H_n D_n^{(M)} b_n^{(M)} \right) - G_n^{(l)} H_n v^{(l)} \right|^2. \quad (291b)$$

Thus, M and $b_n^{(M)}$ the optimal adaptive codebook index and gains over all m is computed first (using the values of $\tilde{l}$ and $G_n^{(\tilde{l})}$ from the previous iteration), and then the optimal codebook index $\tilde{l}$ and gain $G_n^{(\tilde{l})}$ are computed with the new values of M and $b_n^{(M)}$. (During the first iteration, to determine the optimal adaptive codebook parameters, $G_n^{(\tilde{l})}$ is assumed to be zero, as in conventional CELP algorithms).

In a similar manner, (279) can be used assuming knowledge of the future codebook excitation and adaptive codebook excitation (from previous iterations).

Thus, the adaptive codebook parameters are computed first by minimizing

$$\epsilon_{n,n+1}^{(m_n)} = \left| \left(y_{n,n+1} - \mathbf{w}_{n,n+1} - \mathbf{v}^{(i_n)} \mathbf{H}_{n,n+1} \mathbf{H}_b G_n^{(i_n)} \right) - \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{D}_n^{(m_n)} \mathbf{b}_n^{(m_n)} \right|^2 \quad (292)$$

over all m_n and the codebook parameters are then computed (using the new adaptive codebook parameters) by minimizing

$$\epsilon_{n,n+1}^{(l_n)} = \left| \left(y_{n,n+1} - \mathbf{w}_{n,n+1} - \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{D}_n^{(M_n)} \mathbf{b}_n^{(M_n)} \right) - G_n^{(l_n)} \mathbf{H}_{n,n+1} \mathbf{H}_b \mathbf{v}^{(l_n)} \right|^2 \quad (293)$$

over all l_n where

$$\mathbf{w}_{n,n+1} = \mathbf{u}_{n,n+1} + \mathbf{S} \mathbf{H}_{n+1} \mathbf{D}_{n+1}^{(M_{n+1})} \mathbf{b}_{n+1}^{(M_{n+1})} + G_{n+1}^{(i_{n+1})} \mathbf{S} \mathbf{H}_{n+1} \mathbf{v}^{(i_{n+1})}. \quad (294)$$

In this case, the procedure is only used over the first $N/K_c - 1$ subframes out of the total of N/K_c subframes since the IIR filter implementation cannot be trivially represented by a matrix equation since the filters are time varying across the frame boundary (as mentioned previously). On the zeroth iteration, the future excitation is unknown, and the search is based on (277).

5.4. Open Loop/ Closed Loop CELP Optimization

In this section techniques are proposed to re-optimize the short term predictor assuming that the short term predictor is quantized using the multi-stage VQ quantization process developed in Chapter 3. The bit assignments for the various parameters for the reference 4 and 8 kb/s coders are shown in Table 25. Due to the more efficient quantization of the short term predictor, more bits are available for quantization of the excitation. At 4 kb/s it is generally thought that 36 bits per frame to quantize $A(z)$ is excessive.

Table 25: Parameters of the reference 4 and 8 kb/s OC-CELP Codec. At 4 kb/s, $N = 280$ and $K_c = 70$, and at 8 kb/s $N = 140$ and $K_c = 35$. $\gamma_1 = 1$ and $\gamma_2 = 0.9$ at both 4 and 8 kb/s.

Parameter	Symbol	Value
Short term model filter	$A(z)$	MSVQ (M16-16-6) 24 bits
Adaptive Codebook Index	M	$M_{min} = 25, M_{max} = 152$ (7 bits)
Adaptive Codebook Gains	$b^{(M)}$	5 bits ($q = 1$)
Codebook Index	$\tilde{l}$	MSVQ (T16-64-2) 12 bits
Codebook Gain and Sign	$G(i)$	5 bits
		140 bits, 4/8 kb/s

The tree searched multi-stage quantization process generates a number of candidate model filters. By utilizing the IS-CELP technique, these candidate model filters are then used in the closed loop procedure to determine the optimal closed loop model filter using an exhaustive search (over the set of open loop candidates). Thus, rather than using a simulated annealing procedure to find the set of short term filter codebook indices, an exhaustive search is used over the set of candidate indices from the tree searched procedure. The procedure first chooses the best model filter from the set of M_a candidate model filters to minimize the weighted LSP distance measure. The excitation is then computed for the best model filter. The model filter is then re-optimized for the now *known* excitation (as in Fig. 43) over all M_a candidate model filters.

Alternatively, all candidates from the short term filter search procedure can be investigated to determine which leads to minimum weighted mean squared error. This procedure is outlined as follows:

1. Determine the short term filter, and the M_a best sets of candidate indices from the M - L search procedure used in the short term filter quantization procedure.
2. Over the best M_o ($M_o \leq M_a$) model filters, compute the synthetic excitation and based on WMSE, choose the best (minimum WMSE) model filter.

In this case, each model filter (chosen from a subset of those determined in the M - L search procedure) is independently used in the CELP search procedure. Thus, no re-optimization procedure is used. For each model filter from the subset, the adaptive codebook parameters and the excitation parameters for the entire frame are determined. Clearly, the model filter which results in minimal weighted mean squared error is chosen as the best. This technique is referred to as open loop/closed loop CELP (or OC-CELP).

5.5. Objective Results

In this section objective (Weighted SNR_{SEG} and SNR_{SEG}) results are presented for various degrees of optimization.

