

**PITCH-SYNCHRONOUS PROCESSING OF SPEECH
SIGNAL FOR IMPROVING THE QUALITY OF LOW BIT
RATE SPEECH CODERS**

A Thesis
Presented to
The Academic Faculty

by

Ali Erdem Ertan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2003

Copyright © 2003 by Ali Erdem Ertan

PITCH-SYNCHRONOUS PROCESSING OF SPEECH
SIGNAL FOR IMPROVING THE QUALITY OF LOW BIT
RATE SPEECH CODERS


Approved by:

 Thomas P. Barnwell III, Advisor

Dr. Randal T. Abler

 Dr. Mark A. Clements

 Dr. Greg Turk

 Dr. Biing-Hwang (Fred) Juang

Date Approved 12/03/2003

To my family, for their endless love,
to Fatma Çalışkan, for her patience and support,
and to Dr. Thomas P. Barnwell III, for helping me to make my dreams come true

ACKNOWLEDGEMENTS

Everything that has a beginning has an end....

Time really passes very quickly. It has been five years since I first started my Ph.D. study in the Center of Signal and Image Processing (CSIP) group, and now it has come to an end. During these five long years, I have had the opportunity of meeting many brilliant and kind people. I would like to take this opportunity to express my gratitude to these people with whom I have great memories.

I am most grateful to my advisor, Dr. Thomas P. Barnwell III. In all these years, he not only has been an excellent advisor who guided me throughout my thesis, but also has been a father-figure for me by showing the greatest patient and support. He has taught me to find my own path in my research and to present my findings in the most accurate and efficient way. I have always seen the reflections of his dedication to his work and his kindness on all his former graduate students with whom I worked, and I always wish that anybody whom I will meet in the future will see the same reflections on me. It is a great honor and pleasure for me to work with Dr. Thomas P. Barnwell III.

I would like to express my appreciation to Dr. Mark A. Clements for his valuable comments about my thesis, and for his intelligent humor that makes all our group meetings enjoyable. I also would like to express my special thanks to Dr. Biing-Hwang Juang for his suggestions on my thesis and for serving on my thesis reading committee. I would like to thank Dr. Randal T. Abler for serving in my thesis committee and Dr. Greg Turk for not only serving in my thesis committee and but also for giving me the opportunity to learn the fantastic world of the 3-D computer graphics that is one of my long-time hobbies.

I wish to thank our professors in CSIP who always inspire us and also give us the opportunity to work in this wonderful environment.

I feel that I owe a sincere thanks to Dr. Vishu Viswanathan from Texas Instruments for

giving me the chance to work in my dream job. My time in Texas Instruments as an intern was a great experience and pleasure for me, and I am looking forward to join them again. I especially would like to express my appreciation to Dr. Alan V. McCree for guiding me in my internship and then for giving valuable suggestion that also helped shaping this work. I also want to thank Texas Instruments for supporting me in all these years.

I would like to thank all the brilliant and kind people of the CSIP group; especially Jon Arrowood, Volkan Cevher, Macid Fozunbal, John Glotzbach, Bahadır Güntürk, Sangkeun Lee, Jenfang Samuel Li, Elliott Moore II, Robert Morris, Raviv Ranch, Sourabh Ravindran, Gail Rosen and Phil Spencer Whitehead. I especially want to express my appreciation to Soner Özgür for his great friendship, to Xin Zhong for his endless discussions about life, and to Sunil Shukla for long discussions about the work in this thesis and his nice friendship. I especially wish to thank Venkatesh Krishnan for proof-reading my thesis and for bugging me to turn this document into a much better one, and Robert Morris for being a good friend and allowing me to use his software for listening tests.

I am grateful to the staff who provided us administrative support and helped us to solve our problems that we have always left to the very-last minute, especially Marilou Mycko, Christy Ellis, Stacy Shultz, Kay Gilstarp and Charlotte Doughty. I would like to thank to Keith May and Sam Smith who solved all our computer and network related problems.

I also would like to thank all my close friends in Atlanta who always remind me that there is a life outside my cubicle, especially Çağatay Candan, Ümit Batur and Cenk Argon. I am especially grateful to my friends, Cem Baydar, Muzaffer Büyükkaragöz and Yılmaz Acar for their continuous morale support and for being there for me like one of my family.

I wish to thank to Fatma Çalışkan for her patience, support and kindness in these five long years. This thesis cannot happen without her help.

Finally, I would like to express my appreciation to and my love for my family; my mother and my father, Rahmiye and Mevlüt Ertan, my brother - my joy of life - İ. Emre Ertan, my grandmother and my grandfather, Remziye and Hilmi Apaydın, and all my aunts, uncles and their families. I am here because of their support and encouragement, and I hope I make them proud.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	xi
SUMMARY	xviii
I INTRODUCTION	1
1.1 Problem Statement	1
1.2 Contribution of the Thesis	4
1.3 Organization of the Thesis	5
II BACKGROUND	6
2.1 Speech Coding Algorithms	6
2.1.1 Waveform-Matching Speech Coders	6
2.1.2 Parametric Speech Coders	7
2.1.3 Hybrid Coders	15
2.1.4 Parametric/Hybrid Speech Coders	19
2.2 Linear Prediction Methods for Speech Coders	20
2.2.1 Direct-Form All-Pole Filter Estimation Methods	21
2.2.2 Lattice-Form All-Pole Filter Estimation Methods	26
2.2.3 Frequency Domain All-Pole Filter Estimation Methods	30
2.2.4 Constraint Excitation All-Pole Filter Estimation Methods	34
III PITCH-PERIOD ESTIMATION AND PITCH-CYCLE SEGMENTATION	39
3.1 Pitch-Period Estimation	39
3.2 Pitch-Cycle Segmentation	49
3.2.1 Pitch-Cycle Segmentation Based on Prediction Gain Maximization	49
3.2.2 Pitch-Cycle Segmentation Based on Normalized Correlation Maximization	53

IV	CIRCULAR LINEAR PREDICTION MODELING	58
4.1	CLP Analysis	59
4.2	CLP Synthesis	61
4.3	Performance Analysis of CLP Modeling	63
4.3.1	Test Setup	63
4.3.2	Performance Evaluation of CLP Analysis	64
4.3.3	Performance Evaluation of CLP Synthesis	81
4.4	CLP Modeling with Fractional Cycle Length	86
4.5	Performance Improvements for Short Pitch-Cycles	92
4.5.1	Multicycle Circular Linear Prediction Method	93
4.5.2	Pulse Excited-Circular Linear Prediction Method	98
4.6	Linear Prediction of Real-Speech Signals Using the CLP Analysis	115
V	CONSTANT PITCH TRANSFORMATION	130
VI	PITCH-SYNCHRONOUS METHODS FOR SPEECH CODING	138
6.1	The 2.4 kb/s U.S. Military Standard MELP Coder	140
6.1.1	The Encoder	142
6.1.2	The Decoder	146
6.2	The New 2.4 kb/s Improved-MELP coder	148
6.2.1	The Encoder	149
6.2.2	The Decoder	152
6.3	Parametric/Hybrid Coding of Speech Signal Using the new 2.4 kb/s Improved-MELP coder	154
6.3.1	Pitch-Cycle Modification of Speech Signal for Designing a Parametric/Hybrid MELP Coder	155
6.3.2	An Experimental Parametric/Hybrid I-MELP/PCM Coder	161
6.3.3	An Experimental Parametric/Hybrid I-MELP/MP Coder	164
6.4	Subjective Evaluation of the Proposed Coders	168
6.4.1	Description of the Tests	168
6.4.2	Test Setup	169
6.4.3	Initial Test Results	171
6.4.4	Statistical Analysis Method for Evaluating the Test Results	172

6.4.5 Interpretation of the Test Results	175
VII CONCLUSION	185
7.1 Future Work	188
APPENDIX A — PITCH-PERIOD ESTIMATION ALGORITHM AP- PENDICES	191
APPENDIX B — LEVINSON-DURBIN RECURSION	197
APPENDIX C — EQUIVALENCE OF THE CLP ANALYSIS TO COM- MON LINEAR-PREDICTION ESTIMATION ALGORITHMS	201
APPENDIX D — C++ RESEARCH AND DEVELOPMENT ENVIRON- MENT FOR SPEECH CODING APPLICATIONS	213
REFERENCES	221
VITA	227

LIST OF TABLES

Table 1	The MOS scores for different CW sampling rates in the WI model.	15
Table 2	Correlation groups according to normalized correlation coefficient of the primary pitch candidate.	48
Table 3	Formant locations and bandwidths used in the synthetic speech experiments.	64
Table 4	The percentage of unstable filters obtained by the variations of the SPE-CLP method for the ten speakers in the evaluation set.	127
Table 5	The computational complexity of various CPT-ICPT implementations. .	136
Table 6	Bit-allocation table for the 2.4 kb/s U.S. military standard MELP coder.	146
Table 7	The quantization levels of the normalized amplitude of the pulses	167
Table 8	The test cases that are evaluated in the subjective listening tests	170
Table 9	Rating scale for the degradation category rating (DCR) test.	170
Table 10	Rating scale for the comparison category rating (CCR) test.	170
Table 11	The number of phrases spoken by male and female speakers used in the DCR test.	173
Table 12	The average DMOS scores for the DCR test.	173
Table 13	The average CMOS scores for the CCR test.	173
Table 14	Comparison of the quality of the 2.4 kb/s speech coders for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	177
Table 15	Comparison of the quality of the 2.4 kb/s speech coders for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	177
Table 16	Comparison of the quality of the 2.4 kb/s speech coders for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	177
Table 17	Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	180
Table 18	Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	180

Table 19	Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	180
Table 20	Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	182
Table 21	Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	182
Table 22	Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	182
Table 23	Comparison of the various test cases for combined male and female speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	184
Table 24	Comparison of the various test cases for male speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	184
Table 25	Comparison of the various test cases for female speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.	184

LIST OF FIGURES

Figure 1	Excitation+filter model for speech synthesis.	10
Figure 2	Example for mixed excitation spectrum.	13
Figure 3	(a) Single-stage FIR lattice filter and its basic block representation, and (b) single-stage IIR lattice filter and its basic block representation	27
Figure 4	(a) The FIR lattice filter and (b) the IIR lattice filter	28
Figure 5	Repetition of the correctly segmented (a) and arbitrary segmented (b) signal. Prediction gains, P_g , for the segments shown in (a) and (b) are 7.343 dB and 5.522 dB, respectively.	50
Figure 6	Illustration of the problem in pitch-cycle segmentation when signal energy is increasing. Because of the changing energy, the prediction gain is not maximum in the correct end location, marked with the dashed line. . . .	52
Figure 7	The speech spectra with uniformly spread formants (a) and with grouped formants (b).	65
Figure 8	The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the auto-correlation methods for pitch-cycle lengths between 20 and 160 samples. The speech spectrum is the “uniformly spread formants” example.	67
Figure 9	The true spectrum with uniformly spread formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch-period is 60 samples (a), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples (b).	67
Figure 10	The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 40 samples. The speech spectrum is the “grouped formants” example.	69
Figure 11	The true spectrum with grouped formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch-period is 25 samples (a), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 25 samples (b).	69
Figure 12	The speech spectrum used in the performance tests for synthetic unvoiced speech.	70
Figure 13	The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples (solid line) and the autocorrelation method using 200 samples (dash-dotted line) and using 100 samples (dashed line).	71

Figure 14	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for partially-voiced speech signal. The transition frequencies of the spectrum are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively. The speech spectrum is the “uniformly spread formants” example.	73
Figure 15	The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced and partially-voiced speech signals. The speech spectrum is the “uniformly spread formants” example.	74
Figure 16	The true speech spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the transition frequencies are 3000 Hz(a), 2000 Hz(c) and 1000 Hz(e), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the transition frequencies are 3000 Hz(b), 2000 Hz(d) and 1000 Hz(f). The pitch period is 60 samples.	75
Figure 17	The absolute spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method when the analyzed signals are fully-voiced and various partially-voiced synthetic speech signals and the pitch period is 60 samples. The speech spectrum is the “uniformly spread formants” example.	76
Figure 18	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for noisy speech signal. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f), respectively.	78
Figure 19	The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced speech and various noisy speech signals.	79
Figure 20	The true speech spectrum with uniformly spread formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(a), 20 dB(b) and 10 dB(c), and the error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(d), 20 dB(e) and 10 dB(f).	80
Figure 21	A 200 sample stationary real-speech segment (a), and the estimated spectra obtained by the autocorrelation method using 200 samples and the CLP method using 42 and 43 samples (b).	81

Figure 22	The reconstruction SNR for the speech spectrum with uniformly spread formants for the pitch-cycle range between 20 and 160 samples (a) and for the speech spectrum with grouped formants for the pitch-cycle range between 20 and 30 samples (b). The maximum number of iterations in this figure is 25. The SNR is increased by each iteration.	84
Figure 23	The number of iterations required to achieve an average 72 dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the first synthesis method without pre/de-emphasis filter (a) and with pre/de-emphasis filter (b).	84
Figure 24	The number of iterations required to achieve an average 72-dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the second synthesis method. The difference between the formant bandwidths and locations of the original and modified configurations in successive pitch cycles is 0.5% in (a) and 5% in (b).	85
Figure 25	The length of the filter in multiples of τ required to achieve an average 72 dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the third synthesis method.	87
Figure 26	The average reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the thresholds, 0.01, 0.0025 and 0.0001, for the second length-calculation method of the third synthesis method.	87
Figure 27	The true speech spectrum and the estimated spectra obtained by the CLP method using 60 and 61 samples (a); and the absolute error between the true spectrum and the estimated spectra (b). The cycle length is 60.3 samples.	88
Figure 28	The true speech spectrum and the estimated spectra obtained by CLP analysis using 60, 61 and 60.3 samples (a); and the absolute error between the true spectrum and the estimated spectra (b). The cycle length is 60.3 samples.	89
Figure 29	The speech spectra obtained by the autocorrelation method using 200 samples and the CLP method using 42, 43 and 42.7 samples (a), and the absolute spectral difference between the spectrum estimated by the autocorrelation method and the spectra estimated by the CLP method using 42, 43 and 42.7 samples (b).	90
Figure 30	The sample resolution that results into an average 0.5 dB spectral mismatch for pitch-cycle lengths between 20 and 160 samples in 0.5 sample steps.	92
Figure 31	The true spectrum and the estimated spectra obtained by CLP analysis using single pitch cycle ($\tau_0=30$ samples), M-CLP analysis using two pitch cycles ($\tau_{0,1}=30,31$ samples) and M-CLP analysis using three pitch cycles ($\tau_{0,1,2}=30,31,32$ samples) (a); and the absolute error between the true spectrum and the estimated spectra (b).	94

Figure 32	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the M-CLP method with two and three cycles, and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for partially-voiced speech signal. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.	97
Figure 33	The mean (a,c,e) and the standard deviation (b,d,f) of the spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the M-CLP method with two and three cycles and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for noisy speech signal. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.	99
Figure 34	The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the SPE-CLP method and the autocorrelation method for pitch-cycle lengths between 20 and 160 samples. The speech spectrum is the “uniformly spread formants” example.	102
Figure 35	The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the SPE-CLP method and the autocorrelation method for pitch-cycle lengths between 20 and 40 samples. The speech spectrum is the “grouped formants” example.	102
Figure 36	The true spectrum with grouped formants and a first formant with narrow bandwidth and the estimated spectra obtained by the SPE-CLP method and the autocorrelation method when the pitch period is 25 samples (a), and the error between the true spectrum and the estimated spectra by the SPE-CLP and the autocorrelation methods when the pitch period is 25 samples (b).	103
Figure 37	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for various partially-voiced speech signals. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.	104
Figure 38	The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced and various partially-voiced speech signals.	105

Figure 39	The speech spectrum with uniformly spread formants and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods when the pitch period is 60 samples and the transition frequencies are 3000 Hz(a), 2000 Hz(c) and 1000 Hz(e), and the absolute error between the true spectrum and the estimated spectra when the pitch period is 60 samples and the transition frequencies are 3000 Hz(b), 2000 Hz(d) and 1000 Hz(f).	106
Figure 40	The average (a,c,e) and the maximum (b,d,f) spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for noisy speech signals. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.	108
Figure 41	The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method for analysis frame lengths between 20 and 160 samples when used on clean speech and various noisy speech signals.	109
Figure 42	The true speech spectrum with uniformly spread formants and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(a), 20 dB(b) and 10 dB(c), and the absolute error between the true spectrum and the estimated spectra when the pitch period is 60 samples and the SNR is 30 dB(d), 20 dB(e) and 10 dB(f).	110
Figure 43	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method, the M-SPE-CLP method with two and three cycles, and the autocorrelation methods for pitch-cycle lengths between 20 and 60 samples for various partially-voiced speech signals. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.	113
Figure 44	The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method, the M-SPE-CLP method with two and three cycles and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for various noisy speech signals. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.	114
Figure 45	A real-speech signal that illustrates an onset (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for the first (between the vertical lines in (a)) (b), the second (c) and the third (d) pitch cycles. The lengths of the first, the second and the third pitch cycles are 38.4, 47.7 and 49.6 samples, respectively.	120
Figure 46	A stationary real-speech segment (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) (b). The length of the pitch cycle is 51 samples.	121

Figure 47	The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for displayed speech segment.	122
Figure 48	A stationary real-speech segment (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) (b). The length of the pitch cycle is 22.8 samples.	122
Figure 49	The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for the displayed speech segment. . .	123
Figure 50	The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for a partially-voiced speech segment.	123
Figure 51	The LSF tracks obtained by the autocorrelation method (solid line) and the three-cycle M-CLP method (dash-dotted line) for the partially-voiced speech segment displayed in Figure 50.	124
Figure 52	A stationary real-speech segment for a male speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 51 samples.	125
Figure 53	A stationary real-speech segment for a female speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 23.4 samples.	125
Figure 54	The residual signals obtained by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) for a speech segment. . . .	126
Figure 55	The LSF tracks obtained by the autocorrelation method (solid line) and the three-cycle M-SPE-CLP method (dash-dotted line) for a high-pitched speaker (average pitch-period is 22 samples.)	128
Figure 56	A stationary real-speech segment for a female speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the three-cycle M-SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 22.2 samples.	128
Figure 57	Block diagram of constant pitch transformation.	131
Figure 58	(a) A real speech-signal segment, (b) the CPT of the single cycle between the dashed lines in (a), (c) the DFT of the single cycle between the dashed lines in (a), and (d) the DFT of the CPT of the single cycle between the dashed lines in (a). The length of the original cycle is 90 samples and the length of the CP transformed cycle is 160 samples.	132
Figure 59	The L_s (a) and K_s (b) for pitch-cycle lengths between 20 and 128 required to achieve 72 dB average reconstruction SNR.	136
Figure 60	Block diagram of the inverse constant pitch transformation.	136
Figure 61	Block diagram of the analysis system.	139
Figure 62	Block diagram of the MELP model.	141
Figure 63	The decoder of the new 2.4 kb/s improved MELP coder.	152

Figure 64	Mapping of the original pitch cycle to the synthesis pitch cycle when the synthesis pitch cycle overlaps with the original pitch cycle completely (a), when the significant part of the synthesis pitch cycle overlaps with the current pitch cycle (b), when the significant part of the synthesis pitch cycle overlaps with the next original pitch cycle (c). The left-to-right diagonal lines in (b) and (c) illustrate the overlapping segments of the selected original pitch cycle and the synthesis pitch cycle, and the right-to-left diagonal lines in (b) and (c) illustrate the overlapping segments of the the next original cycle and the synthesis cycle.	157
Figure 65	Flowchart of the mapping algorithm.	158
Figure 66	The input speech segment (top), the same segment processed by the pitch-cycle modification algorithm (middle) and the same segment encoded by the 2.4 kb/s improved MELP coder (bottom).	161
Figure 67	Flowchart of the pitch-cycle modification algorithm.	162
Figure 68	The bit rate variation (bottom) in one of the sentence in the evaluation set (top).	168
Figure 69	The system overview.	215
Figure 70	Implementation illustration of LSF and its extraction and quantization methods.	216
Figure 71	Memory management for three subframe analysis.	217
Figure 72	Snapshot of the visualization tool.	220

SUMMARY

Recent advances in low bit rate speech coding have resulted in a family of speech coders with very good quality and high intelligibility in the past decade. However, these state-of-the-art coders still have problems in modeling and encoding of the transition regions in the speech signal. This problem not only affects the overall quality of these coders but also prohibits further quality improvements with increasing bit rate. One of the reasons for this failure is the stationary assumption used in the estimation of the model parameters in these regions. For these cases, the parameter estimates are often flawed, and the use of these parameters in synthesis may result in audible artifacts.

This thesis introduces new methods that can capture the spectral characteristics efficiently from the shortest possible speech signal segments - individual pitch cycles. As a result, it is possible to capture the perceptually important information in both stationary and transition regions. For this purpose, this thesis presents a new class of linear-prediction methods and a residual signal representation method that both use single pitch cycle. A new 2.4 kb/s speech coder is also developed using these proposed methods. Listening tests proved that this new coder performs better than the current state-of-the-art speech coder, especially for female speech.

In addition, the quality of the current parametric speech coders is improved by encoding the waveform of the excitation signal in transition regions. For this purpose, a new algorithm that modifies the original signal such that it becomes time-synchronous with the synthetic signal and the waveform of both signals become similar is introduced. This new algorithm allows the use of both fully-parametric representation and waveform encoding of the excitation signal in the same coder to encode different parts of the speech signal. Listening tests have proved that the speech coders using this method have better synthetic speech quality.

CHAPTER I

INTRODUCTION

1.1 Problem Statement

Speech is the easiest and most efficient way for us to express our thoughts, feelings and needs to others. For this reason, one of the most important inventions of all times is the telephone, which allows people to communicate with others at distant places using speech. In the last fifty years, digital communication systems are rapidly replacing the analog communication systems. This change also requires the use of digitized speech. However, a good-quality linear PCM speech signal can only be represented with 128 kbit/s, and that requires a transmission bandwidth that is not practical with today's standards for wireless personal communication systems. In addition, military applications demand transmission of speech signals even at bit rates below 3 kb/s. For these reasons, both commercial and military applications require the representation of the digitized speech signal in a compressed form. The research area that deals with this problem is called speech coding, and the algorithms that compress the speech signal is known as speech coders.

A speech coder basically has three elements: the speech model, the parametric analysis and the parameter coding. The speech models define a set of rules that specifies how the perceptually important characteristics of the speech signal are captured by a set of parameters and how a speech signal is synthesized using these parameters. In a coding application, a speech model must capture the characteristics of as many different speech sounds as possible. If the model cannot capture important characteristics of a particular sound, the synthesized signal may sound artificial or may not be even perceived as speech. However, as computational complexity is always another major design factor in a speech coding algorithm, the model must also be mathematically tractable and allow the real-time implementation of the coding system. In addition, the model also should not be very sensitive to the errors in the quantization of the parameters. The second important element

of a speech coder is the parameter analysis. The analysis algorithms estimate the model parameters from the input speech signal. Even with the best model, if the parameters are not estimated correctly, the synthetic speech signal may have audible artifacts. In addition, the computational complexity is also a major concern for analysis algorithms. Finally, the last element of a speech coder is the parameter coding that deals with the quantization of the parameters required for the digital communication systems. The primary goal of these algorithms is to quantize the parameters such that the synthesized speech obtained by the quantized parameters sounds perceptually identical to the one obtained by the unquantized parameters.

Various design choices, such as the communication channel capacity (bit rate), the delay and the computational complexity, limit the capabilities of these three elements. As an example, a model may require a set of parameters that can only be quantized above a certain bit rate. The synthetic speech quality from this model may be very high above this data rate, but may degrade rapidly when the bit rate is reduced. On the other hand, the synthesized speech quality obtained by another model may be moderate at high bit rates even when the parameters are quantized by the best algorithms. However, such a model may be still acceptable at ultra low bit rates. In addition to these examples, even when the synthesized speech quality of a model is very high, the analysis and quantization algorithms may be constrained by the computational complexity, and as a result, the model may not be used to its full potential.

Historically, early fully-parametric speech coders had simple models that were mathematically tractable and had very low computational complexity. Although the synthesized speech signals obtained by these coders were intelligible, they were far from natural. In the last decade, advances in the digital signal processors have allowed researchers to extend the existing models by adding a few more parameters so that it is now possible to synthesize a broader class of speech sounds resulting in more natural-sounding synthetic speech. Together with more complex analysis and quantization methods, these new coders can synthesize high-quality speech signal at and above 2.4 kb/s [79].

In spite of these advances, the best speech quality that can be obtained by state-of-the-art fully-parametric speech coders is still limited. Increasing the bit rate beyond a certain point does not improve the quality of these speech coders and the quality usually saturates around 4 kb/s, as the synthesized speech obtained by the quantized model parameters is as good as the one obtained by using the unquantized parameters. This observation means the source of the problem must be either the speech model or the analysis methods. One of the reasons for the quality saturation with increasing bit rate is the stationary assumption in transition regions. In a parametric coder, the model parameters are usually estimated on a frame-by-frame basis using a segment of speech in which the signal is assumed to be stationary. Although this assumption is often valid, it fails in segment transitions, stop consonants and for short events. For these cases, the parameter estimates are often flawed, and the interpolation of the resulting parameter values in the synthesis process results in audible artifacts. As a result, although the segmental quality of the synthesized speech is high, these isolated artifacts decrease the overall quality significantly. This is one of the main reasons why the quality of parametric speech coders does not increase linearly with the bit rate and saturates around 4 kb/s.

This thesis presents new methods for improving the quality of the current low bit-rate speech coders and that also allow the design of scalable “bit-rate vs. quality” speech coders. To achieve this goal, this thesis introduces new methods that can not only capture the spectral characteristics of the speech signal efficiently, but can also do it from the shortest possible speech signal segment. As a result, it is possible to accurately capture perceptually important information in both stationary and transition regions. The periodic nature of the voiced speech signal is a perfect match for such a model, and both analysis and synthesis can be performed on individual pitch cycles. In addition, a new pitch-cycle modification algorithm is developed that allows the synthetic speech signal to maintain time-synchrony with the original signal. As a result, a speech coder using this modification algorithm can make a seamless transition in the encoding of the excitation signal from a fully parametric representation to a waveform encoding technique.

1.2 Contribution of the Thesis

This thesis focuses mainly on processing of the individual pitch cycles of the speech signal in a pitch-synchronous fashion. The following new methods are introduced in this thesis:

- 1- An accurate pitch-period estimation algorithm based on finding the pitch track that minimizes the pitch-prediction residual energy in the speech and residual signals.
- 2- A new method for pitch-cycle boundary detection based on the maximization of the prediction gain.
- 3- A new method for pitch-cycle segmentation based on the maximization of the normalized correlation.
- 4- Re-introduction and in-depth analysis of the circular linear prediction method and its variations for modeling speech signals using individual pitch cycles with fractional cycle length.
- 5- Re-introduction of constant pitch transformation to generate an alternative representation of a single-cycle residual signal with fractional cycle length.
- 6- A 2.4 kb/s improved mixed excitation linear prediction (MELP) coder that uses the proposed techniques.
- 7- A pitch-synchronous speech modification algorithm that makes input signal time-synchronous with the synthetic speech signal obtained by the new 2.4 kb/s improved MELP coder and removes the phase information from the input signal such that the input signal waveform is similar to the synthetic speech signal waveform.
- 8- An experimental speech coder that encodes only the voiced speech segments with the new 2.4 kb/s improved MELP coder and transmits waveform of the residual signal of the unvoiced and transition segments.
- 9- An experimental speech coder that encodes the voiced speech segments with the new 2.4 kb/s improved MELP coder and encodes the unvoiced and transition segments with a variable-rate multi-pulse coder.

- 10- A C++ programming framework and a collection of tools that simplifies the implementation and testing of speech coding algorithms.

1.3 Organization of the Thesis

The thesis is organized as follows: Chapter 2 presents the background material on various speech coding algorithms and common linear-prediction methods. In Chapter 3, the detailed description of a new pitch-estimation algorithm and two pitch-cycle segmentation algorithms are presented. Chapter 4 focuses on the description and in-depth analysis of the circular linear prediction (CLP) modeling and its variations. The details of the constant pitch transformation (CPT) are given in Chapter 5. Integration of these proposed techniques into a 2.4 kb/s MELP coder, an experimental MELP/PCM coder and a variable rate MELP/MP coder is discussed in Chapter 6. Finally, the concluding remarks are given in Chapter 7.

Appendices at the end of this thesis present additional materials about the algorithms proposed in this thesis. In Appendix A, the proof of the equivalence of the pitch-prediction residual error minimization of a pitch track and the normalized correlation maximization of the same pitch track is given. In addition, details about the decision logic used in the new pitch-estimation algorithm are also presented. Appendix B presents the Levinson-Durbin recursion, a technique commonly used to solve the linear equations resulting from the autocorrelation and the CLP method. The equivalence of the CLP method and many common linear-prediction techniques when used to model an infinitely periodic signal is presented in Appendix C. Finally, the research and development environment used in the implementation of all of the algorithms described in this thesis is given in Appendix D.

CHAPTER II

BACKGROUND

This chapter presents background material for speech coding algorithms. Besides, an extensive summary of linear-prediction algorithms is also presented in this chapter.

2.1 Speech Coding Algorithms

Historically, speech-coding algorithms are roughly categorized into two groups: waveform-matching coders and parametric coders. Waveform-matching coders encode the original speech signal directly and the decoder generates an approximation of the original signal. On the other hand, parametric coders extract a number of parameters for a model that characterizes the speech signal so that a similar sounding speech signal can be synthesized using that model in the decoder [13]. In parametric coders, the waveform of the synthesized signal is usually quite different from the original signal. The waveform-matching coders can encode very high-quality speech at bit rates above 16kb/s [11], and the quality of the synthesized speech degrades rapidly below this data rate. State-of-the-art fully parametric coders such as MELP and WI coders can compress highly intelligible speech signal at bit rates less than 2.4 kb/s with good quality [56, 57, 37, 81]. In the 80s', a third speech-coding method that combines the ideas from the other two methods emerged. Often known as hybrid coders, these coders extract a set of parameters and encode a waveform to be used in conjunction with these parameters in the decoder. As a result, these hybrid coders such as CELP and MP-LP coders encode speech signal efficiently at bit rates between 4.8 and 16 kb/s [2, 42, 3].

2.1.1 Waveform-Matching Speech Coders

Waveform-matching coders are also known as nonparametric coders. These coders make relatively few assumptions based on the nature of the speech signal. Therefore, these coders

can encode all speech sounds, maintain the time synchrony between the original and synthesized signals, and are robust to background noise. The computational complexity and algorithmic delay of such coders are also relatively low. The main drawback of these coders is the high bit rate requirement. Typical nonparametric coders operate at bit rates between 16 and 32 kb/s, and the quality decreases rapidly below this rate [13]. These algorithms were studied extensively in the 70's and 80's. Many such algorithms were presented in the form of waveform coding [30], subband coding [13] and transform coding [85] of the speech signal. Pulse code modulation (PCM) [13], adaptive delta modulation (ADM) [30] and a form of adaptive differential pulse code modulation (ADPCM) [8] are good examples of waveform-coders. As current coders achieve similar quality as a 32 kb/s ADPCM coder at much lower bit-rates, speech coding with waveform-matching techniques has not been an active research area for more than two decades. However, these algorithms are still used in conjunction with sophisticated perceptual masking models to achieve transparent-quality coding of wide-band audio signals [60].

2.1.2 Parametric Speech Coders

Parametric coders assume that perceptually important characteristics of the speech signal can be captured with a set of parameters and these parameters can be used to synthesize the speech signal. The performance of these coders depends on the accuracy of the speech model and efficiency of quantization of the model parameters. Furthermore, the characteristics of the coders such as noise robustness and computational complexity are quite different from one model to another. In the literature, several parametric models such as channel vocoders [66] and formant vocoders [13] have been reported. However, because of their unique modeling properties, only sinusoidal coders [50] and linear prediction coders (LPC) [13] are the most well-studied and well-known models among all.

2.1.2.1 Sinusoidal Coders

The sinusoidal model assumes that the speech signal can be represented as a sum of sinusoidal components having arbitrary frequencies, amplitudes and phases such that

$$\hat{s}[n] = \sum_{l=1}^L \gamma_l e^{j\omega_l n + \phi_l} \quad (1)$$

where $\hat{s}[n]$ is the approximated speech signal, L is the number of sinusoidal components in the speech spectrum, and γ_l , ω_l and ϕ_l are the amplitude, frequency and phase of the l^{th} sinusoidal component. A speech coder based on this model basically extracts these parameters from the speech signal and transmits them to the decoder. The decoder then synthesizes the speech signal using (1). In the literature, two well-known coding methods that use this model, but with different assumptions have been reported: the sinusoidal transform coder (STC) introduced by McAulay and Quatieri [50] and the multi-band excitation (MBE) coding introduced by Griffin and Lim [20]. While the STC method strictly uses this sinusoidal model, the MBE method partitions the speech spectrum into non-overlapping bands with a bandwidth equal to the fundamental frequency and centered around the harmonics and makes a voiced/unvoiced decision for each band. For this reason, (1) is only used for voiced bands.

In practice, both coding methods assume that the speech signal is a periodic signal with a known pitch period. This assumption restricts the frequency locations of sinusoidal components to be at (or close to) the harmonics of the fundamental frequency of the signal. The STC algorithm obtains the magnitude and phase components from the peaks of the fast Fourier transform (FFT) of the signal around the harmonic locations, and the MBE algorithm obtains the magnitude of a band from the integration of the squared magnitude of the frequencies in the band. The MBE algorithm also makes a voiced/unvoiced decision based on whether the band has a harmonic structure or not. Although the sinusoidal model is most suitable for voiced speech, McAulay et al. [50] reported that it is also possible to capture the characteristics of unvoiced speech and transitional events when the speech spectrum is sampled at every 100 Hz in every 10 ms intervals.

The coders based on both the STC and the MBE algorithms can encode speech signal

with very high quality at and above 8.0 kb/s [50, 20]. The number of bits at these rates allows these coders to encode both magnitude and phase information accurately. However, below this data rate, other techniques have to be used to encode or represent the phase information [51, 52]. This is especially important for the STC algorithm, since incorrect phase information usually results in a reverberant quality. Furthermore, the alternative phase recovery techniques usually result in a buzzy speech quality that can be eliminated by randomizing the phase of the sinusoids in the high-frequency part of the speech spectrum [53]. Since the MBE algorithm already incorporates this feature, the quality of these coders is still very good at 4.8 kb/s [21]. At bit rates lower than 4 kb/s, accurate encoding of the magnitude is also a problem. In different studies, McAulay et al. [53], Yeldener et al. [84] and Das et al. [12] showed that an all-pole model whose frequency response fits to the envelope of the logarithm of the harmonic magnitudes can be used to model the magnitude of the spectrum. They reported that the speech coders using these techniques have a very good speech quality and high intelligibility at 2.4 kb/s. Furthermore, a 4.15 kb/s MBE coder, also known as improved MBE (IMBE), was selected by Inmarsat as the standard for satellite voice communications [9].

2.1.2.2 Linear Prediction Coders

The linear prediction model assumes that the speech signal is produced by filtering a source excitation with an all-pole filter. This idea stems from the fact that the upper vocal tract of the human speech-production system acts as an acoustical spectral shaping filter, and a simplified model of the vocal tract, a series of concatenated loss-less acoustical tubes, results in an all-pole filter. (This model does not take into account the nasal tract, coupling effects with the glottis, and internal losses in the vocal tract; hence, it is an oversimplified model.) The speech coders based on this model estimate the coefficients of this all-pole filter and a suitable excitation on a frame-by-frame basis, and then transmit this information to the decoder. Since all-pole filtering of a signal in the absence of an input signal is exactly the same as predicting the signal from the linear combination of the past samples, the term *linear prediction* is used for this model. The class of parametric coders based on this model

is also known as linear prediction coding (LPC) vocoders [13].

Parametric LPC coders not only use this excitation+filter model but also represent the excitation signal parametrically. This excitation modeling technique requires classification of the speech signals. Historically, the early parametric speech coders classify speech signals into two categories: voiced and unvoiced speech. In this model, it is assumed that voiced speech is obtained by exciting the vocal tract with glottal pulses, and unvoiced speech is generated by forcing air through a constriction in the vocal tract. As a result, it is possible to capture the characteristics of speech signal with the voicing information, the pitch-period (for voiced speech), the linear prediction coefficients and the gain of the signal. The synthesizer of this model is shown in Figure 1. In a typical LPC coder, these parameters are estimated in every 20-25 ms and transmitted to the decoder. The decoder then synthesizes the speech signal on a frame-by-frame basis. To obtain a smoothly evolving speech signal, the frame is usually divided into smaller subframes and the parameters used in the synthesis of these subframes are obtained from the interpolation of the estimated parameters. The autoregressive (AR) parameter estimation techniques are often used to find the linear-prediction coefficients. An extensive review of these techniques is presented in the next section. The pitch-period and voicing information is often estimated by techniques based on correlation of the speech signal or harmonic tracking [27].

Although the synthesized speech of LPC coders is highly intelligible, it is far from natural. The synthetic speech often sounds mechanical and tense, and it has a buzzy quality. Furthermore, the misclassification of the speech signal results in thumps for unvoiced speech

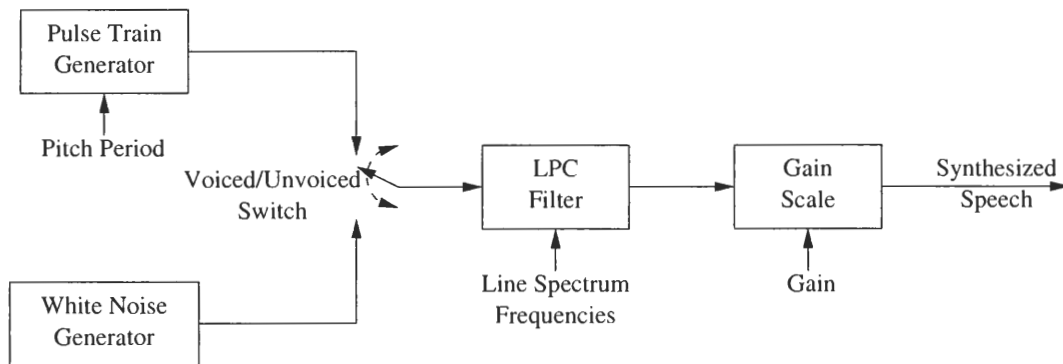


Figure 1: Excitation+filter model for speech synthesis.

and whispered quality for voiced speech signals. Because of its simplicity, the algorithm is not robust to background noise in harsh (military) environments. However, as the model has relatively few parameters, it is possible to encode this information at and below 2.4 kb/s, which makes it suitable for military communication. As a result, a 2.4 kb/s version of this coder, also known as LPC-10, was adopted as the Federal Standard in 1984 [76].

Many researchers tried to improve the quality of this model. Sambur et al. [65] reported that the buzzy quality of this model can be reduced somewhat by replacing the impulses in the impulse train with pulses whose frequency response is flat and whose energy is dispersed within the pitch cycle. However, the buzzy quality still cannot be completely eliminated with this method because of the strong periodicity among consecutive pitch cycles. In another experiment, Kang et al. [33] showed that this kind of strong periodicity is only observed for angry and tense speakers and the amount of periodicity is less in the high-frequency part of the spectrum for normal speech. As a result, they pointed out that the speech spectrum usually contains a harmonic structure at the low-frequency bands and a noisy structure in the high frequency bands. The transition frequency between those two regions usually depends on the mood of the speaker and the type of sound [33]. Kang also showed that adding a fixed amount of noise to the high-frequency part of the voiced excitation signal improves the quality of the Federal Standard LPC-10 coder [33]. In an earlier experiment, Makhoul et al. [46] described an LPC coder that uses the summation of a low-pass filtered impulse train and high-pass filtered white-noise sequence as the excitation signal. The cut-off frequency of the filters was selected such that no harmonic structure was seen in the speech spectrum after the cut-off frequency. They found that this technique decreases the buzzy quality although it also introduces noisiness in the speech. Recently, McCree et al. [56] introduced mixed excitation linear prediction (MELP) coding that partitions the speech spectrum into a number of non-overlapping bands and estimates the voicing strength of each band. The normalized correlation coefficient of the signal at the estimated pitch lag determines the voicing strength of the signal being analyzed. A band is declared voiced or unvoiced according to its voicing strength. McCree et al. [56] reported that a five-band model produces natural-sounding speech because of its ability to synthesize broader

class of speech sounds such as voiced fricatives (the spectrum of a voiced fricative sound is shown in Figure 2). This model also makes the speech coder robust to background noise [56]. McCree et al. [56] also reported another feature to improve the naturalness of the speech signal, particularly that of high-pitched speakers. This improvement was obtained by destroying the strong periodicity in the synthesized speech when erratic glottal pulses are observed in the residual signal. When the excitation signal has strong periodicity among successive pitch cycles in such cases, short isolated tones are audible in the synthesized speech. To correct this problem, each pitch-cycle length in those regions is jittered with a random number, uniformly distributed $\pm 25\%$ of the original cycle length, so that the new pitch-cycle length is equal to the average pitch-period plus the random number. When the correlation level of the low-pass filtered speech is close to the voicing threshold, aperiodic pulses are used in the excitation signal [56]. In addition to these enhancements, Unno et al. [78] described a detection algorithm and a synthesis method for stop consonants within a MELP coder. They reported that the addition of this new feature improves the clarity of the synthesized speech. A 2.4 kb/s version of the MELP algorithm that uses a five-band model and aperiodic pulses was adopted as the new U.S. military standard in 1996 [79]. Besides these enhancements, this MELP coder also encodes the first 10 harmonic magnitudes of the residual signal's Fourier series using a vector quantizer. The quality and intelligibility of this 2.4 kb/s coder exceeds the CELP-based Federal Standard, FS-1016, at 4.8 kb/s even in noisy environments. Recently, an improved version of this coder, which utilizes a better pitch-period estimator and a very high-quality noise suppressor, was also adopted as the new 2.4-1.2 kb/s NATO standard [80]. Finally, Stachurski et al. [74] reported that it is possible to achieve near-toll quality at 4 kb/s with the MELP model. This coder encodes the Fourier series magnitudes in the entire speech spectrum and updates the pitch and bandpass voicing information twice in a 20 ms frame. This coder uses improved techniques for parameter quantization as well.

Another widely used technique to encode the residual signal is called waveform interpolation (WI) introduced by Kleijn [37]. This technique uses the fact that pitch cycles in a speech signal evolve slowly in time. Hence, this method samples and transmits these

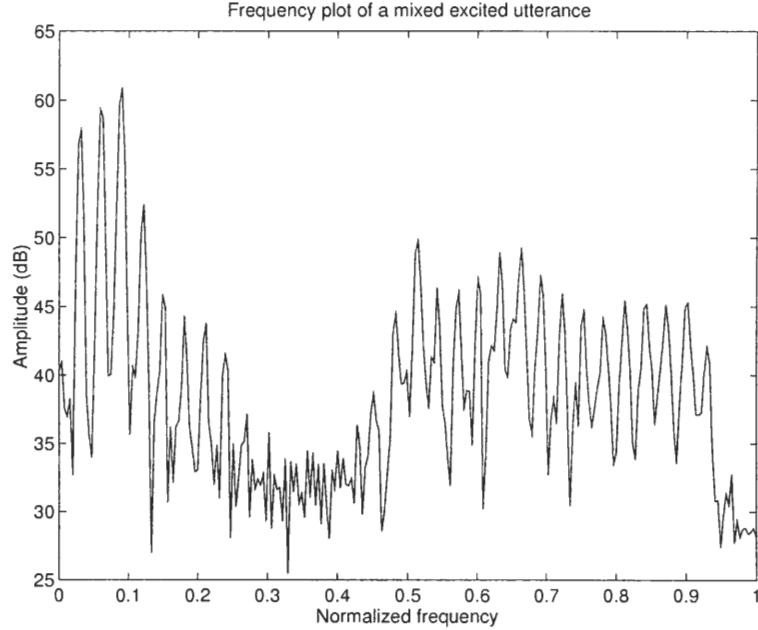


Figure 2: Example for mixed excitation spectrum.

waveforms only at regular time intervals and obtains the rest by interpolation. To do this operation, the WI method requires that a pitch-cycle waveform and a phase function must be known for each time instant. This leads to a two-dimensional surface representation, $u(t, \phi)$, where the pitch-cycle waveforms, normalized to 2π , are displayed along the ϕ axis and the evolution of these waveforms is displayed along the time axis, t . The waveforms displayed along the ϕ axis are also known as characteristic waveforms (CW). Since pitch-cycle length also changes in time, the phase function, ϕ , is also a function of time and defined as

$$\phi(t) = \phi(t_0) + \int_{t_0}^t \frac{2\pi}{p(t')} dt', \quad (2)$$

where $p(t)$ is the pitch period at time instant t . A signal with any waveform shape with a time varying pitch can be written as $u(t, \phi(t))$, and the one-dimensional signal can be recovered as

$$r(t) = u(t, \phi(t)). \quad (3)$$

This result also suggests that the CW surface is determined only on a particular trajectory, periodic with 2π along ϕ axis. The $\phi(t_0)$ becomes obsolete when even a slight error occurs in the transmission of $p(t)$. Therefore, the CW surface must be defined everywhere such that the quality of the reconstructed signal is invariant to any phase offset that causes the phase

function to deviate from the trajectory on which the CW surface is defined. To obtain the entire CW surface, Kleijn proposed two methods in [37]: In the first method, each sample for a particular ϕ is threaded equally spaced along the t axis, and band-limited interpolation is applied to the known samples to obtain the rest of the CW surface. This method is also known as continuous sampling. In the second method, it is assumed that the CW as a function of ϕ is known only at specific time instances. Therefore, the CWs are extracted by applying a pitch-cycle length rectangular window to the residual signal and normalizing its length to 2π . This method is known as discrete sampling. In almost all up-to-date WI speech coders, the second method is preferred because of its simplicity. Experimentally, Kleijn [37] found that sampling the CW surface at 40-50 Hz and transmitting the pitch cycles in the form of the gains and indices of an adaptive and a stochastic codebook at 3 kb/s generate very high-quality speech signal for voiced speech. To obtain similar quality for unvoiced speech, the CW surface must be sampled at 400 Hz, which is not practical for low bit rate speech coding [38]. To overcome this problem, the CW is generated for each sample of the original signal using the discrete-sampling method, and then the CW surface is filtered with a low-pass filter along the t axis to obtain the slowly evolving waveform (SEW) that contains the voiced speech part of the CW surface. When the filter's cut-off frequency is set to 20 Hz, the SEW can be sampled at 40 Hz and transmitted more accurately. The remaining signal is called rapidly evolving waveform (REW), which usually contains the unvoiced information. By allowing a slight aliasing, the REW can be sampled at 160 Hz and transmitted with a rough spectral magnitude representation without a significant loss in quality. The WI model is also very similar to the MELP model in terms of modeling the excitation signal; both models use a mixture of pulse train and white noise sequence. However, as the MELP model forced a band to be either voiced or unvoiced, the WI model can parametrically assign a mixture of periodic pulses and noise for each part of the spectrum. Although the WI model seems better in this respect, Kang et al. [34] reported that its quality is distinctly noisy for male speech because of the excessive noise in high frequency bands. This suggests that estimating the voicing degree information from the SEW/REW signals is not as robust as estimating it from bandpass filtered signals. A version of this coder

was also a candidate in the recent 2.4 kb/s U.S. military standard competition [40]. One of the main advantages of the WI model is its scalability. Kleijn et al. [38] reported that the quality of the synthesized speech was improved by increasing the sampling rate of the CW surface. Table 1 presents the mean opinion scores (MOS) for different sampling rates. These results suggest that transparent quality can be obtained at high CW sampling rates; however it is also possible to argue that sampling the CW surface at this high rate is also similar to waveform encoding of the residual signal. Finally, because of the pitch-cycle operations, the computational complexity of a WI coder is usually very high compared to the other well-known techniques.

Table 1: The MOS scores for different CW sampling rates in the WI model.

CW Sampling Rate (Hz)	50	100	200	400
MOS Scores	2.3	2.8	3.6	4.0

2.1.3 Hybrid Coders

Like the LPC coders, hybrid coders also use the excitation+filter model. In addition, they find and encode a suitable excitation signal waveform to complement this filter instead of modeling the excitation signal parametrically. Hybrid speech coders usually obtain and encode the excitation signal in two different ways. In the first method, the residual signal is obtained by filtering the speech signal with the inverse of the all-pole filter and is quantized directly such as in a form of ADPCM coders [13]. However, these coders are also often considered as waveform-matching coders, because the linear-prediction filter is only used as a pre-whitening filter that removes the short-time correlation in the signal and is not used for modeling purposes. In the second method, an excitation signal that generates the “best” synthesized speech signal when filtered with the all-pole filter is found by searching a set of possible excitation signals. This method is known as the analysis-by-synthesis (AbS) method.

A typical AbS coder first does a linear-prediction analysis on a frame-by-frame basis, typically in every 20-30 ms. Furthermore, the correlation between the successive pitch cycles are removed with the inverse of a pitch-prediction filter [62]. Then, each candidate

excitation signal is filtered with the pitch-prediction and all-pole synthesis filters and the resulting signal is subtracted from the original signal to find an error signal. The error signal is then filtered with a perceptual-weighting filter and the energy of the filtered error signal is computed for each candidate excitation signal. Finally, the excitation signal that results in the lowest weighted-error energy is selected and transmitted to the decoder. The weighting filter is obtained from the linear-prediction filter by moving its roots towards the origin by a constant factor without changing their angles [4]. This would result in a filter with the same formant frequencies but with wider bandwidths. This filter typically emphasizes the frequency regions with low energy (the valleys between the peaks of the spectrum) and de-emphasizes the frequency regions with large energy (peaks of the spectrum). This method exploits the masking property of the human auditory system that masks the noise in the frequency region centered on a high-energy sinusoid [82].

The most well-known and widely-studied AbS methods are the multi-pulse linear prediction (MP-LP) [2] and code-excited linear prediction (CELP) [3] algorithms. The regular-pulse excitation linear prediction (RPE-LP) method is also another well-known AbS algorithm [42] that has found its way to various standards. Since these methods directly encode an excitation signal, most hybrid coders are sometimes considered as different forms of waveform-matching coders. Thus, they allow efficient encoding of various speech sounds, maintain the time synchrony between the original and synthesized signals, and are robust to background noise to some degree when the bit rate is sufficient. They can compensate for the deficiencies in the all-pole filter estimation as well. The main problem of the AbS coders is their high computational complexity. To solve this problem, often sub-optimum methods are employed in searching for the best excitation signal. In addition, hybrid coders, particularly CELP coders, are generally scalable between 5 and 16 kb/s [54], but the synthesized speech quality degrades rapidly below 4.8 kb/s; the bit rate below this level is not sufficient to quantize the excitation signal directly. This makes them not suitable for low bit rate speech-coding applications.

The main idea behind the MP-LP algorithm introduced by Atal et al. [2] is to encode the excitation signal with a set of pulse locations and amplitudes. A typical MP-LP algorithm

computes the energy of the filtered error signal for each possible set of pulse locations and selects the pulse set that results in the minimum error. However, the complexity increases exponentially with the number of pulses and approximately 15-20 pulses are required for a 20 ms frame [72]. To reduce this large search space, the frame is divided into smaller subframes and only a small number of pulses are searched within each subframe. In addition, to decrease the computational complexity further, an iterative algorithm is used to search for only one pulse at a time while preserving the location of the pulses obtained in the previous iterations [2]. This method also results in a set of linear equations in which the unknowns are the pulse amplitudes. As a result, the pulse amplitudes and the filtered error energy can be found by a closed-form solution. This basic algorithm can synthesize high-quality speech at bit rates between 9.6 and 16 kb/s [72]. The performance of such coders can be improved in various ways. Singhal et al. [72] reported that including the pitch-prediction filter decreases the required number of bits while maintaining similar quality. Furthermore, they also extended the method such that the linear-prediction coefficients and amplitude locations are obtained simultaneously in a closed-form solution [72]. Hasib et al. [22] developed another algorithm which also optimizes the coefficients of the pitch-prediction filter as well as the amplitude of pulses and the linear-prediction filter coefficients. Very high-quality speech can be obtained using these methods around 8-9.6 kb/s. Finally, Unno et al. [77] also reported a variable bit rate MP-LP coder that can operate between 5.5 and 10 kb/s without sacrificing the quality of the synthesized speech.

The RPE-LP algorithm is a simplified version of the MP-LP algorithm. In this method, regularly spaced pulses are used to encode the excitation signal [42]. As the pulses are regularly spaced, the number of possible pulse locations is equal to the number of samples between the pulses. As a result, a RPE-LP algorithm finds the amplitudes of all possible pulse sets the same as the MP-LP algorithm and transmits only the location of the first pulse and the amplitudes of that pulse set. Although this method does not improve the performance over the MP-LP method, the computational complexity is decreased significantly without affecting the speech quality much when encoding speech at similar bit rates.

Furthermore, as the algorithms are almost identical, all algorithms that enhance the performance of the MP-LP algorithm can also be used with the RPE-LP method. A 13 kb/s RPE-LP speech coder that also incorporates a pitch-prediction filter was adopted as the speech-coding standard for time-division multiple access (TDMA) digital cellular telephony by the global system for mobile communications (GSM) subcommittee of the European Telecommunications Standards Institute (ETSI) [11, 24].

The most extensively studied form of the AbS algorithm is the CELP algorithm. Instead of finding a set of pulses, a stochastic codebook of different excitation signals is searched such that the selected excitation signal generates the best speech signal when filtered with the pitch-prediction and the linear-prediction filters [3]. Each entry of the codebook consists of a different Gaussian white-noise sequence. Although the initial algorithm introduced by Atal et al. had an extraordinary complexity that prohibited a real-time implementation, this method proved that obtaining very high speech quality at bit rates between 5 and 9.6 kb/s is indeed possible. In the following years, several researchers proposed different codebook designs that reduced the computational complexity and storage requirements significantly and also improved the performance [19]. As a result, the real-time implementation of high-quality CELP coders became possible. Furthermore, Rose et al. [63] proposed the use of an adaptive codebook whose entries are generated from the previous frame's excitation signal. Later, Kleijn et al. [39] used the same idea to replace the pitch-prediction filter in a CELP algorithm. This method is now used in almost all current CELP coders. Several variations of the CELP algorithm were adopted in various standards operating between 16 and 4.8 kb/s such as ITU's 8 kb/s G.729 standard based on conjugate structure-algebraic CELP method (CS-ACELP), ITU's 16 kb/s G.728 standard based on low-delay CELP method (LD-CELP) and 4.8 kb/s U.S. Federal Standard, FS-1016, [64, 10, 1]. Since the CELP algorithm is beyond the scope of this thesis, only a brief summary is given here. Interested readers may refer to the review paper written by Gersho [19] for more information.

2.1.4 Parametric/Hybrid Speech Coders

In the recent years, there has been a growing interest in designing a toll-quality speech coder at 4 kb/s. The main challenge for achieving this goal is that none of the speech coding techniques described above is suitable for this purpose. Although the parametric coders have very high segmental speech quality in stationary segments, they often fail to represent transition regions and short events accurately. For this reason, the quality of these coders usually saturates at about 4 kb/s and does not approach toll-quality. On the other hand, although the hybrid coders are capable of encoding all kinds of signals including transition effects, they fail to make high-quality encoding of stationary segments at 4 kb/s, as they also try to capture the perceptually unimportant phase information.

To solve this problem, several researchers have proposed techniques that use the best of these two methods to encode different parts of speech signal: parametric speech coding methods for stationary segments and hybrid speech coding methods for transition segments. However, the main problem in this approach is switching between these two encoding methods. As parametric coders do not encode the phase information, the synthesized signal is not time-synchronous with the original signal and the waveform shapes of original and synthesized signals are quite different. On the other hand, hybrid coders preserve the phase information, and as a result synthesized signal is time-synchronous with original signal and their waveform shapes are similar. For this reason, switching between these two encoding methods usually introduces audible artifacts in synthesized speech.