A number of cases are considered (both at 4 and 8 kb/s). The parameters of Table 24 are used unless explicitly stated otherwise. The reference system is a conventional CELP codec, with three cases considered:

1. No quantization of $A(z)$.
2. Quantization of $A(z)$.
3. Quantization and interpolation of $A(z)$. $A(z)$ is interpolated for each subframe and $W(z)$ is computed open loop for each subframe.

The performance in terms of Weighted SNR_{SEG} (WSNR_{SEG}), SNR_{SEG} (over 20 millisecond segments) at both 4 and 8 kb/s is shown in Table 26.

The first optimization considers $A(z)$ unquantized and optimized closed loop as discussed in Section 5.2.1. Two cases are considered:

1. No re-optimization of $B(z)$. During the first iteration, $B(z)$ is computed to minimize the weighted mean squared error, and this value of $B(z)$ is maintained through subsequent iterations (Equation (291a) with $G_n^{(l)} = 0$).
2. Re-optimization of $B(z)$. In this case, $B(z)$ is re-optimized given the stochastic excitation from previous iterations (Equation (291a)).

The simulations results are shown in Table 27a and 27b and in Fig. 45. Clearly re-optimization of $B(z)$ is desirable. After 7 iterations, re-optimization of $B(z)$ outperforms no re-optimization by at least 0.5 dB (at both 4 and 8 kb/s).

The second optimization considers $A(z)$ quantized and optimized closed loop as discussed in Section 5.2.2. Again two cases are considered:

1. Re-optimization of subframe parameters without knowledge of the future excitation information (Equation (291)).
2. Re-optimization of subframe parameters with knowledge of the future excitation information (Equations (292)-(294)).

Table 26: Performance of reference system. Three reference systems were considered, (R1): No quantization of $A(z)$, (R2): Quantization of $A(z)$, and (R3): Quantization and interpolation of $A(z)$.

System	4 kb/s		8 kb/s	
	WSNR _{SEG} dB	SNR _{SEG} dB	WSNR _{SEG} dB	SNR _{SEG} dB
R1	11.13	8.76	14.88	13.03
R2	11.02	8.62	14.66	12.77
R3	11.50	8.91	15.08	13.09

The results are shown in Table 28a and 28b. Using tentative future excitation information results in a rather modest improvement of approximately 0.2 dB. Quantization of $A(z)$ results in a loss of approximately 0.2 dB at 4 kb/s and approximately 0.4 – 0.5 dB at 8 kb/s (compared to the unquantized case) as shown in Fig. 46.

The third optimization considers $A(z)$ quantized, interpolated every subframe, and optimized closed loop as discussed in Section 5.2.2. Again two cases are considered:

Table 27: Performance of optimized (closed loop) system with unquantized short term filter ($A(z)$). (a): No closed loop re-optimization of $B(z)$. (b) Closed loop re-optimization of $B(z)$.

N_l	4 kb/s		8 kb/s	
	WSNR _{SEG} dB	SNR _{SEG} dB	WSNR _{SEG} dB	SNR _{SEG} dB
1	11.43	9.20	15.26	13.64
3	11.47	9.28	15.43	13.88
5	11.46	9.27	15.47	13.93
7	11.45	9.26	15.48	13.95

(a)

N_l	4 kb/s		8 kb/s	
	WSNR _{SEG} dB	SNR _{SEG} dB	WSNR _{SEG} dB	SNR _{SEG} dB
1	11.43	9.20	15.26	13.64
3	11.80	9.66	15.80	14.29
5	11.91	9.79	15.90	14.42
7	11.95	9.83	15.95	14.49

(b)

Figure 45: Performance of optimized (closed loop) system with unquantized short term filter ($A(z)$). (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents no closed loop re-optimization of $B(z)$ (from Table 27a) and curve (ii) represents closed loop optimization of $B(z)$ (from Table 27b). (0 iterations represents the conventional CELP codec. The error bars are computed with respect to the improvement over the conventional CELP codec).

1. $A(z)$ quantized open loop. In this case no re-optimization of $A(z)$ is performed.
2. $A(z)$ quantized closed loop.

The results shown in Table 29a and 29b and in Fig. 47. From these results it is clear that a large portion of the gain is due to re-optimization of the short term filter.

Comparing Tables 26 (R2) and 29b (7 iterations) it is apparent that a gain of between 1.3 and 1.5 dB (SNR_{SEG}) is available using the sequential optimization

Table 28: Performance of optimized (closed loop) system with quantized short term filter ($A(z)$). (a): Re-optimization of subframe parameters without knowledge of the future excitation information. (b) Re-optimization of subframe parameters with knowledge of the future excitation information.

N_l	4 kb/s		8 kb/s	
	WSNR_{SEG} dB	SNR_{SEG} dB	WSNR_{SEG} dB	SNR_{SEG} dB
1	11.25	8.95	14.87	13.09
3	11.66	9.46	15.38	13.74
5	11.76	9.58	15.52	13.90
7	11.79	9.63	15.58	13.97

(a)

N_l	4 kb/s		8 kb/s	
	WSNR_{SEG} dB	SNR_{SEG} dB	WSNR_{SEG} dB	SNR_{SEG} dB
1	11.25	8.95	14.87	13.01
3	11.88	9.67	15.48	13.86
5	11.99	9.82	15.63	14.05
7	12.03	9.86	15.70	14.14

(b)

Figure 46: Performance comparisons with optimized (closed loop), quantized $A(z)$ and unquantized $A(z)$ (from Tables 27b and 28a). (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents the unquantized case and curve (ii) represents the quantized case.

techniques developed in this thesis. It is also evident, that interpolation of the short term model filter to minimize weighted mean squared error is also beneficial, and results in a gain of approximately 0.3 dB (SNR_{SEG}) over no interpolation (from Table 28a and 29b). Informal listening tests indicate that the perceptual speech quality gain is more than the approximately 0.3 dB would indicate.

The final system considers the use of MSVQ of the short term filter. It was determined experimentally that the IS-CELP method was inferior to the OC-CELP method especially for a large number of iterations and/or large M_o . Intuitively,

Table 29: Performance with quantized, interpolated short term filter ($A(z)$). Re-optimization of subframe parameters without knowledge of the future excitation information. (a): $A(z)$ quantized open loop. In this case no re-optimization of $A(z)$ is performed. (b): $A(z)$ quantized closed loop.

N_l	4 kb/s		8 kb/s	
	WSNR _{SEG} dB	SNR _{SEG} dB	WSNR _{SEG} dB	SNR _{SEG} dB
1	11.50	8.91	15.08	13.09
3	11.69	9.13	15.37	13.41
5	11.71	9.15	15.37	13.43
7	11.71	9.15	15.39	13.44

(a)

N_l	4 kb/s		8 kb/s	
	WSNR _{SEG} dB	SNR _{SEG} dB	WSNR _{SEG} dB	SNR _{SEG} dB
1	11.75	9.28	15.34	13.50
3	12.12	9.76	16.82	14.08
5	12.25	9.93	15.93	14.24
7	12.25	9.94	15.98	14.29

(b)

Figur. 7: Performance with quantized, interpolated short term filter $A(z)$. (a) 4 kb/s. (b) 8 kb/s. Curve (i) represents $A(z)$ quantized (and interpolated open loop) (Table 29a) and curve (ii) represents $A(z)$ quantized (and interpolated) closed loop (Table 29b).

for a large number of iterations, OC-CELP should be superior since all candidate model filters are tried, which is not necessarily the case in the IS-CELP method. However, the joint optimization of the adaptive codebook excitation and the stochastic codebook excitation is not performed in OC-CELP method discussed thus far. However, joint optimization of the adaptive codebook and stochastic excitation can be performed by including the adaptive codebook search in the M - L algorithm for the MSVQ excitation codebook search. That is, rather than selecting only the best adaptive codebook index and adaptive codebook parameters from the adaptive codebook search, the best M_b (where $M_b \leq M_c$) candidates are maintained and used in the first stage of the excitation codebook search. The performance of the reference 4 and 8 kb/s OC-CELP coders (with parameters as shown in Table 25) for various values of M_b and M_o is shown in Table 30 and in Fig. 48.

The improvements in the segmental signal to noise ratio (from Fig. 48) are approximately 1.2 and 1.8 dB at 4 and 8 kb/s, respectively. It is evident that slightly better performance may be attainable by using larger values of M_o and M_b .

The improvements due to the OC-CELP optimizations are similar to the improvements obtained with iterative sequential optimization. However, the complexity of OC-CELP is much more manageable than the iterative sequential optimization techniques.

5.6. Subjective Results

In this section, the results of subjective tests which were conducted to evaluate the performance of the codebook design methods (for the excitation and for the short term filter) and CELP optimization procedures developed in this thesis are presented.

Comparisons are made against 32 kb/s ADPCM (G.721), 64 kb/s μ -law PCM (G.711), 8 kb/s VSELP (IS-54), and 4.8 kb/s FS-1016.

A-B comparisons were conducted for IS-54 [51] versus 8 kb/s, 7 kb/s, 6 kb/s and 5 kb/s OC-CELP optimized coders with (unoptimized) codec parameters as shown in Table 31. The A-B preference tests were conducted as discussed in Appendix B over 14 listeners and 16 sentences (8 male, 8 female) for a total of 224 scores per test.

Table 30: Performance of OC-CELP as a function of M_o and M_b . MSVQ with ($M_a = 16$), non-interpolated $A(z)$. (a) 4 kb/s. (b) 8 kb/s.

M_o	WSNR _{SEG} dB (vs. M_b)				SNR _{SEG} dB (vs. M_b)			
	$M_b = 1$	2	4	16	$M_b = 1$	2	4	16
1	11.22	11.44	11.51	11.61	8.53	8.71	8.83	8.96
2	11.32	11.50	11.65	11.83	8.65	8.85	9.00	9.20
4	11.46	11.64	11.79	12.02	8.78	9.00	9.18	9.43
8	11.55	11.76	11.93	12.17	8.90	9.13	9.34	9.62
16	11.63	11.84	12.05	12.28	8.99	9.25	9.46	9.76

(a)

M_o	WSNR _{SEG} dB (vs. M_b)				SNR _{SEG} dB (vs. M_b)			
	$M_b = 1$	2	4	16	$M_b = 1$	2	4	16
1	14.34	14.64	14.86	15.04	12.09	12.41	12.65	12.85
2	14.49	14.81	15.07	15.30	12.27	12.62	12.91	13.14
4	14.65	14.97	15.24	15.56	12.45	12.81	13.11	13.45
8	14.79	15.14	15.41	15.76	12.61	13.02	13.30	13.67
16	14.92	15.27	15.58	15.93	12.77	13.16	13.50	13.88

(b)

Figure 48: Performance of OC-CELP as a function of M_o and M_b . Quantized, non-interpolated $A(z)$. (a) 4 kb/s (from Table 30a). (b) 8 kb/s (from Table 30b). (i) $M_b = 1$ (ii) $M_b = 2$ (iii) $M_b = 4$ (iv) $M_b = 16$.

Table 31: Parameters of the 5, 6, 7, and 8 kb/s OC-CELP Codec.