This problem can be approached in several ways. Shlomot et al. [69] reported a parametric/hybrid coder that uses a sinusoidal coder and a multi-pulse coder to encode the voiced segments and transition segments, respectively. The unvoiced segments are synthesized with spectrally shaped noise in this coder. To reduce the switching artifacts at onsets (i.e. transition from the multi-pulse coder to the sinusoidal coder), the decoder adjusts the phase of the sinusoids such that they are in phase at the location of the pitch pulse transmitted with the multi-pulse coder in the previous frame. When a transition occurs from the sinusoidal coder to the multi-pulse coder, the encoder modifies the original signal such that it is time-synchronous with the synthesized speech. In the formal listening tests, it was

found that the quality of the 4 kb/s coder is very close to that of the CELP based 5.3 kb/s G.723.1 coder. In other work, Stachurski et al. [73] proposed to use the MELP and the CELP coders to encode voiced segments and transition/unvoiced segments of the speech signal, respectively. Similar to Shlomot's method, when the main coder switches from the CELP coder to the MELP coder, the decoder adjusts the phase of the harmonics such that they are in phase at the location of the pitch pulse transmitted with the CELP coder. However, instead of modifying the original signal, the MELP coder estimates and transmits an alignment phase to the decoder such that the synthesized speech is always time-synchronous with the original speech. Furthermore, Stachurski et al. also proposed applying a zero-phase equalization filter to the original signal to remove the phase information from the CELP's target signal. They reported that the quality of the synthesized speech is equivalent to that of the 32 kb/s ADPCM coder in the listening tests. Katugampala et al. [35] proposed a similar parametric/hybrid coder that uses a sinusoidal coder to encode the voiced segments and a CELP coder to encode the transition segments. In this coder, a pitch pulse location parameter that specifies the position of the pitch pulse relative to the frame boundaries and a pitch shape parameter that determines the rough shape of the pitch pulse in the residual signal are transmitted to the decoder in voiced frames. As a result, the synthesized signal in the sinusoidal coder is both time-synchronous with the original signal and the waveform shapes of the both signals are sufficiently close so that there is no need to make any change on the original signal to avoid switching artifacts. In informal A/B comparison listening tests, this new sinusoidal/CELP coder is consistently preferred over the 5.3 kb/s G.723.1 standard coder, but the 8 kb/s G.729 coder is slightly preferred over this new coder.

2.2 Linear Prediction Methods for Speech Coders

Application of linear prediction to speech analysis and synthesis dates back to late 60s'. Initially, Itakura and Saito applied linear prediction to speech analysis in the context of maximum likelihood approach [28]. In the following years, several researchers investigated different aspects of the linear prediction analysis and applied this technique successfully to different speech processing applications [44]. Furthermore, Fant developed a mathematically

tractable speech production model that results in filtering a source with a series of linear filters [18]. These filters are used to model glottal shaping, vocal-tract shaping and lip radiation. It is usually assumed that these three filters can be combined into a single all-pole linear prediction filter. As discussed in the previous section, this discovery opened up many opportunities that enable efficient coding of speech signal at low bit rates.

As one of the main contributions of this thesis is a new linear prediction method that also makes the well-known linear prediction methods to estimate the same set of prediction coefficients under a certain assumption, these common methods are summarized in this section. These methods can be classified into following groups:

- Direct-form all-pole filter estimation methods
- Lattice-form all-pole filter estimation methods
- Frequency domain all-pole filter estimation methods
- Constraint excitation all-pole filter estimation methods

For the interested readers, a detailed description and analysis of linear prediction method can be found in the book written by Markel and Gray [47] and in the survey paper written by Makhoul [44].

2.2.1 Direct-Form All-Pole Filter Estimation Methods

In the absence of an input signal, the application of an all-pole filter is basically the same as the prediction of the output signal from the linear combination of its past samples, which can be written as

$$\hat{x}[n] = \sum_{k=1}^p a_k \hat{x}[n - k], \quad (4)$$

where $\hat{x}[n]$ is the predicted signal, p is the order of the filter, and a_k is the k^{th} coefficient of the inverse of the prediction filter, $A(z)$, defined as

$$A(z) = 1 - \sum_{k=1}^p a_k z^{-k}. \quad (5)$$

The all-pole filter, $\frac{1}{A(z)}$, is known as the prediction filter or the synthesis filter. This technique is also known as linear prediction (LP) of the signal.

If the error between the predicted signal and the original signal is white Gaussian noise, the optimum filter coefficients can be obtained by minimizing the sum of the squared error between the original and the predicted signal, ε_{LP} , which can be written as

$$\begin{aligned}\varepsilon_{LP} &= \sum_{n=n_1}^{n_2} e^2[n] \\ &= \sum_{n=n_1}^{n_2} (x[n] - \hat{x}[n])^2,\end{aligned}\tag{6}$$

where $e[n]$ is the prediction error defined as $x[n] - \hat{x}[n]$, and $x[n]$ is the original signal [23]. This assumption is reasonable for speech signals, because the excitation signal for unvoiced speech can be modeled with white noise and the excitation signal for voiced speech can be modeled with an impulse train which is only correlated at lags equal to integer multiples of the pitch period. Since ε_{LP} is a quadratic function of predictor coefficients, it has a global minimum. Therefore, the optimum predictor coefficients can be found by minimizing the ε_{LP} with respect to the filter coefficients and solving the resulting linear equations obtained as

$$\sum_{k=1}^p a_k \sum_{n=n_1}^{n_2} \hat{x}[n-k] \hat{x}[n-l] = \sum_{n=n_1}^{n_2} x[n] \hat{x}[n-l] \quad l = 1, \dots, p.\tag{7}$$

Although these equations are linear, $\hat{x}[n]$ is unknown before the filter coefficients, a_k 's, are computed. To simplify (7), (4) can be written as

$$\hat{x}[n] = \sum_{k=1}^p a_k x[n-k],\tag{8}$$

which makes (7)

$$\sum_{k=1}^p a_k r(k, l) = r(0, l) \quad l = 1, \dots, p,\tag{9}$$

where $r(k, l)$ are correlation coefficients defined by

$$r(k, l) = \sum_{n=n_1}^{n_2} x[n-k] x[n-l].\tag{10}$$

It is also possible to combine all linear equations defined by (9) in a matrix-vector representation as

$$\begin{bmatrix} r(1,1) & r(2,1) & \dots & \dots & r(p,1) \\ r(1,2) & r(2,2) & \dots & \dots & r(p,2) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ r(1,p) & r(2,p) & \dots & \dots & r(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ \vdots \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(0,1) \\ r(0,2) \\ \vdots \\ \vdots \\ r(0,p) \end{bmatrix}, \quad (11)$$

or

$$\mathbf{R}\mathbf{a} = \mathbf{r}, \quad (12)$$

where \mathbf{R} is the matrix on the left side of (11), \mathbf{a} is the vector of the unknown predictor coefficients, $[a_1 \dots a_p]^T$, and \mathbf{r} is the vector on the right side of (11). As a result, the filter coefficients can be obtained by either solving the equations given in (9) or by calculating $\mathbf{a} = \mathbf{R}^{-1}\mathbf{r}$.

The two most common methods used in speech coders, the autocorrelation and covariance methods, use this formulation. These methods make different assumptions on the nature of the analyzed signal that effects how n_1 and n_2 are set in (10). The autocorrelation method assumes that the signal is a stationary ergodic stochastic process. As a result, the correlation coefficients can be computed from the signal as

$$r(k, l) = \sum_{n=-\infty}^{\infty} x[n-k]x[n-l]. \quad (13)$$

Since the boundaries of the summation are $\pm\infty$, any offset added to both k and l does not change the result. Therefore, the correlation coefficients can be computed as

$$r(k, l) = r(|k-l|) = r(i) = \sum_{n=-\infty}^{\infty} x[n]x[n-i], \quad (14)$$

where i is equal to $|k-l|$. In practice, it is also assumed that the correlation coefficients can be estimated from finite length signal. Therefore, before the analysis, the original signal is first multiplied by a finite length window to generate a new finite length signal:

$$x_w[n] = \begin{cases} x[n]w[n] & 0 < n < N-1 \\ 0 & \text{elsewhere,} \end{cases} \quad (15)$$

where $w[n]$ is the window function with N non-zero samples. As a result, the correlation coefficients are computed as

$$r(i) = \sum_{n=-\infty}^{\infty} x_w[n]x_w[n-i] = \sum_{n=i}^{N-1} x_w[n]x_w[n-i]. \quad (16)$$

This simplification actually turns the autocorrelation method into a minimized mean squared estimator. The performance of the autocorrelation method depends largely on the choice and length of the windowing function. Multiplying a rectangular window with the signal generates discontinuities across the boundaries, which also introduces a bias to the filter coefficients. This may severely decrease the effectiveness of the method. For this reason, a tapered window like Hamming window is usually multiplied with the signal prior to the analysis. Although application of a tapered window introduces its own distortion, the bias resulting from the application of the rectangular window reduces significantly. A review of this bias analysis can be found in [15]. In speech coding applications, a 20-30 ms Hamming window is usually used as the window function. This choice is also consistent with the assumption that the speech signal can be considered stationary in 20-30 ms segments.

The main advantage of the autocorrelation method comes from the computation of $r(k, l)$ by $r(|k - l|)$. In the matrix representation of the equations, this simplification turns \mathbf{R} into a positive-definite Toeplitz matrix:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(p-1) \\ r(1) & r(0) & r(1) & \dots & r(p-2) \\ r(2) & r(1) & r(0) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ r(p-1) & r(p-2) & \dots & \dots & r(0) \end{bmatrix} \quad (17)$$

This property always guarantees that the solution of the equations would result in a stable filter [23], and the equations can be solved efficiently by a recursive technique, called Levinson-Durbin recursion, without inverting \mathbf{R} . The Levinson-Durbin recursion is explained in Appendix B.

The covariance method does not make any assumption about the signal outside the modeling region, and thus, finds a set of coefficients that models the signal best in the given

finite duration in the least squares sense. Therefore, the covariance method is more accurate and better for modeling short segments, since there is no bias because of the application of a window function. In the covariance method, the correlation coefficients are computed as

$$r(k, l) = \sum_{n=p}^{N-1} x[n-k]x[n-l]. \quad (18)$$

Despite the modeling accuracy advantages, \mathbf{R} in (12) may not be positive definite and, therefore, the prediction filter may not be stable, which is unacceptable for speech coding and synthesis applications. In addition, the prediction error usually increases when the error signal has a pitch pulse in the beginning of the analysis frame. As a result, the performance of the covariance method is more dependent on the placement of the analysis window than the autocorrelation method [61]. For these reasons, the autocorrelation method is generally used in almost all state-of-the-art speech coders. A detailed discussion of both methods can be found in [47].

In addition to these two well-known methods, another linear prediction method, known as modified covariance method, is also reported in the literature [23]. In the modified covariance method, the sum of squared backward prediction error is also minimized as well as the sum of squared forward prediction error defined in (6) with respect to the predictor coefficients. The backward prediction error is defined as

$$e_b[n] = x[n-p] - \sum_{k=1}^p a_k x[n-p+k], \quad (19)$$

for a p^{th} order prediction filter. In this case, the sum of squared error, ε_{MC}^2 , is defined as

$$\begin{aligned} \varepsilon_{MC}^2 &= \sum_{n=p}^{N-1} e^2[n] + \sum_{n=p}^{N-1} e_b^2[n] \\ &= \sum_{n=p}^{N-1} \left[x[n] - \sum_{k=1}^p a_k x[n-k] \right]^2 + \sum_{n=p}^{N-1} \left[x[n-p] - \sum_{k=1}^p a_k x[n-p+k] \right]^2. \end{aligned} \quad (20)$$

To find the optimum predictor coefficients, ε_{MC}^2 is minimized with respect to prediction coefficients, and the resulting set of equations are obtained as

$$\sum_{k=1}^p a_k [r(k, l) + r(p-k, p-l)] = r(0, l) + r(p, p-l) \quad l = 1, \dots, p \quad (21)$$

The solution of (21) gives the optimum set of predictor coefficients that not only optimally predict the current sample from the past p samples but predict the current sample from the future p samples optimally as well. However, since only forward prediction is used in speech synthesis, modified covariance method is rarely used in speech coding applications.

2.2.2 Lattice-Form All-Pole Filter Estimation Methods

Lattice filters have many unique features including modularity, low sensitivity to quantization effects and a simple test to ensure the stability of an all-pole filter implementation. Because of these features, they find their way into many digital signal processing applications involving both signal analysis and synthesis. A lattice filter has a cascade structure made of concatenated basic stages that contain two inputs and two outputs. In the basic lattice structure, one of the inputs is first delayed by one sample and the outputs are generated by the linear combination of these one delayed and one non-delayed inputs within each of these stages. The update equations for the j^{th} stage in an FIR filter are defined as

$$e_j^+[n] = e_{j-1}^+[n] + \Gamma_j e_{j-1}^-[n-1], \quad (22)$$

$$e_j^-[n] = e_{j-1}^-[n-1] + \Gamma_j e_{j-1}^+[n], \quad (23)$$

where $e_j^+[n]$ and $e_j^-[n]$ are the output signals and denoted as forward and backward prediction error, respectively, and Γ_j is the partial correlation coefficient of the j^{th} stage. The signals, $e_{j-1}^+[n]$ and $e_{j-1}^-[n]$, are the input signals for this stage. Similarly, the equations for a single stage IIR filter is defined as

$$e_{j-1}^+[n] = e_j^+[n] - \Gamma_j e_{j-1}^-[n-1], \quad (24)$$

$$e_j^-[n] = e_{j-1}^-[n-1] + \Gamma_j e_{j-1}^+[n], \quad (25)$$

where $e_j^+[n]$ and $e_{j-1}^-[n]$ are the input signals, $e_{j-1}^+[n]$ and $e_j^-[n]$ are the output signals. There are also other lattice structures for all-pole filtering such as three multiplier and Kelly-Lochbaum forms [23], however they are not covered in this section. Interested readers may refer to [23] for more details on these structures. A single stage in an FIR filter and an IIR filter is also illustrated in Figure 3a and Figure 3b, respectively. An FIR and IIR lattice filter can be constructed by concatenating the associated basic stages by feeding one stage's

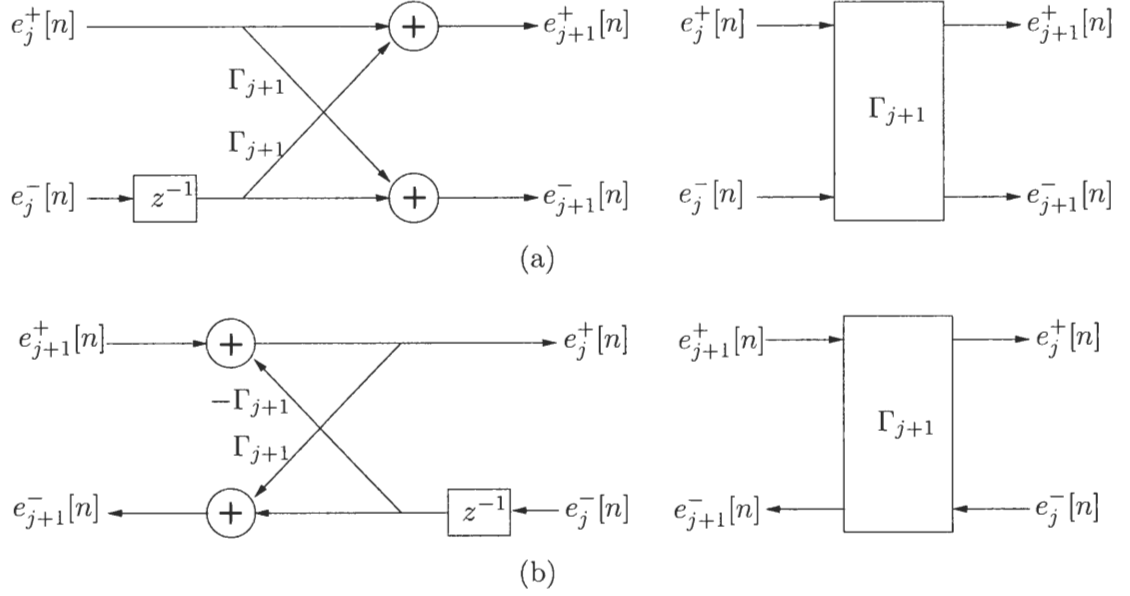


Figure 3: (a) Single-stage FIR lattice filter and its basic block representation, and (b) single-stage IIR lattice filter and its basic block representation

outputs to next one's inputs. An illustration to an FIR and IIR lattice filter is provided in Figure 4.

One of the key properties of this structure is its modularity: It is possible to change the order of the filter by adding or eliminating individual stages without computing the rest of the coefficients. Parametric signal modeling that does not require a fixed model order is one of the applications that benefits from this property; the number of stages can be increased until the prediction error decreases below a threshold. Another important property is the ability to make a simple test to check the stability of an all-pole filter implementation: when the absolute value of all partial correlation coefficients in the filter is less than one, the filter is always stable. This property is especially important for speech coding and synthesis applications implemented using fixed-point arithmetic [23]. Since update equations require a single multiplication and addition per sample, the stability of the filter is always guaranteed as long as the numerical representations of absolute of the partial correlation coefficients do not exceed one. This can be easily obtained by a lattice filter. Besides, all direct-form and lattice-form filters are interchangeable as well. Two recursive methods, known as *step-down recursion* and *step-up recursion*, are used to change the direct-form predictor coefficients to partial correlation coefficients and vice versa [23].

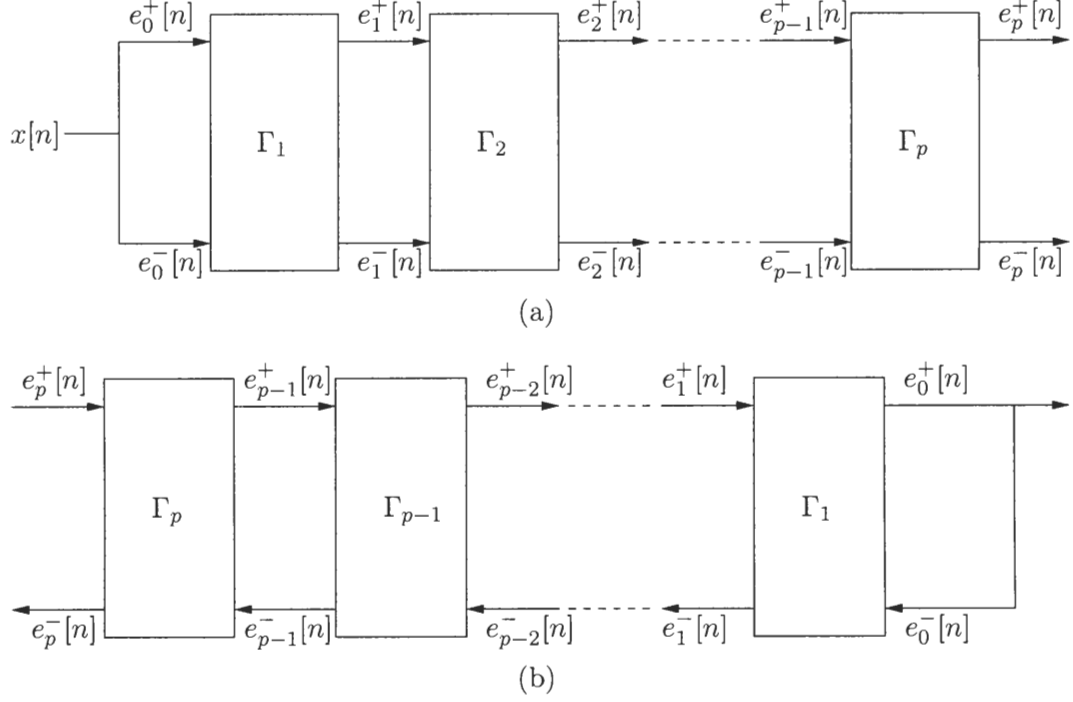


Figure 4: (a) The FIR lattice filter and (b) the IIR lattice filter

There are various ways to estimate the partial correlation coefficients for the purpose of linear prediction of the signal. One way is to use the reflection coefficients obtained in the Levinson-Durbin recursion. The reflection coefficients obtained by this method can be used directly as Γ_j 's in (24) and (25) for all-pole filtering. In addition, there are other methods to obtain the partial correlation coefficients directly as well. In these methods, the objective is to find a set of partial correlation coefficients in the FIR filter illustrated in Figure 4 that whitens the input signal. Since, the error signals are assumed to be white Gaussian noise, the optimum partial correlation coefficient at the j^{th} step can be obtained by minimizing the sum of squared error of either error term or the summation of both error terms. The sum of the squared error terms are defined as

$$\varepsilon_j^+ = \sum_{n=j}^{N-1} |e_j^+[n]|^2, \quad (26)$$

$$\varepsilon_j^- = \sum_{n=j}^{N-1} |e_j^-[n-1]|^2, \quad (27)$$

where ε_j^+ and ε_j^- are sum of the squared forward prediction error and the backward prediction error in the j^{th} step, respectively, and N is length of the modeling region. The three

techniques that minimize these error terms are:

- forward covariance method that minimizes ε_j^+ ,
- backward covariance method that minimizes ε_j^- and,
- Burg's method that minimizes the summation of both error terms.

In forward covariance method, the solution of the minimization of (26) with respect to Γ_j^+ results in the Γ_j^+ as

$$\begin{aligned}\Gamma_j^+ &= -\frac{\sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\sum_{n=j}^{N-1} |e_{j-1}^-[n-1]|^2} \\ &= -\frac{\sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\varepsilon_{j-1}^-}.\end{aligned}\tag{28}$$

Similarly, in backwards covariance method, the solution of the minimization of (27) with respect to Γ_j^- would result in

$$\begin{aligned}\Gamma_j^- &= -\frac{\sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\sum_{n=j}^{N-1} |e_{j-1}^+[n]|^2} \\ &= -\frac{\sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\varepsilon_{j-1}^+}.\end{aligned}\tag{29}$$

Finally, the Burg's method minimizes summation of both error terms,

$$\begin{aligned}\varepsilon_j^B &= \varepsilon_j^+ + \varepsilon_j^- \\ &= \sum_{n=j}^{N-1} |e_j^+[n]|^2 + \sum_{n=j}^{N-1} |e_j^-[n-1]|^2\end{aligned}\tag{30}$$

with respect to Γ_j^B , which results in

$$\begin{aligned}\Gamma_j^B &= -\frac{2 \sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\sum_{n=j}^{N-1} |e_{j-1}^+[n]|^2 + \sum_{n=j}^{N-1} |e_{j-1}^-[n-1]|^2} \\ &= -\frac{2 \sum_{n=j}^{N-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\varepsilon_{j-1}^+ + \varepsilon_{j-1}^-}.\end{aligned}\tag{31}$$

In all three methods, it can be seen that the partial correlation coefficients can be obtained using the error signals obtained in the previous iteration. Since, the filter acts as a whitening filter, the iteration starts with setting $e_0^+[n]$ and $e_0^-[n]$ to $x[n]$, which makes both ε_0^+ and ε_0^- equal to the energy of the input signal.

The partial correlation coefficients generated by all three methods are usually different from one another. In addition, they are usually different from the reflection coefficients obtained by the Levinson-Durbin recursion after an autocorrelation analysis and from the partial correlation coefficients obtained by the step-down recursion after a covariance analysis. The modeling efficiency of each method depends on the input signal. Furthermore, among all these methods, only the partial correlation coefficients obtained by the Burg's method guarantees the stability of the all-pole filter. A detailed analysis of all these methods can be found in [23]. These modeling techniques are rarely used in speech coding applications. They have slightly larger computational complexity than that of the autocorrelation method. Furthermore, all of the predictor coefficients as well as the reflection coefficients are optimized in a single step in the direct all-pole filter estimation methods, while the partial correlation coefficients are only optimized for the updated error signal. Therefore, it is more likely that the reflection coefficients obtained by the direct all-pole filter estimation methods models the signal better than the partial correlation coefficients obtained by the lattice all-pole filter estimation methods for fixed order prediction filters.

2.2.3 Frequency Domain All-Pole Filter Estimation Methods

The frequency-domain formulation of linear prediction can be derived from the time-domain formulation discussed in Section 2.2.1. Although this new derivation results in a different way of computing the same set of predictor coefficients with the autocorrelation method, it requires the knowledge of the frequency-domain representation of the analyzed signal, which can usually be obtained by time-frequency transformations. As a result, although it seems that it is beneficial to use time-domain formulations of the linear prediction method to avoid the additional computational complexity resulting from the time-frequency transformation, the frequency-domain linear prediction methods provides a better understanding of the spectral modeling property of the linear prediction method. In this section, it is assumed that the signal being analyzed is a stationary ergodic process as in the case of autocorrelation method. Therefore, the analysis is still carried out on the input signal. The analysis for non-stationary case can be found in [45].

In Section 2.2.1, the squared sum of the prediction error is defined as

$$\begin{aligned}\varepsilon_{LP} &= \sum_{n=-\infty}^{\infty} e^2[n] \\ &= \sum_{n=-\infty}^{\infty} (x[n] - \sum_{k=1}^p a_k x[n-k])^2.\end{aligned}\tag{32}$$

Using Parseval's theorem, the sum of squared prediction error can be written as

$$\begin{aligned}\varepsilon_{LP} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega}) - \sum_{k=1}^p a_k X(e^{j\omega}) e^{jk\omega}|^2 d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 \left| 1 - \sum_{k=1}^p a_k e^{jk\omega} \right|^2 d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 A(e^{j\omega}) A^*(e^{j\omega}) d\omega,\end{aligned}\tag{33}$$

where $A(e^{j\omega})$ is the frequency response of inverse of the prediction filter. The optimum predictor coefficients can be found by setting the partial derivative of ε_{LP} with respect to the prediction coefficients to zero, and solve the linear equations. The partial derivative of ε_{LP} with respect to the prediction coefficients is computed as

$$\begin{aligned}\frac{\partial \varepsilon_{LP}}{\partial a_l} &= -\frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 \left[\left(1 - \sum_{k=1}^p a_k e^{jk\omega} \right) e^{-jl\omega} + \left(1 - \sum_{k=1}^p a_k e^{-jk\omega} \right) e^{jl\omega} \right] d\omega \\ &= -\frac{2}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 \left(\cos(\omega l) - \sum_{k=1}^p a_k \cos(\omega |k-l|) \right) d\omega = 0,\end{aligned}\tag{34}$$

where l is between 1 and p . (34) results in a set of following equations

$$\sum_{k=1}^p a_k R(|k-l|) = R(l) \quad l = 1, \dots, p,\tag{35}$$

where the correlation coefficients, $R(i)$, are defined as

$$R(|k-l|) = R(i) = \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 \cos(\omega i) d\omega.\tag{36}$$

The set of equations defined in (35) is exactly same as the ones obtained by the autocorrelation method. Therefore, the resulting linear prediction filter is always stable and the equations can be solved by the Levinson-Durbin recursion.

Equation (33) also gives an insight to the spectral modeling properties of linear prediction. To present these properties, it is better to rewrite (33) as

$$\begin{aligned}
\varepsilon_{LP} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega \\
&= \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{|X(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega \\
&= \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P_x(e^{j\omega})}{\hat{P}_x(e^{j\omega})} d\omega,
\end{aligned} \tag{37}$$

where $H(e^{j\omega})$ is the all-pole synthesis filter, G^2 is the scalar gain term, $P_x(e^{j\omega})$ is the squared magnitude of the frequency domain representation of the speech signal and $\hat{P}_x(e^{j\omega})$ is the squared magnitude of the frequency response of the synthesis filter. When G^2 is set to the sum of the squared prediction error, the following result is obtained:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P_x(e^{j\omega})}{\hat{P}_x(e^{j\omega})} d\omega = 1. \tag{38}$$

Equation (38) implies three important results. The total error is determined by the sum of the ratios of the two spectra at instantaneous frequencies. As a result, the modeling accuracy is uniform throughout the whole spectrum and is not dependent on the spectrum shape. On the other hand, the integration also ensures that the arithmetic mean of the ratio of the two spectra at instantaneous frequencies would be one, and hence, the ratio would be larger than one for some frequencies and smaller than one for the others. As a result, the contribution to total error is more when $P_x(e^{j\omega})$ is larger than $\hat{P}_x(e^{j\omega})$, hence, the modeling is usually better in the regions where $P_x(e^{j\omega})$ is large. That's the reason why the spectral estimation with linear prediction methods is usually better around formant resonances than the valleys in between. However, that is also the reason why $\hat{P}_x(e^{j\omega})$ becomes very large compared to $P_x(e^{j\omega})$, when there is an interaction of a harmonic with a narrow-bandwidth formant resonance. Finally, (38) also implies that the linear prediction filter is a spectral flattening filter. In the ideal case, the ratio of the two spectra should be one throughout the spectrum.

In practice, it is impossible to obtain a continuous frequency domain representation of the original signal. Besides, as discussed before, the speech signal can be considered stationary only for a short time. Therefore, this information is usually obtained either by a

bank of filters or through a discrete Fourier transform (DFT) of a windowed speech signal. In both cases, (37) reduces to

$$\varepsilon_{LP} = \frac{G^2}{N} \sum_{n=0}^{N-1} \frac{P_x(e^{j\omega_n})}{\hat{P}_x(e^{j\omega_n})}, \quad (39)$$

where N is the number of spectral locations on the unit circle. As a result, the correlation coefficients are computed as

$$R(i) = \frac{1}{N} \sum_{n=0}^{N-1} P(\omega_n) \cos(\omega_n i). \quad (40)$$

This method is known as *discrete-spectrum linear prediction* modeling method [45]. As the spectrum of the voiced speech consists of only harmonics at the integer multiples of the fundamental frequency, it is a natural candidate for this method. However, as pointed out in [45] and [14], the sampling of the spectra at the harmonics of the fundamental frequency results in aliasing in the correlation coefficients that leads to discrepancies in the spectral estimation. This problem worsens with the increase in the fundamental frequency. When there are only a few harmonics in the speech spectrum, this method tends to model the large magnitude harmonics with separate poles. This results in a prediction filter with a frequency response that contains formant resonances with very narrow bandwidths. However, this problem is not restricted to discrete-spectra linear prediction modeling method. All other time-domain linear prediction methods suffer from the same problem when the pitch period is very short. As a result, the performance of this method can be considered as good as the common time-domain linear prediction estimation methods discussed in the Section 2.2.1.

Several other methods were also proposed in literature to improve the performance of the frequency-domain linear prediction methods especially for high-pitched speakers. Hermansky et al. suggested to use transformed spectral samples before the error the minimization to modify the dynamic range [25]. They claimed that when a *root spectral transform*, defined as $T[x] = x^{\frac{1}{r}}$, is used with $r > 2$, it is possible to decrease the sensitivity of the modeling method to the interaction of a harmonic with a narrow-bandwidth formant resonance, especially for high-pitched speakers. In another work, Hermansky suggested to fit a continuous parabolic function to the samples of the speech spectrum, and compute the correlation coefficients from this continuous function [26]. He claimed an improvement in the spectral

matching when used on both clean and noisy speech. In another approach, El-Jaroudi et al. introduced *discrete all-pole* modeling method (DAP) and used the Itakura-Saito distance measure in the discrete-spectra all-pole modeling method to improve the correlation matching property of the all-pole modeling [14]. They reported that the estimated spectrum usually fits to the original spectrum better than the one estimated with the discrete-spectra all-pole modeling method. Finally, in their recent work, Kabal et al. introduced a new frequency modeling method that takes the mixed-excitation property of speech into account, where the correlation coefficients for voiced and unvoiced parts of the spectrum is estimated differently [31]. In their work, they claimed that the spectral estimates corresponding to noisy region of the spectrum are more consistent from one frame to another compared to the other common techniques such as the autocorrelation method.

2.2.4 Constraint Excitation All-Pole Filter Estimation Methods

The techniques described in Sections 2.2.1 and 2.2.2 work very well when the prediction error signal is white Gaussian noise and the modeling region is sufficiently long. Unfortunately, the prediction error signal is not entirely Gaussian for voiced speech. As discussed before, voiced speech is generated by exciting the vocal tract with glottal pulses. Therefore, in each pitch cycle of a voiced speech segment, there are a few samples that have significantly larger amplitudes than those of the rest of the samples in the same cycle. Since the excitation signal is assumed white Gaussian noise in all linear prediction algorithms, these algorithms also minimize the energy resulting from these glottal pulses. In this case, although the prediction error in the samples without these pulses is supposed to be minimized, the energy resulting from these glottal pulses is minimized more because of their high energy and the energy of the prediction error in the samples without these pulses still remains large. For low-pitched speakers, there are only few of these pulses in the entire analysis frame. As a result, this is not a problem and the performance of all methods when used on such signals is as good as when used on signals generated by white Gaussian noise. However, when pitch cycles get shorter, this problem becomes more apparent and the frequency response of the estimated filter deviates more from the true frequency response. To correct this problem, two different

methods have been proposed in the literature.

In the first method, the error samples are weighted such that these pulses with large amplitudes receive less weight than the rest of the samples. In this case, (6) is modified as

$$\varepsilon_{WLP} = \sum_{n=p}^N w[n](x[n] - \hat{x}[n])^2, \quad (41)$$

where $w[n]$ is used to weight the squared prediction error and to de-emphasize the samples with large amplitudes resulting from the glottal pulses. The first of these methods was reported by Mizoguchi et al. and named as *selective linear prediction* method [59]. In their algorithm, a new formulation of linear prediction allows them to use successive computation of the least-squares solution of the linear prediction coefficients for each sample of the analysis frame, which also produces the energy of the residual signal as a side product. While computing the prediction coefficients, they reject the input samples that increase the energy of the residual signal. They reported that the residual signal for high-pitched speakers obtained by this method is peakier than the one obtained by conventional methods. In another technique reported by Miyoshi et al. [58], the prediction coefficients are first computed with one of the common methods and then the resulting filter is used as an inverse filter to generate the residual signal. Then, the samples with large energies and the samples just before those samples are excluded from computation of the sum of squared prediction error. They reported that the performance of the algorithm is equivalent to conventional methods for low-pitched speakers and is better for high-pitched speakers. Also, the error in formant estimation of the synthetic speech signal was decreased from 6% to 2% on the average. Finally, Lee proposed a novel method in [43], known as *robust linear prediction* method, that uses a minimax estimator that results in a sum of squared prediction error, ε_{RLP} , written as

$$\varepsilon_{RLP} = \sum_{n=p}^N \hat{e}[n]^2, \quad (42)$$

where $\hat{e}[n]$ is obtained as

$$\hat{e}[n] = \begin{cases} (x[n] - \hat{x}[n])^2 & |x[n] - \hat{x}[n]| \leq c \\ 2c|x[n] - \hat{x}[n]| - c^2 & |x[n] - \hat{x}[n]| > c, \end{cases} \quad (43)$$

and c is a suitably chosen constant. The solution of Lee's method requires iterative techniques. However, he claimed that the formant and bandwidth estimation errors reduced below 0.001% and 0.1% respectively after four iterations for a synthetic speech with high fundamental frequency. Lee also showed that the performance of the robust linear prediction algorithm is independent of the position of the analysis frame using line spectrum frequencies (LSF) obtained from a real speech data.

In the second method, a known excitation signal, $d[n]$, is added to the linear prediction formulation such that this known excitation compensates the glottal pulses that bias the estimation of the filter coefficients:

$$\hat{x}[n] = \sum_{k=1}^p a_k x[n-k] + d[n]. \quad (44)$$

When the sum of squared prediction error using (44) is minimized with respect to the prediction coefficients, the solution results in following equations:

$$\begin{aligned} \sum_{k=1}^p a_k \sum_{n=n_1}^{n_2} x[n-k]x[n-l] &= \sum_{n=n_1}^{n_2} x[n]x[n-l] + \sum_{n=n_1}^{n_2} d[n]x[n-l] \\ \sum_{k=1}^p a_k r(k, l) &= r(0, l) + r_{dx}(0, l), \end{aligned} \quad (45)$$

where $r_{dx}(0, l)$ is the cross-correlation between the known excitation signal and the l sample delayed input signal. Various techniques were reported to improve the performance of linear prediction using this method in the literature. The most straight forward method was first introduced by Erkelens [15]. In this method, the linear prediction coefficients are first obtained by one of the common techniques, and the residual signal is obtained by filtering the original signal with inverse of this estimated filter. Then, the residual signal is used as $d[n]$ in (44) to re-estimate the predictor coefficients. This process continues iteratively until both prediction filter coefficients and residual signal converge. Unfortunately, Erkelens found that this method optimizes both of the residual spectrum and the prediction filter spectrum simultaneously, and the final prediction filter spectrum does not resemble a speech spectrum. As a result, interpolation of the filter parameters from one frame to another in the speech synthesis results in audible artifacts. However, he claimed that in an analysis-by-synthesis coder, the prediction filter coefficients can be re-estimated after the excitation

signal is obtained. He claimed that the SNR of the reconstructed speech with the re-optimized prediction filter is higher than the one obtained with the initial prediction filter. This approach was adopted by various researchers as well. Initially, Singhal et al. used the same technique to re-optimize the filter coefficients simultaneously while obtaining the amplitudes of the pulses in the excitation signal in a MP-LP coder [71]. In this case, $d[n]$ is represented by a set of pulses with unknown amplitudes such as

$$d[n] = \sum_{l=1}^L G_l \delta(n - n_l), \quad (46)$$

where L is the number of pulses, G_l and n_l are the amplitude and position of the l^{th} pulse, respectively. When the pulse locations are known, optimum G_l and prediction coefficients can be found simultaneously by minimizing ε_{LP} with respect to prediction coefficients and G_l and solving the resulting equations. Singhal reported that the synthesized speech quality for female speakers is improved with this method. In another work, Hasib et al. [22] also incorporated the pitch-prediction filter coefficients with unknown gains into (46) to improve the performance. In this case, $d[n]$ is defined as

$$d[n] = \sum_{l=1}^L G_l \delta(n - n_l) + \sum_{m=-M}^M P_m x[n - \tau + m], \quad (47)$$

where $2M + 1$ is the number taps of the pitch-prediction filter, P_m is the gain of the m^{th} coefficient of the filter and τ is the pitch lag. In this case, the pitch-prediction filter coefficients can also be computed simultaneously with the prediction filter coefficients and the gain of the pulses. They reported an improved SNR of the reconstructed signal compared to Singhal's method. Besides, they also claim that the number of low SNR outliers decreased. Serizawa [67] also used a similar technique to find the linear prediction coefficients and the closed-loop pitch parameters simultaneously in a CELP coder. He also claimed to achieve an improvement in the average SNR of the reconstructed speech obtained with this method over the reconstructed speech obtained with sequentially computed linear prediction and pitch-prediction filters. Finally, Kabal et al. used this method to find the best combination of the pitch-prediction and linear prediction filters using

$$d[n] = \sum_{m=-M}^M P_m x[n - \tau + m]. \quad (48)$$

They reported two different methods in which the parameters of the two filters can be either estimated sequentially in an iterative approach or estimated simultaneously as in the case of Hasib et al.'s method [32]. In both methods, they also claimed improved quality and SNR of the reconstruction speech. In all these methods, researchers found that this technique especially improves the quality of the female speech.

CHAPTER III

PITCH-PERIOD ESTIMATION AND PITCH-CYCLE SEGMENTATION

The pitch-period and voicing-degree are arguably the most important parameters in a parametric speech coder. An incorrect estimation of these parameters usually results in severe audible artifacts in the synthesized speech. As the new methods introduced in this thesis are pitch-synchronous methods, the correct estimation of the pitch-period becomes even more crucial. However, the correct estimation of the pitch-period alone does not guarantee working of these algorithms correctly; the pitch-cycle boundaries must also be located accurately. For these reasons, the performance of a pitch-synchronous system depends on both the pitch-period estimation and the pitch-cycle segmentation algorithms.

In the next section, an accurate pitch-period estimation algorithm is presented. This algorithm is based on finding a pitch track in a number of subframes that minimizes the pitch-prediction residual error. In performance tests, this algorithm estimated the correct pitch period most of the time, even in transition regions. In addition, this chapter also presents two new pitch-cycle segmentation algorithms that are capable of segmenting the critically sampled narrowband speech signal into pitch cycles in fractional sample resolution. These algorithms are based on maximizing the linear-prediction gain of a single cycle and maximizing the normalized correlation in successive pitch cycles. Both methods are found to be very accurate in segmenting the speech signal.

3.1 Pitch-Period Estimation

One of the main characteristics of the voiced speech is its quasi-periodic nature. These signals can be said to resemble periodic signals whose waveform shapes and period lengths change slowly with time. All parametric and hybrid coders use the quasi-periodic nature of speech to eliminate the redundancy in consecutive cycles in the speech signal. The length

of each cycle is called the *pitch period*, and one period signal is referred as *pitch cycle*. The term *fundamental frequency* is also used to refer the frequency of the pitch cycles.

As discussed before, the voiced/unvoiced decision and pitch period are arguably the most important parameters affecting the synthesized speech quality in a parametric speech coder. Incorrect estimation of these parameters not only results in audible artifacts in the synthesized speech, but also effects the estimation accuracy of the other speech parameters (such as Fourier series magnitudes [79].) Therefore, pitch-period and voicing-state estimation have long been two of the main research areas in speech coding. A summary of common estimation techniques can be found in [27] written by Hess.

Common pitch-period estimation methods can be divided into two groups: time-domain methods based on correlation and frequency-domain methods based on harmonic tracking. Correlation methods use the fact that the autocorrelation function of a periodic signal has the same period as the signal, and the lag with maximum correlation ideally gives the pitch period of the signal. On the other hand, the Fourier transform of a periodic signal is only non-zero at the harmonics of the fundamental frequency, and frequency-domain techniques try to find the fundamental frequency whose harmonics structure is similar to that of the speech signal. When the maximum correlation in the autocorrelation function is low or no harmonic structure is detected in the spectrum, the speech signal is considered unvoiced.

Even the simplest forms of these algorithms can find the voicing state and pitch period correctly 80-90% of the time. However, there are cases in which finding the pitch period reliably is very difficult even with the most complex algorithms.

Almost all pitch-period detection algorithms assume that the speech signal is a stationary periodic signal. However, this assumption is often not strictly valid; spectral characteristics and energy of the signal and the time between glottal pulses usually vary in time, making the signal quasi-periodic. Especially at onsets and at the end of words, the variations are usually significant, making the signal non-stationary. The estimation of correlation function using time averages may not result in correct pitch-period estimation for these cases. Similarly, the Fourier transform of a time-varying signal may not give the true harmonic structure of the spectrum.

The second problem is associated with a voicing state in the speech signal known as vocal fry, which usually occurs at the end of the words. In these cases, the subglottal pressure is not enough to maintain even a quasi-periodic state, and the length of the associated pitch cycles is often multiples of the average pitch period and sometimes changes significantly from one pitch cycle to the next. A definition for “periodicity” is very difficult in these instances, because the duration of a vocal fry is also limited to very few pitch cycles.

Another problem occurs when the speech spectrum contains a formant with very narrow bandwidth. In this case, the second or third pitch harmonic’s magnitude may be much larger than the magnitude of fundamental frequency, and therefore the estimation algorithms may estimate the pitch period as a submultiple of the true pitch period. This problem is usually referred as the “pitch-halving” problem.

Finally, a common problem, known as pitch-doubling, occurs frequently for the correlation-based estimation algorithms in which an integer multiple of the true pitch period is found as the pitch period. The reason is that a periodic signal is also periodic in multiples of the true pitch period. In reality, even a small amount of noise may result in a slightly higher correlation at multiples of the true pitch period. Therefore, picking the lag that maximizes the correlation function may result in an incorrect pitch-period estimation.

Most of these problems are addressed in the pitch-period estimation algorithm of the new 2.4 kb/s military standard MELP coder [79]. This pitch-period estimation algorithm uses the normalized correlation coefficient as the correlation measure. The normalized correlation, ρ_τ , at lag τ is defined as

$$\rho_\tau = \frac{\langle x_0, x_\tau \rangle}{\sqrt{\langle x_0, x_0 \rangle \langle x_\tau, x_\tau \rangle}}, \quad (49)$$

where the correlation term, $\langle x_k, x_l \rangle$, is calculated as

$$\langle x_k, x_l \rangle = \sum_{n=0}^{L-1} x[n+k]x[n+l], \quad (50)$$

where $x[n]$ is the analyzed signal and L is the number of samples used in the calculation. This correlation measure is robust to energy variations and always bounded by ± 1 . In this algorithm, the correlation function is computed for lags between 40 and 160 using a low-pass

filtered speech signal whose cut-off frequency is 1 kHz. The lag maximizing this correlation function is selected as the initial candidate. Then, this selection is verified by calculating the normalized correlation coefficient of the lags around the initial estimate using another low-pass filtered speech signal with a cut-off frequency of 0.5 kHz. Finally, the estimation is verified once more using a low-pass filtered residual signal and the speech signal. The use of residual signal provides robustness for pitch-halving errors. Furthermore, a pitch-doubling elimination logic is used to reduce the pitch-doubling errors.

Although this algorithm in the MELP coder is very effective and even robust to background noise to some extent, pitch-doubling problems and voicing-state estimation errors still occur, especially at onsets and at the end of the words. Unno et al. [78] reported that the placement of the estimation window becomes crucial in these instances. To correct this problem, they proposed to estimate a set of normalized correlation functions using a sliding estimation window technique in which each correlation function is obtained from a different positioning of the estimation window around the frame's original estimation window location. The lag that maximizes the normalized correlation coefficient in the whole function set is selected as the pitch period and the corresponding analysis window is also used to estimate the other parameters, such as linear prediction coefficients and Fourier series magnitudes. Although this algorithm is very effective in correcting most of the pitch-period estimation problems, it introduces additional delay. In addition, it was also observed that the algorithm sometimes classifies an unvoiced signal as voiced by finding an analysis window location whose maximum correlation is slightly larger than the voicing threshold.

In this section, a new pitch-period estimation algorithm that copes with the problems stated above without introducing further delay and voicing-state estimation errors is described. Previously, McCree et al. [57] reported that a similar but subframe based approach improves estimation accuracy, especially at onsets. This approach is based on finding a pitch track within a frame that minimizes the pitch-prediction residual energy over the frame. This approach is also used in this work.

Before explaining the new pitch-period estimation algorithm, it is beneficial to briefly review the pitch-prediction filters, particularly the one-tap version. This type of filters is

commonly used in the AbS coders to eliminate the redundancy resulting from the periodicity in the signal. The idea is to predict a signal from a gain scaled delayed version of the same signal, which can be written as

$$\hat{x}[n] = \sum_{k=-M}^M \lambda_k x[n - \tau + k], \quad (51)$$

where $\hat{x}[n]$ is the predicted signal, $2M + 1$ is the number of taps in the prediction filter, λ_k is the k^{th} scale factor and τ is the pitch period or the optimum pitch lag. Such multi-tap predictors are usually used to compensate for the fractional pitch periods of the speech signal, which can not be handled by a single-tap predictor. However, since the pitch-prediction idea is used for estimating the average pitch period, a single-tap predictor is sufficient for this purpose. To find the scale term of a single-tap pitch-predictor, the pitch-prediction error, defined as

$$\begin{aligned} \varepsilon_\tau &= \sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2 \\ &= \sum_{n=0}^{N-1} (x[n] - \lambda_\tau x[n - \tau])^2, \end{aligned} \quad (52)$$

is minimized with respect to λ_τ . The solution gives λ_τ as

$$\lambda_\tau = \frac{\langle x_0, x_\tau \rangle}{\langle x_\tau, x_\tau \rangle}, \quad (53)$$

and the resulting pitch-prediction error can be written in closed form as

$$\varepsilon_\tau = \left[\langle x_0, x_0 \rangle - \frac{\langle x_0, x_\tau \rangle^2}{\langle x_\tau, x_\tau \rangle} \right] \quad (54)$$

The optimum pitch lag can be found by computing ε_τ for a range of pitch lags and selecting the one with minimum ε_τ .

To find a pitch track of a frame, this approach can be extended to include multiple subframes. In this case, the track's pitch-prediction error, ε_Γ , for the pitch-track, Γ , is defined as

$$\varepsilon_\Gamma = \sum_{s=1}^K \varepsilon_{\tau_s} = \sum_{s=1}^K \left[\langle x_0, x_0 \rangle - \frac{\langle x_0, x_{\tau_s} \rangle^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle} \right], \quad (55)$$

and the minimized pitch-prediction error, ε_{PPE} , can be found as

$$\varepsilon_{PPE} = \min_{\Gamma} \varepsilon_\Gamma, \quad (56)$$

where K is the number of subframes within the frame, τ_s is the pitch period of the s^{th} subframe, and $\langle x_k, x_l \rangle_s$ for the s^{th} subframe is defined as

$$\langle x_k, x_l \rangle_s = \sum_{n=0}^{L_s-1} x_s[n \pm k] x_s[n \pm l], \quad (57)$$

where $x_s[n]$ is the signal in the s^{th} subframe, L_s is the number of samples within one subframe, and the '+' and '-' sign in $x_s[n \pm k]$ and $x_s[n \pm l]$ are used for forward correlation and backward correlation computation, respectively. The minimization of ε_Γ is exactly the same as the maximization of the frame's pitch-track's normalized correlation coefficient, ρ_Γ , defined as

$$\rho_\Gamma^2 = \frac{\sum_{s=1}^K \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s}}{\sum_{s=1}^K \langle x_0, x_0 \rangle_s} = \frac{\sum_{s=1}^K P_s \rho_{\tau_s}^2}{\sum_{s=1}^K P_s}, \quad (58)$$

where P_s is the s^{th} subframe's energy, equal to $\langle x_0, x_0 \rangle_s$, and ρ_{τ_s} is the normalized correlation coefficient of the s^{th} subframe at τ_s , defined as

$$\rho_{\tau_s} = \frac{\langle x_0, x_{\tau_s} \rangle_s}{\sqrt{\langle x_0, x_0 \rangle_s \langle x_{\tau_s}, x_{\tau_s} \rangle_s}}. \quad (59)$$

(The proof showing the equivalence of ε_Γ minimization and ρ_Γ maximization is given in Appendix A.1.)

This pitch-prediction algorithm uses a main analysis window of a length twice the maximum allowed pitch period so that there are enough samples to find the correlation between at least two pitch cycles. Furthermore, the main analysis window is partitioned into two equal-length regions such that the forward correlation and backward correlation computations from (57) are used for the first and second regions, respectively.

Like most other pitch-period estimation algorithm, this algorithm also requires an exhaustive search to find the optimum pitch track. However, (58) implies that the normalized correlation coefficient of a pitch track is the energy weighted mean of the normalized correlation coefficients of the subframes' pitch lags that are independent of each other. So, one way to find the optimum pitch track is to compute the normalized correlation coefficients of all allowed pitch-period lengths in all subframes using (59), and then pick the pitch lag that maximizes the correlation in each subframe. Unfortunately, this method may result in a pitch track with unnatural pitch variations. To correct this problem, a constraint on

the pitch track can be imposed such that the variation of the pitch-period within a frame is bounded by $\pm 15\%$. This constrains τ_s to be between $0.85\tau_{ps-avg}$ and $1.15\tau_{ps-avg}$, where τ_{ps-avg} is the pseudo-average pitch period of the pitch track. After finding the optimum pitch track, the true average pitch period is calculated as

$$\tau_{avg} = \frac{\sum_{s=1}^K P_s \tau_s}{\sum_{s=1}^K P_s}. \quad (60)$$

Finally, a fractional pitch-period is found in the area of the rounded true average pitch period using the formula given in [79] as

$$\Delta = \frac{C_{0,T+1}C_{T,T} - C_{0,T}C_{T,T+1}}{C_{0,T+1}[C_{T,T} - C_{T,T+1}] + C_{0,T}[C_{T+1,T+1} - C_{T,T+1}]}, \quad (61)$$

where $C_{k,l}$ is equal to $\langle x_k, x_l \rangle$, T is equal to $\lfloor \tau_{avg} + 0.5 \rfloor$, and τ_{FR} is equal to $T + \Delta$. The associated normalized correlation coefficient is obtained as

$$\rho_{T+\Delta} = \frac{(1 - \Delta)C_{0,T} + \Delta C_{0,T+1}}{\sqrt{C_{0,0}[(1 - \Delta)^2 C_{T,T} + 2\Delta(1 - \Delta)C_{T,T+1} + \Delta^2 C_{T+1,T+1}]}}, \quad (62)$$

Note that all samples in the analysis window are used in the calculation of the fractional pitch period and its normalized correlation coefficient.

The pitch-period estimation algorithm can be summarized as follows:

1. ρ_{τ_s} is computed at all allowed integer pitch lags in all subframes using (59).
2. Each integer pitch lag in the allowed pitch-period range is set to τ_{ps-avg} , and a pitch track is obtained by selecting the integer pitch lags among the constrained lags that maximize ρ_{τ_s} in each subframe.
3. Each pitch track's normalized correlation coefficient is calculated as given in (58).
4. The pitch track that maximize ρ_F^2 is selected as the optimum pitch track.
5. The true average pitch period for the selected pitch track is computed by (60).
6. The frame's pitch period and its correlation level are calculated by finding the fractional pitch period and its normalized correlation coefficient using (61) and (62), respectively.

This particular algorithm is very effective in finding the exact pitch period, and it is also observed that its performance is better than that of the one used in 2.4 kb/s U.S. military standard MELP coder, especially at onsets. However, since the algorithm is a correlation based technique, it suffers from occasional pitch-doubling/tripling and pitch-halving errors. It is also observed that the algorithm sometimes miscalculates the pitch period when it changes rapidly. To reduce these problems, the following features are added to the algorithm.

As discussed above, the primary reason for the pitch-halving problem is the interaction of one of the pitch harmonics with a narrow-bandwidth formant resonance where the pitch-period estimation algorithm detects the pitch period as a submultiple of the true pitch period. This narrow-bandwidth formant resonance can be eliminated by filtering the speech signal with the inverse of the linear-prediction filter to obtain the residual signal. No pitch-halving problem is observed when this signal is used in this estimation algorithm. On the other hand, the residual signal often becomes very noisy, especially at the end of words, and it often does not contain enough periodicity information in these cases. For this reason, it is beneficial to use both the speech and the residual signals at the same time and combine the results with decision logic. In addition, it is also known that the high-frequency part of the speech spectrum is often noisy. This may also hamper the pitch-period estimation process especially with a correlation based technique. Therefore, a low-pass filter with a cut-off frequency at 1.5 kHz is applied to both the speech and the residual signals before the estimation process.

To deal with the pitch-doubling problem, the algorithm is modified so that it finds the optimum pitch-track as follows: The allowed integer pitch-period range is first partitioned into octave regions, such that the pitch lag in the first entry of a region is twice the pitch lag in the first entry of the previous region. Then, the steps between two and six in the algorithm given above are performed for each octave band, and the fractional pitch period and its normalized correlation coefficient are calculated in each region. These fractional pitch-period values are assigned as pitch-period candidates, and the one with the highest correlation is assigned as the primary candidate. In this algorithm, if the primary pitch

candidate is detected as multiple of another candidate whose correlation level is larger than 90% of the primary pitch candidate, the primary pitch candidate is replaced by the one obtained in the lower octave region.

A simple onset detection mechanism to change the location of the fractional pitch-period analysis is also incorporated into the algorithm to enhance the estimation at onsets. The whole analysis frame is first partitioned into three regions, and the energy of each region is computed. If the energy change between the first and last region is larger than 6 dB, an onset is assigned to the frame, and the window used in fractional pitch-period computation is moved toward the end of the main analysis window, so that more samples from the high energy regions are used. Experimentally, it was observed that this technique reduces the incorrect pitch-period and voicing-degree estimations, especially for high-pitched speakers.