Parameter	Symbol	Value
Short term model filter	$A(z)$	MSVQ (M16-16-7) 28 bits
Adaptive Codebook Index	M	$M_{min} = 25, M_{max} = 152$ (7 bits)
Adaptive Codebook Gains	$\mathbf{b}^{(M)}$	4 bits ($q = 1$)
Codebook Index	$\tilde{l}$	MSVQ (T16-64-3) 18 bits
Codebook Gain and Sign	$G(i)$	5 bits
Subframes/frame	N_p/K_c	4
Subframe dimension	K_c	67/56/48/42 samples
Coder rate	R	5/6/7/8 kb/s

Results of the A-B listening tests are displayed in Table 32 and in Fig. 49. Using the OC-CELP technique it is apparent that IS-54 quality (clean channel) can be obtained at very near 6 kb/s.

MOS tests were also conducted as outlined in Appendix C. A number of schemes were tested for reference purposes:

1. 8 kb/s VSELP (IS-54) [51];
2. an MNRU of 10 dB [1];

Table 32: Preference for VSELP vs. OC-CELP at various rates. The error is within 7 % with 95 % confidence.

OC-CELP Rate (kb/s)	Preference (for IS-54)	SNR _{SEG} dB
8	33 %	13.44
7	33 %	12.50
6	54 %	11.40
5	61 %	10.02

Figure 49: Preference for VSELP vs. OC-CELP at various rates.

3. 32 kb/s ADPCM [77, 78];
4. 64 kb/s μ -law PCM [78]; and
5. FS-1016 [69].

The results are shown in Table 33 and in Fig. 50. The parameters of the 8 kb/s (OC-CELP-8, CELP-8) and 4 kb/s (OC-CELP-4, CELP-4) codecs are displayed in Tables 31 and 25 respectively. The OC-CELP optimized codecs used $M_b = 16$ and $M_o = 16$. The CELP-8 codecs show a significant performance improvement over the IS-54 standard codec. However, the OC-CELP optimized codecs show no significant subjective improvement (in terms of MOS), while obtaining approximately 1 dB increase in objective measures. There are a number of possible reasons for this lack of noticeable improvement:

1. There is no subjective improvement. Improved distortion measures are thus required to improve the codec quality.
2. There is a subjective improvement although it is quite small and does not show up reliably in the MOS tests. The 1 dB gain is approximately the same gain in SNR_{SEG} when the rate is increased from 7 to 8 kb/s which leads to an improvement which was not noticeable in A-B preference tests (see Table 32). Thus, the improvement afforded by a 1 dB increase in SNR_{SEG} may only show up in formal MOS tests with many (> 100) listeners.

The 8 kb/s CELP-8 and OC-CELP-8 codecs also obtained performance superior to ADPCM although this improvement is not significant due to the rather large confidence interval. However, A-B preference tests indicated that there was a 59 percent preference for the OC-CELP-8 codec over the ADPCM coder which is significant given the confidence interval of the particular A-B preference test (± 6.5 percent). The ADPCM encoder and decoder were implemented by Sun Mi-

crossystems and there is also no guarantee that the ADPCM encoder and decoder are G.721 compliant (although they are claimed to be).

The 4 kb/s (CELP-4 and OC-CELP-4) codecs obtained performance virtually equivalent to FS-1016. Note that the codec parameters and bit assignments were not extensively optimized as with FS-1016 [47].

5.7. Summary

In this chapter two joint search procedures were used to determine the performance improvement attainable by optimizing all the speech coder parameters simultaneously. It was observed that significant gains in objective measures were realized by utilizing these optimization techniques. Corresponding increases in subjective measures were not observed, and it was hypothesized that improved distortion measures and/or more accurate MOS tests are required to obtain a noticeable increase in subjective measures.

Table 33: MOS performance of various codecs.

Coder	Rate	MOS			Error (95 % conf)		
		Total	Male	Female	Total	Male	Female
MNRU Q-10	-	1.33	1.35	1.31	0.10	0.15	0.13
FS-1016	4.8	2.85	3.14	2.56	0.16	0.22	0.20
IS-54	8	3.53	3.74	3.32	0.15	0.21	0.20
ADPCM	32	3.71	3.85	3.57	0.14	0.18	0.19
PCM	64	4.26	4.42	4.10	0.12	0.15	0.19
CELP-8	8	3.88	4.07	3.68	0.14	0.18	0.21
CELP-4	4	2.68	3.00	2.36	0.15	0.21	0.18
OC-CELP-8	8	3.89	3.99	3.79	0.15	0.21	0.20
OC-CELP-4	4	2.75	3.06	2.44	0.14	0.18	0.19

Figure 50: MOS performance of various codecs.

Chapter 6.

Conclusions

In this research, improved search techniques coupled with improved codebook design methodologies were proposed to improve the performance of conventional CELP coders operating at rates between 4 and 8 kb/s. Particular emphasis was placed on the trade-off between performance, rate and complexity to arrive at a codec which may be implemented at a relatively low complexity.

Efficient, robust quantization of the short term predictor is required to arrive at a high quality, robust codec. Multi-stage vector quantization (MSVQ) of the short term filter was proposed. Joint codebook design techniques coupled with improved search procedures were utilized to obtain an efficient, high quality quantizer.

Structured codebook design procedures for independent gaussian random variables were compared to general codebook design procedures. It was determined—both theoretically and in practice— that little is lost in using significant structure in the codebooks, while greatly reducing the search complexity. Excitation codebook design procedures for a wide variety of codebook structures were presented and it was empirically verified that little is lost in adding significant structure to the excitation codebooks.

Improved CELP search procedures which attempt to jointly optimize the short term filter, the adaptive codebook, and the excitation were proposed. Although the computational complexity is rather large, the improvements obtained are significant.

6.1. Contributions to Knowledge

This section lists the key contributions of the work performed. The key contributions to knowledge are:

1. High quality low bit rate, low computational complexity and low memory size encoding of the short term filter at approximately 20-30 bits per frame. Low bit rate encoding based on a weighted LSP distortion measure was shown to obtain superior performance to previous work. It was also shown that in practice the performance of multi-stage quantization depends more on the number of distortion computations rather than on the actual parameter of the multi-stage code. At constant complexity, the performance of different codes (with very different memory size) but similar computational complexity performed similarly. A number of joint codebook design methodologies were investigated to both speed up convergence and guarantee convergence. Various parameter sets and distortion measures were investigated which obtained similar performance. A spectral distortion of 1 dB was obtained for rates as low as 22 bits/frame.
2. Codebook design methodologies for low complexity CELP excitation search algorithms. Severe codebook structure was shown to impose extremely small performance degradation both in practice and in theory (gaussian case). Multi-stage quantization was again utilized to obtain good performance, complexity and rate trade-offs. Low

complexity highly structured codes were shown to obtain performance as good as random unstructured codes.

3. The development of a reduced search, reduced complexity (open loop/closed loop) adaptive codebook search procedure.
4. Enhanced CELP search procedures to jointly optimize the parameters of a CELP codec. The performance improvements of locally optimal search procedures were shown to give an improvement of between approximately 1 and 2 dB in segmental signal to noise ratio.

6.2. Suggestions for Future Work

Considerable work was done on optimization of the various codebooks to obtain a high quality, low rate codec. Suggestions for future research include:

1. Coder parameter optimization. Optimization of the trade-offs between the various codebook dimensions and codebook sizes. Extensive simulation and listening tests are required.
2. Improving robustness to errors and lost packets. Frame reconstruction techniques, forward error correction, and error detection are required to obtain a robust codec in the presence of channel errors. A concise channel model or actual channel measurements are required to determine good trade-offs between the rate required for forward error correction and error detection. The techniques employed in joint source/ channel coding could also be employed to enhance the robustness of the codec to channel errors.
3. Improved distortion measures. The improvement in objective measures often did not lead to a corresponding increase in subjective measures. Thus, further work is required to obtain an objective measure which is more closely correlated with objective measures. Distortions

which take into account the manner in which the average human ear operates are required for improved performance. Postfiltering may also be investigated further to improve synthetic voice quality.

4. **LAR weighting methods.** Unweighted MSE with the LARs were observed to perform as well as weighted distortion measures with LSPs if the Likelihood Ratio was utilized in the final selection process. Suitable weighting for LARs may improve the performance so selection based on the LR is not necessary. The LARs are somewhat simpler to compute than the LSPs.
5. **Application of the techniques to stereo coding and or wideband (7 kHz) speech coding.** Development of new structures to jointly encode the left and right channels simultaneously.
6. **Real time implementation.** Once the codec is fully optimized, a real time version may be implemented in VLSI or on a fixed or floating point digital signal processor.
7. **Formal MOS with approximately 100 (or more) listeners** may also be performed to accurately determine the improvement in quality of the techniques developed in this thesis.

Appendix A. Characteristics of the Downsampling Filter

The downsampling filter is utilized to reduce the rate of the TIMIT database from 16 kHz to 8 kHz. The speech samples are filtered through the downsampling filter and then every second sample is discarded. The frequency response of the downsampling filter is presented in Fig. 51. It is a 129 point FIR filter designed using the Remez exchange algorithm.

Figure 51: Frequency response of the downsampling filter.

Appendix B. A-B Preference Tests

The A-B preference tests were carried out according to the suggestions of [72, 1]. The listeners were discarded if they could not differentiate accurately between an MNRU of 20 and 25 dB. Persons choosing the lower SNR value more than 4 times out of 16 trials were judged to be either poor listeners and/or unmotivated and not included in the test results.

The directions presented to each listener is as shown in Fig. 52. Sixteen phrases (8 male, 8 female), were used for each preference test as shown below:

1. (Female): Young children should avoid exposure to contagious diseases.
2. (Male): December and January are nice months to spend in Miami.
3. (Female): Tradition requires parental approval for under-age marriage.
4. (Male): His shoulder felt as if it were broken.
5. (Female): Daphne's Swedish needlepoint scarf matched her skirt.
6. (Male): Please dig my potatoes up before frost.
7. (Female): A chosen few will become Generals.
8. (Male): The emblem depicts the Acropolis all aglow.
9. (Female): Kindergarten children decorate their classrooms for all holidays.
10. (Male): Special task forces rescue hostages from kidnappers.
11. (Female): Michael colored the bedroom wall with crayons.
12. (Male): Withdraw all phony accusations at once.
13. (Female): Project development was proceeding too slowly.
14. (Male): That noise problem grows more annoying each day.
15. (Female): Combine all the ingredients in a large bowl.

16. (Male): Bob papered over the living room murals.

The ordering of the phrases and the ordering of the test conditions were randomized.

DIRECTIONS FOR THE A-B PREFERENCE TESTS

Thank you for participating in this test.

These tests are being conducted to determine the performance of speech coding systems operating at rates between 4 and 8 kb/s.

Your task is to listen to short segments of distorted speech and to determine which segment (A or B) sounds perceptually superior. Each test segment is independent of previous segments. You must listen to A and B (once) before making your choice. Choose A or B by pressing the left mouse button after pointing the cursor at the A or B button. You should not spend an unnecessary amount of time on each utterance. You are to rate A and B based on your own subjective criteria which should include:

- intelligibility of the A or B utterances;***
- background noise in the A or B utterances;***
- naturalness of the A or B utterances;***
- irritating buzziness, pops, or artifacts.***

You will be given an example trial before commencing. You may ask the instructor questions after the example trial.