The results obtained from these two signals are combined with a finely tuned decision logic to obtain the frame's pitch period and correlation level. First, the two signals are classified as one of the four groups according to their primary pitch candidates' correlation level. These groups are tabulated in Table 2. The decision logic primarily obtains the frame's pitch period and correlation level from the signal which has the highest correlation level. However, there are few exceptions when this rule is not used: Even when the correlation level of residual signal is higher than that of speech signal, the estimation results from the speech signal is used when the primary pitch candidates of both signals are sufficiently close. Furthermore, when both signals are weakly correlated, or one of the signals is weakly correlated and the other is uncorrelated, the one whose primary pitch candidate is closer to the average pitch period is used. The average pitch period is updated only when the signal is strongly correlated and the gain of the signal is larger than a pre-determined threshold, as in [79]. In the case that both signals have the same degree of correlation, the pitch period and correlation level is obtained from the speech signal. The decision logic also uses a second pitch-doubling elimination algorithm when the signal used is at least moderately correlated. This elimination algorithm is based on the comparison of the correlation degree of the signal at the primary pitch candidate and the pitch candidates found in the lower octave regions. It was observed that almost all of the remaining pitch-doubling errors are

eliminated by this method. This pitch-period estimation algorithm makes very accurate estimation using this logic. The details and pseudo-code of the decision logic can be found in Appendix A.2.

Table 2: Correlation groups according to normalized correlation coefficient of the primary pitch candidate.

$\rho_{T+\Delta}$	Correlation Level
> 0.8	Strong Correlation (SC)
$0.65 < 0.8$	Moderate Correlation (MC)
$0.5 < 0.65$	Weak Correlation (WC)
< 0.5	No Correlation (NC)

As an extension of this algorithm, when the pitch period is estimated more than once in a frame, a smoothing procedure is applied to the pitch-period and voicing-degree estimates:

- If only one estimate is declared voiced among all voicing-degree estimates within a frame, and if it is not the last estimate, it is declared as unvoiced.
- If an unvoiced estimate is detected between two voiced estimates, both their pitch-period values and normalized correlation coefficients are interpolated and substituted in place of the unvoiced estimate.

This procedure is very effective in correcting the estimation errors, especially when the signal is weakly correlated.

The computational complexity of this algorithm is mainly dependent on the maximum allowed integer pitch lag and the total number of allowed integer pitch lags. The computation of the normalized correlation coefficients accounts for 95% of the computational complexity of the whole algorithm. Although, this algorithm is not implemented in fixed-point, the computational complexity of the algorithm is estimated around 12.5 WMOPS (Weighted Million Operations per Second) when the allowed pitch-period range and the frame update rate is the same as 2.4 kb/s U.S. military standard MELP coder.

Experimentally, it was found that the final algorithm estimated the voicing degree and the pitch period correctly almost all the time. The remaining errors mostly occurred at onsets for the low-pitched speakers especially when less than half of the analysis frame

contained the voiced part. Throughout the test sequence, no pitch-halving problem was observed except once where the pitch-period length exceeded the allowed pitch-period range.

3.2 Pitch-Cycle Segmentation

This section describes two high-resolution segmentation algorithms that segment the speech signal into individual pitch cycles. The prediction gain and the normalized correlation between successive pitch cycles are used as the periodicity measures in these algorithms. The prediction gain measure is one of the successful methods used for pitch-cycle boundary detection in literature [36]. In addition, the normalized correlation coefficient is used in pitch-period estimation algorithms as the periodicity and voicing-degree indicators [79], which also makes it a good candidate for a periodicity measure in a pitch-cycle segmentation algorithm.

Since both the CLP method and the CPT of the pitch cycles require exact periodicity, the proposed algorithms obtain the boundaries of the pitch cycles within the speech signal in fractional steps, defined by an upsampling factor of the original signal, N . Both algorithms described here assume that the starting location of the segment is known and the problem is to find the ending location that results in a segment containing only a single pitch cycle of the speech signal.

3.2.1 Pitch-Cycle Segmentation Based on Prediction Gain Maximization

In this method, an arbitrary length of a signal is repeated many times by concatenation and a linear-prediction algorithm is applied to find the predictor coefficients and prediction gain. The predictability of the signal decreases with the increasing discontinuity across the boundaries of the repeated signal, as shown in Figure 5. The prediction gain can be maximized only when the discontinuity across the borders of the segment is minimized. For voiced speech signals, the discontinuity is minimized when the repeated segment is equal to single pitch cycle. This method is also reported to be a good approach when the speech signal is not exactly periodic; this method results in the least discontinuity across its boundaries [36].

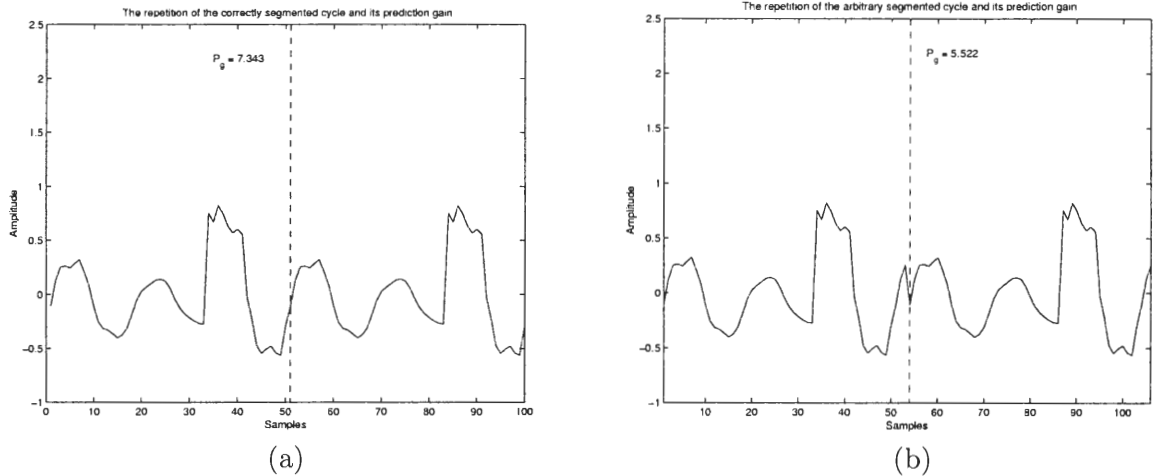


Figure 5: Repetition of the correctly segmented (a) and arbitrary segmented (b) signal. Prediction gains, P_g , for the segments shown in (a) and (b) are 7.343 dB and 5.522 dB, respectively.

Kleijn uses two variations of this method in [36] and [37]. In the first variation, a set of Fourier series coefficients is fitted to the speech signal with the desired length, and autocorrelation coefficients and the resulting prediction gain are computed from these Fourier series coefficients. Since the length of pitch cycles is rarely an integer for a narrowband-critically sampled speech signal, these steps are repeated in fractional steps around an initial integer pitch-period estimate. In the second method, the signal is first upsampled, and all autocorrelation coefficients between zero and the oversampling factor times the prediction order are computed. Then, fractional prediction coefficients and their prediction gains are computed for each fractional pitch candidate. In both methods, the length that maximizes the prediction gain is selected as the cycle length [36]. Kleijn reported that the results obtained by both methods are slightly different. The problem of both methods is the computational complexity: In the first method, the computational complexity increases significantly for long pitch-cycle lengths because of the Fourier series coefficient computation. On the other hand, the number of linear equation sets that must be solved for each fractional pitch-period candidate increases linearly with the fractional resolution of the analysis in the second method.

The new pitch-cycle segmentation algorithm generates results similar to those in Kleijn's first method, without the necessity of the Fourier series coefficients computation. As will

be discussed in Chapter 4, the CLP analysis is exactly the same as the generation of the predictor coefficients from the Fourier series coefficients for infinitely periodic signals. Therefore, the signal is first upsampled by a factor of N , and then the CLP analysis is performed to obtain the predictor coefficients and the associated predictor gain for all fractional pitch-cycle length candidates around the initial pitch-period estimate. The cycle length that maximizes the predictor gain is selected as the length of the pitch cycle. Using this approach, there is no need to compute the Fourier series coefficients or to solve the many linear equations required in Kleijn's second method. The only difference that may result in different results than in Kleijn's first method is the non-ideal nature of the low-pass interpolation filter applied after the upsampling of the signal.

The algorithm itself is usually very efficient for finding the correct pitch-cycle boundaries. However, any mistake also results in incorrect segmentation locations in the subsequent pitch cycles. In performance experiments, it was observed that the energy changes in the signal and rapidly decaying impulse responses result in incorrect segmentation locations. To correct those problems, two more criteria are included in the algorithm:

1. The first criterion deals with the energy changes within the signal. In such cases, when a pitch cycle starts at a location whose amplitude is not close to zero, the algorithm usually finds the end location whose amplitude is also close to that of the initial location, but not the cycle's exact boundary location, as illustrated in Figure 6. To correct this problem, the cycle boundaries are forced to be close to zero-crossing locations. Fortunately, when this criterion is forced once at the onsets, the algorithm naturally finds the rest of the cycle boundaries close to zero crossings. Onset handling is discussed later in this section.
2. When the impulse response dies rapidly within the pitch cycle before the cycle reaches the end, the prediction gain of the shorter than correct fractional cycle lengths may be larger than that of the correct cycle length. To correct this problem, a weighting function is included such that longer cycle lengths are favored over the shorter ones. Since prediction gain usually drops significantly for the cycle lengths that are longer

than the correct one, the weighted prediction gain of the correct cycle length is still greater than that of the longer cycle lengths. As a result, the weighting function does not increase the probability of selecting a cycle length longer than the correct one.

The onsets, detected by the pitch-period estimation algorithm described in this chapter, are handled with a special logic in this pitch-cycle segmentation algorithm: At first, the pitch-cycle boundaries are estimated with the algorithm as usual. Then, the pitch cycle with the highest energy is detected. Within this cycle, some number of peak locations with the greatest absolute amplitudes is found, and the widths of the peaks - the distance between the two zero-crossing locations that include the peak - are calculated. A score is assigned to each peak according to its amplitude and width, where the peak with the largest amplitude and width gets the largest score. The zero-crossing location just before the peak with the largest score is assigned as the initial point of the pitch cycle. The rest of the cycle boundaries within the frame are obtained by forward and backward processing of the algorithm described above. Furthermore, the frame's first cycle's initial location, obtained by backward processing, is set to the previous frame's last pitch cycle's end location. Finally, when the resulting first pitch-cycle length is less than the minimum allowed pitch-period length, the pitch cycle is merged with the next pitch cycle. Since this segment is usually

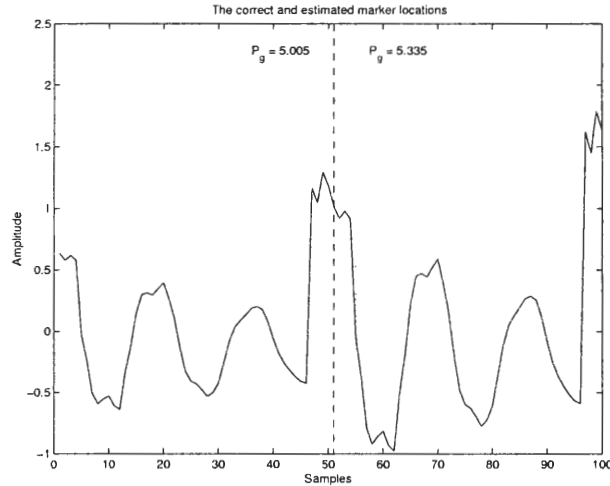


Figure 6: Illustration of the problem in pitch-cycle segmentation when signal energy is increasing. Because of the changing energy, the prediction gain is not maximum in the correct end location, marked with the dashed line.

in the silence or unvoiced sections of the speech signal, these modifications do not degrade the algorithm's performance.

These two added criteria together with the onset handling result in a very effective pitch-cycle segmentation algorithm. Most of the segmentation errors are eliminated. Also, the computational complexity is reduced compared to previous implementations of this idea. However, the computational complexity of this algorithm is still very high. Experimentally it was found that the search range around the initial pitch-period estimate must be between ± 3 samples in 0.1 sample steps to track rapid pitch changes in transition regions and to have enough segmentation accuracy for the subsequent pitch-synchronous algorithms. This requirement results in 61 possible end locations, for which a different linear-equation set must be solved. Fortunately, it is possible to compute the correlation coefficients recursively. It is estimated that the computational complexity of a fixed-point implementation is between 11.2 and 2.2 WMOPS depending on the initial pitch period between 20 and 160 samples, respectively. As solving the linear equations accounts for most of the computational complexity, the overall complexity is much larger for shorter pitch-period lengths. In a typical speech signal having average pitch-period between 40 and 60 samples, the overall computational complexity of the algorithm is between 6 and 4.2 WMOPS, respectively.

3.2.2 Pitch-Cycle Segmentation Based on Normalized Correlation Maximization

The pitch-period estimation algorithm described in the previous section and the pitch-period estimation algorithm in the 2.4 kb/s military standard MELP coder described in [79] both use the normalized correlation function to find the voicing-degree and associated pitch period of the speech signal. In both algorithms, the correlation values are calculated from a fixed length of signal. To use this measure in a pitch-cycle segmentation algorithm, the definition of the normalized correlation coefficient is modified slightly as

$$\rho_\tau = \frac{\langle x_0, x_\tau \rangle_\tau}{\sqrt{\langle x_0, x_0 \rangle_\tau \langle x_\tau, x_\tau \rangle_\tau}}, \quad (63)$$

where the correlation term, $\langle x_k, x_l \rangle_\tau$, is calculated as

$$\langle x_k, x_l \rangle_\tau = \sum_{n=0}^{\tau-1} x[n+k]x[n+l], \quad (64)$$

where τ is the length of the segment used to calculate this measure. Note that this modification synchronizes the length of the segment used in calculating the measure to the lag whose normalized correlation is calculated. Ideally, this measure has the maximum value only when τ is equal to the pitch-cycle length in a periodic signal.

As described in the pitch-period estimation algorithm, this measure is robust to energy changes in the signal. Furthermore, a small amount of noise does not affect the normalized correlation function significantly. Finally, this method does not require any special starting location as long as the signal shape does not change significantly from one cycle to the next. However, this method requires at least two complete pitch cycles for reliable segmentation. As a comparison, the pitch-cycle segmentation algorithm based on the prediction gain maximization does not have this requirement.

Since the residual signal does not contain the spectral shaping effects of the vocal tract, the correlation computation is not affected by the spectral changes in the signal. Because of this, this segmentation algorithm mainly uses the residual signal. However, as the residual signal has an (almost) flat spectrum, the noise in the high-frequency region of the residual spectrum reduces the correlation of the signal, sometimes significantly. To reduce the effect of the noise, the residual signal is also filtered with a low-pass filter with a cut-off frequency of 1.5 kHz before it is used in this algorithm. As stated above, this algorithm assumes that the starting location of a pitch cycle is known. For this starting location, an initial pitch period, τ_i , is calculated by interpolating the pitch-period estimates of the previous and current frames. Then, (63) is used to calculate ρ_τ for the pitch lags between $\tau_i - \Delta_D$ and $\tau_i + \Delta_D$, and the pitch lag with the largest ρ_τ is selected as the integer length of the pitch cycle, τ_c . In addition, if the maximum correlation occurs at $\tau_i \pm \Delta_D$, the τ_i is modified as $\tau_i = \tau_i \pm \Delta_D$, and the procedure is repeated again. The modification of the initial pitch period allows the tracking of rapid pitch-cycle length changes, especially at onsets. Experimentally, it is observed that the maximum normalized correlation value drops significantly at the end of words and in transition regions for high pitch-speakers. To address these problems, when ρ_{τ_c} is found to be less than 0.9, these steps are repeated using a low-pass filtered speech signal. The cut-off frequency of the low-pass filter is the

same as the one used to filter the residual signal. Finally, when ρ_{τ_c} is greater than 0.8, a fine-search is performed on the N times upsampled signal using the same measure. However this time, the fractional cycle lengths between $\tau_c - \Delta_d$ and $\tau_c + \Delta_d$ are searched in $\frac{1}{N}$ steps. Experimentally, it was found that the accuracy of the algorithm is exceptional when Δ_D and Δ_d are set to eight and two samples, respectively. It was also observed that ρ_{τ_c} obtained from the speech signal becomes less than 0.8 only when the pitch cycle is at the end of word (voiced to silence transition) or in the middle of a rapid transition region (voiced to voiced), especially for high-pitched speakers. To eliminate potential problems for these cases, the algorithm does not stop the segmentation process for an additional two pitch cycles. For the rapid transition case, ρ_{τ_c} becomes greater than 0.8 in almost all test cases after two cycles. As a result, the algorithm continues segmenting the speech signal without interruption.

This algorithm segments only voiced speech signals as there is no periodicity in the unvoiced sections. For this reason, detecting the onsets becomes very important in order not to miss capturing any pitch cycles. The segmentation algorithm does not completely rely on the pitch-period estimation algorithm for onset detection. In experiments, it was observed that the pitch-period estimation algorithm sometimes fails to detect the onsets, especially for low-pitched speakers. Therefore, the algorithm uses its own onset detection mechanism when the pitch-period estimation algorithm indicates an onset or when the energy of the signal is increased by 6 dB from the previous unvoiced frame to the current frame.

The onset detection algorithm first segments the speech signal blindly in both the current and the next frame (the onset detection method requires a frame delay.) The segmentation algorithm starts from the end of the current frame, and segments the current frame by going backwards and segments the next frame by going forward. Furthermore, these two frames are segmented multiple times using each integer pitch-period length between 20 and 160 samples as the initial pitch period. This blind segmentation can also be seen as finding a pitch track for each initial pitch period, since the segmentation algorithm finds the length of pitch cycles as well as the segmentation locations. In this onset detection, the normalized correlation of the pitch track, ρ_Γ , is also calculated just as it is done for

the pitch-period estimation algorithm (i.e. the energy weighted mean of the normalized correlation coefficients of the pitch cycles.) Finally, the pitch period is again estimated as the energy weighted mean of the pitch-cycle lengths in the pitch track with the largest track's normalized correlation.

This blind segmentation algorithm is applied to both the low-pass filtered speech and the low-pass filtered residual signals. For each signal, a pitch track, its average pitch period and its correlation are obtained. In the final segmentation, the average pitch period estimated from the speech signal is used unless the voicing-strength of the speech signal estimated in the pitch-period estimation algorithm is very high (i.e. greater than 0.9.) In this case, the pitch-period estimate from the pitch-period estimation algorithm is used. Furthermore, when the average pitch-period estimates obtained from the speech and residual signals are different, the pitch period and normalized correlation of the pitch-track estimates from the residual signal is used only when the following conditions are met:

- When the ρ_{Γ} estimate obtained from the residual signal is moderately or strongly correlated but the ρ_{Γ} estimate obtained from the speech signal is weakly or non-correlated.
- When the ρ_{Γ} estimate obtained from the residual signal is moderately or strongly correlated, the ρ_{Γ} estimate obtained from the speech signal is moderately correlated, and ρ_{Γ} estimate obtained from the speech signal for the pitch track whose initial pitch period is the same as the pitch-period estimate obtained from the residual signal is high.

Finally, when the correlation level of the signal is greater than 0.7 or the voicing-strength of the signal obtained in the pitch-period estimation algorithm is greater than 0.9, an onset is detected. However, if the first pitch cycle with high correlation is after the middle of the next frame, the segmentation process is deferred until the next frame. Otherwise, the residual signal is searched for a suitable place for the initial location of the segmentation. Since, this algorithm was initially designed for using in a pitch-cycle modification algorithm, low-energy sections of the residual signal are preferred as the segmentation locations. Once

this initial location is determined, the speech signal in the current frame is segmented using the algorithm described above.

In the experiments, it was observed that this algorithm is very accurate when N is set to 10. Unlike the previous algorithm, no segmentation mistake was observed. There is only a rare case in which the algorithm stops segmenting when there is a rapid transition and a large energy variation in a voiced segment of the speech signal. However, the “two-cycle defer” rule eliminates almost all of these problems. The onset detection mechanism also proved to be very effective even for low-pitched speakers. The segment locations found by this algorithm is suitable for making pitch-cycle modifications. In addition, with a slight modification to boundary locations of the pitch cycles, it can also be used with other algorithms such as the CLP analysis.

The computational complexity of the algorithm is between 4 and 3.6 WMOPS depending on the length of the pitch period. As most of the computational complexity is from the computation of the normalized correlation coefficients, the computation complexity is similar for all pitch-period range. Furthermore, the correlation functions $\langle x_0, x_0 \rangle_\tau$ and $\langle x_\tau, x_\tau \rangle_\tau$ can be computed recursively for each candidate pitch-cycle lengths. The (almost) fixed computational-complexity also makes this algorithm a good candidate to be used in a speech coder.

CHAPTER IV

CIRCULAR LINEAR PREDICTION MODELING

The idea of circular linear prediction (CLP) modeling originated from a special case in linear prediction. This special case occurs only when the input signal is an infinite periodic signal, and in this case, well-known linear-prediction methods find the same set of prediction coefficients. In CLP modeling, each individual pitch cycle in the speech signal is represented by an infinite signal, and this signal is used to compute the correlation coefficients. For this special case, the autocorrelation method using an infinite window and the covariance method using only one cycle of this periodic signal both generate identical sets of coefficients. Therefore, the CLP modeling marries the more accurate modeling properties of the covariance method with the stability guarantee property of the autocorrelation method. The quasi-periodic nature of the voiced speech makes this modeling technique a reasonable candidate for speech modeling. This method can also be used with sufficiently long unvoiced speech and properly segmented stop consonants as well.

The CLP modeling method was first introduced by Barnwell as an alternative to the autocorrelation and the covariance method in the late '70s [6, 7]. In the initial experiments, the CLP method was found to be effective when compared to the autocorrelation method, while using fewer samples for analysis. Although the initial results were encouraging, this method did not become popular because of its dependency on high-quality pitch-period detectors and pitch-cycle segmentation algorithms.

Although the CLP algorithm assumes that the input signal is an infinite signal, it is possible to obtain the same results by making the analysis using only one cycle using circular operations. Throughout this thesis, the term *circular signal* is used to refer one cycle of an infinite signal.

4.1 CLP Analysis

In CLP analysis, each individual pitch cycle is first repeated to generate an infinite signal with identical cycles so that

$$x[n] = x[n + k\tau], \quad (65)$$

where k is an integer number and τ is the length of one cycle. Since the CLP method is a linear prediction method, the predicted signal is defined the same as in (8). Therefore, the predicted signal is a linear combination of the delayed versions of the original signal, and hence, the predicted signal is also an infinite signal with identical cycles.

As in the case for all LP methods, the sum of the squared error between the original and the predicted signals is also minimized in CLP analysis. Since the signal is infinite, the boundaries of the summation are set to $-\infty$ and ∞ . As a result, the sum of the squared error is written as

$$\varepsilon_{CLP} = \sum_{n=-\infty}^{\infty} \left(x[n] - \sum_{k=1}^p a_k x[n-k] \right)^2. \quad (66)$$

To find the optimum predictor coefficients, first the partial derivative of ε_{CLP} with respect to predictor coefficients must be computed and set to zero as

$$\begin{aligned} \frac{\partial \varepsilon_{LP}}{\partial a_l} &= -2 \sum_{n=-\infty}^{\infty} x[n-l] \left(x[n] - \sum_{k=1}^p a_k x[n-k] \right) = 0 \\ &\quad -2 \sum_{n=-\infty}^{\infty} x[n-l] e[n] = 0 \quad l = 1, \dots, p. \end{aligned} \quad (67)$$

Since $x[n]$ is an infinite signal, it is impossible to evaluate (67). However, since both $x[n-l]$ for all l and $e[n]$ are periodic functions with the same period, their multiplication is also a periodic function with the same period. Therefore, (67) can be written as

$$\frac{\partial \varepsilon_{LP}}{\partial a_l} = -2 \lim_{N \rightarrow \infty} N \sum_{n=0}^{\tau-1} x[n-l] \left(x[n] - \sum_{k=1}^p a_k x[n-k] \right) = 0. \quad (68)$$

This equation can be satisfied only when

$$\sum_{n=0}^{\tau-1} x[n-l] \left(x[n] - \sum_{k=1}^p a_k x[n-k] \right) = 0. \quad (69)$$

As a result, minimizing the sum of the squared error of a infinitely periodic signal is equivalent to minimizing the sum of the squared error in only one pitch cycle of this signal.

When the correlation function is defined as in (10), (69) can be rewritten as

$$\sum_{k=1}^p a_k r(k, l) = r(0, l) \quad l = 1, \dots, p. \quad (70)$$

In addition, since only one period of this infinite signal is used to compute the correlation coefficients, any offset added to both k and l does not change the result:

$$r(k, l) = r(k - \Delta, l - \Delta) = \sum_{n=0}^{\tau-1} x[n - k + \Delta] x[n - l + \Delta]. \quad (71)$$

Therefore, as in the case of autocorrelation method, the correlation coefficients, $r(k, l)$, can be written as $r(|k - l|)$ such that

$$r(k, l) = r(|k - l|) = r(i) = \sum_{n=0}^{\tau-1} x[n] x[n \pm i], \quad (72)$$

Finally, since all cycles are identical to each other, (72) can be written as

$$r(i) = \sum_{n=0}^{\tau-1} x[n] x[(n \pm i)_\tau], \quad (73)$$

where $((\cdot))_Q$ denotes the modulo Q operation.

As in the case with the autocorrelation method, \mathbf{R} in (12) also has a Toeplitz structure when obtained by the CLP analysis. Therefore, the resulting equations can be solved by the Levinson-Durbin recursion, and the stability of the resulting prediction filter is always guaranteed. Avoiding the use of a tapered window also eliminates the associated distortion that occurs mainly for low-pitched speakers. Since the prediction filter can be estimated using significantly fewer samples compared to the autocorrelation method, the CLP method is often more suitable for analyzing the speech signal, especially in transition regions, where the stationary assumption holds only for a short segment of the signal.

The CLP analysis can be interpreted as an autocorrelation analysis with an infinite window or a covariance analysis using $\tau + p$ samples of this infinite signal, where p is the prediction order. In addition, the CLP method is also equivalent to the following common linear-prediction techniques when they model an infinitely periodic signal:

- Modified covariance method.
- Lattice-filter modeling techniques like forward covariance, backward covariance and Burg's method.

- Discrete-spectra linear prediction modeling.

The proof showing the equivalence of these methods to the CLP modeling can be found in Appendix C.

Although the CLP analysis is a good model for periodic signals, it can also be used on unvoiced speech by choosing sufficiently long stationary segment as will be discussed in Section 4.3.

4.2 CLP Synthesis

The circular LP synthesis is performed to generate one period of a speech signal using the predictor coefficients and the circular residual signal, $e[n]$, extracted by applying the inverse of the prediction filter, $A(z)$, to the input circular signal:

$$e[n] = x[n] - \sum_{k=1}^p a_k x[(n-k)\tau] \quad n = 0, \dots, \tau - 1. \quad (74)$$

The synthesis process can be written as the inverse of (74):

$$\hat{x}[n] = \hat{e}[n] + \sum_{k=1}^p a_k \hat{x}[(n-k)\tau] \quad n = 0, \dots, \tau - 1, \quad (75)$$

where $\hat{e}[n]$ is the processed (quantized) residual signal and $\hat{x}[n]$ is the synthesized speech signal.

Although this method can synthesize individual pitch cycles, a problem associated with the lack of the initial conditions required by the all-pole filter must be addressed. This problem can be solved in three different ways [70, 17].

In the first method, the last p output samples required by the all-pole filter is set to zero, and the pitch cycle is synthesized by filtering the residual signal with the all-pole filter. Then, the last p samples of the newly generated speech signal are used as the initial condition for the next iteration. This iterative approach continues until the synthesized speech reaches a steady state or the number of iterations reaches a preset limit. In other words, this method is exactly the same as circularly filtering the circular residual signal many times with the all-pole filter with the initial conditions set to zero.

The second method is virtually identical to the first except that the initial conditions are obtained from the previously synthesized pitch cycle. This method is particularly useful

for real speech, since the vocal-tract filter usually changes slowly between successive pitch cycles. Because of the better initial conditions compared to the first method, this method converges faster than the first method.

The third method requires generation of a FIR filter from the impulse response of the all-pole filter as follows:

$$h_I[n] = G\delta[n] - \sum_{k=1}^p a_k h_I[n-k], \quad (76)$$

where $h_I[n]$ is the impulse response that is equal to zero when n is negative, and G is used to match the gain of the synthesized speech to the original. To synthesize the speech, this FIR filter is applied circularly to the circular residual signal as follows:

$$\hat{x}[n] = \sum_{k=0}^{L_I-1} h_I[k] \hat{e}[(n-k)_\tau] \quad n = 0, \dots, \tau-1, \quad (77)$$

where L_I is the length of the impulse response. In addition, by taking into account the circular indexing of $\hat{e}[n]$ in (77), $h_I[n]$ can be wrapped around itself with a period of τ and this new filter can be used to decrease the computational complexity in (77):

$$\tilde{h}_I[n] = \sum_{k=0}^{\lceil \frac{L_I}{\tau} \rceil} \hat{h}_I[n+k\tau] \quad n = 0, \dots, \tau-1, \quad (78)$$

where $\hat{h}_I[n]$ is obtained by padding zeros at the end of the $h_I[n]$ so that the length of the new filter is equal to an integer multiple of τ . The length of the FIR filter, L_I , can be selected in two ways:

1. An arbitrary integer multiple of τ .
2. An integer multiple of τ that satisfies the condition that the ratio between the energy of $h_I[n]$ within the first and the last τ samples is greater than a predetermined threshold.

The integer multiple in the first length selection method makes this (third) method similar to the first synthesis method; the integer multiple corresponds to the number of iterations in the first method. As a result, the performance of both methods are very similar when the number of iterations in the first method is the same as the integer multiple in the third method. The second length selection method is an adaptive method that selects the length

of the synthesis filter so as to achieve a uniform performance across different synthesis filter and cycle length combinations. Thus, it is beneficial to use the second length selection method where uniform performance is crucial.

A detailed performance evaluation of each of these three synthesis methods is given in the next section.

4.3 Performance Analysis of CLP Modeling

In this section, the performances of CLP analysis and synthesis under different conditions are evaluated. In these tests, synthetic speech is used to provide objective estimates of how well the CLP analysis can estimate the all-pole filter and how well the CLP synthesis generates the synthetic speech. In these tests, the CLP analysis method is compared to the autocorrelation method, since the autocorrelation method is the technique of choice in almost all state-of-the-art speech coders. The tests include various conditions including purely-voiced speech, partially-voiced speech, unvoiced speech and noisy speech conditions. The spectral estimation performance of both methods is evaluated. In the CLP synthesis performance tests, the SNR of the synthesized speech signal is computed for different parameters for each synthesis method. For objective comparisons, only purely-voiced synthetic speech signals are used in these tests.

4.3.1 Test Setup

The synthetic speech signals used in these tests are generated by filtering an excitation signal with a 10^{th} order all-pole filter. To simulate different artificial vocal-tract configurations, the tests were repeated for 5247 different combinations of the first and second formant locations and bandwidths, which are tabulated in Table 3. The excitation signal in these tests is either a zero-mean periodic impulse train with known period length or a mixture of a low-pass filtered periodic impulse train and a high-pass filtered white noise signal. The experiments were repeated for pitch-cycle lengths between 20 and 160 samples for all vocal-tract filter configurations. The sampling rate of the synthetic speech was 8 kHz.

Table 3: Formant locations and bandwidths used in the synthetic speech experiments.

First formant location (Hz)	250, 350, 450, 550, 650, 750
First formant bandwidth (Hz)	30, 40, 50, 60, 70, 90, 110, 140, 160
Second formant location (Hz)	800, 900, 1000, 1100, 1200, 1400, 1600, 1900, 2200
Second formant bandwidth (Hz)	30, 40, 50, 60, 70, 80, 90, 110, 140, 170, 200

4.3.2 Performance Evaluation of CLP Analysis

The performances of the autocorrelation and the CLP methods were evaluated using purely-voiced, unvoiced, partially-voiced and noisy speech signals. The performance evaluation of both methods using these different types of signals provides insights on how well the two methods will perform on real-speech signals. In all these tests, the autocorrelation analysis used 200 samples of the synthetic speech signal unless stated otherwise, and a Hamming window is multiplied with the signal prior to the analysis. The CLP method is always performed on single pitch cycle with the analysis region length set to the known pitch-period length. In these spectral estimation performance tests, since the synthetic signals are zero mean, only the frequencies between $\frac{4000}{2\tau}$ and 4000 Hz are used to compute the average spectral mismatch and to obtain the maximum spectral mismatch.

In this section, two particular speech spectra are used to illustrate the overall results. The first speech spectrum is used to illustrate the performance of the methods for a spectrum with uniformly distributed formants and with formants whose bandwidths are not too narrow. In this example, the first and second formant locations are 450 and 1200 Hz, and the bandwidths of these formants are 50 and 80 Hz, respectively. The spectrum of this example is shown in Figure 7a and this example is referred as *the speech spectrum with uniformly spread formants* throughout this chapter. The second speech spectrum is used to illustrate the performance of the methods when the spectrum has closely packed formants and a first formant with narrow bandwidth, and when the period length of the signal is short (i.e. < 30 samples.) In this second example, the first and second formants are located at 350 and 800 Hz, respectively, and the bandwidths of these formants are 30 and 70 Hz. The spectrum of this second example is shown in Figure 7b, and this example is referred as *the speech spectrum with grouped formants*. In both spectra, the third and fourth formants

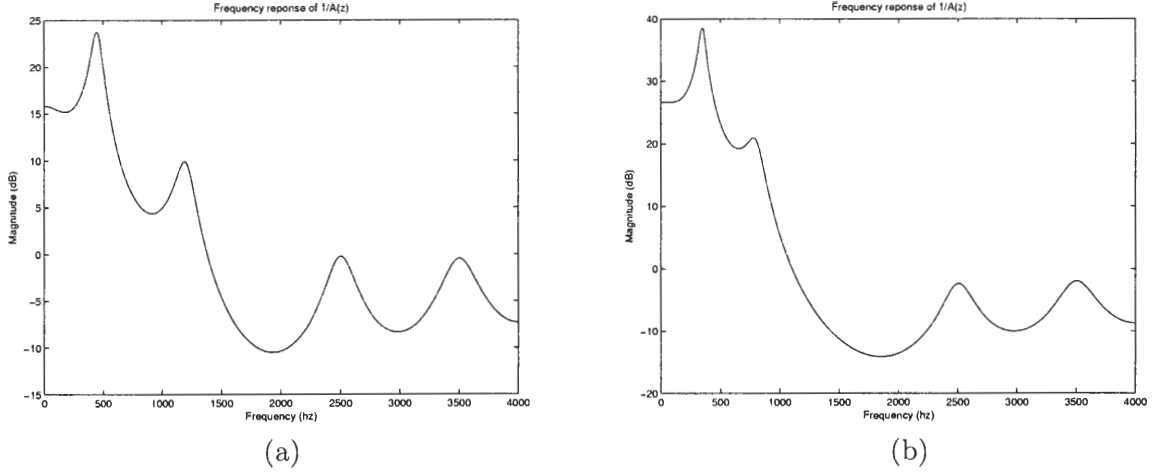


Figure 7: The speech spectra with uniformly spread formants (a) and with grouped formants (b).

are located at 2500 and 3500 Hz with bandwidths of 120 and 150 Hz, respectively.

In the first test, the performances of the CLP analysis and the autocorrelation analysis were compared using fully-voiced synthetic speech signals. The excitation signal in this test was a zero-mean impulse train sequence with known period length. The performance of the CLP analysis and autocorrelation analysis are compared in terms of average spectral mismatch and maximum spectral mismatch. The maximum spectral mismatch was obtained as the maximum of the absolute spectral mismatch between the true and the estimated spectra at all frequencies. For completeness, the autocorrelation analysis was repeated for every possible window location to eliminate the performance dependency on window placement. The minimum, maximum and the mean of the average spectral mismatch of all window locations were calculated for the autocorrelation method and used in the performance evaluation. For the CLP analysis, the analysis frame was set to the known pitch period.

In this experiment, it was observed that the performance of the autocorrelation analysis in all vocal-tract configurations depended heavily on the window location when the number of samples in one cycle was larger than about 80 samples. In such cases, the window length is simply not long enough, and the distortion associated with the convolution of the window spectrum and the speech spectrum increases with increasing cycle length. Both the

average and the maximum spectral mismatches obtained by CLP analysis are always better than those of the autocorrelation method, and the mismatches monotonically decrease with increasing cycle length. When the formants are spread uniformly in the spectrum, the performance of the two methods is almost identical so long as the pitch cycle length is less than 80 samples. This is a particularly important result because it shows that it is possible to capture the speech spectrum from significantly fewer samples using the CLP analysis as compared to the autocorrelation analysis. This result is illustrated in Figure 8, which shows the average spectral mismatch and maximum spectral mismatch obtained by both methods for the pitch-cycle lengths between 20 and 160 samples. The spectrum of the signal in this example is the speech spectrum with uniformly spread formants. The true spectrum and the estimated spectra obtained by both the CLP method and the autocorrelation method and the absolute estimation mismatch between the true spectrum and estimated spectra are shown in Figure 9a and Figure 9b, respectively, when the pitch-period length is 60 samples.

When the formants in the speech spectrum are closely packed, one of the formants has a very narrow bandwidth, and the period length of the signal is less than 30 samples, it was observed that the estimation performances of both methods were very bad. This observation is illustrated in Figure 10 in which the speech spectrum with grouped formant locations example is used. In this example, the spectral mismatch even becomes as high as 26 dB at one particular frequency, around the first formant. Unfortunately, although the prediction gain is always higher for the CLP method, both the average and the maximum spectral mismatch are larger than those of the autocorrelation method. The true spectrum and the estimated spectra obtained by both the CLP and the autocorrelation methods, and the absolute estimation mismatch between the true spectrum and the estimated spectra for the same vocal-tract configuration are shown in Figure 11a and Figure 11b, respectively. The pitch-period length is 25 samples in this example. In the autocorrelation analysis, the convolution of the window's spectrum with the speech spectrum spreads the energy of the harmonics to the neighboring frequencies and results in a spectral estimation with a first formant with slightly wider bandwidth. Possible solutions for this problem are presented in Section 4.5. In addition to these two observations, it was also observed that both average and

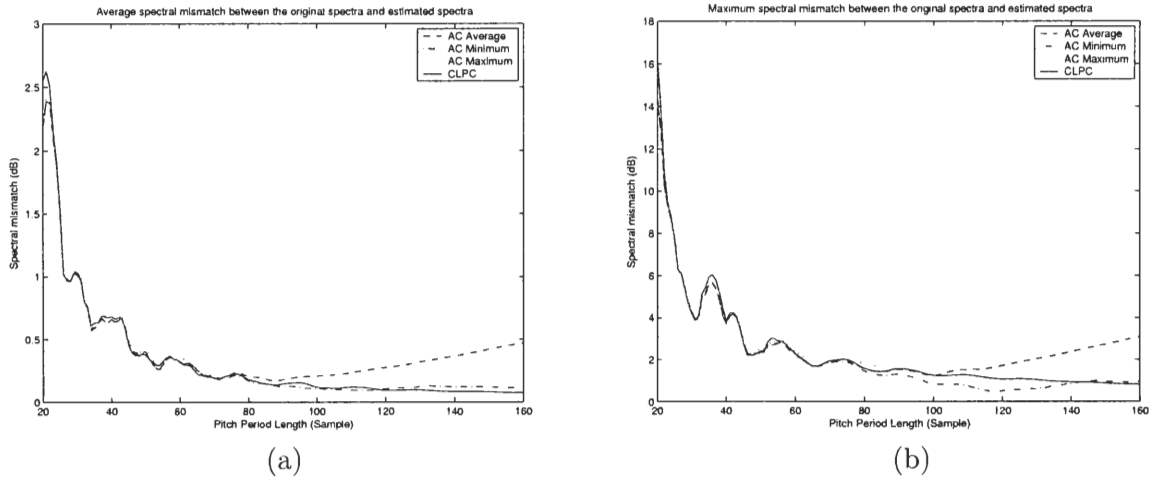


Figure 8: The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples. The speech spectrum is the “uniformly spread formants” example.

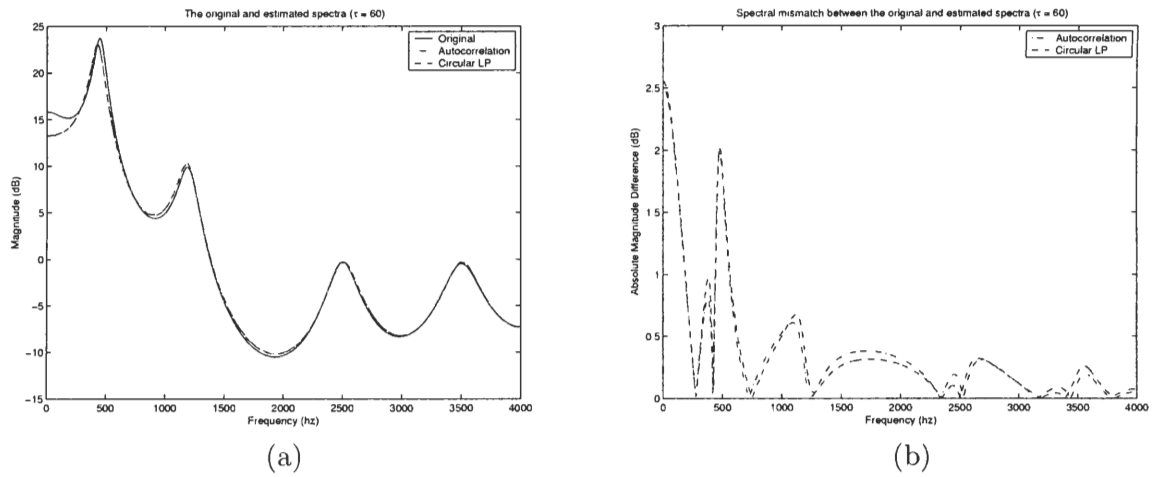


Figure 9: The true spectrum with uniformly spread formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch-period is 60 samples (a), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples (b).

maximum spectral mismatch increases when the frequency of one of the signal harmonics is very close to one of the first two formant locations. Furthermore, in these cases, the mismatch usually increases even more with a decrease in the bandwidth of the formants.

To evaluate the performance of both methods for unvoiced speech, a synthetic speech signal was generated by filtering a white-noise sequence with a 10^{th} order all-pole filter whose frequency response is similar to that of a fricative sound, as shown in Figure 12. Since there is no periodic component in this signal, the period length in the CLP analysis is called as *analysis frame length* for this experiment. Since it is not possible to achieve a reliable performance evaluation from a single analysis frame due to noise, the experiment was repeated 100 times for different frames and the mean and the standard deviation of the average spectral mismatch in these 100 frames were calculated for both autocorrelation and CLP analysis methods. These two performance metrics were used to obtain average performance of both methods and their performance variations in these 100 frames. This procedure was then repeated for all analysis frame lengths between 20 and 160 samples for the CLP method. The autocorrelation method was evaluated twice using different numbers of samples in the analysis frame. In the first case, 200 samples were used as was done for the purely-voiced synthetic-speech signal tests. In the second case, the autocorrelation analysis was performed using 100 samples to illustrate the effect of the analysis window length on the performance.

The mean and the standard deviation of the average spectral mismatch between the true spectrum and the estimated spectra by both methods are shown in Figure 13a and Figure 13b, respectively. The results of the autocorrelation method using 200 and 100 sample analysis windows are illustrated by the dashed-dotted and dashed lines in these figures, respectively. Despite the periodicity assumption in the CLP method, the spectral estimation performance of the CLP method is always better than that of the autocorrelation method when the analysis frame length is the same for both methods. The performance of CLP method using a 110-sample analysis frame matches the performance of the autocorrelation method using 200-sample window. Furthermore, the mean of the average spectral mismatch

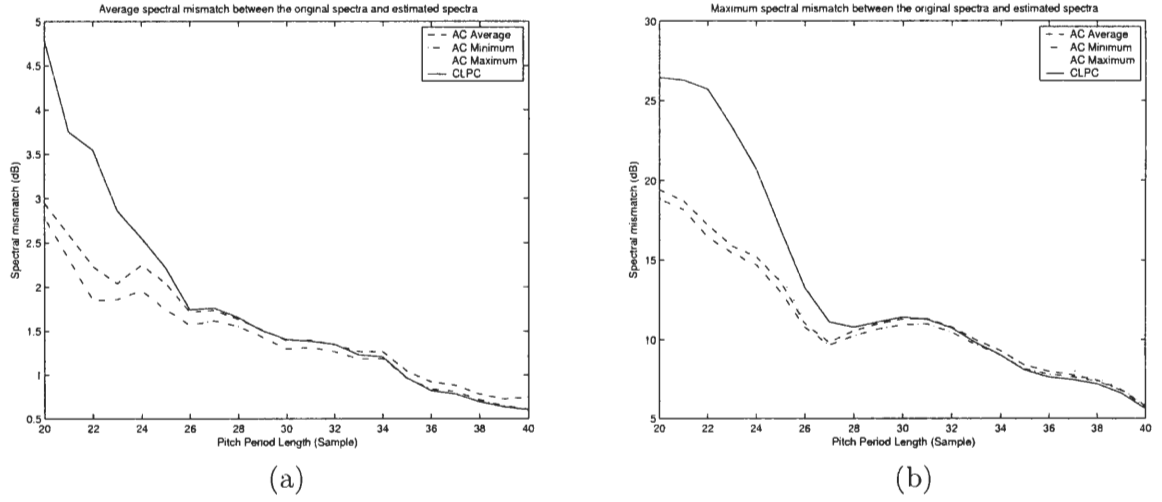


Figure 10: The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 40 samples. The speech spectrum is the “grouped formants” example.

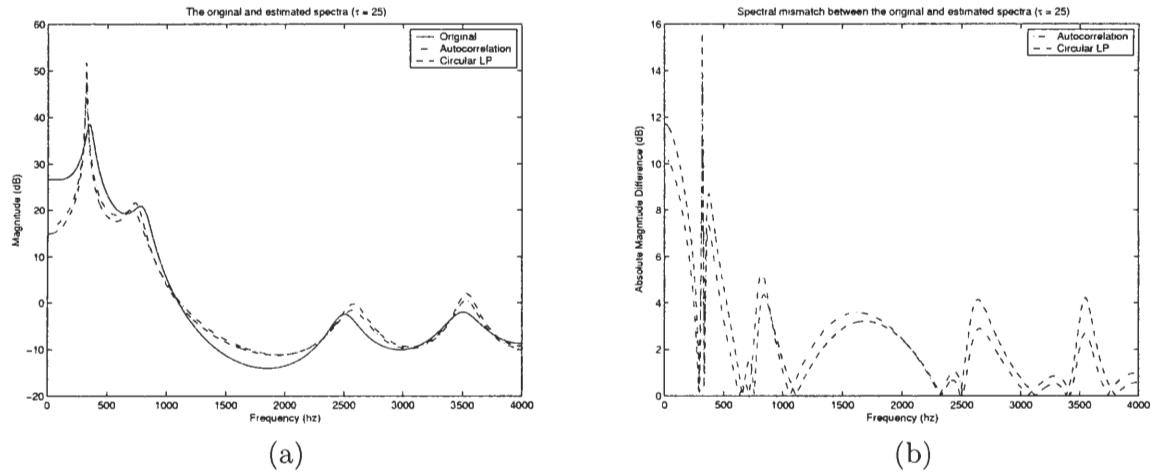


Figure 11: The true spectrum with grouped formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch-period is 25 samples (a), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 25 samples (b).

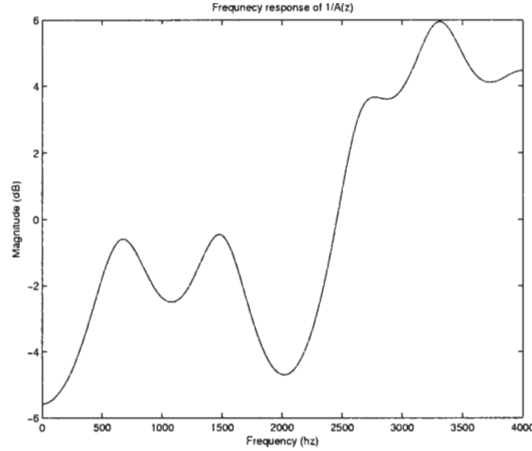


Figure 12: The speech spectrum used in the performance tests for synthetic unvoiced speech.

increases by 0.5 dB when the analysis window is reduced to 100 samples in the autocorrelation method. In addition, the standard deviation in the average spectral mismatch decreases with increasing number of samples in the analysis frame for the CLP analysis method. That is also an indicator to a uniform performance over these 100 frames. These results prove that it is possible to estimate the linear-prediction coefficients with the CLP method reliably using about half number of the samples required for the autocorrelation method. However, it is also evident that for short analysis frames, the mean and the standard deviation of the average spectral mismatch are very high. As a result, the CLP and the autocorrelation methods are both unsuitable for estimation of the linear-prediction coefficients from short analysis frames.

In the third test, the performances of the CLP and the autocorrelation methods were compared with partially-voiced synthetic-speech signals. Since real-speech signals are usually partially voiced, the results of these tests are a good indicator for the performance of the methods on real-speech signals. In this test, the excitation signal was generated by the summation of low-pass filtered impulse train and high-pass filtered white-noise sequences. The low-pass and high-pass filters were complementary filters with the same cut-off frequency, f_c , and summation of both filters resulted in an all-pass filter with constant linear phase. In other words, the generated speech signal's spectrum consisted of two bands in which the low-frequency band and the high-frequency band had harmonic and noise structures,

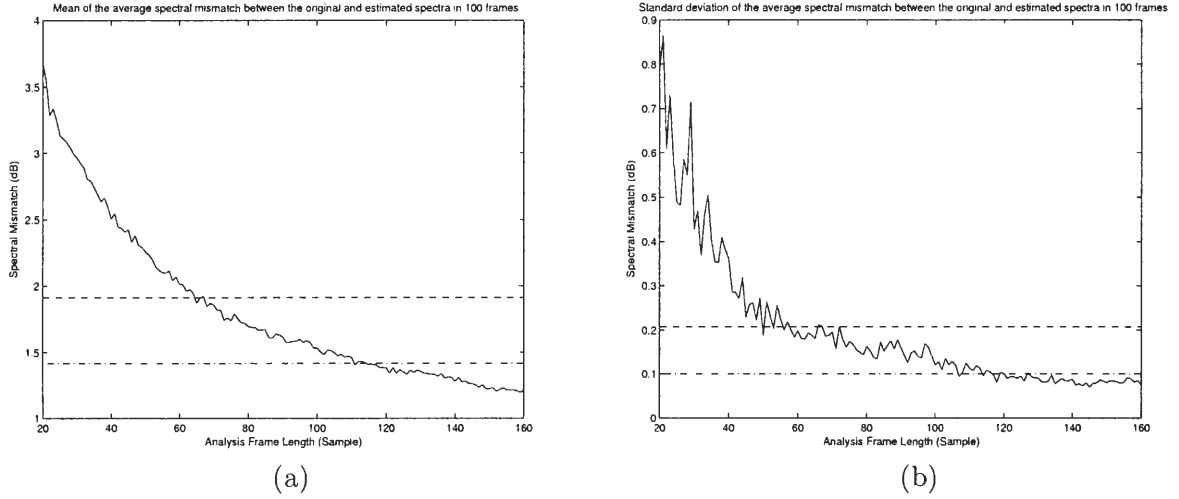


Figure 13: The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples (solid line) and the autocorrelation method using 200 samples (dash-dotted line) and using 100 samples (dashed line).

respectively, and the transition frequency from one band to the next was f_c . Therefore, the term *transition frequency* is also used to denote the cut-off frequency of the filter pairs. In this test, three different sets of low-pass/high-pass filters with different cut-off frequencies at 3000, 2000 and 1000 Hz were used to generate three different types of excitation signals. Since the excitation signal was not purely voiced, it was not possible to obtain the performances of both methods reliably from a single analysis frame. Therefore, the experiment was repeated for 100 different frames, and the mean and the standard deviation of the average spectral mismatch between the original and the estimated spectra obtained by both methods are used as the performance metrics for the evaluation of both methods.

In this experiment, it was observed that the average spectral mismatch obtained by both methods always increases as the transition frequency decreases for all vocal-tract configurations. However, it was also observed that the CLP method is more sensitive to the amount of the noise in the excitation signal. This observation is illustrated in Figure 14 using the speech spectrum with uniformly spread formants. In this figure, the right and left columns show the mean and the standard deviation of the average spectral mismatch obtained by both methods for the pitch-cycle lengths between 20 and 160 samples, respectively. The three rows are the results from the three transition frequencies in decreasing order. In this

figure, it is seen that the CLP method performance is still better than that of autocorrelation method for pitch cycles longer than 100 samples as in the case of purely-voiced speech. However, for pitch cycles shorter than 100 samples, both the mean and the standard deviation of the average spectral mismatch obtained by the autocorrelation method are consistently better than those obtained by the CLP method. The CLP method is more sensitive to the transition frequency; the average spectral mismatch increases as much as 1 dB on the average when a purely-voiced speech spectrum is replaced with a partially-voiced speech spectrum with the transition frequency at 3 kHz. However, the mismatch increases less when the transition frequency is decreased further. This observation is illustrated in Figure 15a obtained using the speech spectrum with uniformly spread formants. Furthermore, it is observed that the standard deviation also increases with decreasing transition frequency for the CLP method while it is almost constant for the autocorrelation method. In addition, this performance variation for the CLP method is almost constant for pitch-cycle lengths longer than 100 samples but increases for short pitch cycles with decreasing transition frequency, which is illustrated in Figure 15b. This observation is also consistent with the one obtained in the synthetic unvoiced speech test in which the standard deviation of the average spectral mismatch always increases with decreasing pitch-cycle length. A new method which reduces the problem with the spectral estimation variation of the CLP method for short pitch cycles is described in Section 4.5.

These results show that the autocorrelation analysis has better performance than the CLP analysis for partially-voiced speech because of the sensitivity of the CLP analysis to noise in the excitation signal. However, when the true spectrum and the estimated spectra obtained by both methods are compared, it is also seen that the mismatch is mostly occurred in the high-frequency part of the spectrum that contains only noise. It is also observed that the spectral mismatch further increases when the energy of the noisy band is much less than the energy of the band with harmonics. In these cases, long analysis frame allows the autocorrelation method to make a better spectral estimate in these regions. This observation is illustrated in Figure 16 using the speech spectrum with uniformly spread formants. The pitch-cycle length is 60 samples in this illustration. In this figure, the figures

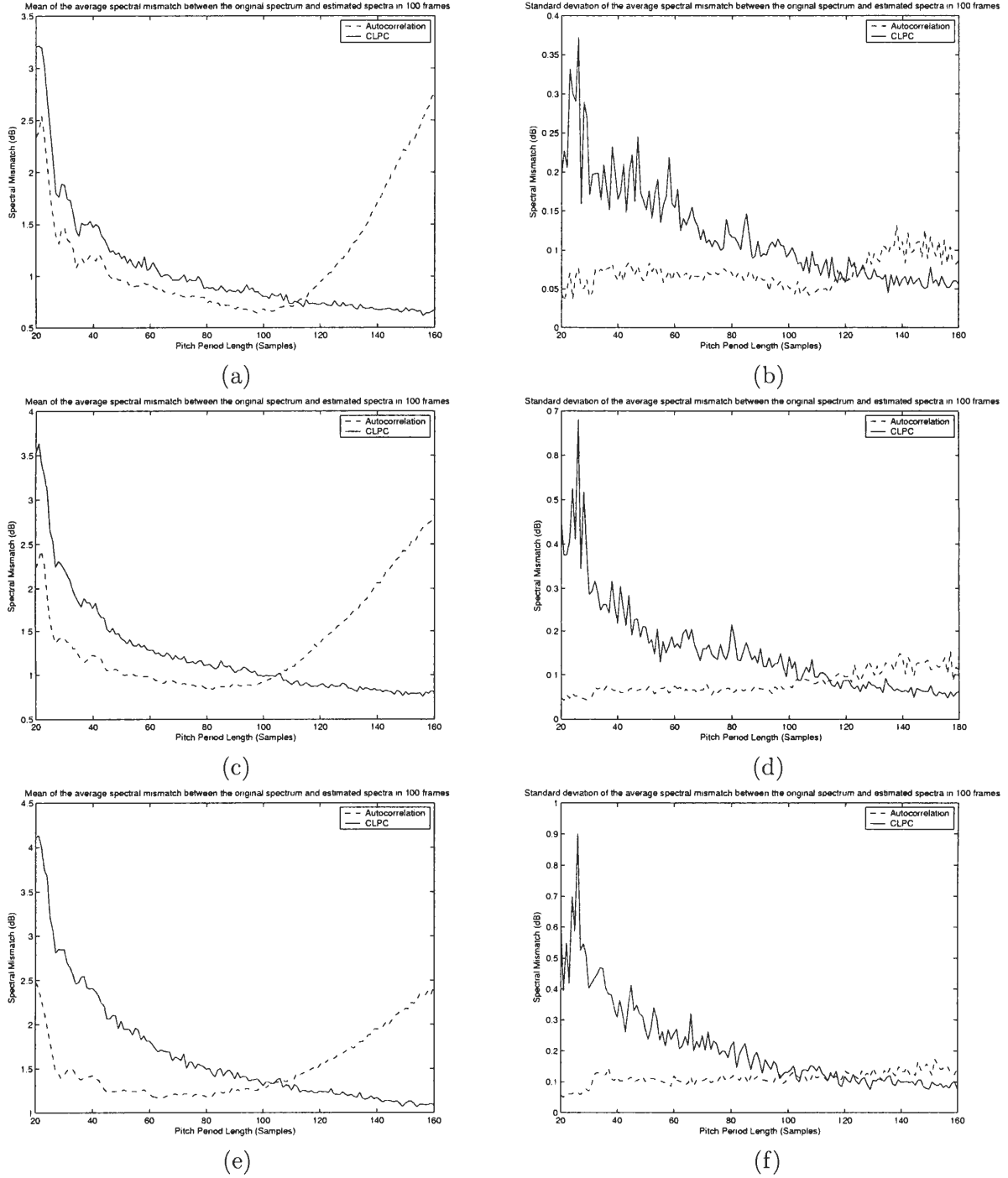


Figure 14: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLPC and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for partially-voiced speech signal. The transition frequencies of the spectrum are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively. The speech spectrum is the “uniformly spread formants” example.