Figure 52: Directions for the A-B preference tests.

Appendix C. Mean Opinion Score (MOS) Evaluations

MOS tests were conducted as close as possible to the recommendations of [72, 1, 79]. The phrase ordering and the ordering of males and females was random. Eight TIMIT sentence pairs were utilized in the MOS test (4 male, 4 female):

1. (Female): Young children should avoid exposure to contagious diseases/ Weatherproof galoshes are very useful in Seattle.
2. (Male): December and January are nice months to spend in Miami/ If people were more generous there would be no need for welfare.
3. (Female): Tradition requires parental approval for under age marriage/ Laugh, dance, and sing as fortune smiles upon you.
4. (Male): His shoulder felt as if it were broken/ Military personal are expected to obey government orders.
5. (Female): Daphne's Swedish needlepoint scarf matched her skirt/ The fog prevented them from arriving on time.
6. (Male): Please dig my potatoes up before frost/ I'd ride the subway, but I haven't enough change.
7. (Female): A chosen few will become Generals/ Only the most accomplished artists achieve popularity.
8. (Male): The emblem depicts the Acropolis all aglow/ Only lawyers love millionaires.

In a formal MOS test, more speakers would be required.

The directions presented to each listener is as shown in Fig. 53. Although the MOS scores could conceivably show considerable variation due to the relatively small number of phrases utilized, the standard deviation was computed using a

t-distribution. Thus, strictly speaking, the MOS and deviation is only over the set of 8 TIMIT phrases.

DIRECTIONS FOR THE MOS (Mean Opinion Score) TESTS

Thank You for participating in this test.

These tests are being conducted to determine the performance of speech coding systems operating at rates between 4 and 8 kb/s. You will be listening to a number of recorded speech samples, spoken by several different talkers, and rating how good you think they sound.

On each trial, a sample will be played. After you have listened to a particular passage, you should select a particular rating and enter the rating by pointing the cursor at the correct button and press the left mouse button (once). Select the button corresponding to your rating for the particular passage.

Over the session you will hear samples of varying quality. Take into account your total impression of each sample, rather than concentrating on one particular aspect.

The quality of the speech should be rated according to the scale below:

Rate each passage by choosing the word from the scale which best describes the quality of the speech you heard. You should not spend more than 1 or 2 seconds before entering your score.

There will be 72 trials plus 8 practice trials at the beginning. You may ask the instructor questions after the example trial.

Figure 53: Directions for the MOS (Mean Opinion Score) tests.

References

- [1] "Modulated Noise Reference Unit (MNRU)," CCITT Recommendation P.81, Blue Book, Volume V, 1988.
- [2] W. LeBlanc, S. Hanna, and S. Mahmoud, "Performance of a Low Complexity CELP Speech Coder Under Mobile Channel Fading Conditions," *Vehicular Technology Conference*, 1989.
- [3] B. Bhattacharya, W. P. LeBlanc, S. A. Mahmoud, and V. Cuperman, "Tree Searched Multi-Stage Vector Quantization of LPC Parameters For 4 kb/s Speech Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1992.
- [4] W. P. LeBlanc, V. Cuperman, B. Bhattacharya, and S. A. Mahmoud, "Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 kb/s Speech Coding," *Submitted to: IEEE Transactions on Acoustics, Speech, and Signal Processing*, May 1992.
- [5] W. P. LeBlanc, V. Cuperman, and S. A. Mahmoud, "Joint Design of Multi-Stage VQ Codebooks for LSP Quantization with Applications to 4 kb/s Speech Coding," *Submitted to: Speech and Audio Coding for Wireless and Network Applications*, May 1992.
- [6] W. LeBlanc and S. Mahmoud, "Structured Codebook Design in CELP," *International Mobile Satellite Conference*, pp. 667–672, 1990.
- [7] W. LeBlanc and V. Cuperman, "Sequential Optimization in CELP for Speech Coding at 4 kb/s," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 105–106, 1991.
- [8] W. LeBlanc and V. Cuperman, "Speech Coding at 4 and 8 kb/s Based on Iterative Sequential CELP Optimization," *Globecom*, 1991.

- [9] W. LeBlanc and V. Cuperman, "Sequential Optimization in CELP For Speech Coding at 4 kb/s," *Submitted to: IEEE Transactions on Acoustics Speech and Signal Processing*, 1991.
- [10] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. The University of Illinois Press, 1949.
- [11] C. E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," *IRE National Convention Record*, pp. 142–163, 1959.
- [12] R. G. Gallager, *Information Theory and Reliable Communication*. New York, New York: Wiley, 1968.
- [13] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, New Jersey: Prentice Hall, 1971.
- [14] R. E. Blahut, "Computation of Channel Capacity and Rate-Distortion Functions," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 460–473, July 1972.
- [15] W. A. Finamore and W. A. Pearlman, "Optimal Encoding of Discrete-Time Continuous-Amplitude Memoryless Sources with Finite Output Alphabets," *IEEE Transactions on Information Theory*, vol. IT-26, pp. 144–155, March 1980.
- [16] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 129–137, March 1982.
- [17] J. Max, "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory*, vol. IT-6, pp. 7–12, March 1960.
- [18] H. Gish and J. N. Pierce, "Asymptotically Efficient Quantizing," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 676–681, September 1968.