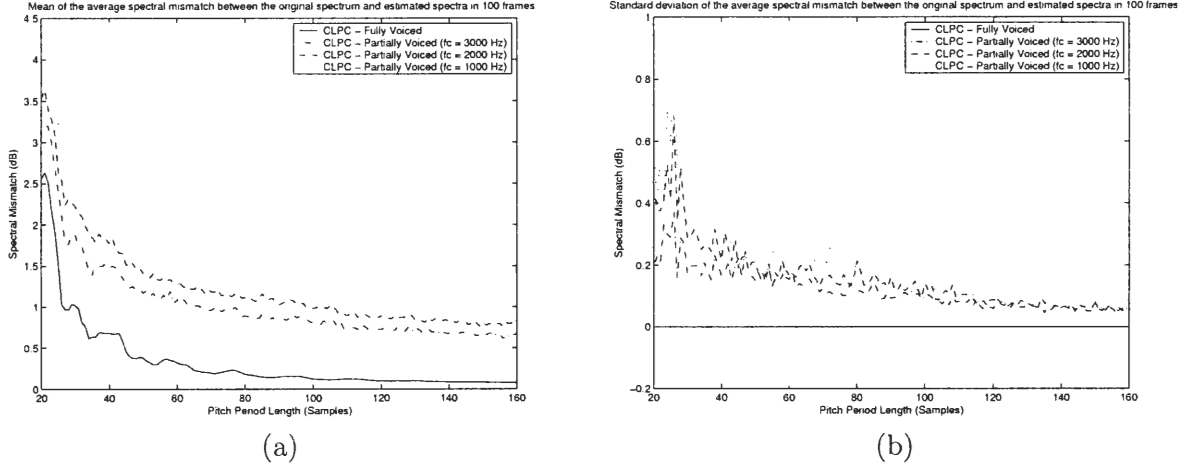


Figure 15: The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced and partially-voiced speech signals. The speech spectrum is the “uniformly spread formants” example.

on the left column show the true spectrum and the estimated spectra obtained by both methods, and the figures on the right column illustrate the absolute spectral mismatch between the true and the estimated spectra. The three rows are the results from the three transition frequencies in decreasing order. As it can be seen from these figures, the largest spectral mismatch occurred around the last formant frequency in which the energy is much lower than those of the regions around the first two formants. Finally, the absolute spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for the signals generated with the excitation signals with different transition frequencies is shown in Figure 17. This figure shows an increase in the mismatch with decreasing transition frequency.

A final synthetic-speech signal test was conducted to obtain the performance of both analysis methods when used to model noisy signals. In this experiment, the fully-voiced synthetic-speech signal was generated first, and then, white-noise was added to obtain the noisy synthetic-speech signal. The energy of the white-noise sequence was adjusted such that the signal-to-noise ratio (SNR) of the final signal was one of the following: 30, 20, 10, 5 and 0 dB. Although the results from all noisy signals with these SNR ratios were

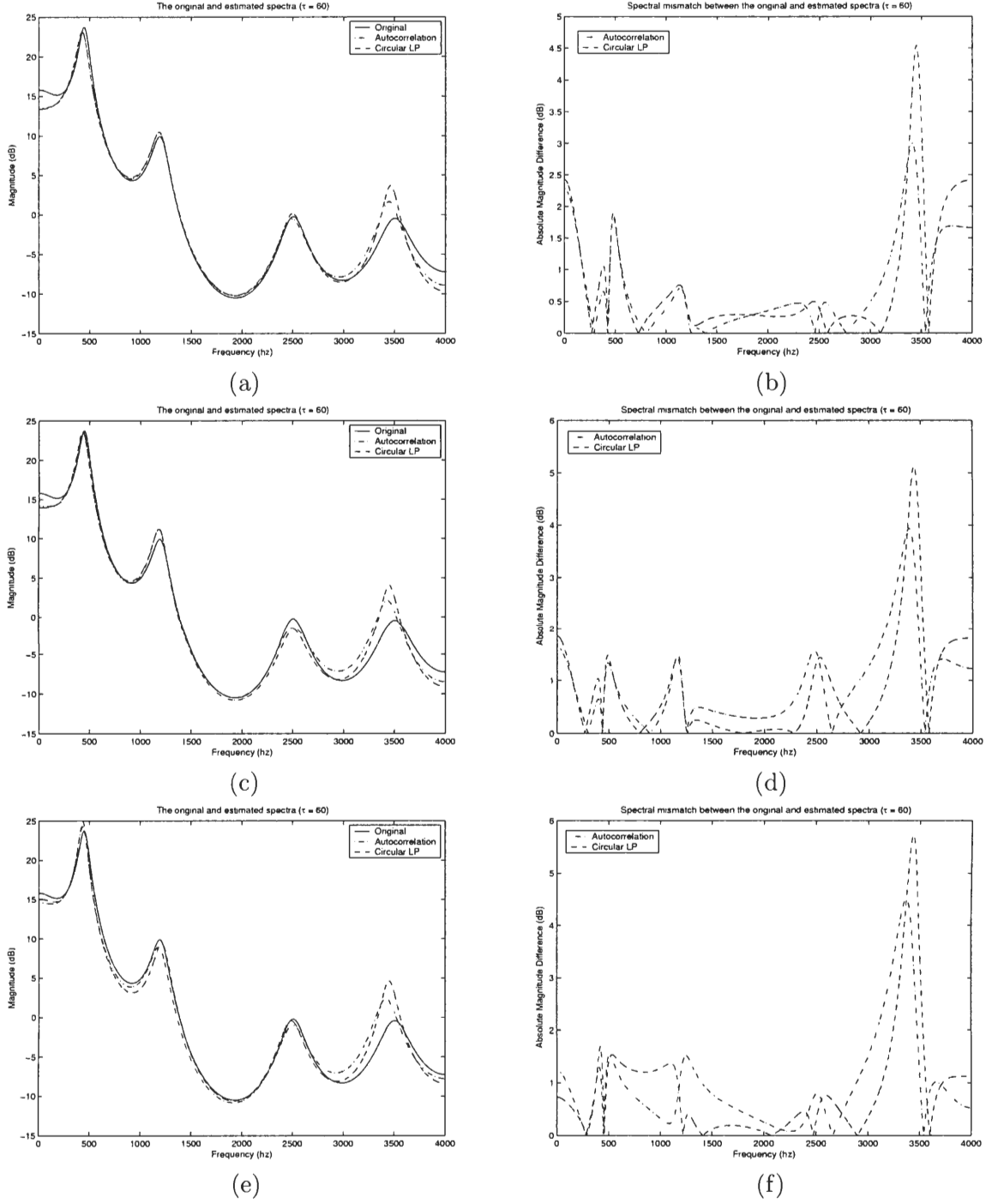


Figure 16: The true speech spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the transition frequencies are 3000 Hz(a), 2000 Hz(c) and 1000 Hz(e), and the absolute error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the transition frequencies are 3000 Hz(b), 2000 Hz(d) and 1000 Hz(f). The pitch period is 60 samples.

three rows are the results obtained using the signals with different SNR in decreasing order. From these graphs, it is observed that the performance of both methods are equivalent when the SNR of the signal is 30 dB and the performance in that case is very close to that of the clean speech. The spectral estimation performance of both methods is also uniform within these 100 frames. When the SNR is reduced to 20 dB from 30 dB, the average spectral mismatch is increased by 1 dB for both methods on the average. The autocorrelation method also has a slightly better performance than the CLP method when the pitch-cycle length is less than 100 samples. However, the average spectral mismatch increases rapidly as the pitch-cycle length increases as in the previous cases. The standard deviation of the average spectral mismatch in these 100 frames is constant for the autocorrelation method (around 0.1 dB) that shows a uniform performance for all cycle lengths. This is also the case for the CLP method when the pitch cycles are longer than 100 frames. However, the standard deviation increases with the decreasing cycle length since the number of samples is not enough to find the filter coefficients reliably because of the noise. When the SNR of the signal is reduced to 10 dB from 30 dB, the average spectral mismatch is increased by 3 dB for both methods on the average, and the estimated spectrum is usually no longer similar to the true spectrum. The spectral estimation performance variation is also increased for both methods, however the variation increases more when the CLP method is used to model a signal with short pitch cycles. The change in the mean and the standard deviation of the average spectral mismatch for the CLP method for different SNR of the signals are illustrated in Figure 19a and Figure 19b, respectively. A method to reduce the estimation performance variation for short pitch cycles is presented in Section 4.5.

As in the case for the synthetic partially-voiced speech, the performance of the autocorrelation method seems better than that of the CLP method for the analysis of noisy speech signals. However, from Figure 18, it is also clear that the CLP still retains the better performance property for pitch cycles longer than 80 samples. Besides these observations, the analysis of the estimated spectra obtained by both methods under different noise conditions also gives more insights on the spectral estimation performance of both methods. As an example, the true spectrum and the estimated spectra obtained by both methods and the

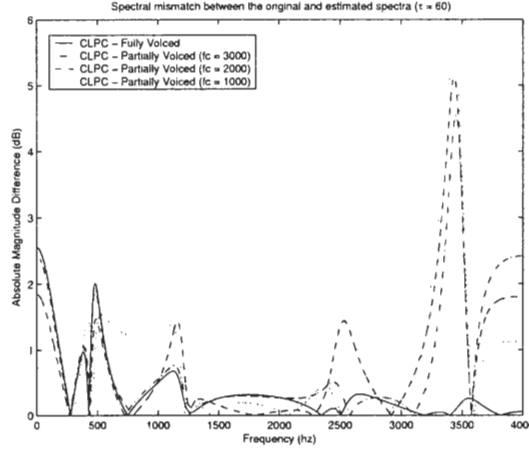
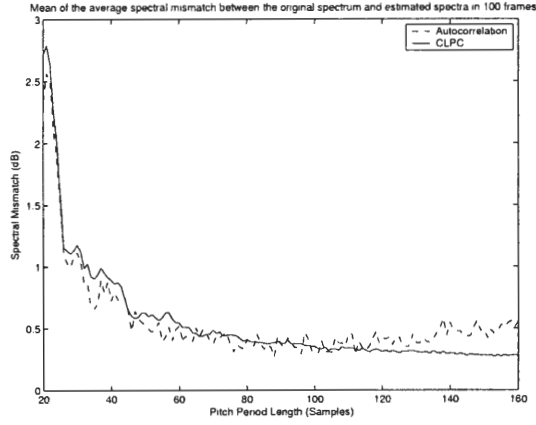


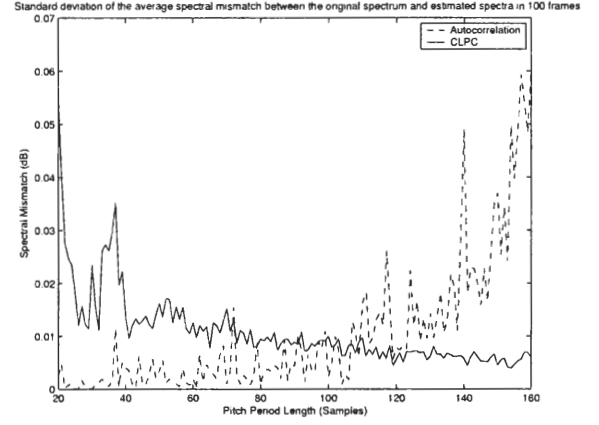
Figure 17: The absolute spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method when the analyzed signals are fully-voiced and various partially-voiced synthetic speech signals and the pitch period is 60 samples. The speech spectrum is the “uniformly spread formants” example.

evaluated, only the results from the noisy speech signals with an SNR of 30, 20 and 10 dB are discussed here, as the results from the signals with 10 dB SNR already illustrates the spectral estimation performance of both methods under high noise. Furthermore, the experiment was repeated for 100 different frames, and the mean and the standard deviation of the average spectral mismatch between the original and the estimated spectra obtained by both methods were used as the performance metrics to compare the methods.

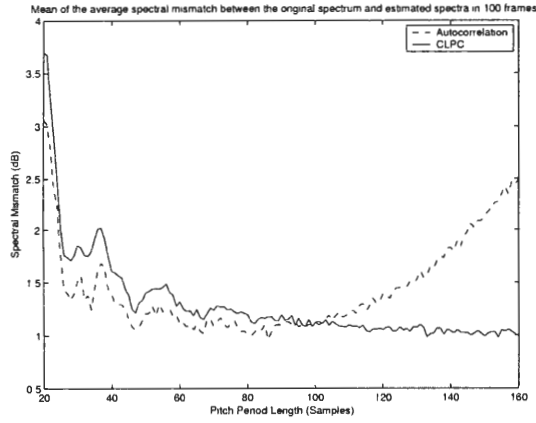
Throughout the evaluation set, it was observed that the average spectral mismatch always increases with decreasing SNR of the signal. However, the spectral estimation variation of the CLP method for noisy signals is not as high as for partially-voiced signals. Furthermore, it was also observed that the performance of the autocorrelation method decreases very rapidly for pitch-cycles longer than 80 samples. The convolution of the window spectrum with the noisy speech spectrum spreads not only the energy of the harmonics but also the energy of the noise into the neighboring harmonics, which results in much worse estimation accuracy in these regions. These observations are illustrated in Figure 18 using the speech spectrum with uniformly spread formants. In this figure, the right and left columns show the mean and the standard deviation of the average spectral mismatch obtained by both methods for the pitch-cycle lengths between 20 and 160 samples, respectively. The



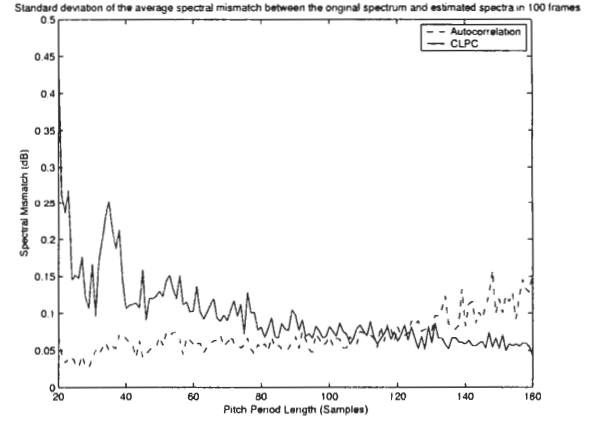
(a)



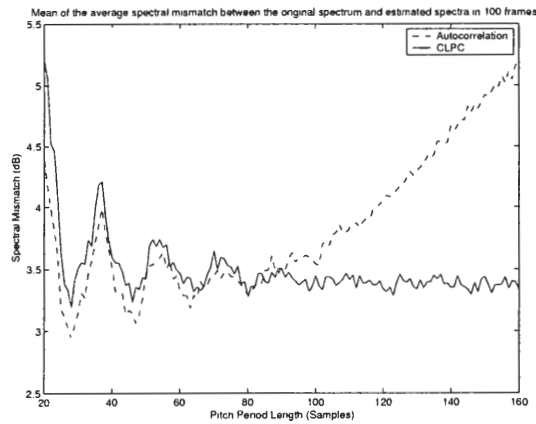
(b)



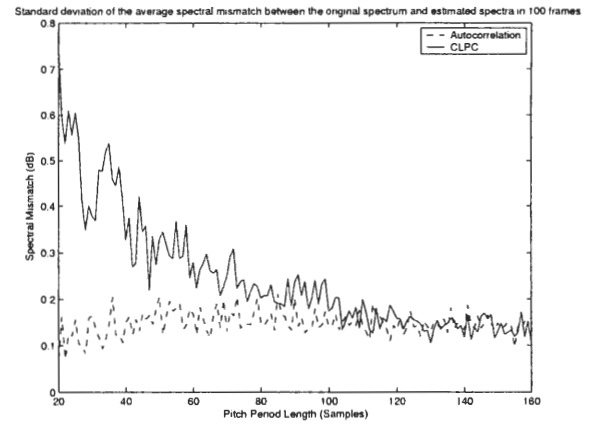
(c)



(d)



(e)



(f)

Figure 18: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for noisy speech signal. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f), respectively.

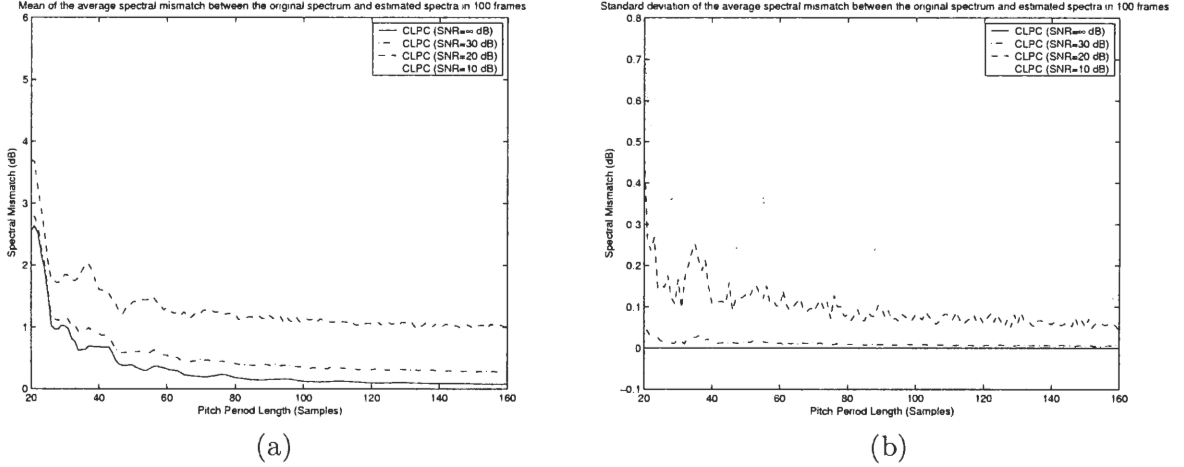


Figure 19: The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced speech and various noisy speech signals.

absolute spectral difference between the true spectrum and the estimated spectra are given in Figure 20 for the signals with SNR of 30, 20 and 10 dB. The pitch period is 60 samples in this example. The performance of both methods are similar, especially when the SNR of the signal is 20 and 30 dB. When the SNR of the signal is 30 dB, the estimated spectra obtained by both methods are very close to the true spectrum, which also supports the results above. When the SNR is decreased to 20 dB, the spectral mismatch is increased almost uniformly throughout the spectrum. In addition, it is evident that the estimation is better in the high-energy regions than low-energy regions. The estimated spectra obtained by both methods are clearly different from the true spectrum when the SNR of the signal is reduced to 10 dB. However, the CLP method estimates the formant locations and bandwidths better than the autocorrelation method, especially in the high-frequency regions. Modeling only the frequency locations at harmonics and avoiding the convolutional effects of the windowing function's spectrum allows the CLP method to perform better than the autocorrelation method for this case.

A final evaluation was performed on narrowband real-speech signal. It was observed that when the pitch period is longer than 60 samples, the CLP method usually estimates the linear-prediction coefficients reliably. However, when the pitch-period length has a fractional

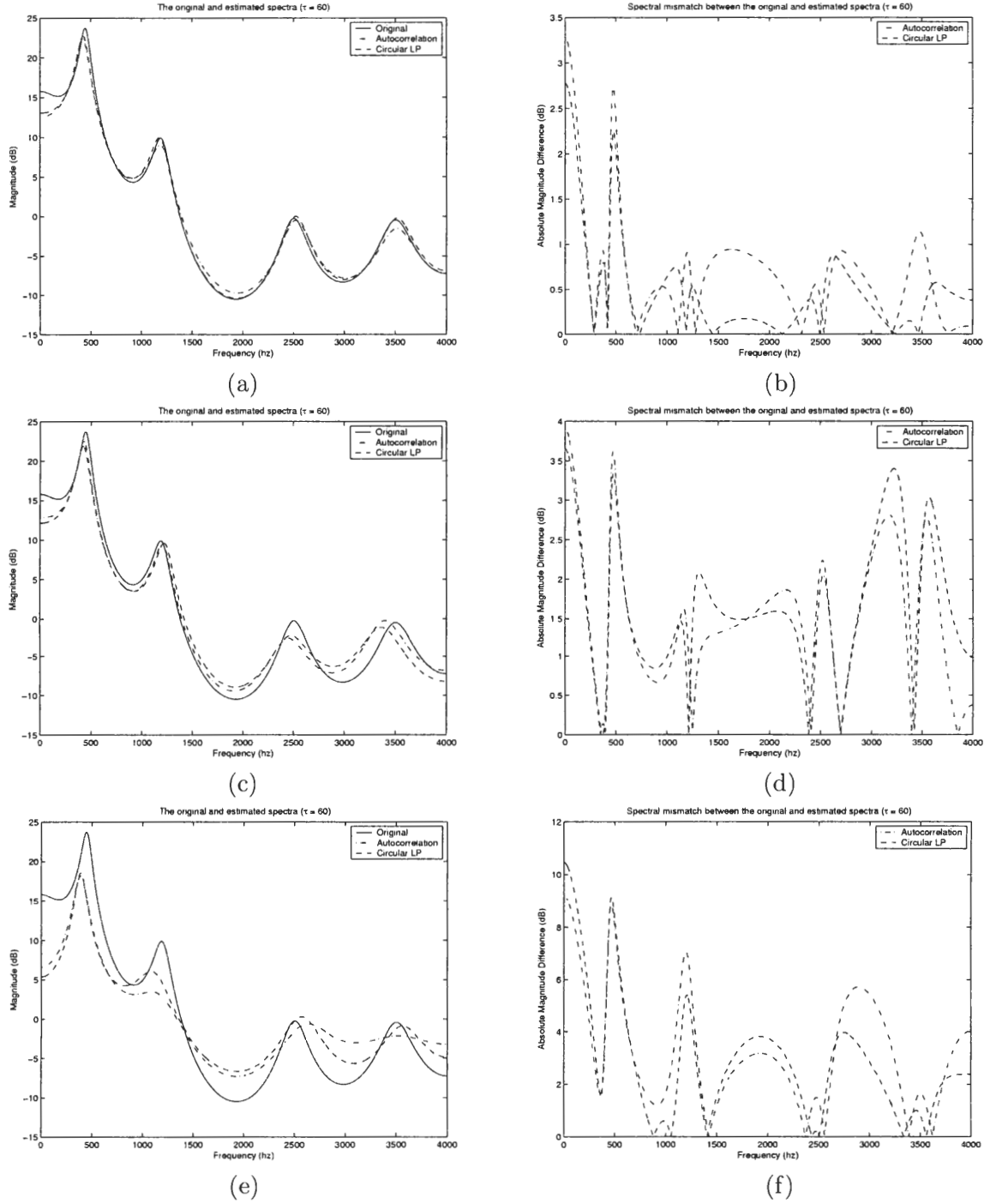


Figure 20: The true speech spectrum with uniformly spread formants and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(a), 20 dB(b) and 10 dB(c), and the error between the true spectrum and the estimated spectra obtained by the CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(d), 20 dB(e) and 10 dB(f).

part that is close to 0.5, the linear-prediction coefficients estimated by the CLP method may be significantly different from the ones estimated by the autocorrelation method even if the signal is stationary. This effect worsens when the pitch-period is much shorter than 60 samples, which is the case for high-pitched speakers. This problem is illustrated in Figure 21. In this example, a fairly stationary speech signal with an average pitch period around 43 samples is used to evaluate both analysis methods. A hamming window is applied to 200 samples prior to the autocorrelation analysis and the CLP analysis is performed twice on two single pitch cycles of length 42 and 43 samples as shown in Figure 21a between the dashed lines. As it can be seen from the Figure 21b, the spectral estimates of CLP method in both cases are slightly different from that of the autocorrelation method, especially in the high-frequency region. These inaccurate spectral estimates occasionally introduce audible artifacts in the synthesized speech as well. The reason for this problem and a solution is given in the Section 4.4.

4.3.3 Performance Evaluation of CLP Synthesis

In this section, the performance of the three synthesis methods discussed in Section 4.2 are illustrated in terms of the reconstruction SNR calculated from a single cycle of the original and the synthesized signals. For each vocal-tract filter combination and the pitch-cycle

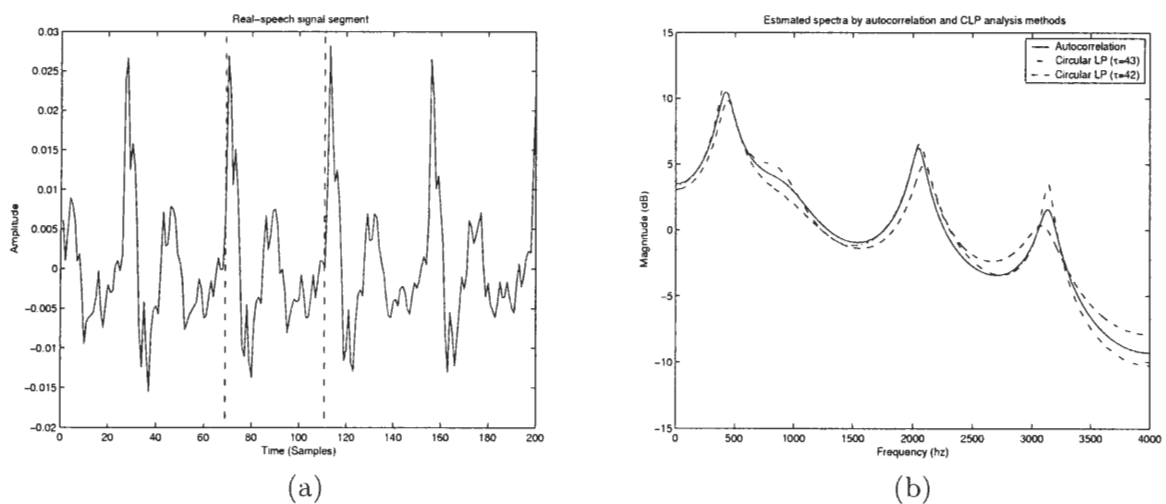


Figure 21: A 200 sample stationary real-speech segment (a), and the estimated spectra obtained by the autocorrelation method using 200 samples and the CLP method using 42 and 43 samples (b).

length in the test setup, a CLP analysis was performed on the synthetic speech to estimate the prediction filter, and a circular residual signal was obtained using (74). Then, this signal was filtered with the estimated all-pole filter using one of the synthesis methods described above to reconstruct the synthetic signal. After energy matching, the SNR between the original pitch cycle and the reconstructed pitch cycle was measured in dB. To obtain consistent results, only purely-voiced synthetic speech was used in these experiments. In addition, the application of pre-emphasis filter prior to CLP analysis and de-emphasis filter after CLP synthesis was also evaluated. The pre-emphasis filter was a single-tap FIR filter whose z-transform is equal to $1 - 0.95z^{-1}$.

The main purpose of this test was to observe the reconstruction performance of the three methods on different vocal-tract/pitch-cycle length combinations for a range of parameters (i.e. number of iterations for the first two methods and the length of the synthesis filter in the third method). Furthermore, it has been previously reported that the reconstructed speech is always perceptually indistinguishable from the original speech when the reconstruction SNR is greater than 72 dB [83]. As a result, one of the other objectives in these tests is to determine the number of iterations required to achieve 72 dB reconstruction SNR on the average for a given pitch-cycle length.

The synthesis performance of the first method depends on various factors including the number of iterations, the spectral characteristics of the synthesis filter and the length of the pitch cycle. In general, the reconstruction SNR always improves in the first few iterations and then saturates. This saturation point and the number of required iterations to reach that point depend on the spectral characteristics of the synthesis filter and the length of the pitch cycle. Furthermore, the number of required iterations to achieve 72 dB SNR for a particular vocal-tract configuration usually decreases with increasing cycle length. As an example, the reconstruction SNR for the speech spectrum with uniformly spread formants is shown in Figure 22a for the number of iterations between 1 and 25, where the SNR is forced to saturate at 200 dB when it is larger than that level. Using all vocal-tract configurations in the test setup, the number of iterations to achieve an average 72 dB reconstruction SNR for the entire pitch-cycle range is calculated and shown in Figure 23a.

Although the saturation level is larger than 72 dB for almost all vocal-tract configurations and cycle-length combinations, there are a few cases in which the saturation SNR is much less than 72 dB. This low SNR is observed only when the pitch-cycle length is less than 30 samples and a first formant with very narrow bandwidth (less than 50 Hz) is present in the frequency response of the all-pole filter. The reconstruction SNR for the speech spectrum with grouped formants is shown in Figure 22b for the number of iterations between 1 and 25. As shown in this figure, the reconstruction SNR is still less than 20 dB even after 25 iterations when the cycle length is less than 25 samples. One way to eliminate this problem is to use pre/de-emphasis filters before CLP analysis and after CLP synthesis, respectively. In this case, it was observed that the SNR increased to 40 dB when these filters were used for the same vocal-tract/pitch-cycle length combinations. However, when the pre/de-emphasis filters were used, more iterations are required before it saturates. Usually, it takes 2-4 more iterations to achieve the same SNR for long pitch-cycle lengths when pre/de-emphasis filters are used. The number of iterations required to achieve 72 dB reconstruction SNR when the pre/de-emphasis filters are used is shown in Figure 23b.

To observe the improvements with the second synthesis method, the experimental setup was modified as follows: after generating the synthetic-speech signal with the known vocal-tract configuration, a second synthetic-speech signal was generated by slightly modifying the same configuration such that the locations and bandwidths of the formants were multiplied by a jitter factor and this number was randomly added to or subtracted from the original formant locations and bandwidths. The initial conditions required for the synthesis were obtained from this second signal. A number of different jitter factors were evaluated in the generation of the second signal: 0.5%, 1%, 2%, 5% and 10% change. In this experiment, it was observed that the saturation points observed in the first method were not changed. The problem associated with the low saturation SNR for the speech spectrum with a first formant with very narrow bandwidth and short pitch-cycle length still remained. However, depending on the similarities between the two vocal-tract filter configurations, the required number of iterations to achieve average 72 dB reconstruction SNR decreases. In these tests, it was observed that the performance of the first method was identical to the performance

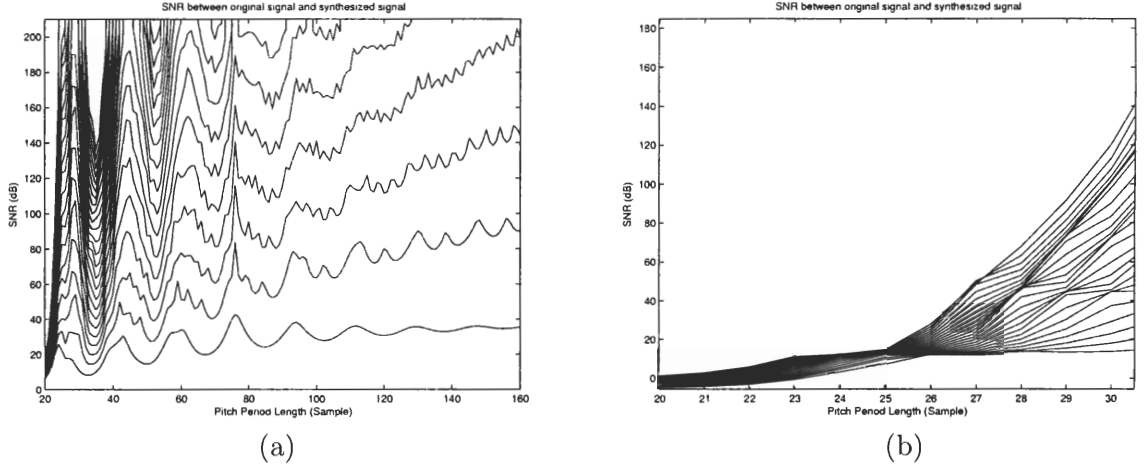


Figure 22: The reconstruction SNR for the speech spectrum with uniformly spread formants for the pitch-cycle range between 20 and 160 samples (a) and for the speech spectrum with grouped formants for the pitch-cycle range between 20 and 30 samples (b). The maximum number of iterations in this figure is 25. The SNR is increased by each iteration.

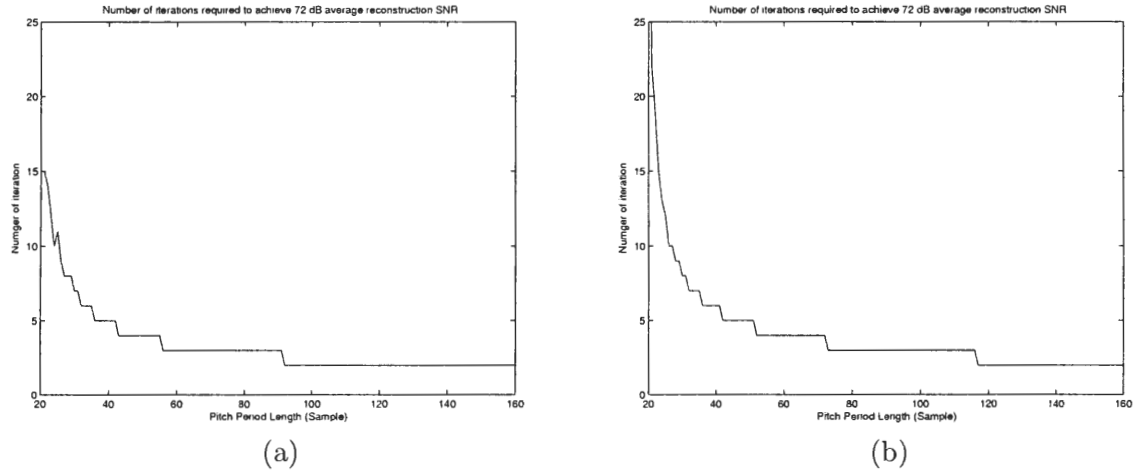


Figure 23: The number of iterations required to achieve an average 72 dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the first synthesis method without pre/de-emphasis filter (a) and with pre/de-emphasis filter (b).

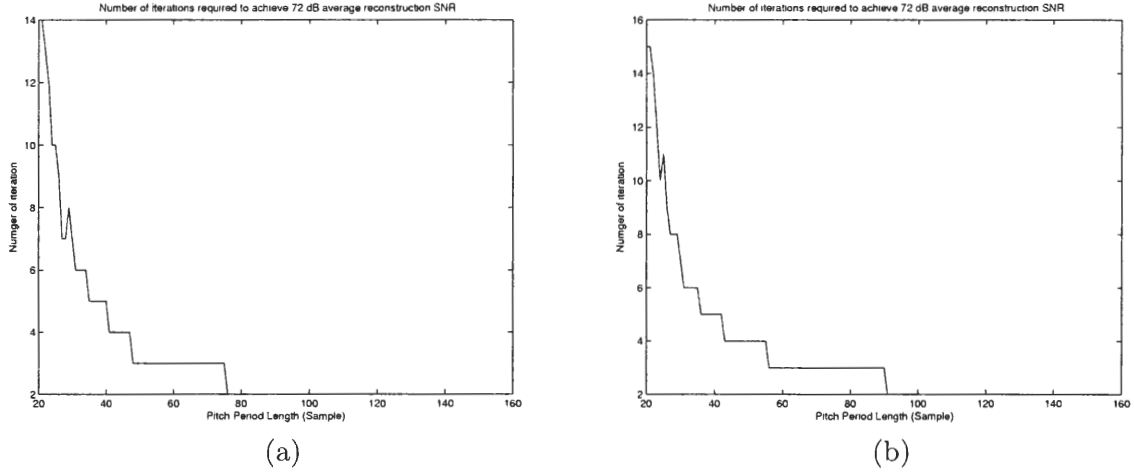


Figure 24: The number of iterations required to achieve an average 72-dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the second synthesis method. The difference between the formant bandwidths and locations of the original and modified configurations in successive pitch cycles is 0.5% in (a) and 5% in (b).

of the second method if the difference between the formant locations and bandwidths of the two vocal-tract configuration was larger than 1%. To illustrate the decrease in the number of iterations, the number of iterations required for an average 72 dB reconstruction SNR is shown in Figure 24a and 24b when the difference between the formant locations and bandwidths of the original and modified configurations are 0.5% and 5%, respectively.

In the performance evaluation of the third synthesis method, it was observed that the first length calculation method performs equally well with the first two synthesis methods, since the integer multiple corresponds to the number of iterations in the first two method. For this reason, this technique does not solve the problem associated with the combination of the short pitch-cycle lengths with the speech spectrum with a first very narrow-bandwidth formant as well. The number of integer multiples of τ to achieve average 72 dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples is given in Figure 25. On the other hand, it is possible to control the degree of reconstruction error with the threshold selection with the second length-calculation method. By setting the threshold to 0.0001, it is possible to achieve 45 dB reconstruction SNR for the speech spectrum with the grouped formants and for a short pitch-cycle length. However, the length of the filter in such cases becomes as large as 2000τ . The average reconstruction SNR for the thresholds 0.01, 0.0025,

and 0.0001 for pitch-cycle lengths between 20 and 160 samples is shown in Figure 26. Although the reconstruction SNR is lower than 72 dB in most cases, the performance variation among different vocal-tract configurations are very small. As a result, it is beneficial to use this method where uniform performance is crucial.

4.4 CLP Modeling with Fractional Cycle Length

In the previous sections, the CLP modeling method was shown to work very well with the synthetic speech when the cycle length is an integer. Unfortunately, the length of the pitch cycles in a narrowband critically sampled real-speech signal is rarely an integer. If the method cannot work well on such signals, it has no practical use. In the previous section, it was shown that the CLP method does not always work very well for real-speech signals with fractional cycle lengths. The following experiment was performed to illustrate the performance of the CLP method for signals with fractional cycle lengths. Here, the excitation signal was generated from an impulse train with known pitch-period at a sampling rate that was 10 times the original sampling rate. Then, this signal was filtered with a low-pass filter, and decimated by a factor of 10 to obtain the excitation signal in the original sampling rate. The excitation signal was filtered with the all-pole filter to generate the synthetic purely-voiced speech signal with fractional cycle lengths in 0.1 sample steps. In this experiment, a purely-voiced speech signal with a period length of 60.3 samples was generated using this method. The spectrum is the one with uniformly spread formants described in the previous section. CLP analysis was performed using both 60 and 61 samples. The true spectrum and the estimated spectra obtained by CLP analysis with 60 and 61 samples and the absolute error between the true spectrum and the estimated spectra are displayed in Figure 27a and Figure 27b, respectively. In both cases, the performance of the CLP method is simply not acceptable despite the sufficiency of the number of samples used in the estimation.

Apparently, this problem occurs mainly because of the discontinuity at the boundaries of the cycle. When the circular residual signal is calculated, a visible spike is observed across the boundary. The problem can be corrected by finding the exact periodic pitch cycle in

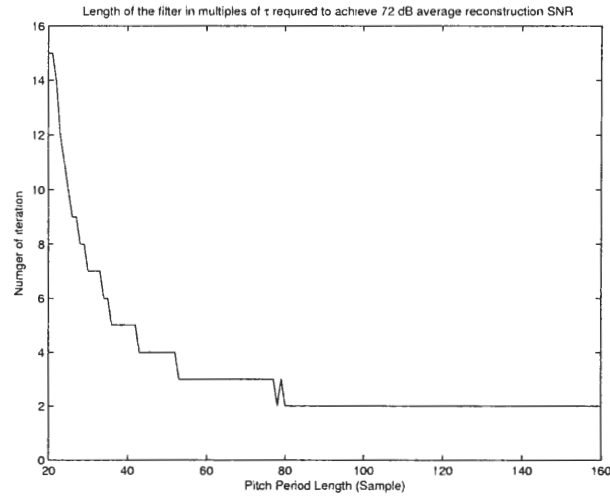


Figure 25: The length of the filter in multiples of τ required to achieve an average 72 dB reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the third synthesis method.

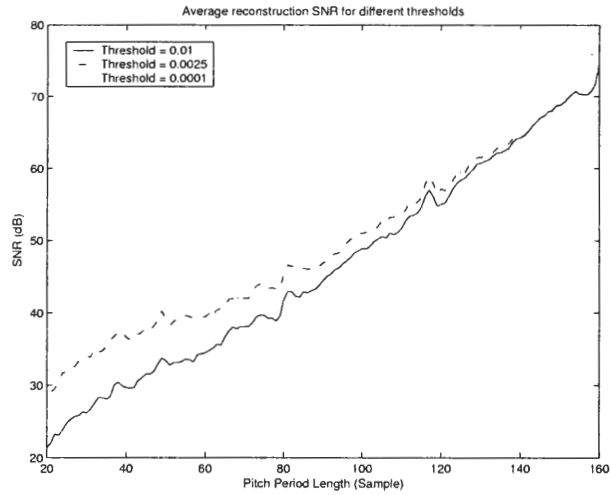


Figure 26: The average reconstruction SNR for pitch-cycle lengths between 20 and 160 samples for the thresholds, 0.01, 0.0025 and 0.0001, for the second length-calculation method of the third synthesis method.

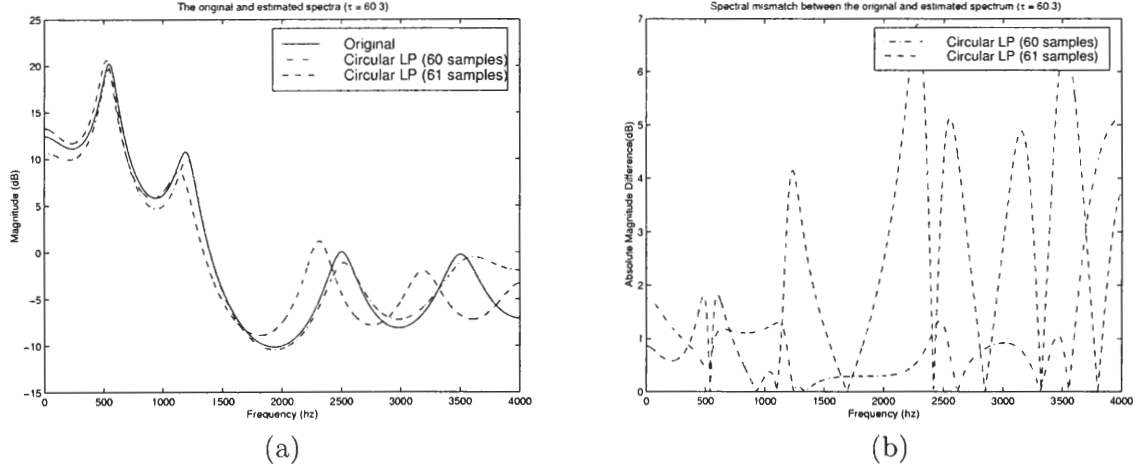


Figure 27: The true speech spectrum and the estimated spectra obtained by the CLP method using 60 and 61 samples (a); and the absolute error between the true spectrum and the estimated spectra (b). The cycle length is 60.3 samples.

the upsampled signal and generating a new signal by concatenating a specific number of cycles of this circular signal such that the decimation of this new signal is exactly periodic in the original sampling rate.

Assume $x_N[n]$ is a circular signal, bandlimited to π/N , whose sampling rate is N times the original sampling rate. The required number of identical cycles, m , to be concatenated to generate an exactly periodic signal at the original sampling rate can be found as

$$m = \frac{N}{\text{GCD}(N, Nf)}, \quad (79)$$

where $\text{GCD}(x, y)$ is the greatest common divisor of x and y , and f and τ_0 are the fractional and integer parts of the cycle length, $\tau = \tau_0 + f$, and Nf is a positive integer number. As a result, the new truly periodic signal, $x_m[n]$, is generated by concatenating m cycles of $x_N[n]$ and decimating it by N . The correlation coefficients in (71) are computed by simply replacing $x[n]$ by $x_m[n]$ and τ by $m(\tau_0 + f)$. Alternatively, to avoid downsampling, the same calculation can be performed on the upsampled signal, $x_N[n]$. Since $x_N[n]$ is bandlimited to π/N , there is no need to apply a low-pass filter before decimation to generate $x_m[n]$. Considering this fact, the correlation coefficients can also be calculated as

$$r(i) = \sum_{n=0}^{m(\tau_0+f)-1} x_N \left[\frac{nN}{m} \right] x_N \left[\left(\left(\frac{nN}{m} \pm iN \right) \right)_{N(\tau_0+f)} \right] \quad (80)$$

To demonstrate the usefulness of this new method, the previous experiment was repeated by performing CLP analysis using 60.3 samples. The true speech spectrum and the estimated spectra obtained by CLP analysis using 60, 61 and 60.3 samples and the absolute error between the true spectrum and the estimated spectra are displayed in Figure 28a and Figure 28b, respectively. The proposed method works very well with fractional cycle lengths.

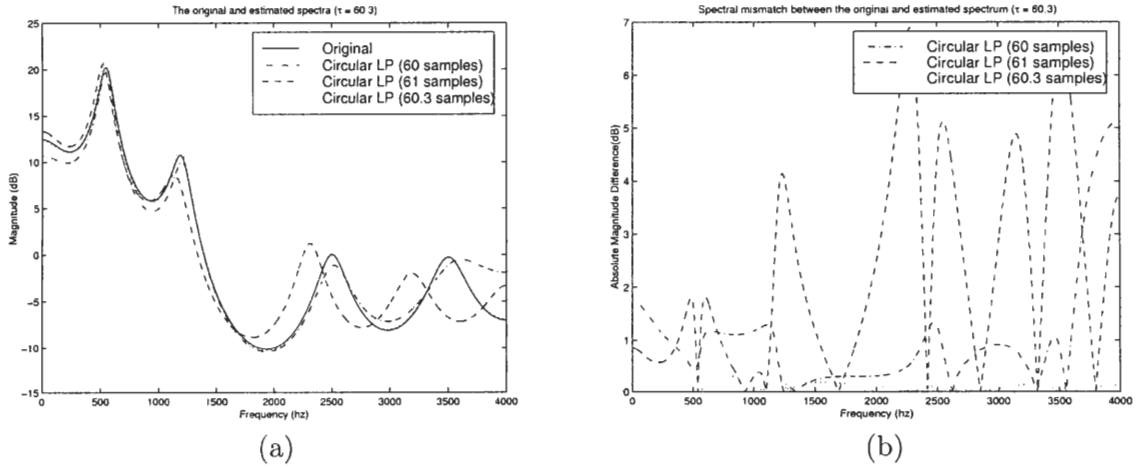


Figure 28: The true speech spectrum and the estimated spectra obtained by CLP analysis using 60, 61 and 60.3 samples (a); and the absolute error between the true spectrum and the estimated spectra (b). The cycle length is 60.3 samples.

To observe the performance of the new CLP method, the real-speech example given in previous section was revisited and the same cycle was modeled with the CLP method using 42.7 samples. The spectra estimated by the autocorrelation method and the CLP method using 42, 43 and 42.7 samples for this case are displayed in Figure 29. The absolute spectral difference between the spectrum estimated by the autocorrelation method and the spectra estimated by the CLP method using different cycle lengths is also shown in this figure. The estimated spectra by the CLP method using 42.7 samples are much closer to the spectrum obtained by the autocorrelation method, especially in the high-frequency region.

To take advantage of this method, the process to obtain circular residual signals and the synthesis process must also be modified such that it can use $x_m[n]$ or $x_N[n]$. The circular residual signal at the upsampled signal rate, $e_N[n]$, can be obtained using two different methods:

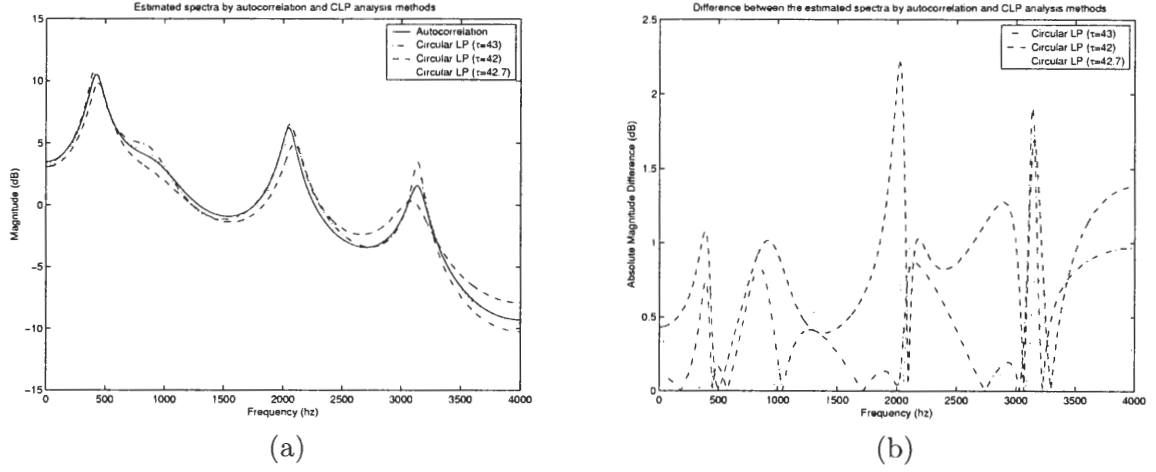


Figure 29: The speech spectra obtained by the autocorrelation method using 200 samples and the CLP method using 42, 43 and 42.7 samples (a), and the absolute spectral difference between the spectrum estimated by the autocorrelation method and the spectra estimated by the CLP method using 42, 43 and 42.7 samples (b).

- After $x_m[n]$ is obtained, the $x[n]$ and $e[n]$ in (74) are replaced by $x_m[n]$ and $e_m[n]$, respectively, and $e_m[n]$ is computed for the samples between 0 and $m(\tau_0 + f) - 1$. Then, $e_m[n]$ is upsampled by a factor of N and low-pass filtered to obtain $e_N[n]$.
- Instead of filtering $x_m[n]$ with the inverse of the all-pole filter, $A(z)$, only the N/m indexed samples of $x_N[n]$ are filtered with the inverse of $A(z^N)$ such that

$$e_N[n] = x_N[n] - \sum_{k=1}^p a_k x_N \left[((n - kN))_{N(\tau_0 + f)} \right] \quad (81)$$

and the rest of the samples are obtained with interpolation. This method also uses the fact that $x_N[n]$ is bandlimited to π/N and does not need to be low-pass filtered before decimation to obtain $x_m[n]$.

The synthesis process in (75) is modified similarly and thus can also be done in two ways:

- After $\hat{e}_N[n]$ is obtained, m cycles are concatenated and downsampled to obtain $\hat{e}_m[n]$. Then, $\hat{x}_m[n]$ is computed using (75) by replacing $\hat{x}[n]$ with $\hat{x}_m[n]$ and $\hat{e}[n]$ with $\hat{e}_m[n]$ for the samples between 0 and $m(\tau_0 + f) - 1$. Then, $\hat{x}_m[n]$ is upsampled by a factor of N and low-pass filtered to obtain $\hat{x}_N[n]$.
- Instead of filtering $\hat{e}_m[n]$ with the all-pole filter, $A(z)$, only the N/m indexed samples

of $\hat{e}_N[n]$ is filtered with the all-pole filter, $A(z^N)$, such that

$$\hat{x}_N[n] = \hat{e}_N[n] + \sum_{k=1}^p a_k \hat{x}_N \left[((n - kN))_{N(\tau_0+f)} \right] \quad (82)$$

and the rest of the samples are obtained with interpolation. This method also uses the fact that $\hat{e}_N[n]$ and $\hat{x}_N[n]$ are bandlimited to π/N and does not need to be low-pass filtered before decimation to obtain $\hat{e}_m[n]$.

To find a proper upsampling factor, the experimental setup described in Section 4.3 was used but the excitation signal was generated with fractional cycle lengths using the technique described early in this section. In this experiment, an upsampling factor of 20 was used, which resulted in a sample resolution of 0.05 samples. In this test, a synthetic signal with known all-pole filter and fractional pitch-cycle length was generated first, and a CLP analysis with correct pitch-cycle length was performed. The estimated spectrum was used as the reference spectrum. Then, the CLP analysis was repeated for all possible pitch-cycle lengths within a ± 0.5 sample distance of the true pitch-cycle length in 0.05 sample steps, and the average spectral mismatch between the reference spectrum and the newly estimated spectrum was calculated. This experiment was repeated for all vocal-tract combinations in the test set and for all pitch-cycle lengths between 20 and 160 samples in 0.5-sample step. The sample errors that resulted in an average spectral mismatch of 0.5 dB among all vocal-tract configurations in the test set for all pitch cycle-lengths are shown in Figure 30. From this experiment, it was concluded that the estimated spectrum is sufficiently close to the original spectrum if the mismatch between the true pitch-cycle length and the cycle length used in CLP analysis is less than 0.05 samples. In other words, an upsampling factor of 10 is required for accurate spectral estimation with the CLP method. It was also verified in the informal listening tests that no audible artifacts were introduced in the synthesized speech when the original signal was upsampled 10 times and the pitch cycles are obtained from this signal to be used in CLP analysis.

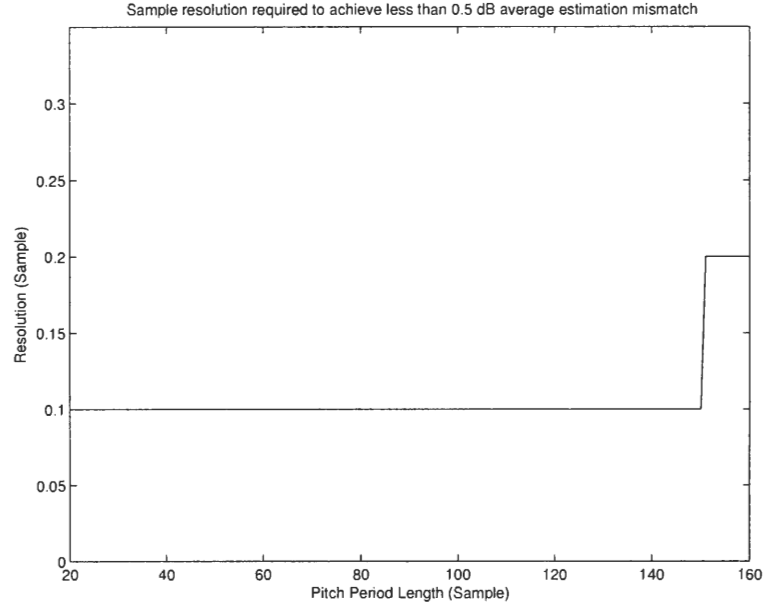


Figure 30: The sample resolution that results into an average 0.5 dB spectral mismatch for pitch-cycle lengths between 20 and 160 samples in 0.5 sample steps.

4.5 Performance Improvements for Short Pitch-Cycles

The problem of the linear-prediction estimation methods with short pitch cycles can be explained in two ways. First, as discussed in Section 2.2, the white-Gaussian noise prediction error assumption fails for voiced speech signal, which has an excitation signal consisting of impulses with large amplitudes separated by the pitch period. This problem gets worse with decreasing pitch-cycle lengths, which results in an increase in the spectral estimation mismatch. Second, in the frequency-domain linear-prediction modeling methods [44], the algorithms try to minimize the ratio between the squared magnitude of the true spectrum and the estimated spectrum by fitting an linear-prediction spectrum at the known harmonics. For short cycle lengths, there are only a few harmonics in the spectrum, and these methods tend to model the harmonic with the largest magnitude with a narrow bandwidth formant. The overall error is very high for these cases.

As discussed in Section 2.2.4, numerous approaches have been proposed to cope with this problem. In this section, two new methods are introduced to improve the performance for short pitch cycles within the circular processing framework. The first method uses multiple

cycles with slightly varying cycle lengths, and hence it is denoted as multicycle CLP (M-CLP) analysis. The second method assumes that the excitation - or pulse locations that form the excitation - is known, and therefore the equations can be reformulated accordingly as discussed in Section 2.2.4. This second method is denoted as pulse excited-CLP (PE-CLP) analysis.

4.5.1 Multicycle Circular Linear Prediction Method

The idea of M-CLP method is based on improving the performance by increasing the number of harmonics in the linear-prediction analysis in frequency domain. In real speech, the pitch-cycle length changes slowly in successive pitch cycles. If it is assumed that the variation in the spectral shape of the vocal-tract is negligible in C successive pitch cycles, it is possible to sample the speech spectrum at the harmonics of C different fundamental frequencies. Including more harmonics in discrete-spectra linear-prediction analysis increases the accuracy of the linear-prediction modeling. Since the CLP analysis is exactly the same as the discrete-spectra linear-prediction modeling of an infinite signal, the same result can be obtained by the CLP analysis using the following formula:

$$r(i) = \frac{1}{\sum_{c=0}^{C-1} \tau_k} \sum_{k=0}^{C-1} r_c(i), \quad (83)$$

where τ_k is the cycle length of the k^{th} pitch cycle and $r_k(i)$ is the i^{th} correlation coefficient of the k^{th} pitch cycle computed by circular correlation defined in (71). This method can be interpreted in a different way as well: C different infinitely periodic signals with different pitch-cycle lengths are going to be predicted with the same set of linear predictor coefficients such that

$$\hat{x}_c[n] = \sum_{k=1}^p a_k x_c[n-k] \quad c = 1, \dots, C. \quad (84)$$

In this case, the sum of squared prediction error for all signals can be written as

$$\varepsilon_c = \sum_{n=0}^{\tau_c-1} (x_c[n] - \hat{x}_c[n])^2 \quad c = 1, \dots, C. \quad (85)$$

When summation of all ε_c is minimized with respect to predictor coefficients, the following set of linear equations are obtained

$$\sum_{k=1}^p a_k \sum_{c=1}^C r_c(|k-l|) = \sum_{c=1}^C r_c(l) \quad l = 1, \dots, p, \quad (86)$$

where

$$r_c(i) = \sum_{n=0}^{\tau_c-1} x_c[n]x_c[(n \pm i)\tau_c]. \quad (87)$$

This resulting set of equations is exactly the same as the ones obtained using (83).

The improvement in the estimation accuracy is illustrated in Figure 31 using one to three pitch cycles in M-CLP analysis. This figure shows the absolute spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP and the M-CLP methods. The cycle lengths are 30, 31, and 32 samples in this example. As expected, since the combined harmonics are distributed more evenly above 1.5 kHz, the accuracy of the estimation is better in this region. Although this method decreases the average spectral mismatch, the spectral mismatch may still be high in the low frequencies, which affects the performance. The performance of autocorrelation analysis is also similar to the M-CLP algorithm in similar conditions. Despite using multiple cycles, the M-CLP method still achieves the same results using almost half of the samples used in autocorrelation analysis for this particular example. Therefore, it is still a viable alternative to the autocorrelation method in the transition regions.

The M-CLP method can also be used to improve the estimation performance with short pitch cycles. In Section 4.3, it was shown that the spectral estimation performance of the

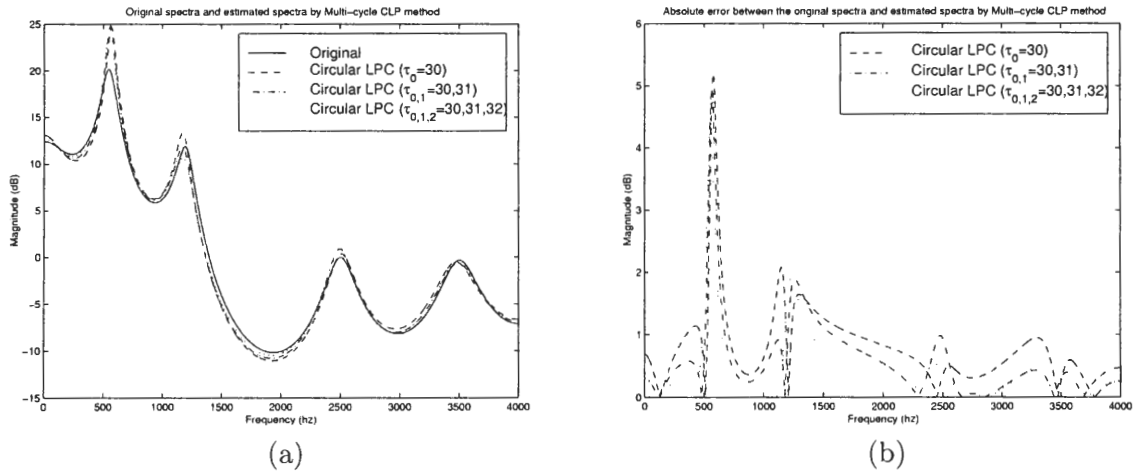


Figure 31: The true spectrum and the estimated spectra obtained by CLP analysis using single pitch cycle($\tau_0=30$ samples), M-CLP analysis using two pitch cycles ($\tau_{0,1}=30,31$ samples) and M-CLP analysis using three pitch cycles ($\tau_{0,1,2}=30,31,32$ samples) (a); and the absolute error between the true spectrum and the estimated spectra (b).

CLP method is always inferior to that of the autocorrelation method when the analyzed pitch-cycle length is less than 80 samples for partially-voiced speech signals regardless of the transition frequency. In addition, the variation in the spectral estimation performance of the CLP method is always greater than that of the autocorrelation method for both partially-voiced and noisy speech signals when the pitch-cycle length is less than 80 samples. In these cases, the M-CLP method can be used to improve the performance and decrease the spectral estimation performance variability for both types of signals, even when the length of the analyzed pitch cycles are the same. To investigate possible performance improvements, the experiments with the synthetic partially-voiced and noisy speech signals were repeated again for the M-CLP method using two and three cycles and compared to the results of the CLP and autocorrelation methods. However, since the main purpose of the CLP method is to estimate the LP coefficients reliably from the shortest possible signal segment, the total number of samples in the analysis was restricted to 120 samples. For this reason, M-CLP analysis using two cycles was performed only when the pitch-cycle length is between 20 and 60 samples. The longest pitch cycle was further restricted to 40 samples when three cycles were used in M-CLP analysis. As discussed before, the noise in the excitation signal of the partially-voiced speech signals and the presence of noise in the noisy speech signals make it impossible to estimate the methods' performance reliably from a single frame. Therefore, the mean and the standard deviation of the average spectral mismatch obtained from 100 frames were used as the performance indicators of these methods.

In the experiments with synthetic partially-voiced speech signals, it was observed that the M-CLP method using both two and three cycles decreases both the average spectral mismatch and the spectral estimation performance variation for all vocal-tract configurations. This observation is illustrated in Figure 32 for the speech spectrum with uniformly spread formants. In this figure, the graphs on the left and right column show the mean and the standard deviation of the average spectral mismatch obtained using the CLP method, the autocorrelation method and the M-CLP method using two and three cycles for the pitch-cycle lengths between 20 and 60 samples, respectively. The three rows are the results from

three transition frequencies in decreasing order. These graphs show that the average spectral mismatch always decreases with the increasing number of cycles in the M-CLP analysis for all transition frequencies. The decrease in the average spectral mismatch is close to 0.4 and 0.5 dB on the average for the M-CLP method using two and three cycles, respectively. As a result, when the pitch-cycle length is longer than 25 samples, the performance of the M-CLP method using two cycles and the M-CLP method using three cycles are very close to that of the autocorrelation method for the transition frequencies 3000 and 2000 Hz, respectively. For shorter pitch cycles, the performance of the autocorrelation method is still better than that of the M-CLP method, especially when the transition frequency is 2000 Hz. When the transition frequency is around 1000 Hz, the average spectral mismatch is also decreased on the average when the M-CLP method is used instead of the CLP method for the short pitch cycles. However, the improved performance is still worse than that of the autocorrelation method. In this case, the excitation signal is very noisy and it is impossible to estimate the spectrum above 1000 Hz reliably from short pitch cycles as shown in Section 4.3. For this reason, averaging the correlation coefficients of the individual pitch cycles found by CLP analysis does not help to reduce the average spectral mismatch. The other important observation in this experiment is the decrease in the standard deviation of the average spectral mismatch with the increase in the number of cycles used in M-CLP analysis for short pitch cycles, as shown in the Figure 32. Even when the transition frequency is 1000 Hz, the spectral estimation performance variation of the M-CLP method is very close to that of the autocorrelation method. This result implies that when the transition frequency is higher than 2000 Hz, the average spectral mismatch does not change much from one analysis frame to another. The estimated linear-prediction filter coefficients with the M-CLP method is also close to the ones estimated with the autocorrelation method even for the short pitch cycles. This result shows that the M-CLP method is suitable for real-speech signals.

When the signal is noisy, the M-CLP method also improves the performance over the CLP method for short pitch cycles, however the improvement is not as much as the one for partially-voiced speech signals. In addition, the spectral estimation performance variation

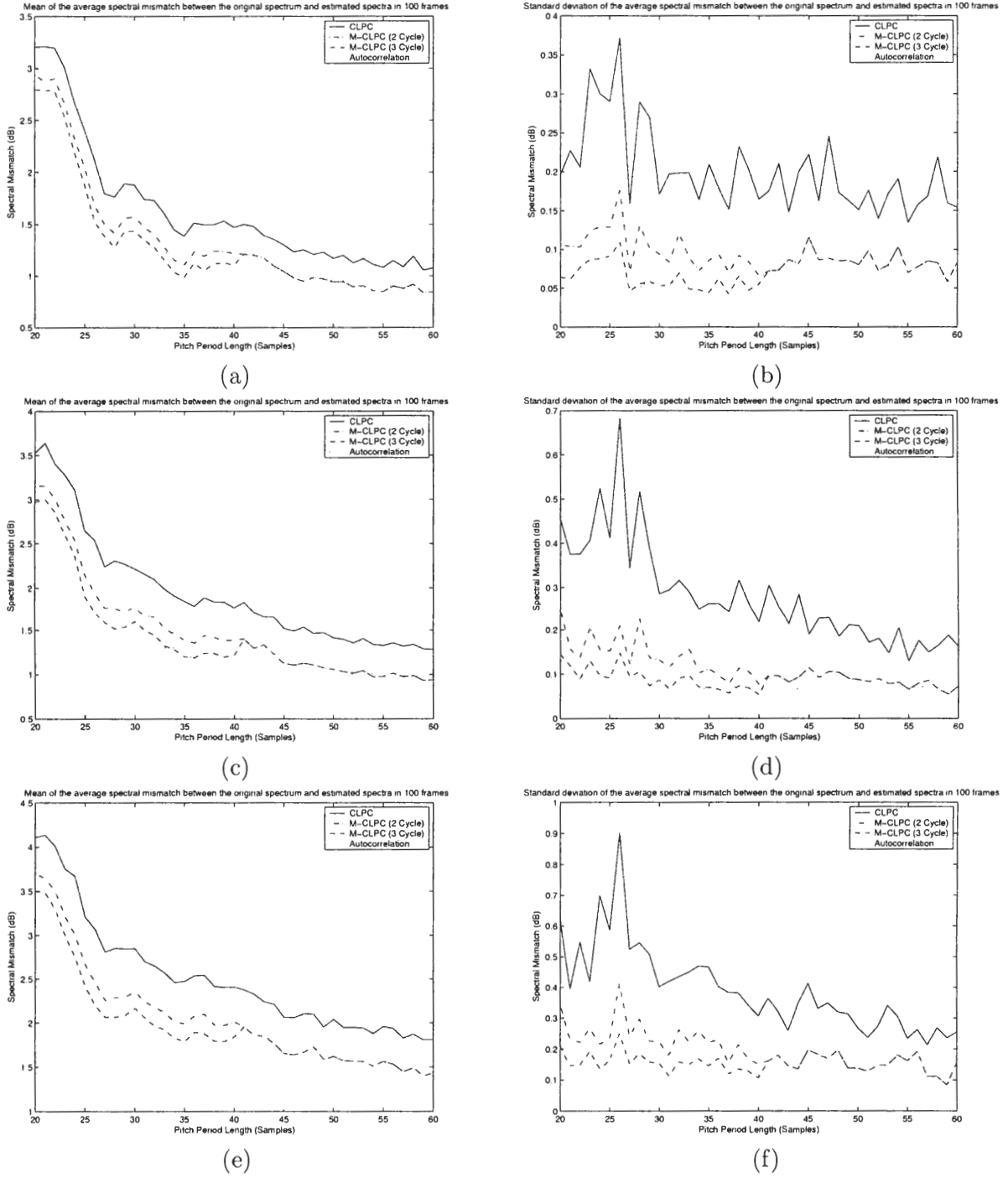


Figure 32: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the M-CLP method with two and three cycles, and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for partially-voiced speech signal. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.

is also reduced such that it is very close to that of the autocorrelation method. These observations are illustrated in Figure 33 using the speech spectrum with uniformly spread formants. In this figure, the graphs on the left and right column show the mean and the standard deviation of the average spectral mismatch obtained by the CLP method, the autocorrelation method and the M-CLP method using two and three cycles for the pitch-cycle lengths between 20 and 60 samples, respectively. The three rows are the results from three SNR cases in decreasing order. As it can be seen from these graphs, the performance of all methods are very close to each other when the SNR of the signal is 30 dB. However, when the SNR of the signal is decreased to 20 and 10 dB, the average spectral mismatch for the M-CLP method with two and three cycles is decreased slightly by 0.2-0.3 dB on the average relative to the one obtained by the CLP method. As a result, the mean of the average spectral mismatch is very close to that of the autocorrelation method when the SNR of the signal is 20 and 10 dB. Furthermore, as in the case for the partially-voiced speech signals, the standard deviation of the average spectral mismatch in the M-CLP analysis also decreases with the increasing number of cycles in the analysis and becomes very close to that of the autocorrelation method. This observation shows that the M-CLP analysis method is as reliable as the autocorrelation method for the short pitch cycles when the analyzed signal is a noisy signal.

4.5.2 Pulse Excited-Circular Linear Prediction Method

In the PE-CLP analysis, it is assumed that the excitation signal has impulses at known locations within the pitch cycle, whose amplitudes are significantly larger than the rest of the samples' amplitudes. This assumption results in a CLP formulation that neglects the impulses with large amplitudes in the excitation signal in the prediction-error minimization. In addition, both the predictor coefficients and the pulse amplitudes are optimized simultaneously in this method. This approach is similar to a version of the MP-LP method that simultaneously finds the predictor coefficients and amplitudes of the pulses. However, this new method also makes circular processing of the signal. In this initial implementation, it is assumed that there is only a single pulse in the excitation signal and its location is

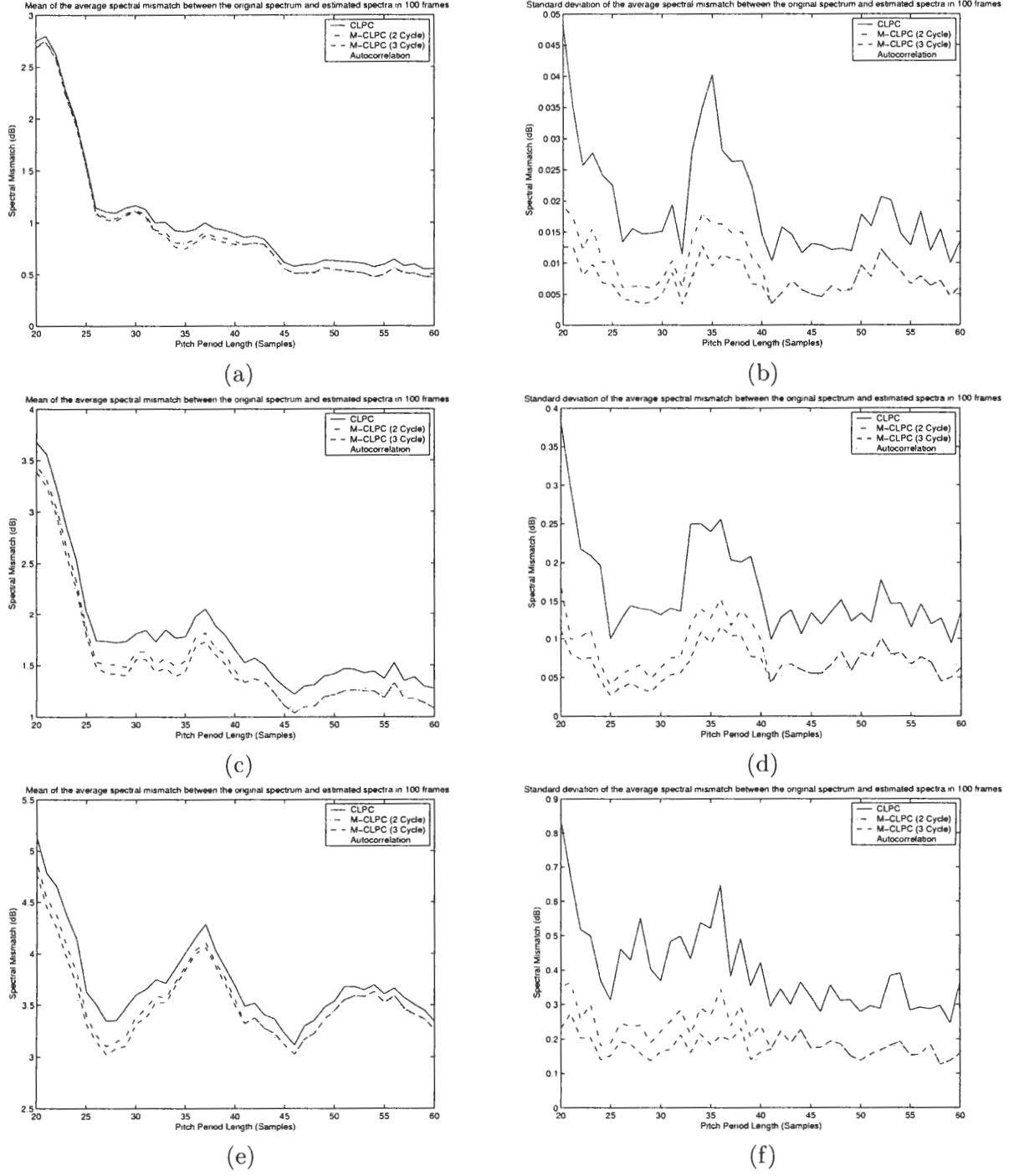


Figure 33: The mean (a,c,e) and the standard deviation (b,d,f) of the spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the M-CLP method with two and three cycles and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for noisy speech signal. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.

known. This assumption, together with the circular signal analysis, modifies (8) into

$$\hat{x}[n] = \sum_{k=1}^p a_k x[(n-k)_\tau] + G\delta[(n-n_i)_\tau]. \quad (88)$$

The unknown coefficients in this formulation are the predictor coefficients and the amplitude of the pulse at the sample n_i in the excitation signal. This technique is denoted as single pulse excited-CLP (SPE-CLP) method. Minimization of the sum of the squared prediction error with respect to predictor coefficients and the amplitude of the pulse results in the following set of equations:

$$\begin{aligned} \sum_{k=1}^p a_k r(|k-l|) + Gr_{dx}(l) &= r(l) \quad l = 1, \dots, p \\ \sum_{k=1}^p a_k r_{dx}(k) + G &= r_{dx}(0), \end{aligned} \quad (89)$$

where $r_{dx}(i)$ is defined as

$$r_{dx}(i) = \sum_{n=0}^{\tau-1} \delta[(n-n_i)_\tau] x[(n-i)_\tau] = x[(n_i-i)_\tau]. \quad (90)$$

These linear equation can be written in a matrix representation as

$$\begin{bmatrix} \mathbf{R} & \mathbf{r}_{dx} \\ \mathbf{r}_{dx}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ G \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ r_{dx}(0) \end{bmatrix}, \quad (91)$$

where \mathbf{R} , \mathbf{r} , and \mathbf{a} are defined in (12), and \mathbf{r}_{dx}^T is equal to the vector, $[r_{dx}(1) \dots r_{dx}(p)]$. The solution of (91) results in both prediction coefficients and the amplitude of the impulse. The main drawback of this algorithm is the loss of the Toeplitz structure in the matrix. Although the equation can still be solved by efficient techniques like Cholesky decomposition, the stability of the filter is no longer guaranteed.

The performance of the SPE-CLP method was evaluated using the experimental setup described in Section 4.3. Despite the use of a zero-mean impulse train in the excitation signal, the SPE-CLP analysis performed exceptionally well in all vocal-tract configurations. To illustrate its performance, Figure 8 is repeated again as Figure 34 including the results of the SPE-CLP analysis, which is obtained using the speech spectrum with uniformly spread formants for pitch-cycle lengths between 20 and 160 samples. The average and the

maximum spectral mismatch for the speech spectrum with grouped formants is also shown in Figure 35. Using the same speech spectrum, the true spectrum and the estimated spectra obtained by the autocorrelation method and the SPE-CLP method are also displayed in Figure 36 for a signal with a pitch-cycle length of 25 samples. The estimated spectrum by the SPE-CLP method is very close to the true spectrum. In this test, it was also observed that the synthesized speech converges to the original much faster in the first two synthesis methods when prediction coefficients are obtained by SPE-CLP analysis. It is even possible to achieve 72 dB reconstruction SNR with the first synthesis method when a speech spectrum with a very narrow-bandwidth first formant is coupled with a short pitch cycle.

In the tests with purely-voiced synthetic speech, the instabilities in the prediction filter were encountered only when the bandwidth of the first formant was very narrow and the pitch-cycle length was very short ($\tau < 25$ samples). In these cases, moving the poles from outside to inside of the unit circle by taking the reciprocal of their magnitudes solved the problem. Furthermore, by slightly sacrificing the performance, applying the pre-emphasis filter solved all of these stability issues in the test set.

To understand the properties of the SPE-CLP method better, the performance tests for the partially-voiced speech signals were repeated for the SPE-CLP method using the test setup described in Section 4.3. In this experiment, it was observed that the SPE-CLP method is very sensitive to noise in the excitation signal. Even when the transition frequency is 3000 Hz, the average spectral mismatch is much larger than that of both the autocorrelation method and the CLP method. This observation is illustrated in Figure 37. In this figure, the graphs on the left and right column show the mean and the standard deviation of the average spectral mismatch obtained using the SPE-CLP method and the autocorrelation method for the pitch-cycle lengths between 20 and 160 samples, respectively. The three rows are the results from three transition frequencies in decreasing order. From these graphs, it can be seen that the average spectral mismatch is much higher than that of the autocorrelation method, especially when the transition frequencies are 2000 and 3000 Hz. In addition, on the contrary to the CLP method, there is only a slight decrease in the

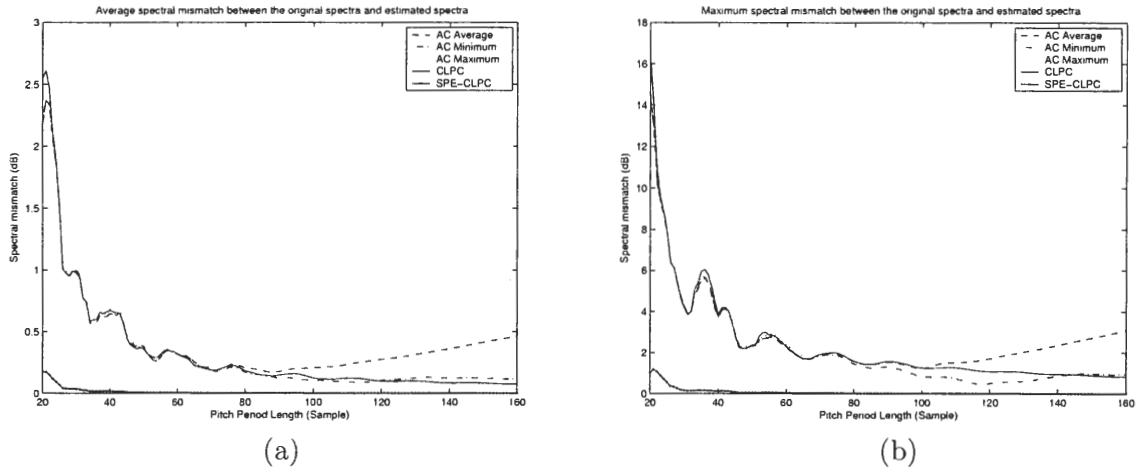


Figure 34: The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the SPE-CLP method and the autocorrelation method for pitch-cycle lengths between 20 and 160 samples. The speech spectrum is the “uniformly spread formants” example.

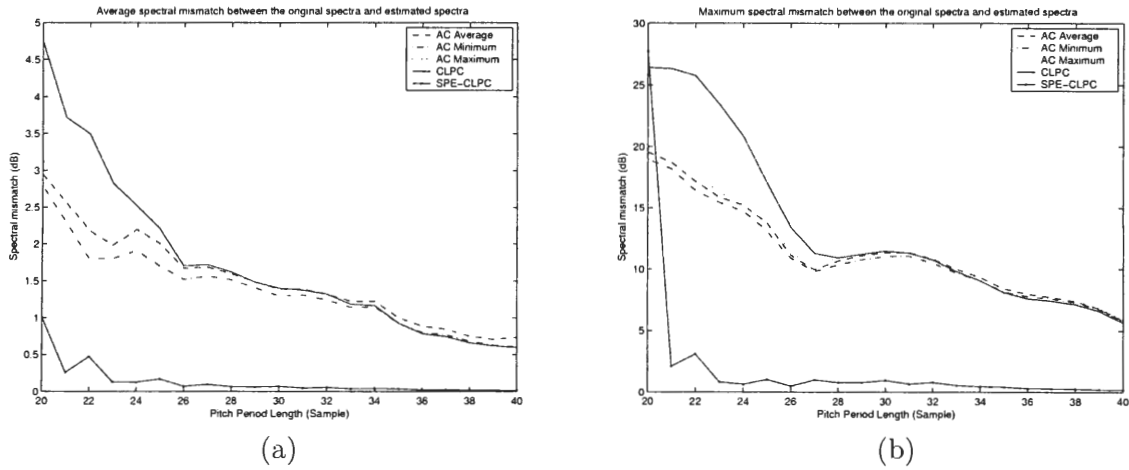


Figure 35: The average (a) and the maximum (b) spectral mismatch between the true spectrum and the estimated spectra obtained by the CLP method, the SPE-CLP method and the autocorrelation method for pitch-cycle lengths between 20 and 40 samples. The speech spectrum is the “grouped formants” example.

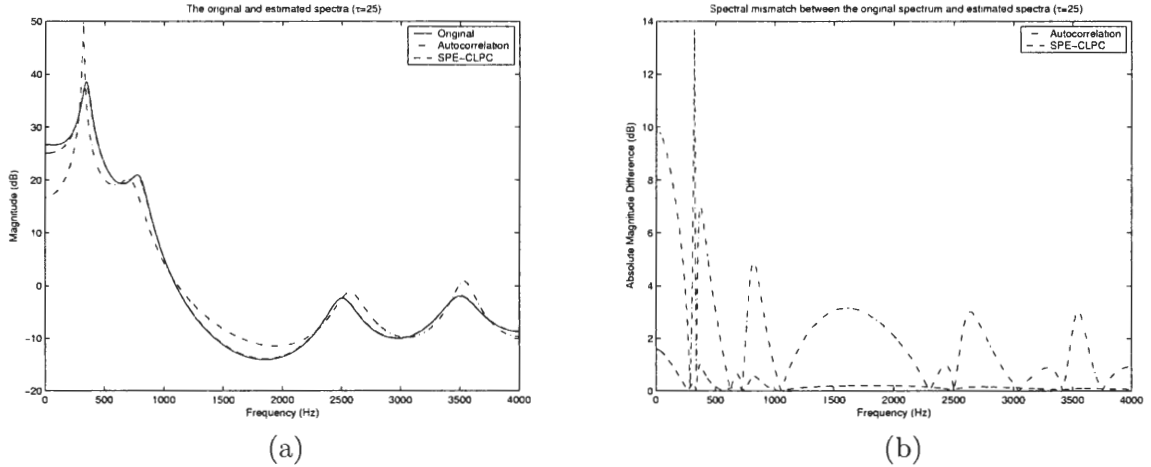


Figure 36: The true spectrum with grouped formants and a first formant with narrow bandwidth and the estimated spectra obtained by the SPE-CLP method and the autocorrelation method when the pitch period is 25 samples (a), and the error between the true spectrum and the estimated spectra by the SPE-CLP and the autocorrelation methods when the pitch period is 25 samples (b).

average spectral mismatch with increasing pitch-cycle length. However, when the transition frequency is 1000 Hz, the average spectral mismatch of the SPE-CLP method is similar to that of the CLP method. This result is expected since the excitation signal is very noisy in this case and omitting the squared error of a single pulse does not change the sum of squared prediction error. The effect of transition frequency on the mean of the absolute spectral mismatch is illustrated in Figure 38a. The spectral estimation performance variation of the SPE-CLP method is also higher than that of both the CLP and the autocorrelation method as shown in Figure 37. Furthermore, the standard deviation of the average spectral mismatch is always larger than that of the autocorrelation method even for the pitch-cycles longer than 100 samples, when the transition frequency is 3000 and 2000 Hz. These results proved that although the performance of SPE-CLP method is much better than those of both the autocorrelation and the CLP methods for the fully-voiced speech signals, it is very fragile when the excitation signal has noise in some of the bands of the spectrum as in the real-speech signals.

The characteristic of the estimated spectrum by the SPE-CLP method is also similar to that of the autocorrelation method and the CLP method for partially-voiced speech signals. The estimation error is usually higher in the frequency bands with noise, especially around

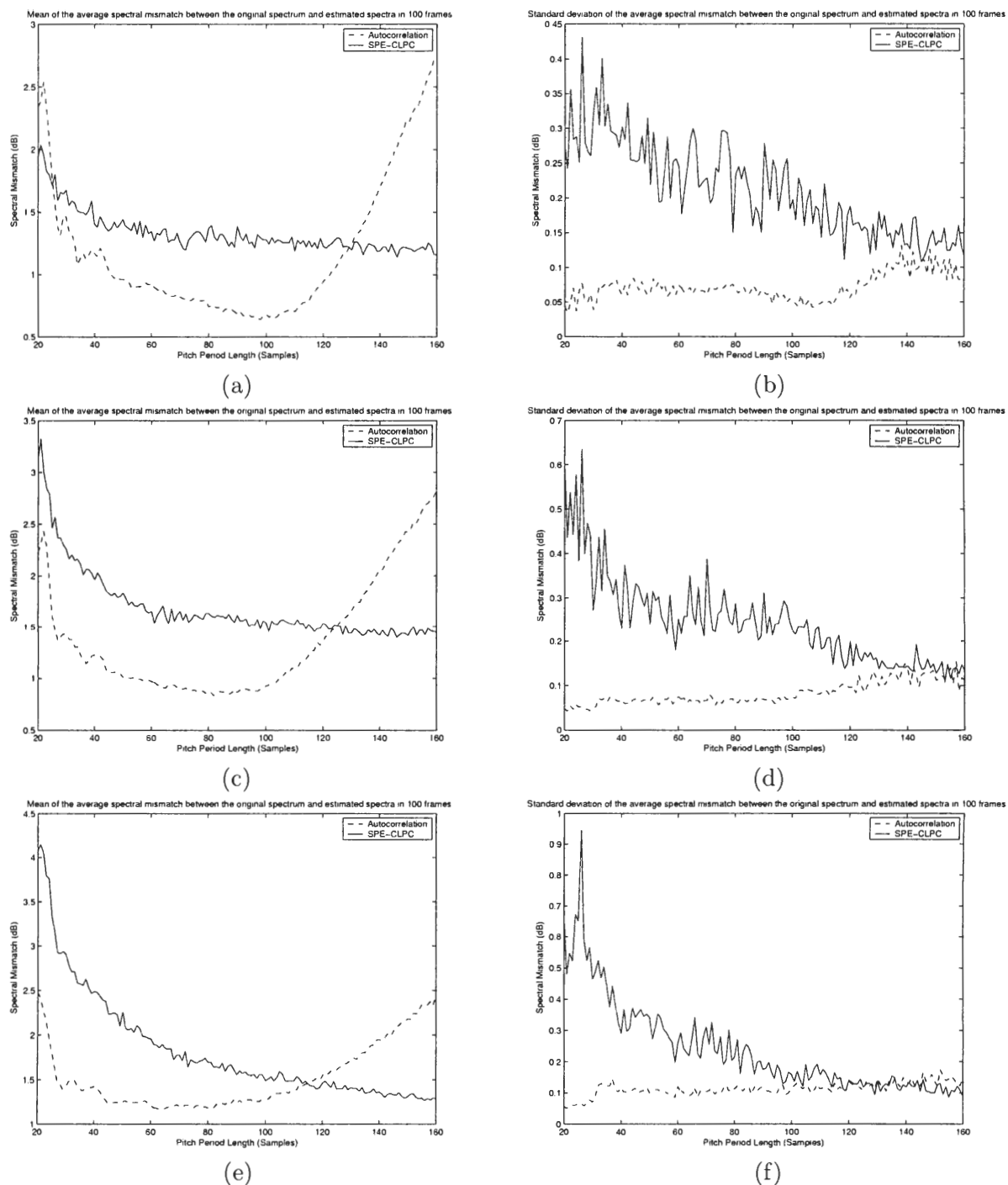


Figure 37: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for various partially-voiced speech signals. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.

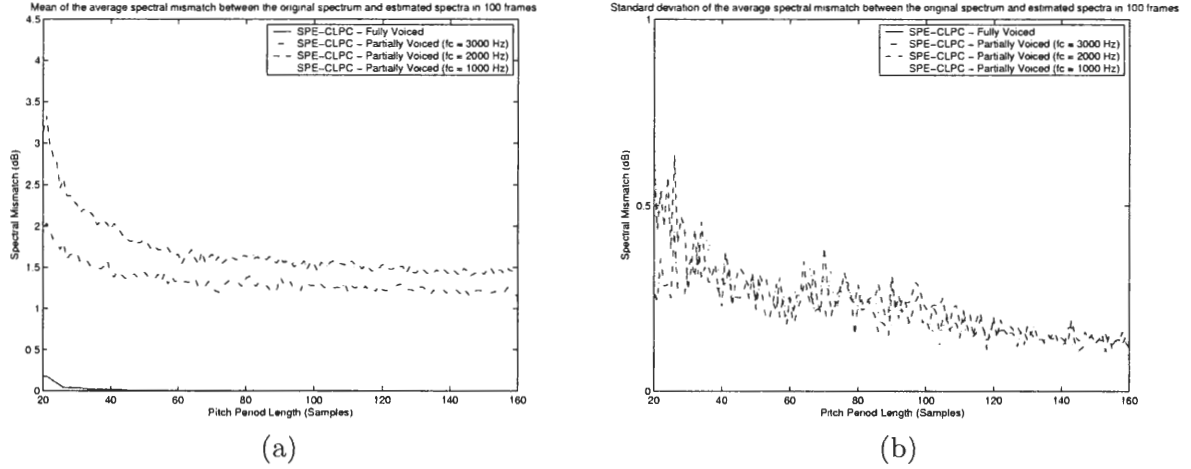


Figure 38: The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method for analysis frame lengths between 20 and 160 samples when the analyzed signals are purely-voiced and various partially-voiced speech signals.

the formant frequencies. It was also observed that absolute spectral mismatch obtained by the SPE-CLP method is larger than that obtained by the autocorrelation method in general. Figure 39 illustrates the estimated spectra obtained by the autocorrelation method and the SPE-CLP method, and the absolute spectral difference between the true spectrum and the estimated spectra by both methods.

As a final experiment for the SPE-CLP method, the performance of this method was evaluated for the noisy speech signals using the same test setup. This experiment revealed that the SPE-CLP method is not sensitive to the noisy speech as it is to the partially-voiced speech. Similar to the clean speech case, the performance of the SPE-CLP method is still better than those of both the CLP and the autocorrelation methods when the SNR of the signal is larger than 20 dB. However, the performance of the autocorrelation method also approached that of the SPE-CLP method when the SNR of the analyzed signal is decreased. The left column of the Figure 40, which displays the mean of the average spectral mismatch, illustrates these observations. However, as in the case of both the CLP and the autocorrelation methods, the average spectral mismatch increases with decreasing SNR as shown in Figure 41a. In this experiment, it was also observed that the spectral estimation performance variation is slightly better than that of the CLP method and close to the

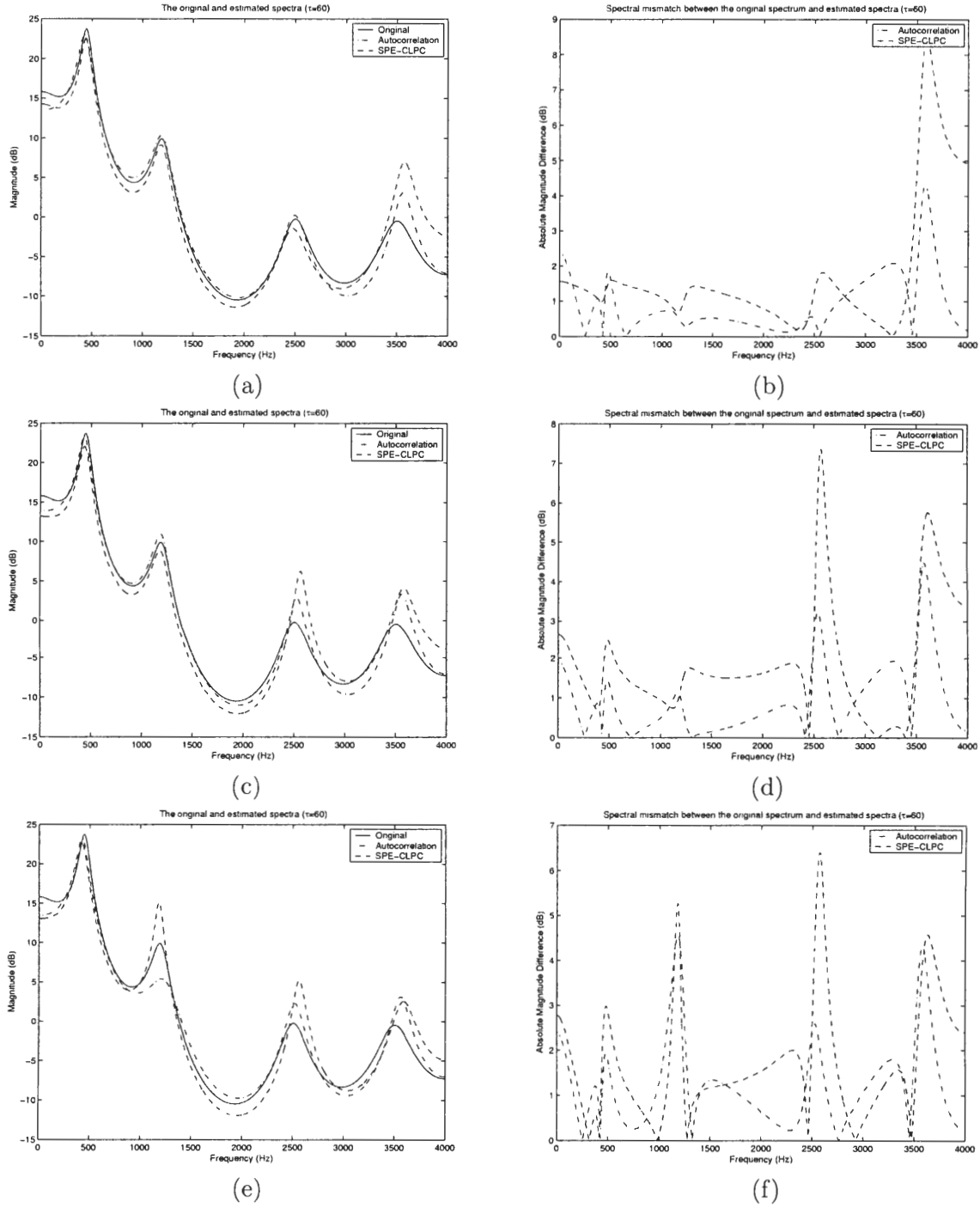
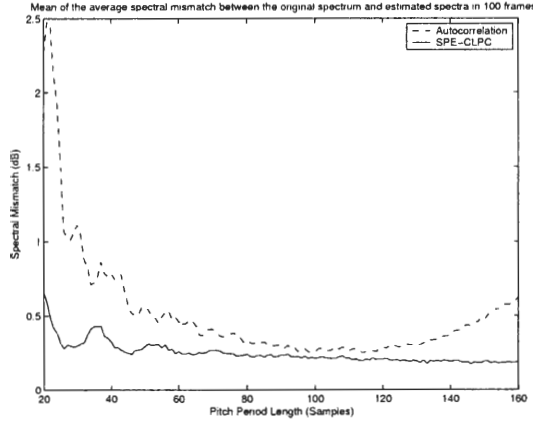


Figure 39: The speech spectrum with uniformly spread formants and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods when the pitch period is 60 samples and the transition frequencies are 3000 Hz(a), 2000 Hz(c) and 1000 Hz(e), and the absolute error between the true spectrum and the estimated spectra when the pitch period is 60 samples and the transition frequencies are 3000 Hz(b), 2000 Hz(d) and 1000 Hz(f).

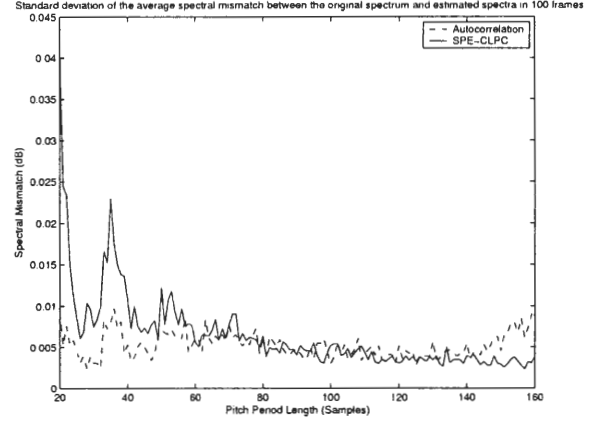
autocorrelation method when the pitch-cycle length is longer than 50 samples. However, the standard deviation of the average spectral mismatch also increases with decreasing SNR as in the case of other methods. This observation is also illustrated in Figure 41b.

To investigate the spectral estimation properties of the SPE-CLP method for noisy speech signals, the speech spectrum with uniformly spread formants and a pitch-cycle length of 60 samples were used as in the previous cases. The result of this analysis is displayed in Figure 42. In this figure, the graphs on the left column show the true and the estimated spectra obtained by the autocorrelation and the SPE-CLP methods and the graphs on the right column display the absolute spectral mismatch between the true spectrum and the estimated spectra. The three rows are the results for the signals with different SNRs in decreasing order. The graphs that display the results for 30 and 20 dB cases prove the results discussed above. In both cases, the estimated spectra by both the autocorrelation method and the SPE-CLP method are sufficiently close to the true spectrum. However, when the SNR is decreased to 10 dB, it is observed that while the autocorrelation method estimates the true spectrum better in the high-frequency region because of the longer analysis frame, the SPE-CLP method estimates the spectrum better in the low-frequency region because of the higher energy of the signal in these regions. This result implies that the SPE-CLP method models the perceptually important frequency regions with high energy better. As a result, the SPE-CLP method may perform better than the autocorrelation method and the CLP method for noisy speech signals.

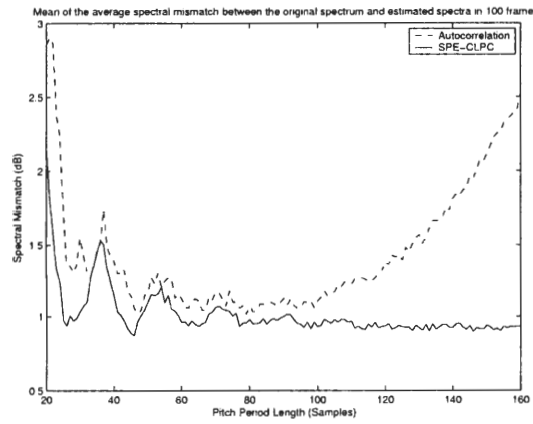
The SPE-CLP method can also be generalized into a multicycle SPE-CLP method (M-SPE-CLP) by minimizing of the summation of the multiple cycles' sum of squared prediction error. As in the case for the CLP method, a single set of linear prediction coefficients are obtained for the prediction of multiple infinitely periodic signals. However, as the amplitude of the known pulse location in each excitation signal is also an unknown for the M-SPE-CLP method, the number of unknowns increases with the number of cycles used in M-SPE-CLP analysis. It is also possible to constrain the unknown amplitudes of the single pulse in all pitch cycles to be same. However, the filter coefficients may not be optimum in this case, and the algorithm may be sensitive to energy variations in the signal.



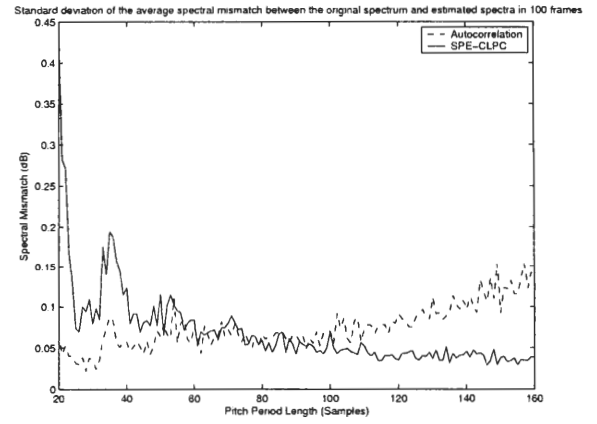
(a)



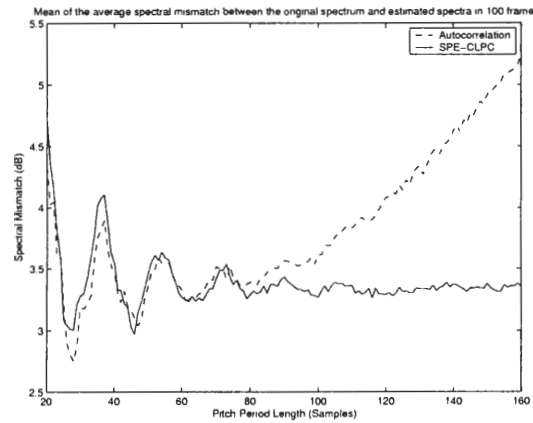
(b)



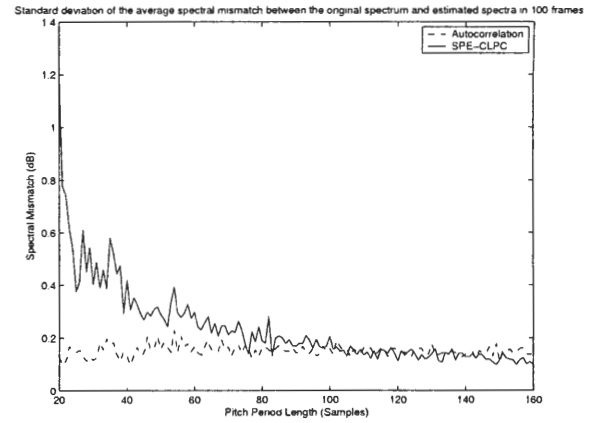
(c)



(d)



(e)



(f)

Figure 40: The average (a,c,e) and the maximum (b,d,f) spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods for pitch-cycle lengths between 20 and 160 samples for noisy speech signals. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.

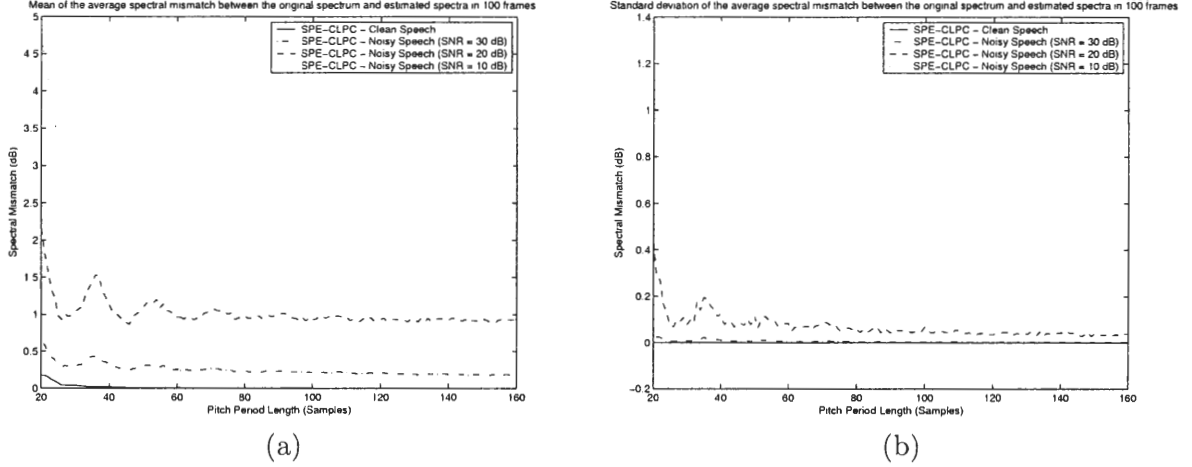


Figure 41: The mean (a) and the standard deviation (b) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method for analysis frame lengths between 20 and 160 samples when used on clean speech and various noisy speech signals.

As in the case for the M-CLP method, the derivation starts with the definition of prediction signal for multiple cycles:

$$\hat{x}_c[n] = \sum_{k=1}^p a_k x_c[((n-k))_{\tau_c}] + G_c \delta[((n-n_{ci}))_{\tau_c}] \quad c = 1, \dots, C, \quad (92)$$

where G_c and n_{ci} are the amplitude and the location of the single pulse in the c^{th} excitation signal, τ_c is the length of the c^{th} pitch cycle, and C is the number of cycles to be modeled with the same set of predictor coefficients. The minimization of the sum of the squared prediction error with respect to predictor coefficients and the amplitudes of the known pulses results in the following set of equations:

$$\begin{aligned} \sum_{k=1}^p a_k \sum_{c=1}^C r_c(|k-l|) + \sum_{c=1}^C G_c r_{cdx}(l) &= \sum_{c=1}^C r_c(l) \quad l = 1, \dots, p \\ \sum_{k=1}^p a_k r_{cdx}(k) + G_c &= r_{cdx}(0) \quad c = 1, \dots, C, \end{aligned} \quad (93)$$

where $r_{cdx}(i)$ is defined as

$$r_{cdx}(i) = \sum_{n=0}^{\tau_c-1} \delta[((n-n_{ci}))_{\tau_c}] x_c[((n-i))_{\tau_c}] = x_c[(n_{ci}-i)_{\tau_c}], \quad (94)$$

and $r_c(i)$ is defined in (87). To simplify the notation in the first set of equations in (93), it can be written as

$$\sum_{k=1}^p a_k \tilde{r}(|k-l|) + \sum_{c=1}^C G_c r_{cdx}(l) = \tilde{r}(l) \quad l = 1, \dots, p, \quad (95)$$

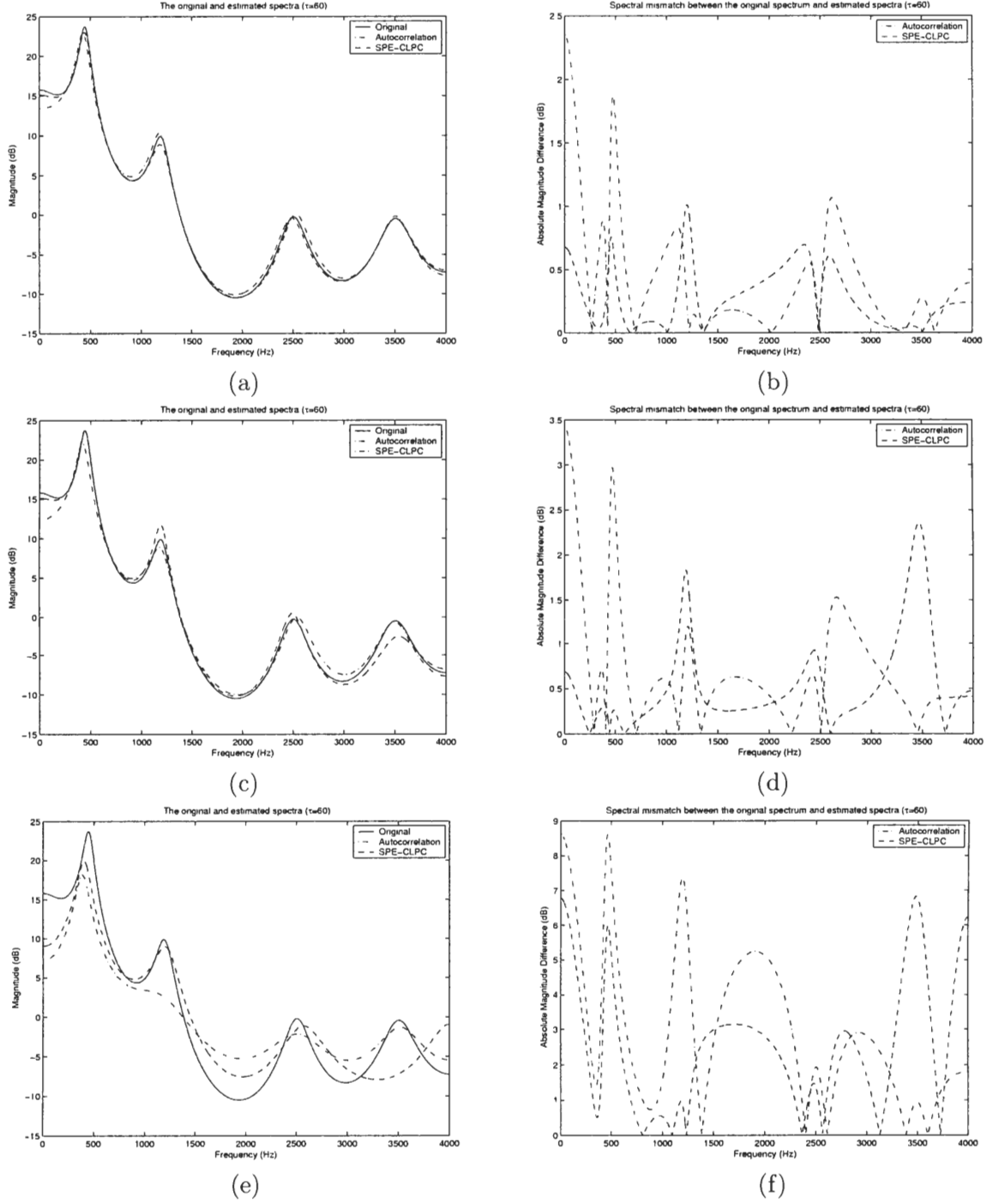


Figure 42: The true speech spectrum with uniformly spread formants and the estimated spectra obtained by the SPE-CLP and the autocorrelation methods when the pitch period is 60 samples and the SNR is 30 dB(a), 20 dB(b) and 10 dB(a), and the absolute error between the true spectrum and the estimated spectra when the pitch period is 60 samples and the SNR is 30 dB(d), 20 dB(e) and 10 dB(f).

where $\tilde{r}(i)$ is defined as

$$\tilde{r}(i) = \sum_{c=1}^C r_c(i). \quad (96)$$

These linear equations can be written in a matrix representation as

$$\begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{R}_{\mathbf{dx}} \\ \mathbf{R}_{\mathbf{dx}}^T & \mathbf{I}_C \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{G}_C \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}} \\ \mathbf{r}_{\mathbf{dx}}^0 \end{bmatrix}, \quad (97)$$

where $\tilde{\mathbf{R}}$ is the summation of the \mathbf{R} of all pitch cycles, \mathbf{I}_C is a c -by- c identity matrix, $\tilde{\mathbf{r}}$ is the summation of the \mathbf{r} of all pitch cycles, $\mathbf{r}_{\mathbf{dx}}^0$ is equal to the vector $[r_{1dx}(0) \dots r_{Cdx}(0)]^T$, \mathbf{G}_C is the vector of the amplitudes of the known pulses in the excitation signals such that $\mathbf{G}_C = [G_1 \dots G_C]^T$, $\mathbf{R}_{\mathbf{dx}}$ is a matrix whose columns are the $\mathbf{r}_{\mathbf{cdx}}$ such that $\mathbf{R}_{\mathbf{dx}} = [\mathbf{r}_{1\mathbf{dx}} \dots \mathbf{r}_{C\mathbf{dx}}]$, and $\mathbf{r}_{\mathbf{cdx}}$ is equal to the vector $[r_{cdx}(1) \dots r_{cdx}(p)]^T$. The solution of (97) results in the optimum predictor coefficients and the amplitudes of the pulses in the excitation signal of the pitch cycles.