- [19] R. C. Wood, "On Optimum Quantization," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 248–252, March 1969.
- [20] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proceedings of the IEEE*, vol. 73, pp. 1551–1588, November 1985.
- [21] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.
- [22] R. M. Gray and H. Abut, "Full Search and Tree Searched Vector Quantization of Speech Waveforms," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1982.
- [23] J. Chen and A. Gersho, "Gain Adaptive Vector Quantization for Medium Rate Speech Coding," *IEEE International Conference on Communications*, June 1985.
- [24] M. J. Sabin and R. M. Gray, "Product Code Vector Quantizers for Speech Waveform Coding," *Globecom*, Dec. 1982.
- [25] M. J. Sabin and R. M. Gray, "Product Code Vector Quantizers for Waveform and Voice Coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, June 1984.
- [26] I. M. Gerson and M. A. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP)," *IEEE Workshop on Speech Coding For Telecommunications*, 1989.
- [27] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 Kbps," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 461–464, 1990.
- [28] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 4.8 Kbps," *International Mobile Satellite Conference*,

- pp. 678–683, 1990.
- [29] B. H. Juang and A. H. Gray, “Multiple Stage Vector Quantization for Speech Coding,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1982.
 - [30] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, “Improved Speech Quality and Efficient Vector Quantization in SELP,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 155–158, April 1988.
 - [31] J. Campbell, V. Welch, and T. Tremain, “An Expandable Error Protected 4800 bps CELP Coder,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 735–738, 1989.
 - [32] G. Davidson and A. Gersho, “Complexity Reduction Methods for Vector Excitation Coding,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.
 - [33] G. Davidson, M. Yong, and A. Gersho, “Real-Time Vector Excitation Coding of Speech at 4800 bps,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1987.
 - [34] J. Makhoul, “Linear Prediction: A Tutorial Review,” *Proceedings of the IEEE*, vol. 63, pp. 561–580, Apr. 1975.
 - [35] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. New York, New York: Springer-Verlag, 1976.
 - [36] N. Levinson, “The Weiner RMS Error Criterion in Filter Design and Prediction,” *J. Math. Phys.*, vol. 25, pp. 261–278, 1946.
 - [37] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

- [38] B. S. Atal, R. V. Cox, and P. Kroon, "Spectral Quantization and Interpolation For CELP Coders," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 69–72, 1989.
- [39] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 661–664, 1991.
- [40] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1985.
- [41] S. Wang and A. Gersho, "Improved Excitation For Phonetically-Segmented VXC Speech Coding Below 4 kbps," *Globecom*, pp. 946–950, 1990.
- [42] W. Granzow and B. Atal, "High-Quality Digital Speech at 4 kbps," *Globecom*, pp. 941–945, 1990.
- [43] I. Gerson and M. Jasiuk, "Techniques For Improving the Performance of CELP-type Speech Coders," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 205–208, May 1991.
- [44] P. Kabal, J. Moncet, and C. Chu, "Synthesis Filter Optimization and Coding: Applications to CELP," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 147–150, 1988.
- [45] J. P. Adoul, P. Mabillean, M. Delprat, and S. Morissette, "Fast CELP Coding Based on Algebraic Codes," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1987.
- [46] J. P. Adoul and C. Lamblin, "A Comparison of some Algebraic Structures for CELP Coding of Speech," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1987.

- [47] T. E. Tremain, J. P. Campbell, and V. C. Welch, "A 4.8 kb/s Code Excited Linear Predictive Coder," *International Mobile Satellite Conference*, pp. 491–496, 1988.
- [48] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Fast Methods for the CELP Speech Coding ALgorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 155–158, August 1990.
- [49] M. Yong and A. Gersho, "Efficient Encoding of the Long-term Predictor in Vector Excitation Coders," *IEEE Workshop on Speech Coding For Telecommunications*, 1989.
- [50] I. M. Trancoso and B. S. Atal, "Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.
- [51] "Cellular Systems Dual-Mode Subscriber Equipment- Network Equipment Compatibility Specification," Electronic Industries Association, Dec. 1988.
- [52] G. Davidson and A. Gersho, "Multiple-Stage Vector Excitation Coding Of Speech Waveforms," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 163–166, April 1988.
- [53] T. Miyano, M. Serizawa, J. Takizawa, S. Ikeda, and K. Ozawa, "Improved 4.8 kb/s CELP Coding Using Two-Stage Vector Quantization With Multiple Candidates (LCELP)," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1992.
- [54] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, "Equations of State Calculations for Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, June 1953.
- [55] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, May 1983.

- [56] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, November 1984.
- [57] P. Kroon, R. J. Sluyter, , and E. F. Deprettere, "A Low Complexity Regular Pulse Coding Scheme With a Reduced Transmission Delay," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.
- [58] N. Phamdo, N. Favardin, and T. Moriya, "Combined Source-Channel Coding of LSP Parameters Using Multi-Stage Vector Quantization," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 36–38, 1991.
- [59] J. B. Anderson and J. B. Bodie, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. IT-21, pp. 379–387, July 1975.
- [60] G. S. Kang and L. J. Fransen, "Low-Bit-Rate Speech Encoders Based on Line Spectrum Frequencies (LSFs)," *Naval Research Laboratory Report No. 8857*, Nov. 1984.
- [61] F. F. Tzeng, "Analysis-By-Synthesis Linear Predictive Speech Coding at 2.4 kbit/s," *Proc. Globecom 89*, pp. 1253–1257, 1989.
- [62] Y. Shoham, "Cascaded Likelihood Vector Coding of the LPC Information," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 160–163, 1989.
- [63] W.-Y. Chan, "The Design of Generalized Product-Code Vector Quantizers," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 389–392, 1992.
- [64] W.-Y. Chan, S. Gupta, and A. Gersho, "Enhanced Multistage Vector Quantization by Joint Codebook Design," *to appear in IEEE Trans. on Communications*, 1992.