As in the case for the M-CLP analysis method, the M-SPE-CLP method also improves the spectral estimation performance of the SPE-CLP method, especially for the partially-voiced speech and noisy speech signals. The improvements in the partially-voiced synthetic speech signals is particularly important because of the sensitivity of the SPE-CLP method to noisy excitation signals. To evaluate the performance of the M-SPE-CLP method, the same test setup described in Section 4.3 was used. Furthermore, similar to the M-CLP tests, the total number of the samples in the analyzed pitch cycles were restricted to 120 samples. As a result, the performance for only the pitch-cycle lengths between 20 and 60 samples were evaluated in the test for the M-SPE-CLP method using two cycles. In the tests for the M-SPE-CLP method using three cycles, the performance of the method was obtained for only the pitch-cycle lengths between 20 and 40 samples.

In the partially-voiced synthetic speech tests, the average spectral mismatch obtained by the M-SPE-CLP method using two cycles was 0.5-0.6 dB lower than the one obtained by the SPE-CLP method. When the M-SPE-CLP method using three cycles was used, the average spectral mismatch was further decreased by another 0.2-0.3 dB. However, the decrease in the mean of the average spectral mismatch with increasing pitch-cycle length

was still less than that obtained by the M-CLP method. Furthermore, the performance of the autocorrelation method was still better than that of the M-SPE-CLP method, especially when the transition frequency is equal to and less than 2000 Hz. Figure 43 illustrates these observations. In this figure, the graphs on the left and right column show the mean and the standard deviation of the average spectral mismatch obtained by the SPE-CLP method, the autocorrelation method and the M-SPE-CLP method using two and three cycles for the pitch-cycle lengths between 20 and 60 samples, respectively. The three rows are the results from three transition frequencies in decreasing order. In this experiment, it was also observed that the spectral estimation performance variation decreases by increasing number of the cycles in the analysis. Although the standard deviation of the average spectral mismatch obtained by the M-SPE-CLP method was not as small as that of the autocorrelation method, it was much less compared to the one obtained by the SPE-CLP method. From these observations, it can be concluded that the multicycle generalization of the SPE-CLP method improves the performance over the SPE-CLP method for partially-voiced speech, however the autocorrelation method and the M-CLP method are still better for this kind of speech signal. As a result, it is expected that the M-CLP method and the autocorrelation method are more suitable linear-prediction methods for modeling real-speech signals.

For noisy speech signals, the M-SPE-CLP method using two cycles slightly improved the performance of the SPE-CLP method that is already better than that of the autocorrelation method for the signals with a SNR higher than 20 dB. For the speech signals with 10 dB SNR, the M-SPE-CLP method using both two and three cycles had similar performance with the autocorrelation method. This observation is illustrated in Figure 44. Furthermore, the spectral estimation performance variation of the M-SPE-CLP method was also similar to the that of the autocorrelation method. As a result, the M-SPE-CLP method for short pitch cycles and the SPE-CLP method for long pitch cycles is a good alternative to replace the autocorrelation method for noisy speech signals.

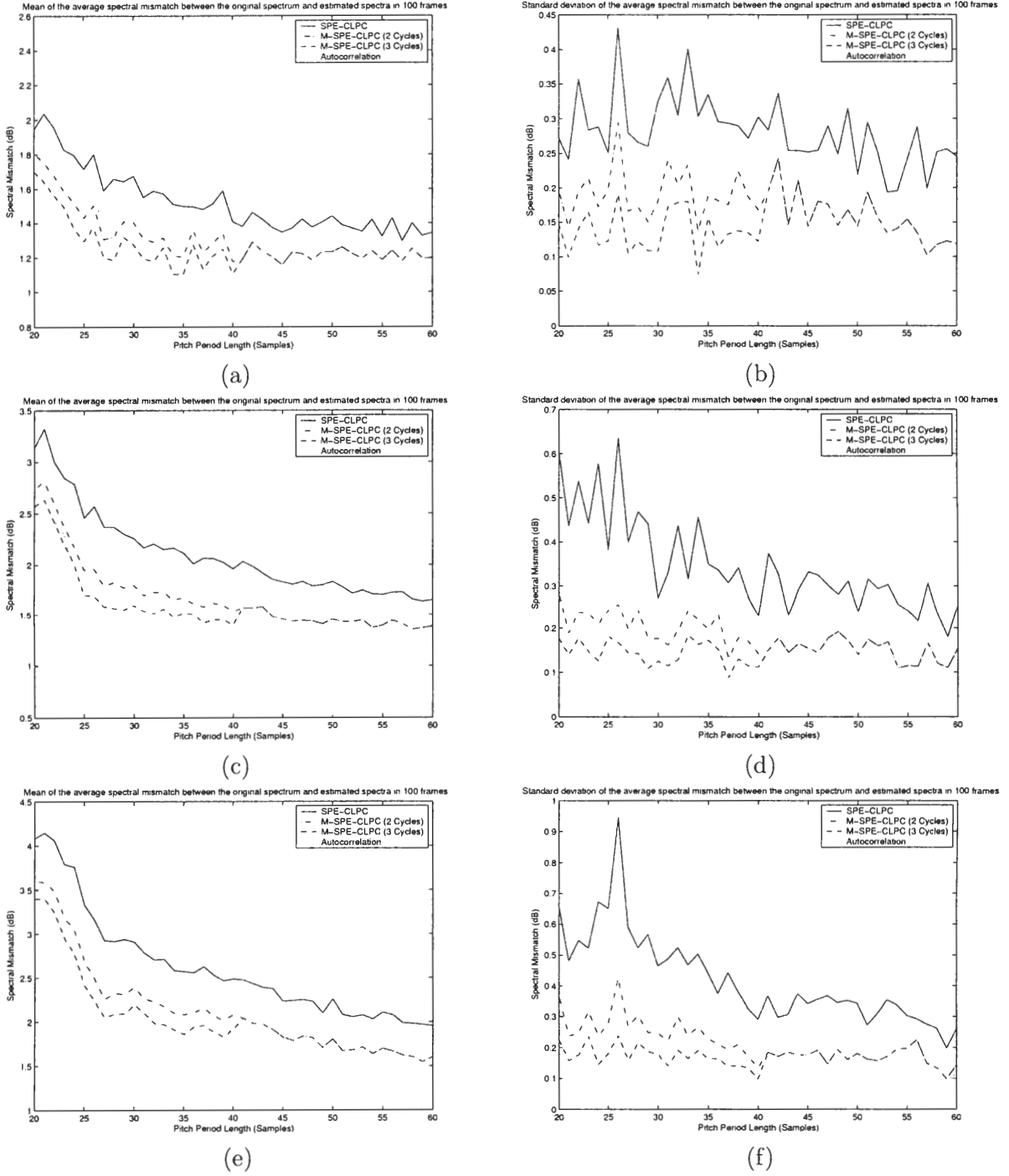


Figure 43: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method, the M-SPE-CLP method with two and three cycles, and the autocorrelation methods for pitch-cycle lengths between 20 and 60 samples for various partially-voiced speech signals. The transition frequencies are 3000 Hz(a,b), 2000 Hz(c,d) and 1000 Hz(e,f) in the first, second and third rows, respectively.

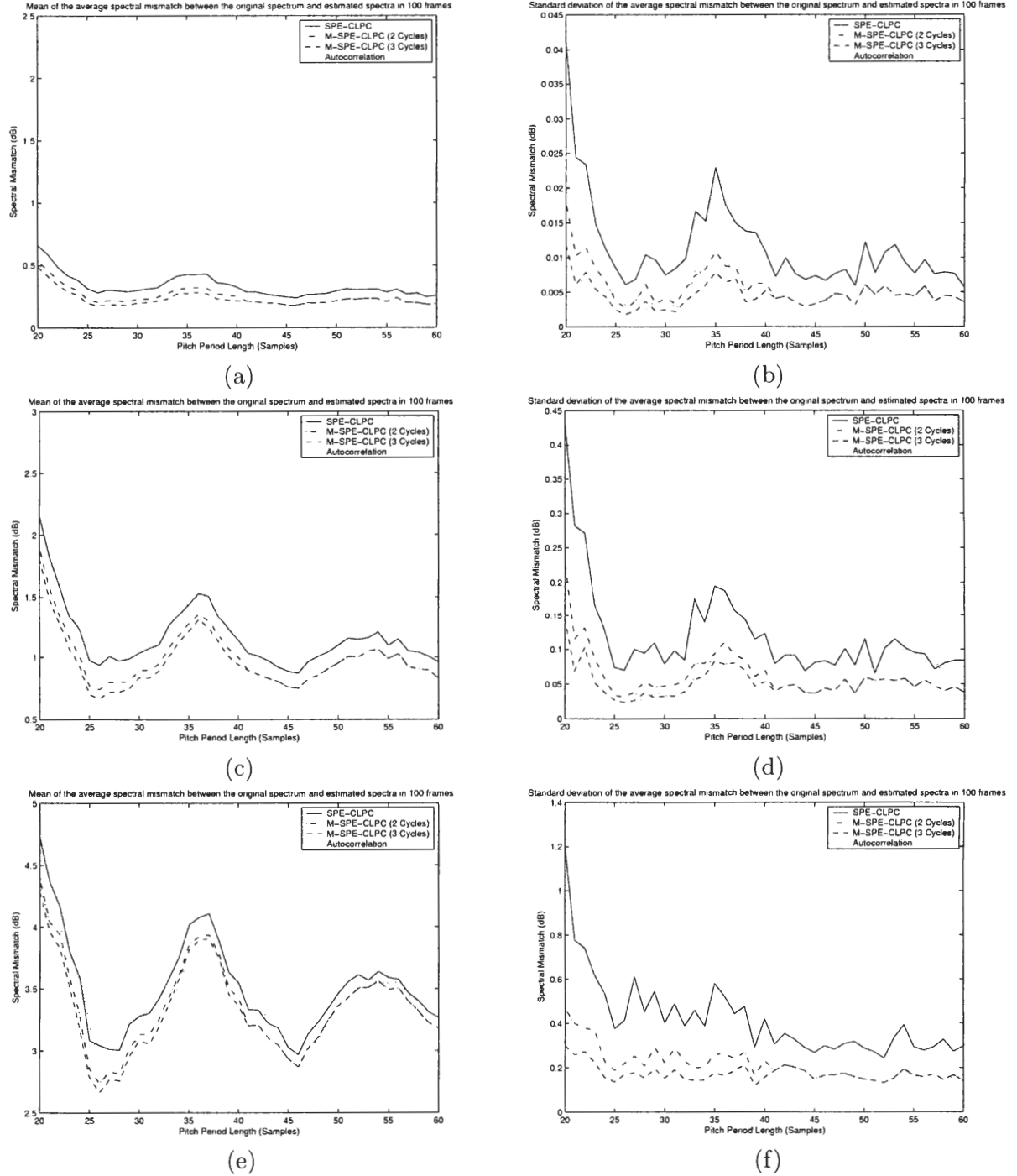


Figure 44: The mean (a,c,e) and the standard deviation (b,d,f) of the average spectral mismatch between the true spectrum and the estimated spectra obtained by the SPE-CLP method, the M-SPE-CLP method with two and three cycles and the autocorrelation method for pitch-cycle lengths between 20 and 60 samples for various noisy speech signals. The SNR of the signals are 30 dB(a,b), 20 dB(c,d) and 10 dB(e,f) in the first, second and third rows, respectively.

4.6 *Linear Prediction of Real-Speech Signals Using the CLP Analysis*

In the previous sections, the theory behind the CLP method and its variations were explained and the performance of each variation was evaluated using various types of synthetic speech signals. These experiments showed the weaknesses and strengths of the proposed methods relative to the autocorrelation method. Furthermore, these results also gave insight into how well these algorithms might perform for real-speech signals. In this section, the modeling of real-speech signal using the CLP method and its variations is discussed. The problems with using the CLP method are discussed and possible solutions are proposed. In addition, the results of the performance tests for real-speech signal are also presented and compared to the ones obtained from the synthetic-speech tests.

The CLP method assumes that the analyzed signal is a perfectly periodic signal. However, because of the changes in cycle length and speech spectrum, and because of the noise in the excitation signal, the speech signal is only *quasi-periodic*. It is sometimes very hard even to visually define the exact boundaries of each pitch cycle in the waveform of the signal. As a result, to obtain the linear-prediction coefficients reliably using the CLP method, the boundaries of the pitch cycles must be selected carefully. One way to achieve this is to use the segmentation algorithm proposed in Section 3.2.1, based on the prediction gain maximization. However, occasional segmentation errors also result in faulty estimation of the linear-prediction coefficients. Furthermore, since that algorithm is sequential, the probability of segmentation errors also increases after a segmentation error has occurred within a voiced segment. The other solution is to use the segmentation algorithm discussed in Section 3.2.2, based on normalized correlation maximization. Although this algorithm is reliable, it is primarily designed for segmenting the residual signal for pitch-cycle modifications. As a result, the cycle boundaries are forced to be in the low-energy sections of the residual signal, which makes it likely that they are in the middle of actual pitch cycles. Therefore, these cycle boundaries are not suitable for the CLP method.

As the CLP method requires an exact periodic signal, the real-speech segment that is best suitable for a CLP analysis should have the following properties:

- If a segment of a speech signal, $x[n]$, starts at the sample, n_s , and the length of the segment is τ_s samples, $x[n_s]$ must be equal to $x[n_s + \tau_s]$ and $x[n_s + \tau_s - 1]$ must be equal to $x[n_s - 1]$. In other words, the amplitude of the first sample of the segment must be equal to the amplitude of the sample just after the last sample of the segment, and the amplitude of the last sample of the segment must be equal to the amplitude of the sample just before the first sample of segment.
- The normalized correlation between the segment including the samples between n_s and $n_s + \tau_s - 1$ and the next one, including the samples $n_s + \tau_s$ and $n_s + 2\tau_s - 1$, must be equal to one. In other words, for a given starting point, n_s , the length of the segment, τ_s , must be selected such that the normalized correlation between the current segment and the next one is equal to one.

For a perfectly periodic signal, the shortest segment that satisfies these properties is a single cycle of this signal. However, the pitch cycles in a critically-sampled narrowband speech signal do not exactly have these properties. The length of a pitch cycle in such signals usually has a fractional part that makes it impossible to find segment boundaries that would satisfy the first property. Furthermore, it is still impossible to find pitch cycles that satisfy the first property, even if the sampling rate is increased by a factor of ten. In addition, even if the pitch period has an integer length, the noise in the partially-voiced speech signals also prohibits the pitch cycles from having both properties. As a result, the best cycle boundaries for the CLP analysis are the ones that satisfy these two properties most closely. These boundary locations can be obtained by the following algorithm.

The speech signal is first upsampled by a factor of ten to generate $x_N[n]$ (which is also required for the CLP analysis,) and then segmented using the segmentation algorithm based on the maximization of the normalized correlation described in Section 3.2.2. Then, the boundaries of each pitch cycle are adjusted separately for the CLP analysis as follows:

- Assume that $x_N[n_c]$ is the first sample of the analyzed pitch cycle whose cycle boundaries are obtained by the segmentation algorithm, and the length of the cycle is τ_c samples at the upsampled rate.

- The samples between $n_c - \frac{\tau_c}{2}$ and $n_c + \frac{\tau_c}{2}$ of $x_N[n]$ are the initial candidates for the new starting location of the pitch cycle, denoted as n_d . For each starting-location candidate, the normalized correlation between the current and next cycles are calculated for the lags between $\tau_c - \Delta_c$ and $\tau_c + \Delta_c$ and the pitch lag that maximizes the normalized correlation coefficient is selected as the pitch-cycle length, τ_d , for the starting location n_d .
- For each starting location candidate, an absolute amplitude mismatch is calculated as the summation of the absolute mismatch between the amplitude of the first sample of the cycle and that of the sample just after the last sample of the cycle, and the absolute mismatch between the amplitude of the last sample of the cycle and that of the sample just before the first sample of the cycle:

$$A_d = |x[n_d] - x[n_d + \tau_d]| + |x[n_d + \tau_d - 1] - x[n_d - 1]|. \quad (98)$$

- The M candidate sample locations with the lowest A_d are selected as the final candidates for the new starting location of the pitch cycle.
- For each final candidate, the prediction gain is calculated for the pitch cycles starting with the sample n_d and ending between $n_d + \tau_d - \Delta_p$ and $n_d + \tau_d + \Delta_p$. The cycle length that maximizes the prediction gain is denoted as τ_{dp} .
- The new starting location is selected as the final candidate starting location whose cycle length obtained by the normalized correlation maximization, τ_d , is the closest one to the cycle length obtained by the prediction gain maximization, τ_{dp} .
- If there are multiple final candidate locations that satisfy this criterion, the one closest to the initial starting location is selected as the new starting point.

To compare the performance of the single cycle and multicycle variations of the CLP and SPE-CLP methods and the autocorrelation method, ten phrases are selected from the TIMIT database, spoken by five male and five female speakers with different average pitch periods (6-11 ms for male speakers and 3.5-8 ms for female speakers.) This selection

provides sufficient speaker variation in the test set. The sampling rate of the sentences is 8 kHz. To remove the dc term and low-frequency noise, a fourth order, Chebyshev-Type II high-pass filter with cut-off frequency at 60 Hz is applied to the signal prior to the analysis. This filtered signal is upsampled by a factor of ten. Finally, the cycle boundary locations are extracted using the method described above. The linear-prediction coefficients are estimated for each segmented cycle using the following algorithms: the CLP method, the M-CLP method using two cycles, the M-CLP method using three cycles, the SPE-CLP method, the M-SPE-CLP method using two cycles, the M-SPE-CLP method using three cycles and the autocorrelation method. Before obtaining the prediction coefficient by the SPE-CLP method, a set of predictor coefficients is found using the CLP method and the pitch cycle is circularly filtered with the inverse of this prediction filter. The “known” sample location for the SPE-CLP method is obtained as the sample with the largest amplitude in this residual signal. For the autocorrelation analysis, a 25 ms Hamming window is placed on the center of the pitch cycle and the windowed signal is used in the analysis. Unvoiced frames are only analyzed with the autocorrelation method.

The performance tests in this section can be categorized into two groups: Objective performance evaluation and informal listening tests. In objective performance evaluation, the frequency response of the linear-prediction filters obtained by all methods were calculated and investigated to determine how well the methods fit the linear-prediction spectrum to the harmonics of the speech signal. The smoothness of the evolution of the estimated spectrum was also evaluated using the line spectrum frequencies (LSF) tracks obtained by each method. In addition, since the stability of the filter is not guaranteed for all variations of the SPE-CLP analysis, the instabilities in the estimated filter were also investigated. All pitch cycles labeled as voiced with the segmentation method were used in this test. The two-cycle multicycle method uses the pitch cycle just before the analyzed cycle, if available. In addition to the previous cycle, the three-cycle multicycle methods use the next cycle as well. In the informal listening tests, the autocorrelation analysis in the 2.4 kb/s military standard MELP coder was replaced by the proposed methods and the quality of the synthesized speech was evaluated. Since the predictor coefficients are estimated once a frame

of 22.5 ms, the pitch cycle that overlaps with center of the analysis frame was used for the CLP and SPE-CLP methods. The pitch cycles used in the multicycle variations were also selected in a similar fashion as in the objective tests. Finally, when the estimated filter obtained by the SPE-CLP method is unstable, the filter is replaced by the one estimated by the CLP method.

As discussed in the previous sections, the main advantage of the CLP method over the autocorrelation method is its ability to estimate the linear-prediction coefficients reliably from a single pitch cycle. This property of the CLP makes it a better analysis method for transition segments. In the experiments with real-speech signal, it was observed that the spectrum estimated by the CLP method always fits the speech harmonics better than that estimated by the autocorrelation method, especially at onsets. This observation is illustrated for a male speaker with average pitch period of 7 ms in Figure 45. Both pitch-cycle length and speech spectrum change very rapidly from one pitch cycle to the next in this example. As the autocorrelation method models the whole transition segment, the estimated spectrum does not fit the speech harmonics in all pitch cycles. However, because of its single cycle modeling, the CLP method models the spectrum much better, especially in the high-frequency region. In this figure, it is also observed that the spectrum estimated by the autocorrelation method is always closer to the spectrum of the pitch cycle with the largest energy. In stationary regions, the spectral estimates of both methods are always very close to the harmonics of the speech signal, and as a result, the spectra estimated by both methods are very close to one another as well. Figure 46, which is obtained one frame after the example given above, shows the spectrum estimation accuracy of both methods for a stationary segment.

The LSF tracks obtained by both methods for a speech segment including the first two examples are given in Figure 47. Except for the first few pitch cycles at the onset, the LSF tracks obtained by both methods are very similar. Furthermore, since the high-frequency region of the speech spectrum becomes noisy at the end of the word, the variations in the CLP method increase near the end of this segment. The performance of the CLP method is also very close to that of the autocorrelation method for short-pitch cycles in stationary

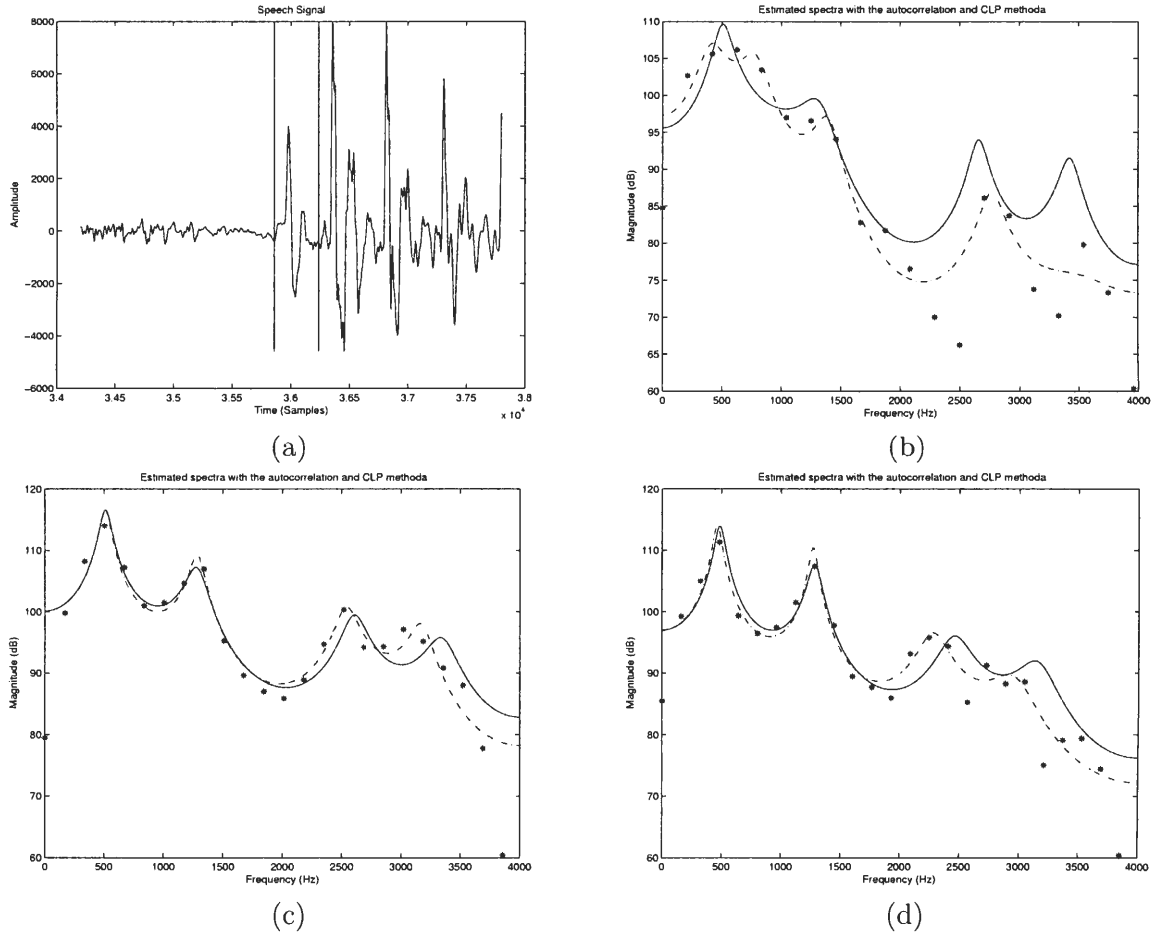


Figure 45: A real-speech signal that illustrates an onset (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for the first (between the vertical lines in (a)) (b), the second (c) and the third (d) pitch cycles. The lengths of the first, the second and the third pitch cycles are 38.4, 47.7 and 49.6 samples, respectively.

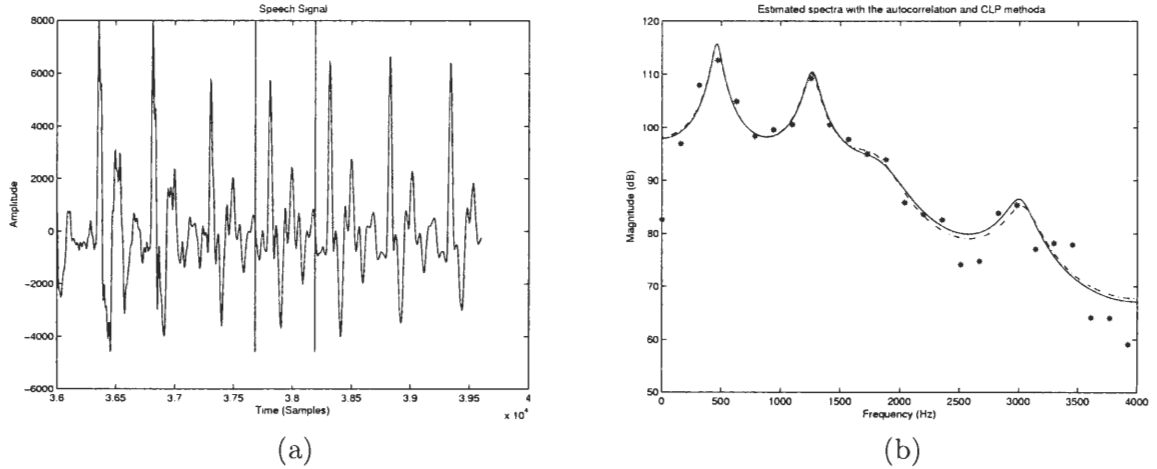


Figure 46: A stationary real-speech segment (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) (b). The length of the pitch cycle is 51 samples.

segments. Especially, if the speech signal is purely-voiced, the spectra estimated by both methods are always close to one another. Figure 48 illustrates an example for a high-pitched speaker. In this example, the average pitch-period is around 23 samples. The LSF tracks obtained by both methods for a segment that includes the short pitch cycle example are shown in Figure 49. Since the speech is purely voiced, there is no variation in the LSF track of the CLP method except a very slight variation in the 8th LSF track. This example proves that the same predictor coefficients can be obtained by using only one tenth of the number of samples used in the autocorrelation method. In all these examples, the CLP method proves to be a very good alternative to the autocorrelation method, especially in the transition regions. However, the performance variation of the CLP method for partially-voiced speech segments observed in the synthetic speech experiments is also seen in the tests with real-speech signal. Figure 50 illustrates this problem. Even for the stationary regions, the LSF tracks between 5 and 10 obtained by the CLP method oscillates around those obtained by the autocorrelation method. It is also important to note that these oscillations only occur for the LSFs that are in the noisy frequency region in the speech spectrum. Fortunately, when the multicycle variation is used instead of the single cycle, it is possible to obtain a smooth-evolving LSFs track as shown in Figure 51.

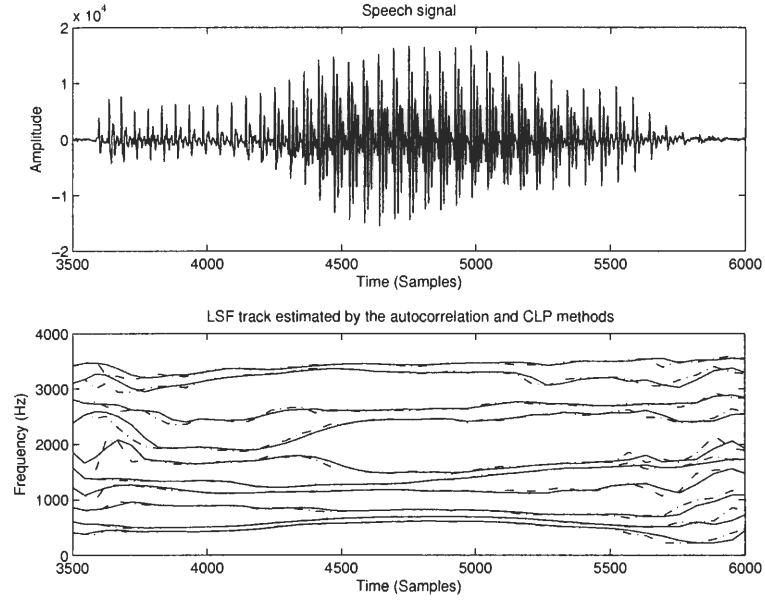


Figure 47: The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for displayed speech segment.

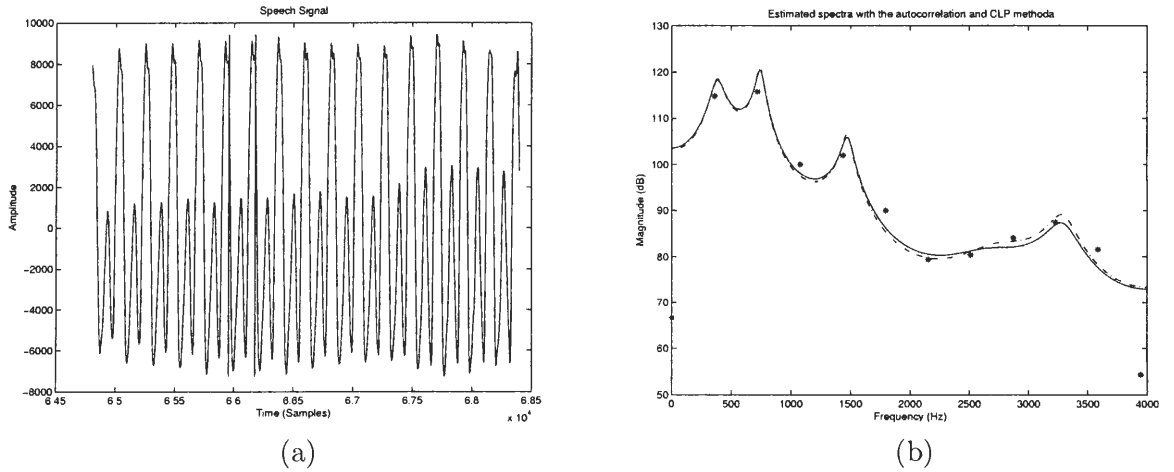


Figure 48: A stationary real-speech segment (a), and the spectra estimated by the autocorrelation method (solid line) and the CLP method (dash-dotted line) (b). The length of the pitch cycle is 22.8 samples.

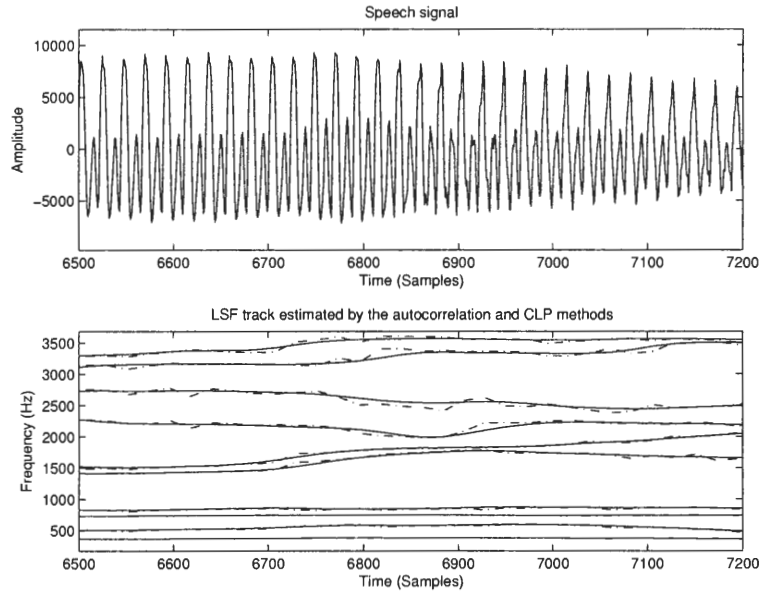


Figure 49: The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for the displayed speech segment.

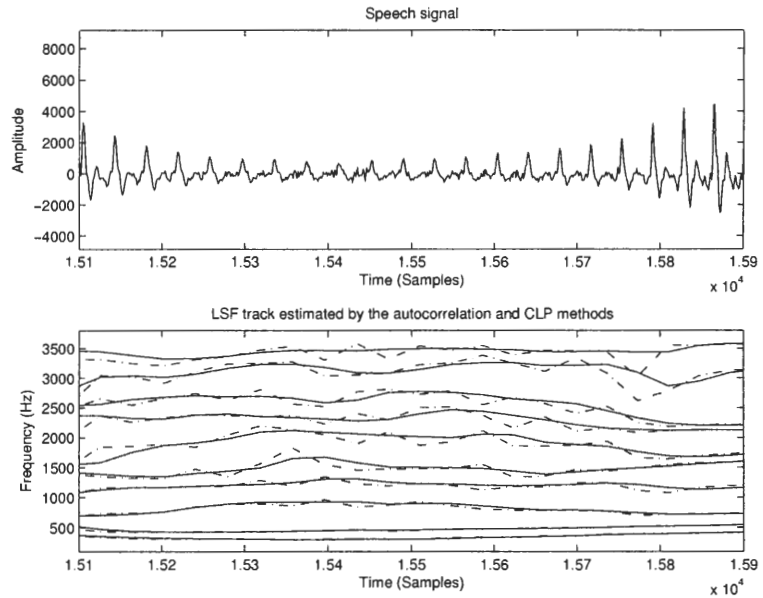


Figure 50: The LSF tracks obtained by the autocorrelation method (solid line) and the CLP method (dash-dotted line) for a partially-voiced speech segment.

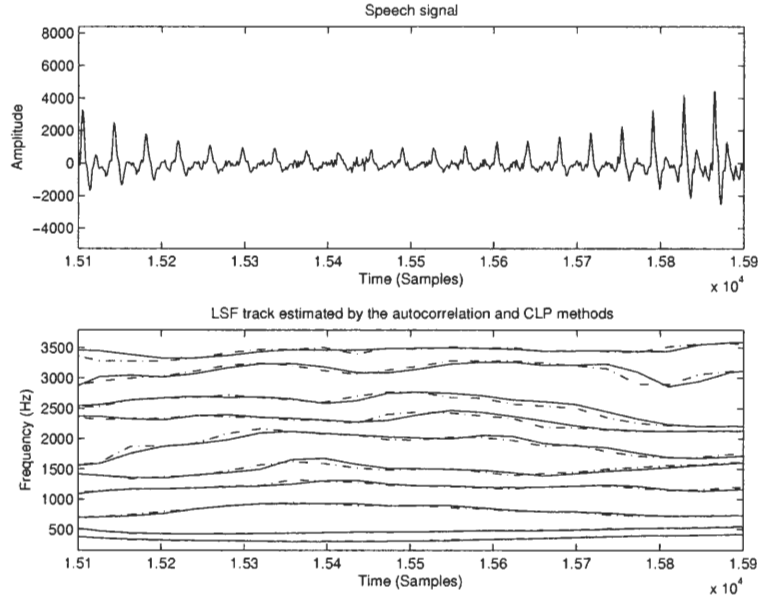


Figure 51: The LSF tracks obtained by the autocorrelation method (solid line) and the three-cycle M-CLP method (dash-dotted line) for the partially-voiced speech segment displayed in Figure 50.

The performance of the SPE-CLP method is similar to that of the CLP method for both transition and stationary regions. As a result, the SPE-CLP method also models the spectrum better than the autocorrelation method in the transition regions, and the performance variation is small for purely-voiced speech signals. In addition, it is also observed that the prediction filter estimated by the SPE-CLP method usually has a wider bandwidth than the ones estimated by the autocorrelation and CLP methods. This observation is illustrated in Figure 52 and Figure 53 for a male and a female speaker, respectively. In the first example, the spectra estimated by both methods are very similar, but careful investigation reveals that the spectrum estimated by the SPE-CLP method is a better fit to the speech harmonics. When the pitch period is short, as in the case for the second example, the autocorrelation and CLP methods tend to model the harmonics with relatively large magnitude using separate formant resonances with narrow bandwidths. However, for this case, the SPE-CLP method models the spectrum much better than the other methods, and the estimated formant bandwidths are much better (wider) than those of the autocorrelation and CLP methods. This estimation property of the SPE-CLP method reduces the distortion related to the spectral estimation.

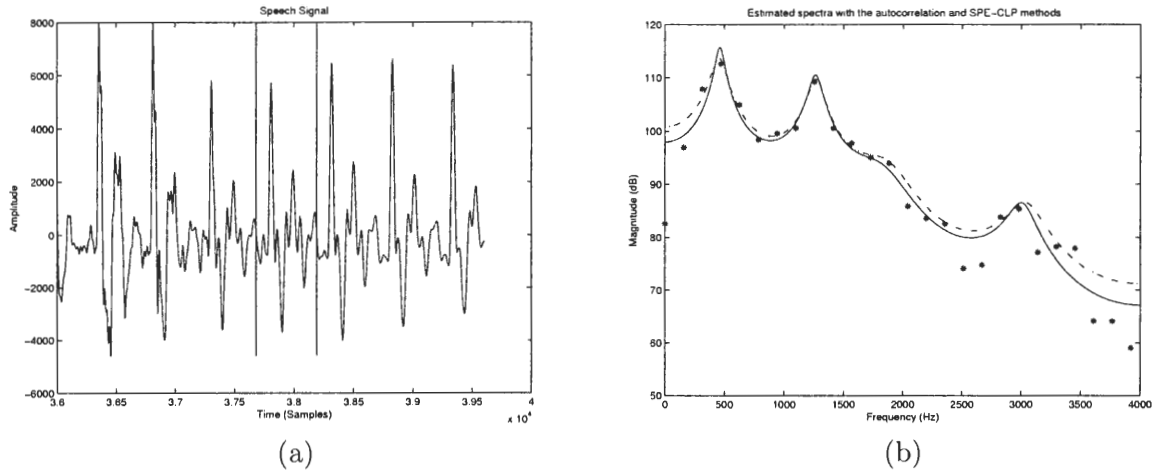


Figure 52: A stationary real-speech segment for a male speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 51 samples.

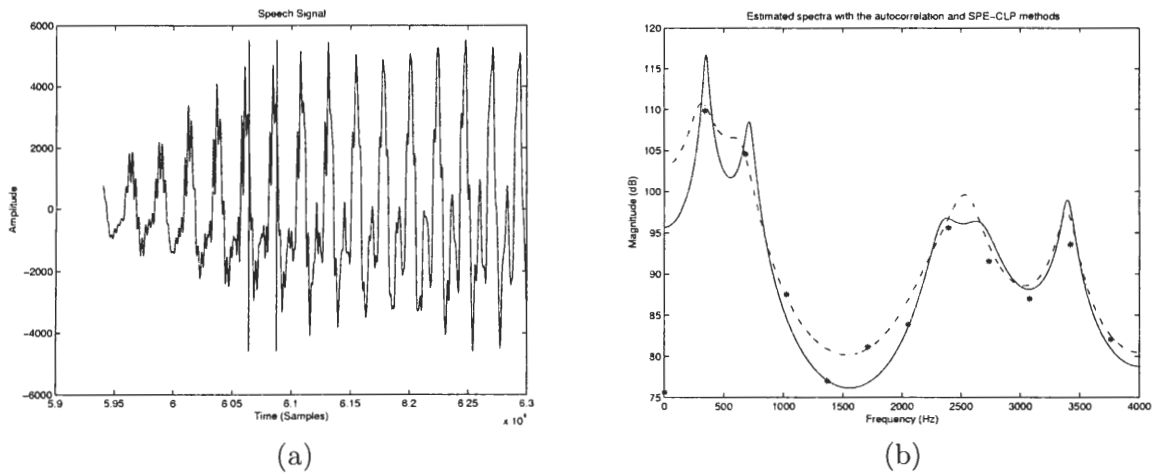


Figure 53: A stationary real-speech segment for a female speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 23.4 samples.

The spectral estimation characteristics of the SPE-CLP method also results in more peaky residual signal whose spectrum is almost flat. As an example, the residual signal obtained by the autocorrelation method and the SPE-CLP method is shown in Figure 54 for a high-pitched speaker. It is clearly seen that while the energy of the pitch cycles is distributed throughout the pitch cycle for the autocorrelation method, the energy is more concentrated around the pitch pulse for the SPE-CLP method.

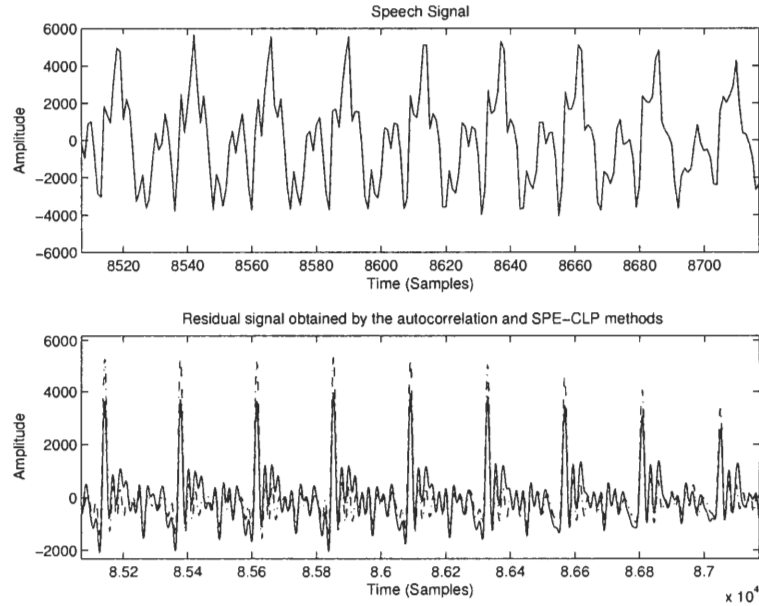


Figure 54: The residual signals obtained by the autocorrelation method (solid line) and the SPE-CLP method (dash-dotted line) for a speech segment.

Unfortunately, as the stability of the filter is not guaranteed, the filter estimated by the SPE-CLP method is occasionally unstable even though the estimated spectrum still models the speech harmonics better than the other methods. It was also observed that the chance of an instable filter estimation increases in high-pitched speakers even for the purely-voiced speech signals. Fortunately, the number of these instabilities reduced significantly with the multi-cycle variation of the SPE-CLP method. The LSF tracks obtained by the autocorrelation method and three-cycle M-SPE-CLP method illustrates this problem, and is shown in Figure 55. As the even and odd line spectrum pairs must be interleaved for a stable filter, the ninth and tenth LSFs violate this property between the samples 6800 and 6950, which is an indication of unstable prediction filter. The waveform and estimated

spectrum for a single cycle of this segment are also given in Figure 56.

In the experiment set, the percentage of unstable filters for ten speakers are given in the Table 4. The estimated unstable filters are always less in the low-pitched speakers than the high-pitched speakers. Furthermore, it was observed that the problem also occurs when the pitch period is much shorter than the average pitch period of the speaker.

Table 4: The percentage of unstable filters obtained by the variations of the SPE-CLP method for the ten speakers in the evaluation set.

Speaker	Average Pitch (Samples)	SPE-CLP	M-SPE-CLP (2 cycles)	M-SPE-CLP (3 cycles)
Male 1	84.86	0.4%	0.0%	0.0%
Male 2	73.61	1.0%	0.0%	0.0%
Male 3	68.98	0.2%	0.0%	0.0%
Male 4	58.61	1.0%	0.0%	0.0%
Male 5	47.46	0.2%	0.0%	0.0%
Female 1	63.24	0.9%	0.2%	0.0%
Female 2	38.38	1.4%	0.6%	0.2%
Female 3	39.43	3.7%	0.7%	0.3%
Female 4	51.65	0.5%	0.2%	0.2%
Female 5	38.99	4.0%	1.3%	0.7%
Male All	N/A	0.5%	0.0%	0.0%
Female All	N/A	2.4%	0.7%	0.3%

The synthesized speech signals using both the autocorrelation method and the variations of the CLP method sound very similar for both male and female speakers. Only in few sentences, careful listening with headphones reveals subtle difference at onsets, especially for female speakers. Similar to the CLP method, the synthesized speech using the prediction filter obtained by the variation of the SPE-CLP method also sound very close to that obtained by the autocorrelation method for male speakers. For female speakers, there is also a subtle difference that is audible throughout the entire sentences. In addition, in all cases, it is very hard to tell which method is the best. In addition to these results, some audible artifacts were observed in some of the sentences when these new methods were used. Two reasons were found for these artifacts. First, when there is a large energy change between two pitch cycles (e.g. a transition from a vowel to a nasal), the segmentation algorithm may divide the pitch-cycles such that half of the analysis cycle is from the high-energy pitch cycle and the other half is from the low-energy pitch cycle. In this case, the

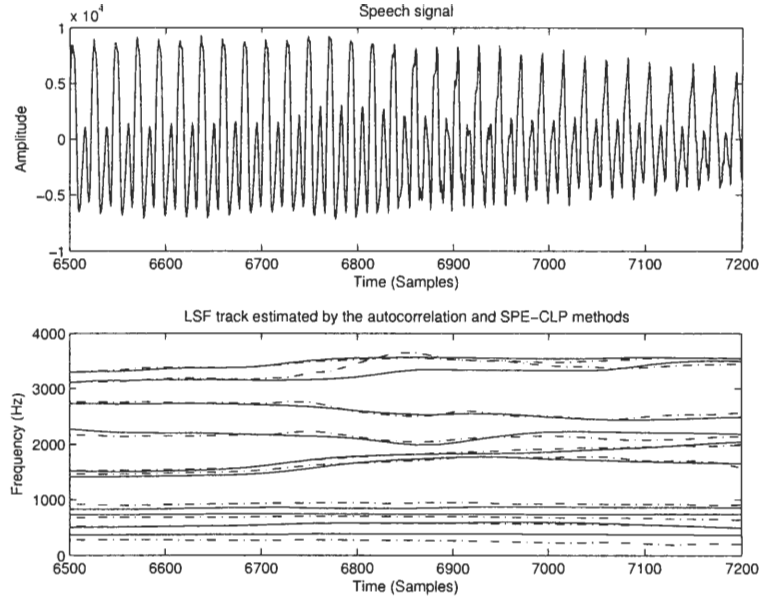


Figure 55: The LSF tracks obtained by the autocorrelation method (solid line) and the three-cycle M-SPE-CLP method (dash-dotted line) for a high-pitched speaker (average pitch-period is 22 samples.)

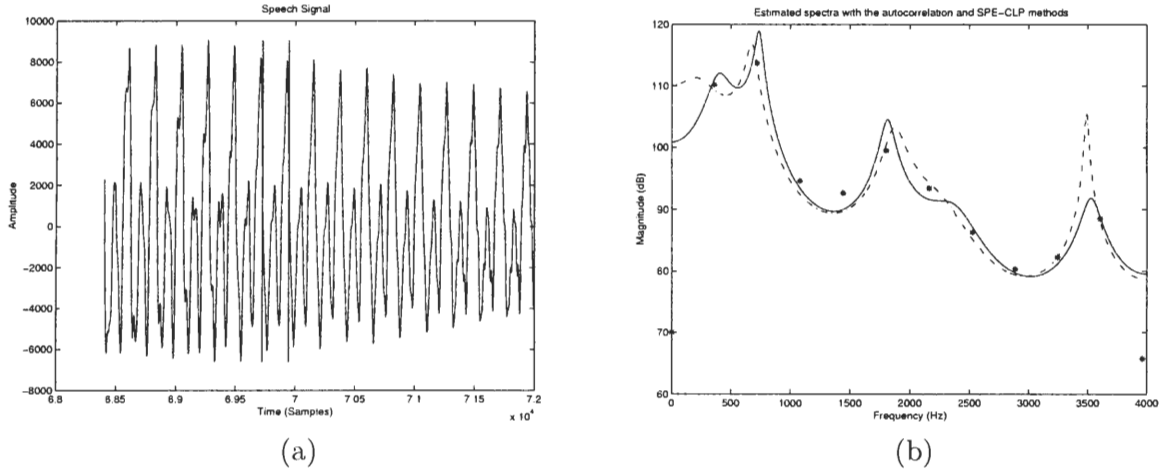


Figure 56: A stationary real-speech segment for a female speaker (a), and the spectra estimated by the autocorrelation method (solid line) and the three-cycle M-SPE-CLP method (dash-dotted line) (b). The length of the pitch cycle is 22.2 samples.

spectra estimated by the variations of the CLP method are not correct, and as a result, a distortion occurs at those segments in the synthesized speech. The second type of distortion originates from the estimation problem in partially-voiced speech, especially at the end of words and on voiced to unvoiced transition segments. However, the multicycle variations often eliminate the second distortion.

CHAPTER V

CONSTANT PITCH TRANSFORMATION

The circular residual signal obtained by filtering the speech pitch cycle circularly with the inverse of the prediction filter still has a fractional cycle length at the original sampling rate. Further processing of this signal at the upsampled rate is also redundant. Therefore, it is useful to find a unified representation for all circular residual signals regardless of the cycle length. The constant pitch transformation (CPT) is used to normalize the cycle length of circular residual signals to a constant integer length, τ_C , to generate (in effect) a monotone signal, $e_C[n]$. This constant length is selected as the largest possible pitch-cycle length at the original sampling rate, so that the original signal can always be recovered without any loss of bandwidth. This process not only removes the redundancy introduced by the upsampling process, but also generates a signal whose harmonics are at fixed locations, making further processing easier. Finally, this signal is also suitable for interpolation in the synthesis process.

The CPT was first introduced by Barnwell in the late '70s to improve the efficiency of the transform domain speech coders [5]. Despite the lack of high-quality pitch-period detectors at that time, the algorithm still improved the performance of such coders. However, it did not become popular because of its dependency on high-quality pitch-period detectors and its high computational complexity. In the early '90s, Kleijn introduced prototype waveform interpolation (PWI), another form of the CPT, to normalize the pitch-cycle length of the residual signal to 2π to produce and encode a two-dimensional surface representing the evolution of the speech signal [37].

The basic block diagram of the CPT is shown in Figure 57. To normalize the length, the circular signal is first upsampled by τ_C and then filtered with a low-pass filter for interpolation. Finally, the filtered signal is decimated by the original cycle length, $\tau_o^N = N(\tau_o + f)$, to obtain the normalized length circular signal. The cut-off frequency of the low-pass filter

is selected as the minimum of π/τ_C and π/τ_o^N to avoid aliasing in the decimation process. The Figure 58 illustrates the result of the CPT. In this example, the length of the cycle between the dashed lines in Figure 58a is normalized to 160 samples by the CPT. The DFT of the cycle and the DFT of the CPT of the same cycle are shown in Figure 58c and Figure 58d, respectively. Note that the length of this cycle is 90 samples and the harmonics of the CP transformed signal is at fixed places.

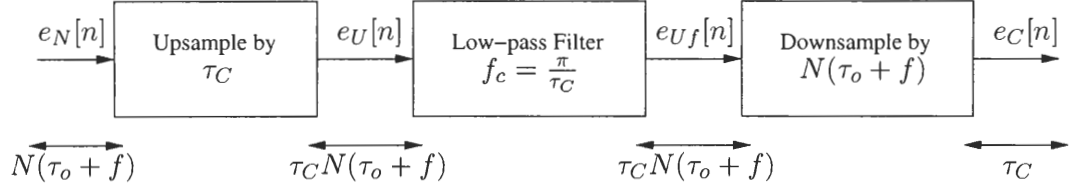


Figure 57: Block diagram of constant pitch transformation.

Direct implementation of this method has very high computational complexity because of the large upsampling factor. The complexity due to filtering in this case is $\tau_C \tau_o^N L$, where L is the length of the low-pass filter. Furthermore, the low-pass filter is usually very long because of the narrow pass-band requirement. Fortunately, because of the decimation, computation of only τ_C samples is required, and only τ_o^N samples of the input signal are nonzero. The computational-complexity requirement in this case decreases significantly to $\tau_C \frac{L}{\tau_C}$.

In CPT, since the input signal is assumed circular, the FIR filter is also applied to the input signal circularly. Therefore, for this special case, it is possible to generate a zero-phase filter with an ideal frequency response. The length of the filter is equal to the length of the upsampled original signal, $\tau_C \tau_o^N$, and all samples of the original circular signal are used in filtering. The CPT using this multi-rate filtering method can be implemented as follows:

- The input signal, $e_N[n]$, is upsampled by τ_C to generate $e_U[n]$:

$$e_U[n] = \begin{cases} e_N \left[\frac{n}{\tau_C} \right] & n = 0, \tau_C, 2\tau_C, \dots, (\tau_o^N - 1)\tau_C \\ 0 & \text{elsewhere} \end{cases} \quad (99)$$

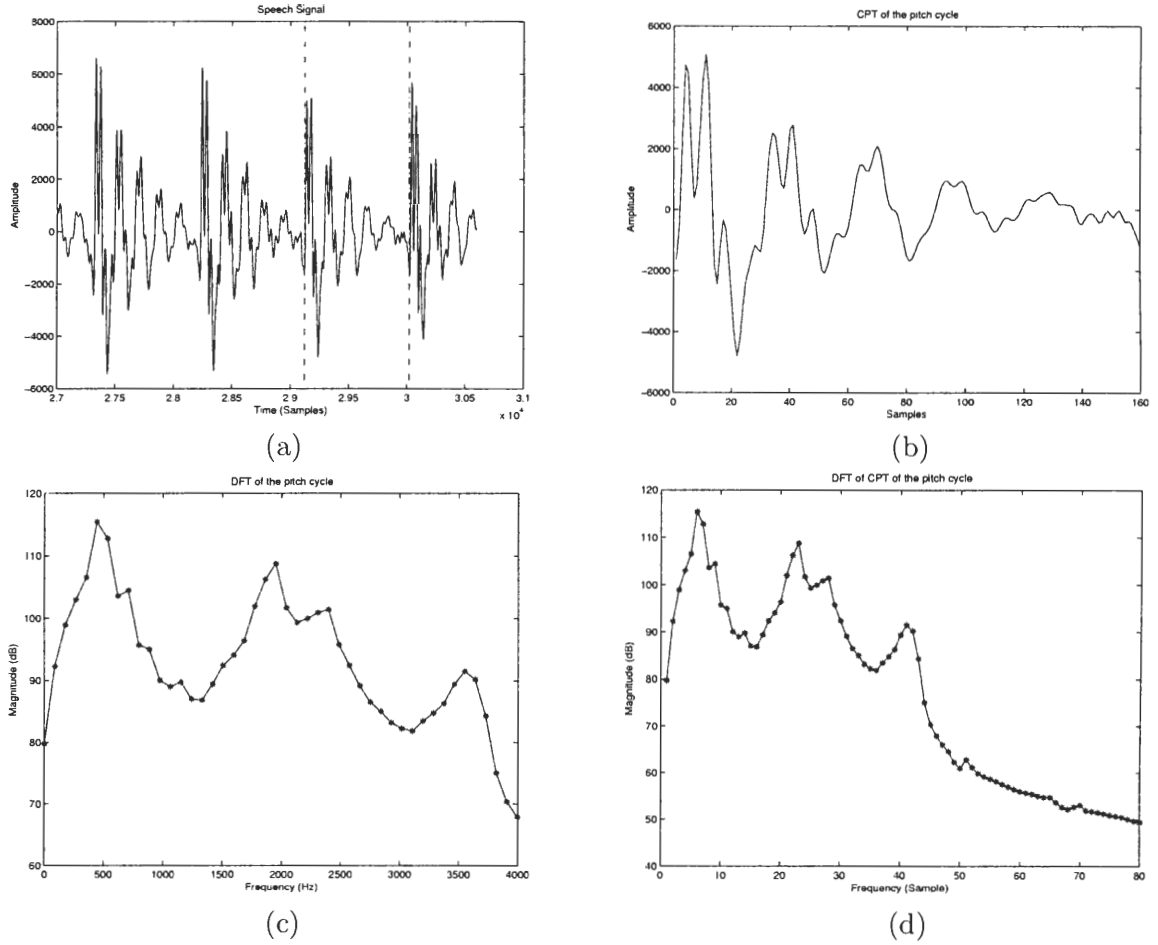


Figure 58: (a) A real speech-signal segment, (b) the CPT of the single cycle between the dashed lines in (a), (c) the DFT of the single cycle between the dashed lines in (a), and (d) the DFT of the CPT of the single cycle between the dashed lines in (a). The length of the original cycle is 90 samples and the length of the CP transformed cycle is 160 samples.

- The $e_U[n]$ is filtered with the ideal filter $h[n]$ to generate $e_{Uf}[n]$.

$$e_{Uf}[n] = \sum_{k=\frac{-\tau_o^N}{2}}^{\frac{\tau_o^N \tau_C - 1}{2}} h[k] e_U[(n+k)_{\tau_o^N \tau_C}]. \quad (100)$$

$e_U[(n+k)_{\tau_o^N \tau_C}]$ is nonzero only when $n+k$ is equal to an integer multiple of τ_C .

This is possible only when

$$k = \left\lfloor \frac{n + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor \tau_C - n + \varsigma \tau_C \quad (101)$$

is satisfied, where ς is an integer number between $-\frac{\tau_o^N - 1}{2}$ and $\frac{\tau_o^N - 1}{2}$ when τ_o^N is odd, and between $-\frac{\tau_o^N}{2}$ and $\frac{\tau_o^N}{2} - 1$ when τ_o^N is even, and ϱ is $\frac{\tau_C \tau_o^N}{2}$. The operation $\lfloor x \rfloor$ is rounding x to the nearest integer toward $-\infty$ or flooring x . When (100) and (101) are combined, the following equation is obtained:

$$e_{Uf}[n] = \sum_{\substack{\varsigma = \frac{-\tau_o^N}{2}, \dots, \frac{\tau_o^N - 1}{2} \\ \tau_o^N = \text{even, odd}}}^{\frac{\tau_o^N}{2} - 1, \frac{\tau_o^N - 1}{2}} h \left[\left\lfloor \frac{n + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor \tau_C - n + \varsigma \tau_C \right] e_U \left[\left(\left(\left\lfloor \frac{n + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor \tau_C + \varsigma \tau_C \right) \right)_{\tau_C \tau_o^N} \right] \quad (102)$$

or

$$e_{Uf}[n] = \sum_{\substack{\varsigma = \frac{-\tau_o^N}{2}, \dots, \frac{\tau_o^N - 1}{2} \\ \tau_o^N = \text{even, odd}}}^{\frac{\tau_o^N}{2} - 1, \frac{\tau_o^N - 1}{2}} h \left[\left\lfloor \frac{n + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor \tau_C - n + \varsigma \tau_C \right] e_N \left[\left(\left(\left\lfloor \frac{n + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor + \varsigma \right) \right)_{\tau_o^N} \right] \quad (103)$$

- The $e_C[n]$ is obtained by decimating $e_{Uf}[n]$ as

$$e_C[n] = e_{Uf}[n \tau_o^N]. \quad (104)$$

It is also possible to obtain $e_C[n]$ directly from $e_N[n]$ as

$$e_C[n] = \sum_{\substack{\varsigma = \frac{-\tau_o^N}{2}, \dots, \frac{\tau_o^N - 1}{2} \\ \tau_o^N = \text{even, odd}}}^{\frac{\tau_o^N}{2} - 1, \frac{\tau_o^N - 1}{2}} h \left[\left\lfloor \frac{n \tau_o^N + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor \tau_C - n \tau_o^N + \varsigma \tau_C \right] e_N \left[\left(\left(\left\lfloor \frac{n \tau_o^N + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right\rfloor + \varsigma \right) \right)_{\tau_o^N} \right] \quad (105)$$

Although this implementation is much more efficient than the direct implementation, the computational complexity still increases linearly by increasing τ_o^N . Furthermore, since $e_N[n]$ is bandlimited to π/N , it is possible to use a non-ideal filter with fewer samples without sacrificing performance. Since the ideal filter is a poly-phase filter, it is possible to construct another poly-phase filter by truncating the ideal filter and applying a window. Experimentally, a Hanning window was found sufficient for this purpose, and generating a Hanning window is computationally more efficient than generating a Kaiser window. In this case, (105) is modified as

$$e_C[n] = \sum_{\substack{\varsigma = \frac{L_s}{2} - 1, \frac{L_s}{2} \\ L_s = \text{even, odd}}}^{\frac{L_s}{2} - 1, \frac{L_s}{2} - 1} h \left[\left[\frac{n\tau_o^N + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right] \tau_C - n\tau_o^N + \varsigma \tau_C \right] e_N \left[\left(\left(\left[\frac{n\tau_o^N + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_C} \right\rfloor \tau_C}{\tau_C} \right] + \varsigma \right) \right)_{\tau_o^N} \right], \quad (106)$$

where L_s is the number of samples of $x_N[n]$ used in filtering, and ϱ is modified to $\frac{\tau_C L_s}{2}$. As discussed in Chapter 4, the reconstructed speech signal is always perceptually indistinguishable from the original signal when the SNR between the original signal and the error between the original and the reconstructed signals is larger than 72 dB. For this reason, the experiment setup described in Section 4.3.1, which is designed to evaluate the performance of the algorithms for different combinations of vocal-tract filter and for pitch-cycle lengths between 20 and 160 samples, is used to find a suitable L_s as a function of pitch-cycle length between 20 and 128 sample that results in an average reconstruction SNR of 72 dB. The value of L_s vs. pitch-cycle length that satisfies this constrain is shown in Figure 59a.

In the inverse CPT (ICPT), the upsampling and the decimation factors in the CPT are interchanged so that the signal with the original cycle length can be recovered, as show in Figure 60. The reconstructed signal, $\hat{e}_N[n]$, can be written in terms of $e_C[n]$ as

$$\hat{e}_N[n] = \sum_{\substack{\varsigma = \frac{K_s}{2} - 1, \frac{K_s}{2} \\ K_s = \text{even, odd}}}^{\frac{K_s}{2} - 1, \frac{K_s}{2} - 1} h \left[\left[\frac{n\tau_C + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_o^N} \right\rfloor \tau_o^N}{\tau_o^N} \right] \tau_o^N - n\tau_C + \varsigma \tau_o^N \right] e_C \left[\left(\left(\left[\frac{n\tau_C + (\varrho - 1) - \left\lfloor \frac{\varrho - 1}{\tau_o^N} \right\rfloor \tau_o^N}{\tau_o^N} \right] + \varsigma \right) \right)_{\tau_C} \right], \quad (107)$$

where K_s is the number of samples of $e_C[n]$ used in filtering and ϱ is modified to $\frac{\tau_o^N K_s}{2}$. K_s can be set to τ_C to generate an ideal filter. However, as in the case of the CPT, it is

possible to use fewer samples in filtering without sacrificing the performance. The K_s that results in an average reconstruction SNR of 72 dB for each pitch-cycle length ranging from 20 to 128 samples is shown in Figure 59b.

Using the results of the experiments shown in Figure 59, it is possible to make both L_s and K_s adaptive depending on the pitch-cycle length. However, to have implementation simplicity, it is beneficial to select fixed values for both L_s and K_s . Therefore, L_s and K_s are set to $13N$ and 20, respectively. These selections ensure the average reconstruction SNR to be around 72 dB for pitch cycles shorter than 100 samples. Furthermore, in informal listening tests, it was also verified that no audible distortions were observed in the reconstructed speech even when the speech signal is partitioned into arbitrary length segments shorter than the maximum allowed pitch-cycle length.

The required computational complexity for different CPT-ICPT implementations and a realization for the 2.4 kb/s U.S. military standard MELP coder ($\tau_C = 160$, $N = 10$) are summarized in Table 5.

Although the multi-rate filtering method with truncated filter has low computational complexity, the filter generation requires significant amount of computation, because of the requirement of the trigonometric function evaluation for each sample of the filter. Even for the truncated filter, the filter lengths are equal to $\tau_C L_s$ and $\tau_o^N K_s$ for CPT and ICPT, respectively. In practice, both filters may have tens of thousands of coefficients. Furthermore, since the cut-off frequency has to be different for each possible cycle length, it is not practical to compute all possible filters and store them in a look-up table. One solution to this problem is to generate a look-up table for the trigonometric functions and then evaluate the first few terms of the Taylor series expansion of the trigonometric function around the stored values in the look-up table. Another sub-optimal solution can be used when the signal is bandlimited to π/N . In this case, after the signal is upsampled by τ_C in the CPT, the signal spectrum does not have any information after $\pi/N\tau_o$ and before π/τ_C (assuming that $N\tau_o$ is larger than τ_C , which is almost always satisfied when N is large.) As a result, instead of generating a low-pass filter with a cut-off frequency at $\pi/N\tau_o$, which is different for each different cycle length, it is possible to generate a single filter with a

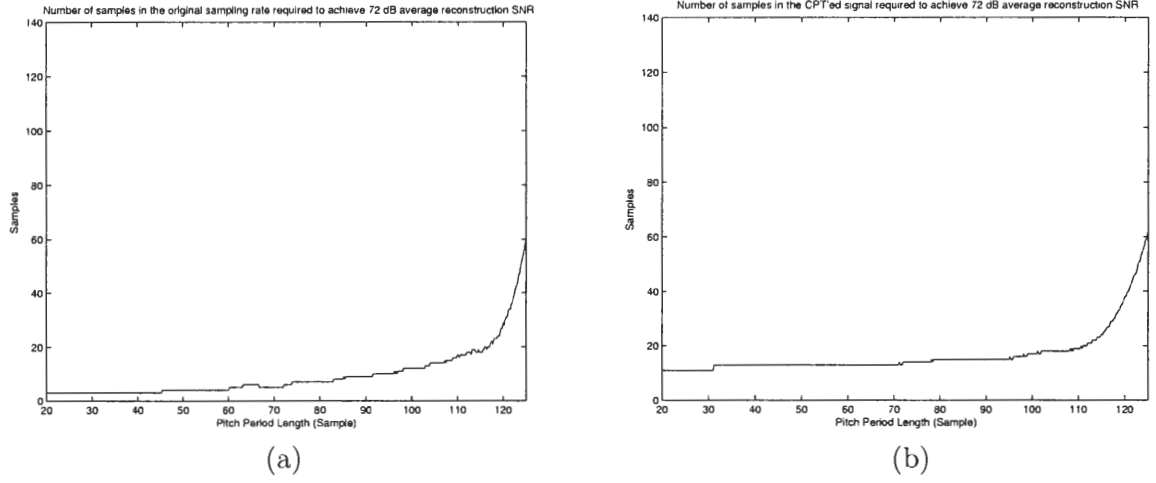


Figure 59: The L_s (a) and K_s (b) for pitch-cycle lengths between 20 and 128 required to achieve 72 dB average reconstruction SNR.

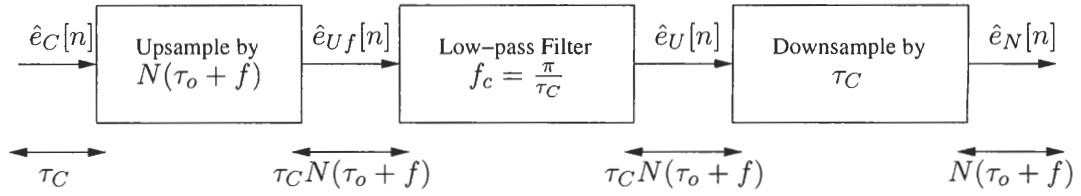


Figure 60: Block diagram of the inverse constant pitch transformation.

Table 5: The computational complexity of various CPT-ICPT implementations.

Implementation Method	Theoretical Computational Complexity	Actual Computational Complexity
Direct	$2(\tau_C \tau_o^N)^2$	6553600 WMOPS
Multi-rate	$2(\tau_C \tau_o^N)$	25.6 WMOPS
Multi-rate with truncated low-pass filter ($L_s = 13N$, $K_s = 20$)	$(\tau_C L_s) + (\tau_o^N K_s)$	9.36 WMOPS

cut-off frequency at π/τ_C . Therefore, the low-pass filter required in the CPT is computed only once in the initialization stage. Similarly, for the ICPT, since the length of the filter depends on the cycle length, a filter with a cut-off frequency at π/τ_C is computed for the maximum possible cycle length in the initialization stage, and at run-time, the filter is truncated and windowed with an appropriate length Hanning window. In this case, only the window coefficients have to be computed in real-time. This method decreases the overall computational complexity of the CPT and ICPT significantly.

There are various uses of the CPT. Since the harmonics of this signal are at fixed locations, the CP transformed pitch cycle is a natural candidate for quantization in both transform domain and parametric coders in terms of Fourier series magnitudes. Besides, this transformation also allows designing simple and effective algorithms which operate pitch-synchronously on speech signal such as filtering for zero-phase equalization. The various uses of the CPT in a speech coder are described in the next chapter.

CHAPTER VI

PITCH-SYNCHRONOUS METHODS FOR SPEECH CODING

A primary goal of this thesis is to develop methods that reduce the audible distortions in the synthesized speech associated with parameter estimation in segment transitions. The methods described in the previous chapters allow an LPC speech coder to capture both slowly and rapidly changing dynamics of the speech signal from individual pitch cycles. The CLP method and the CPT of the circular residual signal capture all of the perceptually important information in the speech signal from a single cycle. As a result, a perceptually indistinguishable replica can be synthesized from this set of parameters.

Figure 61 illustrates the basic speech analysis procedure using the proposed techniques. This procedure starts with an initial pitch-period estimation algorithm. This initial estimate is used in the pitch-cycle segmentation algorithm to partition the speech signal into pitch-cycle length segments so that the subsequent analysis generates the best representation that truly separates the vocal tract effects from the excitation. After the speech signal is segmented, CLP analysis is performed on the individual pitch cycles to obtain the linear-prediction coefficients, and then the pitch cycles are circularly filtered with the inverse of the associated all-pole filter to obtain the circular residual signal. Finally, the CPT is applied to generate a constant and integer length circular residual signal. In addition, for the purpose of speech coding, the final constant-length circular residual signals are aligned at the end of each frame. In early informal listening tests, the synthetic speech signal using the parameters extracted by this model was always perceptually indistinguishable from the input speech signal when the signal was segmented properly. Since these results were encouraging, the next step was to use these methods in speech coding applications.

The parameters extracted in this analysis procedure are exactly the same as those required in a fully-parametric speech coder based on the linear-prediction model. For this

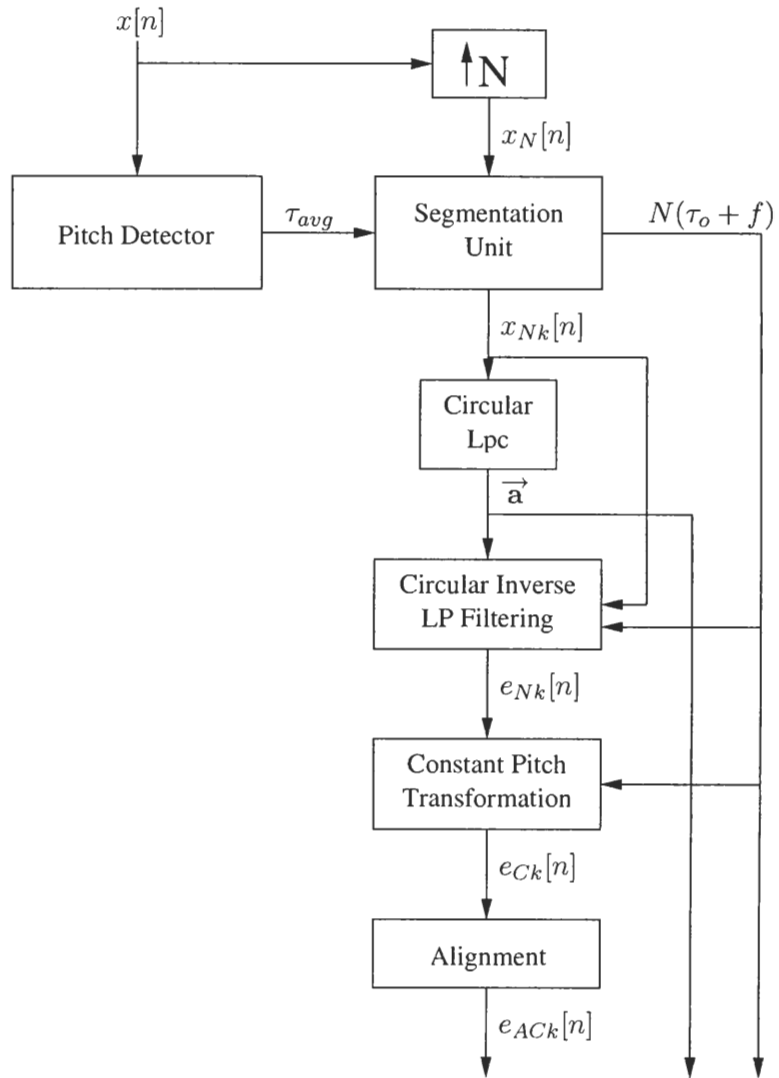


Figure 61: Block diagram of the analysis system.

reason, these steps can easily be incorporated into any speech coder based on this model. In this thesis, the 2.4 kb/s U.S. military standard MELP (MS-MELP) coder was selected as the baseline coder [79]. This coder was designed based on the Ph.D thesis of Alan McCree from the Georgia Institute of Technology [55] and developed as a collaboration between Atlanta Signal Processing Inc. and Texas Instruments. After extensive tests, this coder was found to be the best coder in terms of subjective quality and hardware requirements among the four competitors and selected as the new 2.4 kb/s U.S. military standard in 1996. A variation of this coder that uses a high-quality noise suppression algorithm as a front-end was also selected as the new 2.4 kb/s NATO standard [80]. The details of the MS-MELP coder are summarized in the next section.

In this chapter, a new 2.4 kb/s improved MELP (I-MELP) coder that incorporates the proposed techniques is described. In addition, a new technique that allows the use of a fully-parametric representation and waveform encoding of the excitation signal in the same coding algorithm is presented. To demonstrate the usefulness of this new technique, a speech coder that only encodes voiced frames with the new 2.4 kb/s I-MELP coder and transmits the waveform of the residual signal for unvoiced and transition frames is also introduced. This technique is referred as the parametric/hybrid coding of speech signal using the I-MELP/PCM coder. Furthermore, another experimental coder, referred as I-MELP/MP coder, that also encodes the unvoiced and transition frames with a variable rate multi-pulse coder is described. Finally, the results of the subjective evaluation tests are given at the end of this chapter.

6.1 The 2.4 kb/s U.S. Military Standard MELP Coder

The 2.4 kb/s MS-MELP coder [79] is based on the traditional LPC model, which is illustrated in Figure 1. This basic LPC model is enhanced using five additional features: mixed excitation, aperiodic pulses, an adaptive spectral enhancement filter, a pulse dispersion filter and Fourier series magnitudes. This new model is illustrated in Figure 62.

The most important feature of this model is the mixed excitation that replaces the single binary voicing decision in the traditional LPC coders. The mixed excitation is implemented

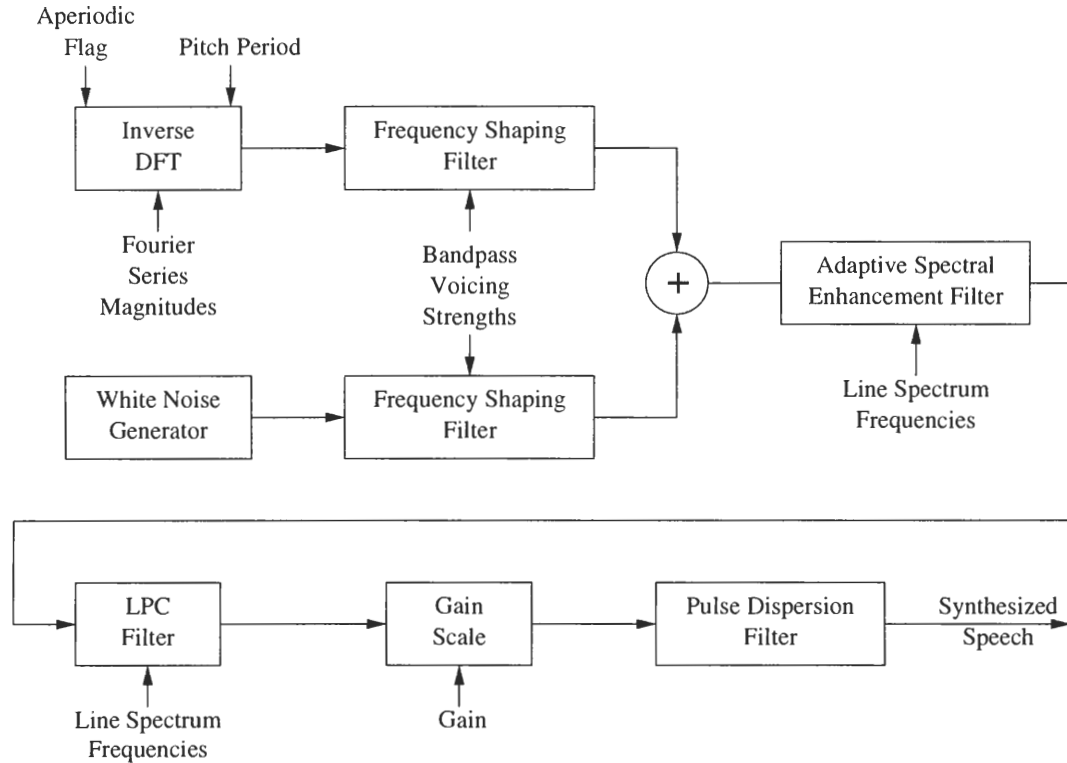


Figure 62: Block diagram of the MELP model.

as the summation of a bandpass filtered impulse train and noise sequence, where the bandpass filters used to filter these two sequences are magnitude complementary filters. This new excitation model reduces the buzzy quality of the LPC coders [56]. Several methods for generating the mixed excitation were reviewed in Section 2.1.2.

The second novelty in this model is the aperiodic pulses. These pulses are used to remove the isolated tones in the synthesized speech that occur in transition regions. This feature allows the model to produce erratic glottal pulses. This feature was also discussed in Section 2.1.2.

The adaptive spectral enhancement filter is an IIR filter whose roots and zeros are derived from the linear-prediction coefficients. This filter is used to match the waveform of the bandpass filtered synthetic speech to that of the input speech in formant regions. The typical formant resonances do not usually decay completely between pitch pulses in both natural and synthetic speech - however the synthetic speech waveform reaches a lower peak-to-valley ratio than the natural speech waveform. The adaptive spectral filter corrects

this behavior [56].

The pulse dispersion filter is used to reduce some of the harsh quality in the synthetic speech. McCree et al. [56] explained that the synthetic speech has large amplitudes for the first few samples in a pitch cycle but decreases rapidly afterwards. For these cases, the synthetic waveform does not match to the input speech especially for the frequency bands that do not have formant resonances. In this new model, the pulse dispersion filter is applied to the synthetic speech continuously to spread the energy of the pitch pulses in the excitation signal through the pitch cycles. The filter used for this purpose is a spectrally-flattened triangle pulse.

Finally, as the 10th order linear-prediction filter cannot capture all the fine details of the vocal-tract shaping filter, the LPC residual signal is not often spectrally flat. For this reason, the first 10 harmonics' magnitudes of the Fourier transform of the residual signal is encoded and transmitted to the decoder. This additional information reduces the spectral mismatch in the perceptually-important low-frequency region of the speech spectrum, which improves the overall quality of the synthetic speech.

The remaining of this section briefly summarizes the encoder and the decoder of this 2.4 kb/s MS-MELP coder. Detailed information about this coder can be found in [79].

6.1.1 The Encoder

The 2.4 kb/s MS-MELP encoder estimates the following parameters: pitch period, voicing state of five frequency bands, gain, linear-prediction coefficients in the form of LSFs, aperiodic flag and Fourier series magnitudes. These parameters are estimated on a frame-by-frame basis in every 22.5 ms (180 samples) from the speech signal sampled at 8000 Hz. The last sample in a frame is used as the reference point and all analysis windows used in the estimation of the parameters are centered on this sample. For this reason, this sample is also referred as the center of the analysis frame. Before the parameter estimation, the input speech is first filtered with a 4th order Chebychev Type-II high-pass filter with a cut-off frequency of 60 Hz. This filter attenuates the low-frequency noise below and around 60 Hz and makes the input signal zero mean. All parameter estimation algorithms uses this

filtered signal instead of the input speech signal in the encoder.

The encoding algorithm starts with an initial pitch-period estimation. This algorithm first calculates the normalized correlation function at the pitch lags between 40 and 160 samples from the low-pass filtered speech signal with a cut-off frequency of 1 kHz. The pitch lag that results in the largest correlation is selected as the initial pitch estimate. Then, the input speech is passed through a filter bank that partitions the speech signal into the following five bands: 0-0.5 kHz, 0.5-1 kHz, 1-2 kHz, 2-3 kHz and 3-4 kHz. The normalized correlation coefficients of the ten lags around the initial pitch estimate and those around the initial pitch estimate found in the previous frame are computed using the signal in the lowest band, and the pitch lag with the largest correlation is selected as the frame's pitch period. The associated normalized correlation coefficient is called as the voicing strength of the lowest band and the voicing strength of the frame. For the remaining bands, the envelopes of the bandpass filtered signals are first calculated, and the normalized correlation coefficient at the frame's pitch period is calculated twice using the bandpass filtered signal and its envelope (the normalized correlation coefficient found from the envelope of the signal is reduced by 0.1 to compensate the bias resulting from the smoothness of the signal [56].) The voicing strength of each band is selected as the larger of these two correlation coefficients. Furthermore, the fractional pitch-period calculation described in Section 3.1 is used to find the fractional pitch period and the associated normalized correlation coefficient of each band. In addition, when the voicing strength of the first band is less than 0.5, the aperiodic flag is set to one. Otherwise, it is set to zero.

The 10th order linear-prediction filter is obtained by the autocorrelation method using 200 samples. A hamming window is first multiplied with the signal prior to the analysis. In addition, the roots of the filter are multiplied by 0.994 for the purpose of bandwidth expansion. The LPC residual signal is obtained by filtering the input signal with the inverse of the linear-prediction filter. The peakiness of the frame is calculated as the ratio of the L2 norm to the L1 norm of the residual signal. When the peakiness exceeds 1.34, the voicing strength of the lowest band is forced to be 1.0. When the peakiness exceeds 1.6, the voicing strengths of the lowest three bands are forced to be 1.0. This residual signal is also

filtered with a low-pass filter with a cut-off frequency of 1 kHz and the resulting filtered signal is used in the last stage of the pitch estimation algorithm. The normalized correlation coefficients of the ten lags around the frame's pitch period are calculated using this residual signal and the lag with the largest correlation is selected as the final pitch period. When the normalized correlation of the final pitch period exceeds 0.6, pitch-doubling elimination logic is used to reduce pitch-doubling errors and to extend the pitch period range down to 20 samples. Otherwise, a fractional pitch refinement is performed around the frame's pitch period using the input speech signal. When the normalized correlation of this refined pitch period is less than 0.55, the final pitch period is assigned as the average pitch period. Otherwise, the pitch-doubling elimination logic is used to calculate the final pitch period using the input speech signal.

The gain of the signal is calculated as the RMS value of the input signal. When the voicing strength of the first band exceeds 0.6, the window length is set to twice the frame's pitch period. Otherwise, the window length is set to 120 samples. The gain term is calculated twice for each frame: once at the end of the frame and once in the middle of the frame. As a result, the gain of the signal is accurately represented in this coder. In addition, the gain values are also converted a logarithmic scale before quantization. When the gain and normalized correlation coefficient at the final pitch period exceed 30 dB and 0.8, respectively, the final pitch is placed in a first-in/first-out buffer that holds the three pitch-period values, and the average pitch period is selected as the median of this buffer.

The linear-prediction coefficients are converted into LSFs, and then the resulting LSFs are forced to have at least a distance of 50 Hz between them. Then, these LSFs are quantized with a 25-bit multi-stage vector quantizer (MSVQ) that uses four stages of 128, 64, 64 and 64 levels respectively. The quantized vector is the sum of the selected vectors, one vector from each stage. The MSVQ search finds the codebook vectors that minimize the weighted Euclidean distance between the unquantized and quantized vectors. The weights are selected as the power spectrum of the linear-prediction filter evaluated at the LSFs. The final pitch period is quantized on a logarithmic scale using a 99 level quantizer. The quantized pitch value is mapped to a 7-bit codeword using a lookup table. The all-zero

codeword represents an unvoiced state occurred when the voicing strength of the first band is less than 0.6. The gain value at the end of the frame (*second gain value*) is quantized with a 5-bit uniform quantizer ranging from 10 to 77 dB. The gain value in the middle of the frame (*first gain value*) is quantized with 3 bits using an adaptive algorithm; when the two gain values and the second gain of the previous frame are sufficiently close to one another, an all-zero code is transmitted. Otherwise, the first gain value is transmitted with a 7-level uniform quantizer ranging from 6 dB below the minimum of the second gain values in the current and previous frames to 6 dB above the maximum of these two gain values. The voicing strength of the first band is inferred from the quantized pitch-period value. The voicing strength of each remaining band is transmitted with a single bit that is set to 1 when the voicing strength of the band exceeds 0.6. Otherwise, it is set to 0.

Finally, the quantized LSFs are transformed back to a direct-form linear-prediction filter and the inverse of this prediction filter is used to obtain quantized-LPC residual signal. A 200 sample hamming window is multiplied with this residual signal, and the resulting signal is zero-padded to increase its size to 512 samples so that the Fourier transform of the signal can be obtained by a 512-point fast Fourier transform (FFT). Finally, the magnitudes of the FFT samples are calculated and the first ten harmonics are found by a peak-picking algorithm. The magnitudes of the harmonics are normalized to have an RMS value of 1.0, and then quantized by a 8-bit vector quantizer. The codebook is searched using a perceptually-weighted Euclidean distance whose weights emphasize low frequencies over high frequencies.

All of the quantized parameters specified above are transmitted to the decoder when the frame is not declared as unvoiced. When the frame is declared as unvoiced, the voicing strength of the four bands, the aperiodic flag and the quantized Fourier series magnitudes are not transmitted, and instead, forward error correction codes are sent to the decoder to protect the first MSVQ index of the quantized LSFs and the second gain value. The bit-allocation table of this coder for the two operating modes is shown in Table 6.

All estimation algorithms used in the encoder use the samples around the center of the analysis frame. As the center location is the last sample of the current frame, half of

Table 6: Bit-allocation table for the 2.4 kb/s U.S. military standard MELP coder.

Parameters	Voiced	Unvoiced
LSFs	25	25
Fourier Magnitudes	8	—
Gain (2 per frame)	8	8
Pitch, Overall Voicing	7	7
Bandpass Voicing	4	—
Aperiodic Flag	1	—
Error Protection	—	13
Sync Bit	1	1
Total Bits / 22.5 ms frame	54	54

the analysis windows always contains samples from the next frame, which introduces an algorithmic delay. The longest window used in the MS-MELP encoder is the one required for the pitch-period estimation algorithm. As the maximum allowed integer pitch period is 160 samples (20 ms) and the algorithm requires at least two pitch cycles for a reliable estimation, the length of the window is 40 ms. As half of this window overlaps with the next frame, the algorithmic delay of the encoder is 20 ms.

6.1.2 The Decoder

The decoder uses the synthesis model illustrated in Figure 62. The procedure starts with decoding the quantized features and generating the bandpass filters to be used in pulse train and noise sequences filtering. In addition, when the frame is voiced and the aperiodic flag is set to one, the jitter is set to 25%. Otherwise, it is set to 0%. In unvoiced frames, the pitch-period is set to a default 50 samples, the jitter is set to 0% and all Fourier series magnitudes are set to 1.0. Before the decoding operation, a small amount of gain attenuation is also applied to the two gain parameters using a power subtraction rule described in [79].

This decoder interpolates the parameters pitch-synchronously for each synthesized pitch cycle. The following parameters are interpolated in the decoder: pitch, LSFs, gain, jitter, Fourier series magnitudes, coefficients of the filters used in pulse train and noise sequence filtering. All parameters are linearly interpolated between the past and current frame values based on the starting location of the new pitch cycle within the frame. In addition, the gain value is interpolated using the first gain and the second gain of the previous frame, when

the starting point of the new pitch cycle is before the middle of the frame. Otherwise, the first and second gain values are used in interpolation. There are two exceptions to these interpolation rules: When there is an onset with high pitch frequency, the new pitch is used immediately. In addition, when there is more than 6 dB difference between the second gain values of the previous and current frames, the LSFs and the pitch period are interpolated using the gain trajectories. These two exceptions improve the synthesis performance of the decoder at onsets and segment transitions.

The synthesis procedure begins by finding the starting location of each new pitch cycle in the frame. The starting location of the first cycle is always one sample after the ending location of the last cycle in the previous frame. The pitch-cycle length is computed as the interpolated pitch period plus the jitter factor times the interpolated pitch period. The jitter factor is calculated as the interpolated jitter times a random number uniformly distributed between -1 and 1. The pitch-cycle length is always rounded to nearest integer. Although the term *pitch cycle* is ambiguous for unvoiced frames, it is still used to specify when the parameters are interpolated within the frame.

After the pitch-cycle locations are determined, the first step in the synthesis procedure is to generate the voiced excitation of each new pitch cycle that is computed from the inverse discrete Fourier transform (DFT) of the interpolated Fourier series magnitudes. After the voiced excitation of the pitch cycle is obtained, the samples in the cycle are multiplied by the square root of the cycle length to obtain a unity RMS signal, and then multiplied by 1000 to obtain a signal with a nominal level. In addition, the samples in the pitch cycle are circularly rotated by ten samples so that the parameter interpolation does not take place at the same location as the pitch pulses with large amplitudes. The noise sequence is generated with a uniform random number generator with an RMS value of 1000. The pulse and noise signals are filtered with the interpolated bandpass filters, and then summed to form the mixed excitation.

The adaptive spectral enhancement filter is also generated pitch-synchronously from the interpolated LSFs. The interpolated LSFs are first converted back to the linear-prediction

filter, and the adaptive spectral enhancement filter is obtained as

$$H_{ASEF}(z) = \frac{A(\alpha z^{-1})}{A(\beta z^{-1})}(1 + \mu z^{-1}), \quad (108)$$

where $A(z)$ is the inverse of the linear-prediction filter, α and β are set to $0.5p$ and $0.8p$, respectively, where p is the signal probability factor computed from the interpolated gain and the background noise estimate and clamped between 0 and 1, and μ is calculated as $\max(0.5k_1, 0)$, where k_1 is the first reflection coefficient also obtained from the linear-prediction filter.

The direct form of the linear-prediction filter is applied to the output of the adaptive spectral enhancement filter, and the resulting signal is scaled with the interpolated gain. In addition, the amplitudes of the first ten samples of each pitch cycle is scaled using a scaling factor obtained by linearly interpolating the scaling factor of the previous and current pitch cycles. Finally, the pulse dispersion filter is applied to the signal continuously.

6.2 The New 2.4 kb/s Improved-MELP coder

The 2.4 kb/s MS-MELP coder usually has high quality. However, the formal subjective listening tests performed in the 2.4 kb/s new U.S. military standard coder competition selection phase revealed that there is usually a significant quality difference between male and female speakers [41]. Experiments on this coder also showed that the pitch-period estimation algorithm makes more pitch-estimation errors for female speakers than for male speakers, especially at onsets. Furthermore, as the decoder constrains the length of pitch cycles to be an integer, the evolution of the pitch-period is not natural in the synthetic speech signal. However, as the length of pitch cycles in male speech is usually longer than that in female speech on the average, this problem appears only for the female speakers. Finally, although the pulse-dispersion filter improves the quality by a slight margin for male speakers, it makes the synthesized speech distinctly noisy for female speakers.

The new 2.4 kb/s I-MELP coder addresses these problems. In addition to all of the features included in the baseline MS-MELP coder, this new coder includes the following properties: an improved pitch-period estimation algorithm, an adaptive selection of linear-prediction method depending on the input signal, a harmonic estimation algorithm using

the CPT of a single cycle and a synthesis method that allows the use of fractional pitch-cycle lengths.

6.2.1 The Encoder

The new encoder estimates the same parameters as the baseline MS-MELP encoder. The frame length is also the same as that of the MS-MELP encoder. As the pitch-cycle segmentation algorithm requires an additional frame-length signal and finding the residual accurately requires the computation of the linear-prediction filter in the next frame, the encoding delay is increased to 35 ms.

This new encoder starts with initial linear-prediction filter estimation with the autocorrelation method using hamming-windowed 200 samples. The LPC residual signal is obtained by filtering the input speech with the inverse of the prediction filter. This resulting signal and the input speech signal are filtered by a low-pass filter with a cut-off frequency of 1.5 kHz. These two signals are used in the pitch-estimation algorithm described in Section 3.1 to obtain the pitch period and the voicing strength of the frame. The voicing strength of the frame is also used as the voicing strength of the lowest band. The three-step pitch-period estimation algorithm used in the MS-MELP coder is replaced by this single-step pitch-period estimation algorithm. The aperiodic flag, the gain, the peakiness, the bandpass voicing strengths and the average pitch period are obtained the same way as they are obtained in the MS-MELP coder.

After obtaining the parameters, the input signal is segmented into non-overlapping pitch-cycle length segments with the pitch-cycle segmentation algorithm based on the normalized correlation function maximization described in Section 3.2.2. The input speech, the residual signal and their low-pass filtered counterparts are all upsampled by a factor of ten to be used in this segmentation algorithm. The boundaries of each pitch cycle are also modified using the cycle boundary modification algorithm described in Section 4.6 so that the pitch cycles with the new boundaries are suitable for the CLP modeling. Furthermore, the correlation between the adjacent pitch cycles is computed in this algorithm as well.

The encoder selects the linear-prediction method depending on the signal within the

frame. In addition, when the M-CLP (or M-SPE-CLP) method is selected, the number of cycles used in analysis is selected adaptively. The pitch cycle that overlaps with the last sample of the frame is referred as the *main pitch cycle* and is used in both the CLP and the M-CLP analysis and the selection of the linear-prediction method. The autocorrelation method is always used when one of the following conditions occurs:

- The voicing strength of the frame is less than a voicing threshold, ρ_{vc} .
- The correlation between the main pitch cycle and the pitch cycle just after the main pitch cycle is less than a cycle-correlation threshold, ρ_{cc} .
- The correlation of only a single cycle in the frame exceeds ρ_{cc} and there is more than 6 dB difference between the energy of the first cycle in the frame and the energy of this single cycle.

The first two criteria avoid the use of the M-CLP (or M-SPE-CLP) method when the speech is either unvoiced or generated by erratic glottal pulses. In addition, as discussed in Section 4.6, when there is a large energy variation between two adjacent pitch cycles such as in a vowel-nasal transition, the cycle boundaries may not be properly selected. In this case, the prediction filter obtained by the CLP method is not reliable and may result in audible artifacts in the synthetic speech. The last criterion deals with this problem and avoids the use of the CLP method in such cases. When the autocorrelation method is not used, the number of cycles used in the M-CLP (or the M-SPE-CLP) method is selected according the correlation of the adjacent cycles. The following steps are used to determine which cycles in the current frame and the next frame will be used in the analysis:

1. The main pitch cycle is always used in the analysis given that the correlation between this cycle and the next one exceeds ρ_{cc} . Note that this cycle is the last one in the current frame and some part of this cycle overlaps with the next frame.
2. When the correlation between the last selected cycle in the *current* frame and the cycle just before this one exceeds ρ_{cc} , the cycle just before the last selected cycle in the current frame is also used in the analysis.

3. When the correlation between the cycle just after the last selected cycle in the *next* frame and the cycle just after this one exceeds ρ_{cc} , this cycle is also used in the analysis.
4. If the number of cycles in the analysis is still less than three and there are still remaining pitch cycles whose correlation with the next cycle have not been compared to ρ_{cc} , the algorithm continues from step 2. Otherwise, it terminates.

These steps ensure the use of multiple cycles when the speech signal is stationary and the use of single cycle when the cycle is on a transient segment (the correlation between adjacent cycles are low in this case.) In early informal listening tests, it was observed that the choice of the thresholds, ρ_{vc} and ρ_{cc} , is very important. Setting these thresholds too low results in audible artifacts in the synthesized speech. In these cases, the boundaries of the pitch cycles cannot be obtained reliably, and the prediction-filter coefficients obtained by the CLP method result in audible artifacts in the synthetic speech. On the other hand, when these thresholds are set too high, the transition regions are always analyzed with the autocorrelation method, and the estimated prediction filter is inferior to the one estimated by the CLP method using properly segmented pitch cycles. In these experiments, when both ρ_{vc} and ρ_{cc} are set to 0.75, no prediction-filter estimation related distortion was observed in the synthetic speech while the transition regions are still modeled with the CLP method.

Finally, the Fourier series magnitudes are also obtained by two different methods depending on the method used in the linear-prediction filter estimation. When the autocorrelation method is used to obtain the prediction filter, the Fourier series magnitudes are obtained by the peak-picking method used in the MS-MELP coder. When the M-CLP (or M-SPE-CLP) method is used, the main pitch cycle is circularly filtered with the inverse of the prediction filter, and then the CPT is applied to normalize its length. The Fourier series magnitudes are obtained from the magnitudes of the first ten frequency samples of the DFT of this constant-length circular residual signal. This method allows the encoder to capture the magnitudes of the harmonics reliably at onsets and transition regions from a single cycle.

The parameters of this coder are quantized with the same quantization methods used

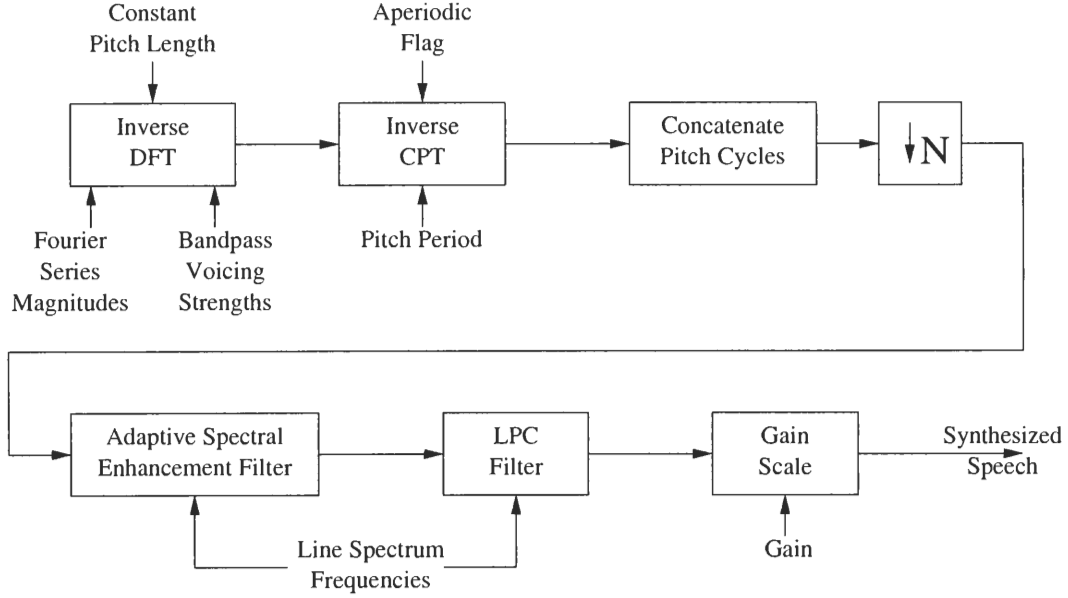


Figure 63: The decoder of the new 2.4 kb/s improved MELP coder.

in the MS-MELP coder. As a result, this encoder is compatible with the decoder of the 2.4 kb/s MS-MELP coder.

6.2.2 The Decoder

This decoder is also similar to the one used in the MS-MELP coder. However, it is modified so that pitch cycles with fractional cycle length in 0.1 sample steps can be synthesized. In addition, the mixed excitation signal is generated in the frequency domain so that filtering of the pulse train and noise sequences with the bandpass filters are eliminated. Finally, as the new pitch-period algorithm decreases the number of pitch-period estimation errors, the pulse-dispersion filter is removed from the decoder to improve the quality of synthesized speech for female speakers. This new decoder is illustrated in Figure 63.

Similar to the MS-MELP decoder, this decoder also finds the starting locations of the new pitch cycles and synthesizes the speech pitch-synchronously. For each new pitch cycle, the parameters are interpolated as described in Section 6.1. The only exception is that the interpolated pitch-cycle length is first multiplied by ten, rounded to the nearest integer, and then divided by ten. As a result, the length of the pitch cycles is always in 0.1 sample steps at 8000 Hz sampling rate in the new decoder.

The decoder first generates the pitch cycle with constant length in the frequency domain. The magnitudes of the frequency samples are set as follows:

$$|X_C[k]| = \begin{cases} 0 & k = 0 \\ |X_{int}[k]| & k = 1, \dots, 10 \\ 1 & k = 11, \dots, \lfloor \frac{\tau}{2} \rfloor \\ 0 & k = \lfloor \frac{\tau}{2} \rfloor + 1, \dots, \frac{\tau_C}{2}, \end{cases} \quad (109)$$

where $|X_{int}[k]|$'s are the interpolated Fourier series magnitudes, τ is the interpolated pitch-cycle length (but not rounded to the nearest integer) at the original sampling rate and τ_C is the length of the constant cycle length. The phase of the frequency samples are generated as follows:

$$\angle X[k] = (1 - \rho_{int})\tilde{\phi}, \quad (110)$$

where ρ_{int} is the interpolated voicing strength of the band that the frequency sample, k , is inside and $\tilde{\phi}$ is a random phase term uniformly distributed between $-\pi$ and π . When the voicing strength of a band is one, $\angle X[k]$ is always equal to zero as in the case for the voiced pitch-cycle generation in the MS-MELP coder. However, when the voicing strength of a band is zero, $\angle X[k]$ is completely random similar to the Fourier transform of an unvoiced speech signal. The frequency samples using this model are simply generated by

$$X[k] = |X[k]|e^{j\angle X[k]} \quad k = 0, \dots, \frac{\tau_C}{2}, \quad (111)$$

and the rest of the frequency samples between $\frac{\tau_C}{2} + 1$ and $\tau_C - 1$ are adjusted so that the resulting function is even. This method allows the decoder to generate a mixed excitation signal in a simple way. The constant-length pitch cycle is obtained by computing the inverse DFT of $X_C[k]$. Then, the inverse CPT is used to obtain the desired-length pitch cycle (interpolated pitch cycle length) at ten times the original sampling rate. After all pitch cycles in the frame are generated in this way, they are concatenated with each other and downsampled by a factor of ten to generate the excitation signal at the original sampling rate.

The rest of the steps in the decoder are exactly the same as the ones in the MS-MELP decoder except that the pulse dispersion filter is not applied to the final synthetic speech.

In the informal subjective listening tests, it was observed that the proposed 2.4 kb/s I-MELP coder's speech quality is similar to that of the MS-MELP coder for male speakers. However, the quality of the new coder for female speakers is distinctly better than that of the MS-MELP coder. The results of the formal listening test comparing the 2.4 kb/s MS-MELP coder and various implementations of the new 2.4 kb/s I-MELP coder are presented in Section 6.4.

6.3 Parametric/Hybrid Coding of Speech Signal Using the new 2.4 kb/s Improved-MELP coder

As discussed in Section 2.1.4, after a certain bit rate, the quality of fully-parametric linear-prediction coders does not improve with the increasing bit rate because of the modeling deficiencies. Although more accurate encoding of Fourier series magnitudes in the entire spectrum with large number of bits improves the quality, it eventually saturates around 4 kb/s. To improve the quality further, it is necessary to encode the transition segments such as onsets and the short events like erratic glottal pulses with waveform-encoding techniques. However, as fully-parametric coders do not preserve the phase information, the synthesized speech signal is not time synchronous with the input speech signal and the waveform shapes of the synthesized and input speech signals are quite different. For these reason, switching between a fully-parametric representation and waveform encoding of the excitation signal may result in audible artifacts in the synthesized speech. Various techniques for eliminating (or reducing) these artifacts were reviewed in Section 2.1.4.

In this section, a new method that allows the use of the I-MELP coder and any waveform-encoding technique within the same coder to encode different parts of speech signal is presented. This novel method relies on the modification of the input speech and no additional parameters are transmitted to the decoder. For this reason, this method is more suitable for parametric/hybrid encoding of the speech signal for variable-rate speech coders that operates down to very low bit rates. In addition to this new algorithm, an experimental I-MELP/PCM coder that encodes the voiced speech segments by the I-MELP coder and

transmits the waveform of the residual signal of the unvoiced/transitional segments is described. Finally, another experimental I-MELP/MP coder that encodes the voiced speech segments by the I-MELP coder and the unvoiced/transitional segments by a MP-LP coder is presented.

6.3.1 Pitch-Cycle Modification of Speech Signal for Designing a Parametric/Hybrid MELP Coder

The basic idea of this pitch-cycle modification algorithm is to modify the input speech signal so that it becomes time synchronous with the synthesized speech signal and waveform shapes of the signals become similar. For this purpose, this algorithm basically changes the location and the length of the pitch cycles in the original signal. In addition, it also applies a zero-phase equalization filter to the pitch cycles so that the resulting signal waveform is similar to the synthetic one. This algorithm is specifically designed to be used in conjunction with the synthesis method in the MELP model. However, it can also be used with any other fully-parametric coder whose decoder synthesizes the speech signal pitch-synchronously. All the changes are performed on the residual signal so as not to modify the spectrum shaping effects of the true vocal-tract filter.

The algorithm starts with finding the starting location of the pitch cycles that are going to be synthesized in the decoder. Since the segments of the speech signal that include erratic glottal pulses will be encoded with waveform-encoding techniques, the aperiodic flag, which is the only feature that randomizes the length of the pitch cycles in the decoder, becomes unnecessary, and is therefore eliminated from the model. As the quantized pitch period and gain values used in the decoder are estimated in the encoder, the same starting locations that are going to be calculated in the decoder can also be found in the encoder. As the new parametric/hybrid coder uses the 2.4 kb/s I-MELP coder for the voiced segments, the pitch-cycle lengths of the synthesized speech signal are always calculated in 0.1 sample steps. As this pitch-cycle modification algorithm synthesizes a new residual signal using the location of these segments, the term *synthesis pitch cycles* are used to denote the segment locations of these pitch cycles.

The second step in this algorithm is to segment the original signal using the pitch-cycle segmentation algorithm based on normalized correlation maximization described in Section 3.2.2. Since this algorithm does not segment the speech signal in unvoiced frames, the input signal is segmented using the same segment boundaries of the synthesis pitch cycles. When the segmentation algorithm encounters an onset, the segment boundaries of the input signal are no longer synchronized to that of the synthesis pitch cycles and the new segment boundaries are obtained according to the location of the pitch pulses in the residual signal. At the end of the words or at voiced-to-unvoiced segment transitions, the segment boundaries of both original pitch cycles and synthesis pitch cycles are re-synchronized. However, special care must be taken in both establishing and destroying the synchronization; when this operation is not performed on low-energy regions, audible artifacts may be introduced in the modified speech. The pitch cycles obtained by the segmentation algorithm are referred as *original pitch cycles* in this section.

After the pitch-cycle boundaries in the synthesized speech and in the input speech are obtained, the next step is to map the original pitch cycles to the synthesis ones. Ideally, the easiest method is mapping the original pitch cycles to the synthesis pitch cycles whose segment boundaries are close to each other within the frame. However, a significant cycle length difference between the original pitch cycle and the mapped synthesis pitch cycle may result in audible artifacts after the modifications described later in this section. Therefore, special care must be taken to avoid such mappings. The mapping of the original pitch cycles to the synthesis pitch cycles is performed as follows. For each synthesis pitch cycle, the goal is to find an original pitch cycle. First, the pitch cycle in the original signal that includes the starting location of the synthesis pitch cycle is found. If this original pitch cycle overlaps completely with the synthesis pitch cycle in time as shown in Figure 64a or if significant portions of both this original pitch cycle and the synthesis pitch cycle overlaps in time as shown in Figure 64b, this original pitch cycle is mapped to the synthesis one. Otherwise, the cycle just after this original pitch cycle is mapped to the synthesis one as shown in Figure 64c. If the length of the original pitch cycle is similar to the length of the mapped synthesis pitch cycle, this initial mapping is preserved. Otherwise, the following

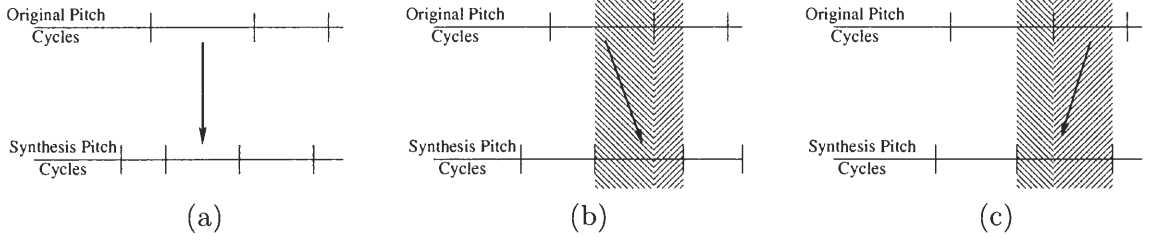


Figure 64: Mapping of the original pitch cycle to the synthesis pitch cycle when the synthesis pitch cycle overlaps with the original pitch cycle completely (a), when the significant part of the synthesis pitch cycle overlaps with the current pitch cycle (b), when the significant part of the synthesis pitch cycle overlaps with the next original pitch cycle (c). The left-to-right diagonal lines in (b) and (c) illustrate the overlapping segments of the selected original pitch cycle and the synthesis pitch cycle, and the right-to-left diagonal lines in (b) and (c) illustrate the overlapping segments of the the next original cycle and the synthesis cycle.

procedure is used to find a suitable original pitch cycle for mapping to the synthesis pitch cycle: Assume that the L^{th} original pitch cycle with a length of τ_L is initially mapped to the K^{th} synthesis pitch cycle with a length of τ_K , and following equation is not satisfied:

$$\left| \frac{\tau_K}{\tau_L} - 1 \right| < \frac{1}{3}. \quad (112)$$

In this case, if the τ_K is much shorter than τ_L (i.e. $\tau_K < \frac{2}{3}\tau_L$), the original pitch cycles in the neighborhood of the initially selected original pitch cycle are searched to find a suitable one whose length is sufficiently close to τ_K . If no such original pitch cycle can be found, the K^{th} synthesized pitch cycle is merged with the subsequent ones until the length of the combined cycles are sufficiently close to τ_L . On the other hand, if the τ_K is much longer than τ_L (i.e. $\tau_K > \frac{4}{3}\tau_L$), the combinations of the original pitch cycles that overlap with the K^{th} synthesized pitch cycle are searched so that the combined length of the selected combination is closest to τ_K . The flowchart of this mapping modification method is shown in Figure 65. This mapping algorithm is very useful when there is a rapid pitch-period change between two frames where the length of the original pitch cycles and synthesis pitch cycles are significantly different.