- [65] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic, 1992.
- [66] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-27, pp. 247-254, June 1979.
- [67] K. Zeger and A. Gersho, "Pseudo-Gray Coding," *IEEE Transactions on Communications*, vol. COM-38, pp. 2147-2158, Dec. 1990.
- [68] K. Paliwal and B. Atal, "Vector Quantization of LPC Parameters in the Presence of Channel Errors," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 33-35, Sept. 1991.
- [69] "Telecommunications: Analog To Digital Conversion of Radio Voice by 4,800 Bit/Second Code Excited Linear Prediction (FS-1016)," National Communications System, Office of Technology and Standards, Feb. 1991.
- [70] N. Farvardin, "A Study of Vector Quantization for Noisy Channels," *IEEE Transactions on Information Theory*, vol. IT-36, pp. 799-809, July 1990.
- [71] P. Lupini and V. Cuperman, "personal communication," March 1992.
- [72] "Subjective Performance Assessment of Digital Processes using the Modulated Noise Reference Unit (MNRU)," CCITT Supplement No. 14 to Recommendation P.81, Blue Book, Volume V, 1988.
- [73] C. Laflamme, J. P. Adoul, H. Y. Su, and S. Morissette, "On Reducing Computational Complexity of Codebook Search in CELP Coder Through the Use of Algebraic Codes," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1990.
- [74] R. Salami and D. Appleby, "A New Approach to Low Bitrate Speech Coding With Low Complexity Using Binary Pulse Excitation (BPE)," *IEEE Workshop on Speech Coding For Telecommunications*, 1989.

- [75] J. J. Shynk, "Adaptive IIR Filtering," *IEEE ASSP Magazine*, pp. 4–21, April 1989.
- [76] D. Parikh, N. Ahmed, and S. D. Stearns, "An Adaptive Lattice Algorithm for Recursive Filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 110–111, February 1980.
- [77] CCITT Recommendation G.721, "32 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," *Study Group XVIII- Report R 26(C)*, Aug. 1986.
- [78] N. S. Jayant and P. Noll, *Digital Coding of Waveforms- With Applications to Speech and Video*. Engle-Wood Cliffs, New Jersey: Prentice-Hall, 1984.
- [79] L. A. Thorpe and P. V. Cloverdale, "Evaluation of Digital Cellular Speech Coders: Listening Test Results," *Prepared for Telecommunications Industries Association (by Bell Northern Research), Committee TR-45.3, Working Group 3, Task Force 2, BNR Case number TR-4184*.
- [80] N. Farvardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Transactions of Information Theory*, vol. 37, pp. 155–160, Jan. 1991.
- [81] S. Wang, A. Sekey, and A. Gersho, "Auditory Distortion Measures for Speech Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 493–496, May 1991.
- [82] N. Lu and D. Morrell, "VQ Codebook Design Using Improved Simulated Annealing Algorithms," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 673–676, May 1991.
- [83] P. Kroon and B. S. Atal, "Pitch Predictors with High Temporal Resolution," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 661–664, 1990.

- [84] J. Marques, I. Trancoso, J. Tribolet, and L. Almeida, "Improved Pitch Prediction with Fractional Delays in CELP Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 665–668, 1990.
- [85] M. Yong, G. Davidson, and A. Gersho, "Encoding of LPC Spectral Parameters Using Switched-Adaptive Interframe Vector Prediction," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 402–405, April 1988.
- [86] W. B. Kleijn, "Continuous Representations in Linear Predictive Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 201–204, May 1991.
- [87] W. Granzow, B. S. Atal, K. K. Paliwal, and J. Schroeter, "Speech Coding at 4 kb/s and Lower Using Single-Pulse and Stochastic Models of LPC Excitation," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 217–220, May 1991.
- [88] J. Chen, "High-Quality 16 kb/s Speech Coding With A One-Way Delay Less Than 2 ms," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 453–456, 1990.
- [89] F. Tzeng, "An Analysis-by-Synthesis Linear Predictive Model For Narrow Band Speech Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 209–213, 1990.
- [90] T. D. Lookabaugh and R. M. Gray, "High-Resolution Quantization Theory and the Vector Quantizer Advantage," *I.E.E.E. Transactions on Information Theory*, vol. 35, pp. 1020–1033, Sept. 1989.
- [91] T. Parsons, *Voice and Speech Processing*. New York, New York: McGraw Hill Book Co., 1986.

- [92] F. Tzeng, "Near Optimal Linear Predictive Speech Coding," *Globecom*, pp. 962–966, 1990.
- [93] D. Wong, B. Juang, and A. Gray, "An 800 bit/s Vector Quantization LPC Vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, pp. 770–780, October 1982.
- [94] S. Roucos, R. Schwartz, and J. Makhoul, "A Segment Vocoder at 150 b/s," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983.
- [95] D. Wong and B. Juang, "Voice Coding at 800 bps and Lower Data Rates With LPC Vector Quantization," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1982.
- [96] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech*. Engle-Wood Cliffs, New Jersey: Prentice-Hall, 1978.
- [97] N. S. Jayant, "High Quality Coding of Telephone Speech and Wideband Audio," *IEEE Communications Magazine*, pp. 10–20, 1990.
- [98] S. Yeldener, A. M. Kondo, and B. G. Evans, "A Variable Rate Speech Compressor for Mobile Applications," *International Mobile Satellite Conference*, pp. 690–694, 1990.

END

1 6 - 1 2 - 9 3

FIN