After mapping the original pitch cycles to the synthesis pitch cycles, the CPT of the original pitch cycles are calculated to normalize the length of each pitch cycle to the constant length, and the DFT of each constant-length pitch cycle is calculated. At this stage,

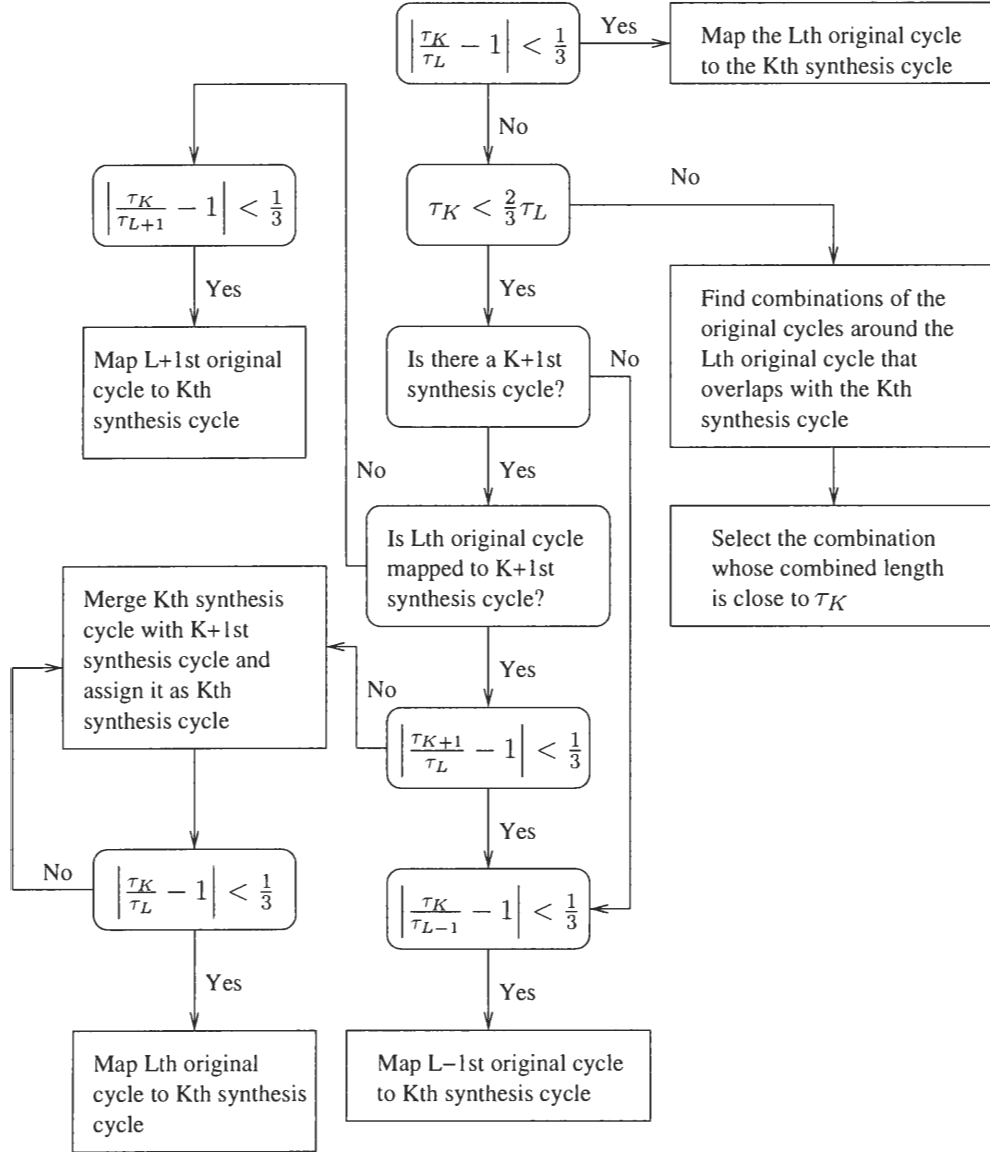


Figure 65: Flowchart of the mapping algorithm.

a zero-phase equalization is applied to the constant-length pitch cycles in the frequency domain when the original pitch cycles are detected as voiced. The purpose of the zero-phase equalization filter is to remove the phase component of the residual signal. However, it also has to be done carefully; simply setting the phase components of the frequency samples to zero introduces buzziness in the modified speech, as this procedure removes the noise in some of the bands in the residual signal. To preserve the correct amount of noise in the residual signal, the following method is used: The frequency samples in the bands with harmonic structure have constant phase in consecutive pitch cycles. On the other hand, the frequency samples in the bands with noisy structure have random phase. In speech signals with both kinds of excitations, both phase terms are present. Using this fact, the frequency domain representation of the K^{th} constant-length pitch cycle with a mixture of both types of excitation can be written as

$$X_K[k] = |X_K[k]|e^{j(\phi^c[k] + \tilde{\phi}_K^r[k])} \quad k = 0, \dots, \tau_C, \quad (113)$$

where $X_K[k]$ is the k^{th} frequency sample of the K^{th} pitch cycle, $\phi^c[k]$ is the phase term of the k^{th} frequency sample that is constant in consecutive pitch cycles, and $\tilde{\phi}_K^r[k]$ is the random phase term of the k^{th} frequency sample. When the frequency samples of M consecutive pitch cycles are normalized, and then averaged, the following equation is obtained:

$$\begin{aligned} Y[k] &= \sum_{m=1}^M e^{j\phi^c[k]} e^{j\tilde{\phi}_m^r[k]} \\ &= e^{j\phi^c[k]} \sum_{m=1}^M e^{j\tilde{\phi}_m^r[k]} \\ &= e^{j\phi^c[k]} |Y[k]| e^{j\tilde{\phi}_M^r[k]}, \\ &= |Y[k]| e^{j(\phi^c[k] + \tilde{\phi}_M^r[k])} \end{aligned} \quad (114)$$

where $|Y[k]|$ is the magnitude of the averaged frequency sample k , and $\tilde{\phi}_M^r[k]$ is another random phase term obtained after averaging. Note that the zero-phase equalization filter can easily be computed by normalizing $Y[k]$ and inverting the phase components. The resulting filter can be written as

$$H_{0\phi}[k] = e^{-j(\phi^c[k] + \tilde{\phi}_M^r[k])}. \quad (115)$$

The zero-phase equalized frequency samples can be obtained as

$$\begin{aligned}
\tilde{X}_K[k] &= X_K[k]H_{0\phi}[k] \\
&= |X_K[k]|e^{j(\phi^c[k]+\tilde{\phi}_K^r[k])}e^{-j(\phi^c[k]+\tilde{\phi}_M^r[k])} \\
&= |X_K[k]|e^{j(\tilde{\phi}_K^r[k]-\tilde{\phi}_M^r[k])} \\
&= |X_K[k]|e^{j\tilde{\phi}_K^r[k]}, \tag{116}
\end{aligned}$$

where $\tilde{\phi}_K^r[k]$ is also a random phase component. As a result, using this technique, it is possible to eliminate the constant phase term while replacing the original random phase with another one. In this pitch-cycle modification algorithm, the zero-phase equalization filter is computed for each original pitch cycle (voiced pitch cycle) from the frequency samples of the cycle itself and the two cycles adjacent to this cycle. After the application of the zero-phase equalization filter, the inverse DFT of all original pitch cycles is computed to obtain the constant-length zero-phase equalized pitch cycles. Since the zero-phase equalization filter is only applied to pitch cycles in voiced speech segments and voiced pitch cycles in transition segments, the unvoiced speech segments and the short events such as stop consonants are not modified. Unfortunately, when the pitch-cycle segmentation algorithm tags voiced pitch-cycles as unvoiced, these pitch cycles are not filtered with the zero-phase equalization filter and an audible artifact is occasionally introduced to the modified signal.

The inverse CPT is applied to the zero-phase equalized constant-length pitch cycles using the *length of the mapped synthesis pitch-cycle* and the resulting pitch cycle is copied to the segment location of the mapped synthesis pitch cycle in the modified residual buffer. Furthermore, each pitch cycle is also circularly rotated by ten samples similar to the decoder. After concatenating the pitch cycles at ten times the original sampling rate, the signal is decimated to obtain the modified residual signal at the original sampling rate. Finally, the linear-prediction filter is applied to the new residual signal, and the resulting signal is properly scaled to match its gain with the input speech signal. An input speech signal segment, the same segment processed with this pitch-cycle modification algorithm and the same segment encoded by the 2.4 kb/s I-MELP coder are shown in Figure 66. The slight difference in the waveforms of the segment processed by the pitch-cycle modification

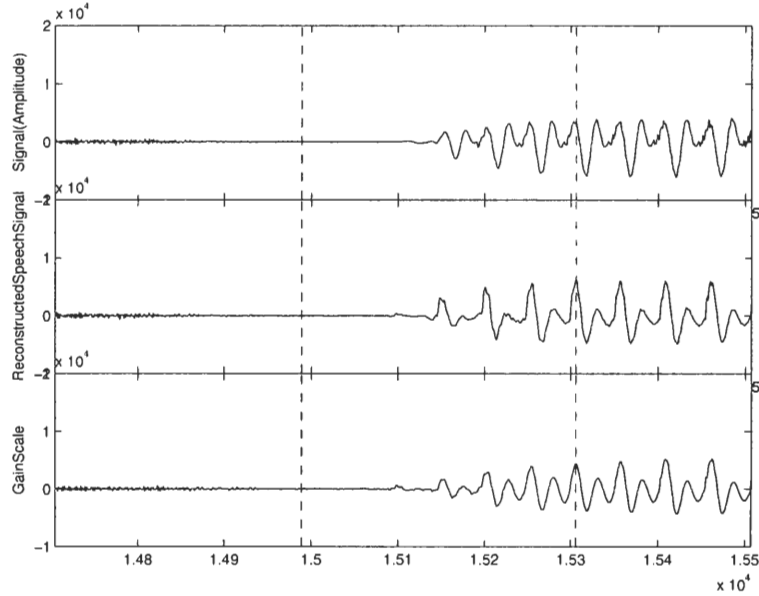


Figure 66: The input speech segment (top), the same segment processed by the pitch-cycle modification algorithm (middle) and the same segment encoded by the 2.4 kb/s improved MELP coder (bottom).

algorithm and the segment encoded by the 2.4 kb/s I-MELP coder results from the adaptive spectral enhancement filter used in the 2.4 kb/s I-MELP coder.

The flowchart of this pitch-cycle modification algorithm is shown in Figure 67. The early informal listening tests were performed on ten sentences from the TIMIT database as described in Section 4.6. In these tests, the modified input speech sounds very close to the input speech. However, it was also observed that any mistake in the pitch estimation and segmentation algorithms also introduces artifacts in the modified speech. The results of the formal listening tests are presented in the next section.

6.3.2 An Experimental Parametric/Hybrid I-MELP/PCM Coder

The pitch-cycle modification algorithm described above allows the use of the low bit rate I-MELP coder and any waveform-encoding technique co-exist in the same speech coder to encode different parts of the speech signal. Since the quality of the synthesized voiced speech obtained by the MELP coder is very good even at 2.4 kb/s, it is logical to use the MELP model for the voiced speech segments, and encode the waveform of the signal for unvoiced and transition segments.

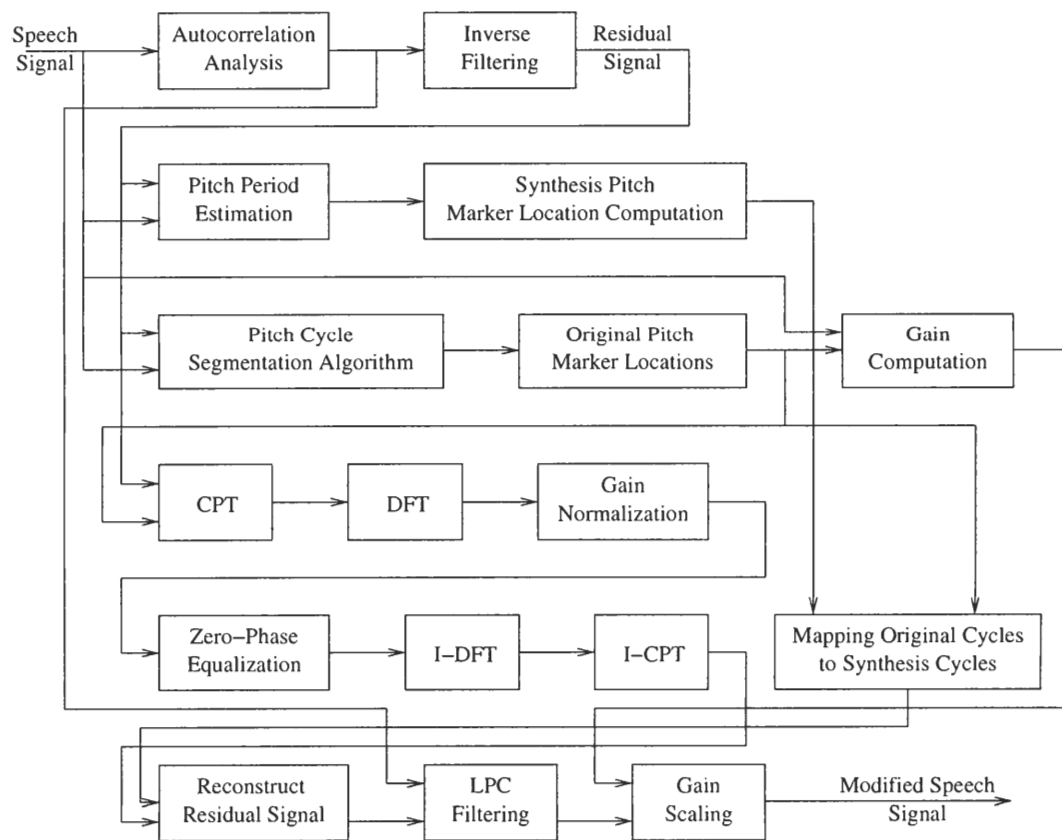


Figure 67: Flowchart of the pitch-cycle modification algorithm.

The easiest way to combine both techniques is to generate a frame of signal with the MELP coder when the voicing strength of the frame is high (i.e. > 0.8) and to use the speech signal generated with the pitch-cycle modification algorithm for the transition and unvoiced frames. Unfortunately, this approach results in audible artifacts in the synthesized speech because of the adaptive spectral enhancement filter. As illustrated in Figure 66, the waveform shapes of the input speech signal modified by the pitch-cycle modification algorithm may be different from the synthesized speech of the MELP coder because of this filter.

In this initial implementation, the 2.4 kb/s I-MELP coder is used to encode the voiced segments and the waveform of the residual signal in unvoiced and transition frames is transmitted to the decoder. When the voicing strength of the frame is less than 0.8, the frame is always declared as unvoiced. When an unvoiced frame is followed by a voiced frame, this frame is declared as transition frame and the new residual signal generated by the pitch-cycle modification algorithm is transmitted to the decoder in this case. For the voiced frames, the MELP model is used to synthesize the excitation signal.

Since the starting location of the synthesis pitch cycles in the decoder is also known in the encoder, the encoder only transmits the residual signal between the starting location of the first synthesis pitch cycle and the ending location of the last synthesis pitch cycle. In unvoiced frames, the original pitch cycle locations are synchronized to the synthesis pitch cycle locations as described above. In addition, these segments are not filtered with the zero-phase equalization filter. For this reason, these unmodified segments are transmitted to the decoder in the form of the CPT of the residual signal segments. For transition frames, the mapping stage in the pitch-cycle modification algorithm decides which part of the input residual signal is mapped to the synthesis pitch cycles and which original pitch cycles are filtered with the zero-phase equalization filter. In addition, the circular rotation of ten samples in the original voiced pitch cycles make the pulse positions at the same place as those in the MELP synthesizer. These processed constant-length pitch-cycles are transmitted to the decoder in transition frames. In unvoiced and transition frames, the inverse CPT is applied to the constant-length pitch cycles and the resulting pitch cycles

with a sampling rate ten times the original sampling rate are copied to the corresponding segment locations of the excitation signal buffer. In the voiced frames, the excitation signal is generated as in the 2.4 kb/s I-MELP coder. The rest of the MELP decoder is used to synthesize the speech signal.

In the informal listening test, the quality of the synthesized speech using this method is slightly better than that of the 2.4 kb/s I-MELP coder. The synthesized speech sounds clearer, especially at onsets and transitions. However, to get full advantage of this parametric/hybrid encoding method, the voiced segments must be quantized more accurately, because the overall quality of the speech coder is still determined by the I-MELP coder. Stachurski et al. [73] proved that it is possible to achieve near-toll quality when a variation the MELP model operating at 4 kb/s is used to encode voiced parts of the speech signal. For this reason, the quality of this initial implementation can be improved with improved quantizers operating at higher bit rates.

6.3.3 An Experimental Parametric/Hybrid I-MELP/MP Coder

This new experimental coder also uses the same techniques described in Section 6.3.2. In addition, instead of transmitting the waveform of the residual signal, the unvoiced and transition frames are also encoded with a variable rate MP-LP coder. The common techniques used in a MP-LP coders were discussed in Section 2.1.3.

As discussed above, the main problem of a MP-LP coder is the computational complexity that grows enormously with the increasing length of the frame and the number of pulses used in the encoding. For this reason, all MP-LP coders encode the speech signal in smaller subframes with a small number of bits. Unfortunately, this approach cannot be used in this coder directly; the effective length of each frame changes according to the starting location of the first pitch cycle and ending location of the last pitch cycle in the frame. For this reason, it is not easy to define the length of subframes suitable for the MP-LP coder.

As encoding the whole frame at once has very high computational complexity, the synthesis pitch cycles must be encoded separately. For this reason, the default pitch period in the unvoiced frames is changed to 45 samples (5.625 ms) so that a 180-sample unvoiced

frame is always partitioned into four synthesis pitch cycles (segments) that are also independent from the starting location of the first pitch cycle in the beginning of the frame. In transition frames, the length of the segments depends on the pitch-period length, and as a result, the MP-LP algorithm is used pitch-synchronously. In addition, as transmitting the pulse location in fractional sample steps requires a very high bit-rate, the beginning and ending location of the pitch cycles are rounded to the nearest integer at the original sampling rate and the locations of the pulses are also transmitted in one sample resolution. Since the number of segments and combined length of all segments are different in each transition frame, the number of required bits to encode this information is also different for each transition frame.

In early tests, it was observed that encoding a pulse in each five sample is not enough to make high-quality encoding of the unvoiced segments. The synthesized speech signal at silence and unvoiced segments always has minor but audible clicks. Although they were not major artifacts, the overall synthetic speech became unpleasant. In experiments, it was found that most of these artifacts can be eliminated when the excitation signal is generated as a white-noise sequence as in the case of the MELP model. However, short events like stop consonants would not be produced in the synthesized speech if all unvoiced segments are synthesized using this model. Unno et al. [78] proposed the detection of stop consonants using the peakiness measure described in Section 6.1. In this coder, the peakiness of each segment is computed, and a segment is declared as unvoiced when the peakiness value is less than 1.34, and the excitation signal for this signal is generated as a white-noise sequence in the decoder. Otherwise, the MP method is used to encode the excitation signal. Experimentally, it was observed that most of the stop consonants were encoded with the MP method, while the rest of the silence and unvoiced segments are generated in the decoder by a white-noise excitation that also eliminated the unpleasant clicks.

The segments that are not declared as unvoiced are encoded with the MP-LP algorithm described in [72] based on the amplitude re-optimization method. In this iterative method, a single pulse location is searched within the subframe (segment) while the pulses obtained in the previous iterations are kept. In the last iteration, the pulse amplitudes are re-optimized

by minimizing the sum of squared weighted error between the input signal and the synthetic signal with respect to amplitude of the pulses and by solving the resulting equations.

As Singhal et al. [72] recommended the use of 3-4 pulse in each 5 ms subframes (40 samples), a pulse is encoded for each nine samples in a segment in this implementation. Furthermore, the minimum and maximum number of pulses is constrained to be four and eight, respectively. However, when the weighted-SNR is still less than 10 dB after the location of eight pulses are computed in a transition frame, the maximum number of pulses is increased by two. These additional pulses are encoded and transmitted to the decoder only when the weighted-SNR exceeds 10 dB with these addition pulses. Otherwise, they are discarded. Furthermore, if the weighted-SNR exceeds 10 dB in any iteration, algorithm terminates and sends the pulses that have already been computed to the decoder. In the early informal listening tests, it was observed that sending more than eight pulses per pitch cycle (segment) did not improve the quality. It was also observed that the weighted-SNR of the segments in the transition frames usually exceeds 12 dB. However, it rarely exceeds 8 dB when the segment is in an unvoiced frame.

The pulse locations and amplitudes are encoded as follows: The total number of pulses in a segment is always between four and ten samples. For this reason, three bits are sufficient to encode the number of pulses in a pitch cycle. In addition, the all-zero code is used to indicate an unvoiced segment. As the Fourier series magnitudes and bandpass voicing strengths are not transmitted to the decoder in the 2.4 kb/s I-MELP coder, the 12 bits that are used to transmit this information can be utilized to transmit the “number of pulses per segment” information to the decoder. As a result, the speech signal can still be encoded at 54 bits/frame when all segments in the frame are declared as unvoiced. After transmitting the number of pulses in the segment, the location of the pulses are transmitted. Before encoding the pulse locations, they are sorted in ascending order. This would limit the possible location of all pulses obtained in a segment to $\frac{N!}{m!(N-m)!}$, where N is the length of the segment and m is the total number of pulses. In this coder, the number of bits required to send this information is first calculated, and the pulse location is coded in the calculated number of bits. As both N and m are also known in the decoder, the required number

of bits to encode the location of pulses can also be calculated in the decoder. To encode the amplitude of the pulses, the energy of the pulses are first normalized to a RMS value of 1. When the segment has a single pulse, the amplitude of that pulse can only be \sqrt{N} when the RMS of the segment is 1. As a result, the amplitude of a pulse in a segment can be at most \sqrt{N} . For this reason, all amplitude values are divided by this number, and the absolute of the resulting values are encoded with a 3 bit non-uniform Lloyd-Max scalar quantizer whose values are listed in Table 7. A single bit is also transmitted for the sign of the amplitude. The quantizer is trained using ten sentences from five male and five female speakers in the TIMIT database.

Table 7: The quantization levels of the normalized amplitude of the pulses

Quantization Level	Quantization Region	Quantized Value
0	0.0 - 0.19	0.14984777517564
1	0.18 - 0.28	0.23778523489933
2	0.28 - 0.37	0.32504533678756
3	0.37 - 0.47	0.41603576751118
4	0.47 - 0.58	0.52109411764706
5	0.58 - 0.72	0.64600907029478
6	0.72 - 0.84	0.77570652173913
7	0.84 - 1.00	0.89630630630631

The resulting coder is evaluated using another ten sentences from the TIMIT database. It was observed that the quality of the synthesized speech is very close to the parametric/hybrid I-MELP/PCM coder described in Section 6.3.2. However, minor artifacts were still audible especially in unvoiced frames. It was observed that the small number of pulses is not enough to encode the remaining unvoiced segments where the peakiness of the segment exceeds 1.34. However, the overall quality is still better than that of the 2.4 kb/s I-MELP coder. The results of the formal subjective listening tests are presented in the next section.

The average bit rate of the coded sentences in the evaluation set is between 3.5 and 4.5 kb/s. However, in transition frames, the bit rate sometimes exceeds 13 kb/s. Figure 68 illustrates the bit rate variation within a sentence in the evaluation set.

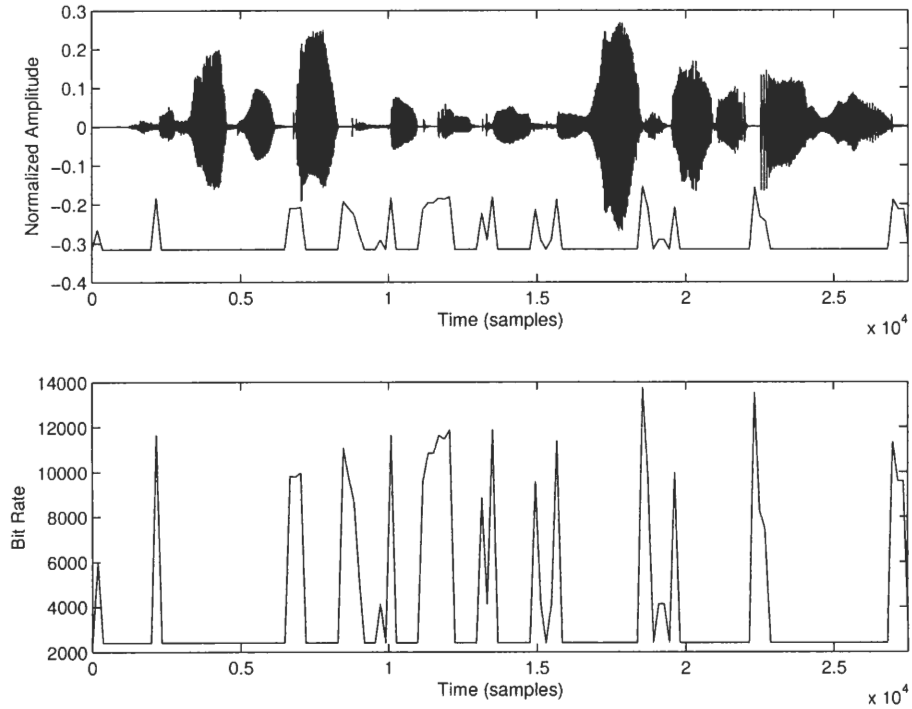


Figure 68: The bit rate variation (bottom) in one of the sentence in the evaluation set (top).

6.4 Subjective Evaluation of the Proposed Coders

The final stage in speech coding research is the quality evaluation of the coders. For this purpose, a series of subjective listening tests were conducted to measure the quality of the proposed coders. These tests included a degradation category rating (DCR) test and a comparison category rating (CCR) test. The output of the pitch-cycle modification algorithm is also compared to the unprocessed speech in these tests. Table 8 presents the conditions evaluated in these two tests.

6.4.1 Description of the Tests

The DCR test is used to determine the amount of degradation in a processed signal when compared to a reference signal [29]. In this test, the subjects are presented the reference signal first, and then the processed signal. Then, the subject judges the degradation in the processed signal by comparing it to the first one. The rating is performed on a 5-point scale shown in Table 9. The final score is known as the degradation mean opinion score (DMOS).

This type of test is sensitive to the degradations introduced by the speech coder and the modification algorithms.

The CCR test is a modified DCR test in which the subject compares the quality of the second sentence relative to the first one using a two-sided rating scale shown in Table 10. This approach eliminates the ordering restriction in the DCR test. The results produced by this test is known as the comparison mean opinion score (CMOS).

6.4.2 Test Setup

The test material includes 64 sentences spoken by four male and four female speakers obtained from the NTT database. The phrases were decimated to 8 kHz before processing with the coders and the algorithms listed in the Table 8. Each phrase was between four to eight seconds. The test material was recorded in quiet environment.

The tests were performed using the subjective listening test software written by Robert Morris. This software is capable of doing the ACR, the DCR and the CCR tests in a single session. In the DCR test, the software requires the subject to play the reference signal and the test signal and to rate the test signal compared to the reference signal using the rating system given in Table 9. The user can listen both signals as many times as they want before deciding the score. For each test case, the software selects the required number of phrases randomly from the 64 phrases and presents these samples to the subject in a random order. To identify the subjects with inconsistent results, a null-set is also introduced in the test session, which compares the reference signal to itself. Although the test setup is very flexible, it has two potential problems:

1. The same phrases may not be used in all test cases.
2. The number of phrases spoken by male and female speakers may not be equal in the test.

However, it is also possible to argue that if the same phrases are used in all test cases, the subjects may rate the phrases according to their quality relative to each other instead of rating them according to the degradation compared to the reference signal. For this reason,

Table 8: The test cases that are evaluated in the subjective listening tests

Test Case	Description
Input Signal	The unprocessed speech signal recorded in quiet environment
Modified Input Signal	The output of the pitch-cycle modification algorithm
I-MELP/PCM	The experimental I-MELP/PCM speech coder
I-MELP/MP	The experimental variable-rate I-MELP/MP speech coder
I-MELP[AC]	The 2.4 kb/s I-MELP coder using the autocorrelation method in all speech frames
I-MELP[CLP]	The 2.4 kb/s I-MELP coder using the adaptive linear-prediction selection method (CLP/M-CLP for voiced frames)
I-MELP[SPECLP]	The 2.4 kb/s I-MELP coder using the adaptive linear-prediction selection method (SPE-CLP/M-SPE-CLP for voiced frames)
MS-MELP	The 2.4 kb/s U.S. military standard MELP coder

Table 9: Rating scale for the degradation category rating (DCR) test.

Description	Rating
Degradation inaudible	5
Degradation perceived but not annoying	4
Degradation slightly annoying	3
Degradation annoying	2
Degradation very annoying	1

Table 10: Rating scale for the comparison category rating (CCR) test.

Description	Rating
Much Better	3
Better	2
Slightly Better	1
Similar	0
Slightly Worse	-1
Worse	-2
Much Worse	-3

the randomization of the phrases for each test case should eliminate this possibility. In addition, when the results obtained from a large number of subjects are averaged, the total number of phrases rated by the subjects may be close for the two genders. In the DCR test, it was observed that this was not the case and the difference between the male speech and female speech used in the same test case was as large as 20%. The CCR test was also performed using the same software. Similar to the DCR test, the software asks the user to listen to two sentences and to rate the second sentence compared to the first one using the rating system given in Table 10. In this test, not only the requested number of phrases for each test case was presented in random order, but also the order of the test and reference sentences were randomized. In addition, the software allows the number of phrases spoken by male and female speakers to be equal in the CCR test.

In both DCR and CCR tests, 12 phrases were used for each test case. The number of phrases spoken by male and female speakers could not be controlled in the DCR test. In the CCR test, 6 phrases were always from male speakers and 6 phrases were always from female speakers. As in the case for the DCR test, a null-set was also introduced in the CCR test as well to identify the users with unreliable test results. The tests were conducted on a Pentium-IV computer equipped with a SoundBlaster-Audigy sound card that has very good D/A converter characteristics. The tests were presented through an Aiwa HP-X222 headphone. All test cases tabulated in Table 8 were used in the DCR test. The unprocessed signal was used as the reference signal. In addition, the following test cases were compared in the CCR test:

- Input Signal vs. Modified Input Signal
- MS-MELP vs. I-MELP[CLP]
- I-MELP[CLP] vs. I-MELP/MP

6.4.3 Initial Test Results

The tests were initially taken by 28 subjects. Six of these subjects were later eliminated because of the unreliable results in the null-sets. The total number of phrases spoken by

male and female speakers are tabulated in Table 11. The averaged DMOS results obtained in the DCR test and the averaged CMOS scores obtained in the CCR test are tabulated in Table 12 and Table 13, respectively.

As these scores are estimates rather than exact results, a statistical analysis has to be performed before making any conclusions. For this reason, a two-sided t-test is used to do the statistical analysis of the results [68]. As two test cases in the DCR test were compared in this analysis each time, a two-sample t-test was used to determine whether the average score of both cases was the same or different at a pre-specified significance level. As the CCR test results in a single CMOS score for all test cases, a single-sample t-test is used to determine whether the final score of the test case has zero mean. In the statistical analysis of each test case, the mean, the standard deviation and 95% of the confidence interval are estimated from the data and t-test is used for the significance test.

6.4.4 Statistical Analysis Method for Evaluating the Test Results

The *null hypothesis* is the original statement. In the DCR test, the null hypothesis claims that the mean of the scores of the two test cases, c_1 and c_2 , are equal as in

$$H_0 : \mu_{c_1} = \mu_{c_2}, \quad (117)$$

where μ_{c_1} and μ_{c_2} are the mean of the scores of c_1 and c_2 , respectively. The *alternative hypothesis* claims that the mean of these two test cases are not equal:

$$H_1 : \mu_{c_1} \neq \mu_{c_2}. \quad (118)$$

For this reason, the test is a two-tailed test. As the test cases in the CCR test generate a single score, the *null hypothesis* is defined as

$$H_0 : \mu_c = 0, \quad (119)$$

where c is the test case. The *alternative hypothesis* for this test is defined as

$$H_1 : \mu_c \neq 0. \quad (120)$$

In the statistical tests, the significance level is related to the degree of certainty required to reject the null hypothesis in the favor of the alternative hypothesis. As the true mean

Table 11: The number of phrases spoken by male and female speakers used in the DCR test.

Condition	The number of phrases spoken by male speakers	The number of phrases spoken by female speakers	The number of phrases spoken by akk speakers
Input Signal	140 (53%)	124 (47%)	264 (100%)
Modified input Signal	135 (51%)	129 (49%)	264 (100%)
I-MELP/PCM	117 (44%)	147 (56%)	264 (100%)
I-MELP/MP	145 (55%)	119 (45%)	264 (100%)
I-MELP[AC]	138 (52%)	126 (48%)	264 (100%)
I-MELP[CLP]	148 (56%)	116 (44%)	264 (100%)
I-MELP[SPECLP]	133 (50%)	131 (50%)	264 (100%)
MS-MELP	130 (49%)	134 (51%)	264 (100%)

Table 12: The average DMOS scores for the DCR test.

Condition	Overall DMOS Score	Male DMOS Score	Female DMOS Score
Input Signal	4.96	4.96	4.96
Modified input Signal	4.00	3.81	4.20
I-MELP/PCM	3.75	3.73	3.77
I-MELP/MP	3.50	3.41	3.60
I-MELP[AC]	3.60	3.64	3.56
I-MELP[CLP]	3.53	3.47	3.59
I-MELP[SPECLP]	3.24	3.18	3.30
MS-MELP	3.34	3.63	3.05

Table 13: The average CMOS scores for the CCR test.

Condition	Overall CMOS Score	Male CMOS Score	Female CMOS Score
Input Signal vs. Modified Input Signal	-0.86	-1.02	-0.69
MS-MELP vs. I-MELP[CLP]	0.23	-0.08	0.55
I-MELP[CLP] vs. I-MELP/MP	0.07	0.05	0.09

of the scores cannot be estimated exactly from a finite number of data, the null hypothesis is rejected only if the probability of the results obtained from the sample data is less than a significance level, ν . In other words, the probability of incorrectly rejecting the null hypothesis when it is actually true is always less than the significance level. In addition, a confidence interval is a range of values in which the true hypothesized quantity is included with a probability of $(1 - \nu)$. The term $(1 - \nu)100\%$ is used to denote the confidence interval.

The sample mean of the test case, c , in both types of tests is found as follows:

$$\bar{X}_c = \frac{1}{\sum_{l=1}^L N_{c,l}} \sum_{l=1}^L \sum_{n=1}^{N_{c,l}} X_{c,l,n}, \quad (121)$$

where $N_{c,l}$ is the number of phrases that the l^{th} subject rates for the test case, c , and L is the number of subjects that takes the test. The sample standard deviation for the test case, c , is calculated as

$$\bar{s}_c = \sqrt{\frac{1}{(\sum_{l=1}^L N_{c,l}) - 1} \sum_{l=1}^L \sum_{n=1}^{N_{c,l}} (X_{c,l,n} - \bar{X}_c)^2}. \quad (122)$$

Before applying the t-test for the comparison of the two test cases, c_1 and c_2 , in the DCR test, the standard error of the difference between the estimated means of the test cases, $s_{\bar{X}_{diff}}$, has to be computed as

$$s_{\bar{X}_{diff}} = \frac{(N_{c_1} - 1)\bar{s}_{c_1}^2 + (N_{c_2} - 1)\bar{s}_{c_2}^2}{N_{c_1} + N_{c_2} - 2} \sqrt{\frac{1}{N_{c_1}} + \frac{1}{N_{c_2}}}, \quad (123)$$

where N_{c_1} and N_{c_2} are the total number of samples in the first and second test cases, respectively, and $\bar{s}_{c_1}^2$ and $\bar{s}_{c_2}^2$ are the variance of the first and second test cases, respectively.

The t-statistics is computed as

$$t_{stat} = \frac{(\bar{X}_{c_1} - \bar{X}_{c_2})}{s_{\bar{X}_{diff}}}. \quad (124)$$

The t-test rejects the null-hypothesis defined in (117) if and only if the following condition is met

$$2 \min(\tilde{t}_{(t_{stat}, N_{c_1} + N_{c_2} - 2)}, 1 - \tilde{t}_{(t_{stat}, N_{c_1} + N_{c_2} - 2)}) \leq \nu, \quad (125)$$

where $\tilde{t}_{(t, df)}$ is the value of the Student's T cumulative distribution function for the value of t and with df degrees of freedom. Note that ν is the significance level. The $(1 - \nu)100\%$

confidence interval is also computed as

$$C_{(1-\nu)100\%} = \bar{X}_{c_1} - \bar{X}_{c_2} \pm s_{\bar{X}_{diff}} \dot{t}_{(1-\frac{\nu}{2}, N_{c_1} + N_{c_2} - 2)}, \quad (126)$$

where $\dot{t}_{(t, df)}$ is the value of the inverse of the Student's T cumulative distribution function for the value of t and with df degrees of freedom. It is also true that if $C_{(1-\nu)100\%}$ includes 0, the null hypothesis can not be rejected. As a result, the estimated mean of the test cases, c_1 and c_2 , cannot be assumed to be different in the significance level of ν .

As the single-sample t-test is used for the test cases in CCR test, the t-test is slightly modified. The t-statistics are computed as

$$t_{stat} = (\bar{X}_c - \mu_c) \frac{\bar{s}_c}{\sqrt{N_c}}, \quad (127)$$

where N_c is the number of samples in the test case, and μ_c is the theoretical mean of the score of the test, c , defined in the null hypothesis. Similar to the (125), the null-hypothesis is rejected if and only if

$$2 \min(\tilde{t}_{(t_{stat}, N_c - 1)}, 1 - \tilde{t}_{(t_{stat}, N_c - 1)}) \leq \nu. \quad (128)$$

In addition, $(1 - \nu)100\%$ confidence interval for a test case in the CCR test is also computed as

$$C_{(1-\nu)100\%} = \bar{X}_c \pm \frac{\bar{s}_c}{\sqrt{N_c}} \dot{t}_{(1-\frac{\nu}{2}, N_c - 1)}. \quad (129)$$

In both of the conducted DCR and CCR test, N_l was always 12 for every subject. However, since the number of male and female test phrases may not be equal in the DCR test, it varied from one subject to another. N_l for male and female speech is tabulated in Table 11. N_l was always 6 for the CCR test when only male or only female speech was used in the tests. In the statistical analysis of the test results, 0.05 is used as the significance level (95% confidence interval) unless otherwise stated. μ_c is assumed to be zero in the statistical analysis of the test cases in the CCR test.

6.4.5 Interpretation of the Test Results

The scores for the test cases in the DCR test and the statistical comparison of various test cases are given between Table 14 and Table 22. Each table presents the mean DMOS score

of the reference and the test coders, the mean score difference between the reference and the test coders and the 95% confidence interval of the estimated mean score difference between the reference and the test coders. All of these quantities are estimated from the sampled data. In addition, a “result” column is also included in the tables that states whether the test coder is significantly better than or equal to or worse than the reference coder in 95% confidence level.

Table 14 presents the results of the comparison between the 2.4 kb/s speech coders including the MS-MELP coder, the I-MELP[AC] coder, the I-MELP[CLP] coder and I-MELP[SPECLP] coder. In the three of these comparisons, the MS-MELP coder is used as the reference coder and the new proposed coders are compared to the MS-MELP coder. Furthermore, the I-MELP[CLP] coder is compared to I-MELP[AC] coder, and the I-MELP[SPECLP] coder is compared to I-MELP[CLP] coder. The comparison results of the same test cases for only male speech and for only female speech are shown in Table 15 and Table 16.

As discussed before, the MS-MELP coder has better speech quality for male speech than for female speech on the average. The results tabulated in Table 15 and Table 16 also prove this fact: While the average DMOS score for the male speech is 3.53, this quantity decreases to 3.05 for the female speech. In the early informal listening tests, this kind of difference was not observed for the proposed 2.4 kb/s speech coders. The formal listening tests also prove that the speech quality difference between the genders are significantly less in the new proposed coders than in the MS-MELP coder. For this reason, the speech quality for female speech in the new 2.4 kb/s I-MELP coders are better than that in the MS-MELP coder. Even when the confidence level is increased to 99%, it can be said that both the I-MELP[AC] and the I-MELP[CLP] coders have still better speech quality than the MS-MELP coder for female speech. From these results, it can be argued that the proposed techniques improve the quality of the baseline MS-MELP coder. For the male speech, the synthetic speech qualities of the MS-MELP coder, the I-MELP[AC] coder and the I-MELP[CLP] coder are not statistically different. However, there is a 0.16 DMOS score difference between the MS-MELP coder and the I-MELP[CLP] coder. The main reason for

Table 14: Comparison of the quality of the 2.4 kb/s speech coders for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
FS-MELP	I-MELP[AC]	3.34	3.60	-0.26	[-0.42 -0.11]	Better
FS-MELP	I-MELP[CLP]	3.34	3.53	-0.19	[-0.34 -0.03]	Better
FS-MELP	I-MELP[SPECLP]	3.34	3.24	0.10	[-0.06 0.23]	Equal
I-MELP[AC]	I-MELP[CLP]	3.60	3.53	0.07	[-0.07 0.22]	Equal
I-MELP[CLP]	I-MELP[SPECLP]	3.53	3.24	0.29	[0.13 0.42]	Worse

Table 15: Comparison of the quality of the 2.4 kb/s speech coders for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
FS-MELP	I-MELP[AC]	3.63	3.64	-0.01	[-0.22 0.20]	Equal
FS-MELP	I-MELP[CLP]	3.63	3.47	0.16	[-0.05 0.37]	Equal
FS-MELP	I-MELP[SPECLP]	3.63	3.18	0.45	[0.23 0.67]	Worse
I-MELP[AC]	I-MELP[CLP]	3.64	3.47	0.17	[-0.03 0.36]	Equal
I-MELP[CLP]	I-MELP[SPECLP]	3.47	3.18	0.29	[0.09 0.46]	Worse

Table 16: Comparison of the quality of the 2.4 kb/s speech coders for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
FS-MELP	I-MELP[AC]	3.05	3.56	-0.51	[-0.73 -0.29]	Better
FS-MELP	I-MELP[CLP]	3.05	3.59	-0.54	[-0.76 -0.32]	Better
FS-MELP	I-MELP[SPECLP]	3.05	3.31	-0.26	[-0.46 -0.04]	Better
I-MELP[AC]	I-MELP[CLP]	3.56	3.59	-0.03	[-0.25 0.18]	Equal
I-MELP[CLP]	I-MELP[SPECLP]	3.59	3.31	0.28	[0.07 0.49]	Worse

this difference is the spectral estimation performance of the CLP method that depends on the accuracy of the pitch-period estimation. Wrong estimation of the pitch period also results in a linear-prediction filter estimate that is not a correct one. As this problem is not existent in the I-MELP[AC] coder, it has almost identical DMOS score with the MS-MELP coder. In addition, it should also be noted that the pulse dispersion filter used in the MS-MELP coder reduces the severity of the distortions in the synthesized speech resulting from the pitch and voicing estimation mistakes. Although this filter is not present in any of the new coders, it is still possible to obtain similar quality with the I-MELP[AC] coder that is also an indication of less pitch-estimation mistakes. When the DMOS scores of both genders are combined, the new I-MELP[AC] and I-MELP[CLP] coders are found *statistically better* than the state-of-the-art 2.4 kb/s MS-MELP coder. These results also prove that it is possible to estimate the linear-prediction filter from individual pitch cycles in the speech signal without introducing any artifacts. However, it is also evident that using the CLP method in a low bit rate speech coder does not have any obvious benefits over the autocorrelation method.

These tests also prove that there is a consistent speech quality difference in the order of 0.3 DMOS between the I-MELP[CLP] and I-MELP[SPECLP] coders. Unfortunately, not only do the pitch-estimation errors affect the performance of the I-MELP[SPECLP] coder, but also the partially-voiced speech has negative impact on this coder. In addition, as the spectral estimates obtained by the autocorrelation method and the SPE-CLP method may be significantly different, switching between these two methods occasionally introduces minor audible artifacts in the synthesized speech. As the spectral estimates obtained by the autocorrelation method and the CLP method are similar, this problem is not present in the I-MELP[CLP] coder. For these reasons, the I-MELP[SPECLP] method is consistently worse than the I-MELP[CLP] coder for both male and female speech and worse than the MS-MELP coder for the male speech in the DCR test.

Table 17 presents the results of the comparison between the 2.4 kb/s I-MELP[CLP] coder, the experimental I-MELP/PCM coder and the variable rate I-MELP/MP coder. In addition, the gender specific results are also tabulated in Table 18 and Table 19 for

male speech and female speech, respectively. As seen in Table 17, the quality of the I-MELP/PCM coder is statistically better than the 2.4 kb/s I-MELP[CLP] coder with an average 0.22 DMOS score difference. The transmission of the modified-residual signal in the transition segments and unvoiced frames reduces the degradation in the synthesized speech and improves the quality. In addition, the degradation resulting from the pitch-estimation mistakes are also reduced in the I-MELP/PCM coder. For this reason, the DMOS score difference is larger in the male speech than the female speech. However, even a 0.18 DMOS score difference is not enough to declare the I-MELP/PCM coder is statistically better than a I-MELP[CLP] coder for female speech. As the pitch-cycle modification algorithm also introduces audible artifacts in the modified speech signal, the quality difference is not that significant on the average, as expected. In addition, as the voiced frames are still encoded with a 2.4 kb/s speech coder, the quality of the overall synthesized speech is still mainly determined by the I-MELP coder.

The I-MELP/MP coder draws a different picture in this DCR test. Although its speech quality is similar to that of the I-MELP/PCM coder when used on the speech signals obtained from a different database, the speech quality is consistently judged worse than that of the I-MELP/PCM coder in this listening test. As discussed before, it was observed that the small number of pulses is not enough to encode the unvoiced segments that was observed more frequently in this test. In addition, the encoding of the amplitudes with a scalar quantizer using three bits occasionally introduces audible artifacts at the transition segments in the synthetic speech. For these reasons, the quality of this coder is consistently worse than that of the I-MELP/PCM coder. This result suggests that better quantization or encoding techniques are required to encode transition regions. Perhaps, a more efficient CELP coder would be a better choice for encoding the transition and unvoiced frames. As a final comment, although the average DMOS scores of the I-MELP/MP coder and the I-MELP[CLP] coder are statistically equal to each other, the nature of the distortions are usually different in these coders.

Finally, the DMOS scores of the unprocessed speech signal, the output of the pitch-cycle

Table 17: Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
I-MELP[CLP]	I-MELP/PCM	3.53	3.75	-0.22	[-0.38 -0.08]	Better
I-MELP[CLP]	I-MELP/MP	3.53	3.50	0.03	[-0.12 0.17]	Equal
I-MELP/PCM	I-MELP/MP	3.75	3.50	0.25	[0.10 0.41]	Worse

Table 18: Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
I-MELP[CLP]	I-MELP/PCM	3.47	3.74	-0.27	[-0.48 -0.05]	Better
I-MELP[CLP]	I-MELP/MP	3.47	3.41	0.06	[-0.14 0.27]	Equal
I-MELP/PCM	I-MELP/MP	3.74	3.41	0.33	[0.09 0.55]	Worse

Table 19: Comparison of the quality of the 2.4 kb/s I-MELP[CLP] coder and the variable rate coders for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
I-MELP[CLP]	I-MELP/PCM	3.59	3.77	-0.18	[-0.38 0.03]	Equal
I-MELP[CLP]	I-MELP/MP	3.59	3.60	-0.01	[-0.22 0.20]	Equal
I-MELP/PCM	I-MELP/MP	3.77	3.60	0.17	[-0.05 0.38]	Equal

modification algorithm and the I-MELP/PCM coder are compared statistically, and the results are tabulated in Table 20 for combined male and female speech, and in Table 21 and Table 22 for only male and only female speech, respectively. As seen in these tables, the quality of the modified speech signal is significantly different from the unprocessed speech signal. Although the average DMOS score is around 4.0, which means that the degradation is not annoying, the modified speech signal should be very close to the unprocessed speech signal. However, as the main purpose of the DCR test is to determine the degradation in the processed signal, it is not surprising that even a slight difference between the unprocessed and the modified speech results in lower scores. In addition, as there is no coding algorithm related degradation in the modified signals, the distortions related to the pitch-cycle modification algorithm using a wrong pitch-period estimation sounds very annoying. As the number of pitch-estimation errors are lower in female speech than in male speech, a score difference of 0.4 DMOS was obtained between the two genders. Furthermore, as there are more pitch-cycles in a frame in female speech at transition regions than in male speech, the initial selection of the segmentation locations are more reliable for female speech than for male speech. This property is the other reason for the large DMOS score difference. However, allowing additional delay in segmentation algorithm will likely reduce this type of distortion in male speech.

The comparison of the modified input speech and the I-MELP/PCM coder reveals interesting results. As the artifacts resulting from the switching between the excitation signal generated in the I-MELP coder and the transmitted waveform of the modified-residual signal is mostly eliminated, the quality difference between these two signal can only result from the encoding of the voiced part with the I-MELP coder. It was observed that the DMOS score difference between these two test cases is very small and statistically insignificant for the male speech. This result suggests that the I-MELP coder already encodes the voiced frames of the male speech very efficiently even at 2.4 kb/s. However, the large DMOS score difference for the female speech in the order of 0.44 DMOS also suggests that there is still a room for improvement in the speech quality of the I-MELP coder for female speech. However, further exploration of the reasons for this difference is beyond the scope of this

Table 20: Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for combined male and female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	4.96	4.00	0.96	[0.85 1.07]	Worse
Modified Input Signal	I-MELP/PCM	4.00	3.75	0.25	[0.09 0.40]	Worse

Table 21: Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for male speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	4.96	3.80	1.16	[0.99 1.32]	Worse
Modified Input Signal	I-MELP/PCM	3.80	3.73	0.07	[-0.15 0.30]	Equal

Table 22: Comparison of the quality of the unprocessed speech and the output of the pitch-cycle modification algorithm and the I-MELP/PCM coder for female speech using the DMOS scores in the DCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_{ref}	\bar{X}_{test}	\bar{X}_{diff}	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	4.96	4.20	0.76	[0.62 0.90]	Worse
Modified Input Signal	I-MELP/PCM	4.20	3.76	0.44	[0.23 0.63]	Worse

thesis.

The results of the DCR test were verified by the CCR tests whose results are tabulated in Table 23 for combined male and female speech and in Table 24 and Table 25 for only male and only female speech, respectively. Similar to the results obtained in the DCR test, the CCR test also proves that the modified speech is *slightly worse* than the input speech, however the difference is smaller for the female speech. This test also reveals that there is no quality difference between the I-MELP/MP coder and 2.4 kb/s I-MELP[CLP] coder, as expected. Finally, as it is also proved in the DCR test, the quality of the I-MELP[CLP] coder is statistically better than that of the MS-MELP coder for female speech while their quality is similar for male speech.

Table 23: Comparison of the various test cases for combined male and female speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_c	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	-0.86	[-0.96 -0.75]	Worse
I-MELP[CLP]	I-MELP/MP	0.07	[-0.01 0.16]	Equal
MS-MELP	I-MELP[CLP]	0.23	[0.10 0.36]	Better

Table 24: Comparison of the various test cases for male speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_c	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	-1.02	[-1.17 -0.87]	Worse
I-MELP[CLP]	I-MELP/MP	0.05	[-0.08 0.19]	Equal
MS-MELP	I-MELP[CLP]	-0.08	[-0.24 0.09]	Equal

Table 25: Comparison of the various test cases for female speech using the CMOS scores in the CCR test. The t-test is used to make the statistical analysis on the data obtained in the listening tests.

Reference Coder	Test Coder	\bar{X}_c	$C_{(95\%)}$	Result
Input Signal	Modified Input Signal	-0.69	[-0.84 -0.54]	Worse
I-MELP[CLP]	I-MELP/MP	0.09	[-0.01 0.20]	Equal
MS-MELP	I-MELP[CLP]	0.55	[0.37 0.72]	Better

CHAPTER VII

CONCLUSION

The main objective of this thesis was to develop techniques that improve the quality of low bit rate speech coders based on the linear-prediction model. Although recent improvements in parametric speech coding have resulted in high-quality low bit-rate speech coders, these coders still have problems with the encoding of the transition regions and short events. These problems also prevent further quality improvements by increasing bit rate. This thesis presents new techniques for accurate analysis and high-quality synthesis of the speech signal using individual pitch cycles. This pitch-synchronous approach allows a speech coder to capture perceptually important characteristics of the speech signal in both stationary and transition regions and to synthesize high-quality speech signal from the parameters of the linear-prediction model.

This thesis first presented the algorithms necessary for a pitch-synchronous analysis and synthesis system. These algorithms include a pitch-period estimation algorithm, two pitch-cycle segmentation algorithms, a new class of linear-prediction methods that estimates the linear-prediction filter from individual pitch cycles with fractional cycle lengths, and a new pitch cycle transformation method that normalizes the length of each pitch cycle to a constant length so that pitch-synchronous algorithms can easily be implemented using the transformed signal. In addition to these new methods, a new pitch-cycle modification algorithm was also introduced to modify the input signal without changing its perceptual characteristics so that different types of coders can be used to encode different segments of the speech signal.

All pitch-synchronous procedures require reliable estimation of the pitch period and the pitch-cycle boundaries. For this reason, an accurate pitch-estimation algorithm and two pitch-cycle segmentation algorithms were proposed in this thesis. The pitch-estimation algorithm first finds a pitch track that minimizes the pitch-prediction residual energy in

a number of subframes. The pitch period of a frame is obtained as the average pitch period of this pitch track. In experiments, it was observed that this new algorithm reduces the number of pitch-period estimation errors especially at onsets and transition regions. This estimated pitch period is later used in the new pitch-cycle segmentation algorithms. The first algorithm uses the linear-prediction gain as the periodicity measure. Although the segmentation accuracy of this algorithm is very high, occasional segmentation errors increase the chance of further segmentation errors in the subsequent pitch cycles. However, this algorithm has the advantage of using only the samples in a single cycle. The other segmentation algorithm uses the normalized correlation between two pitch cycles as the periodicity measure. The segmentation accuracy of this second method is exceptional. However, this algorithm requires two pitch cycles for reliable segmentation.

This thesis re-introduced the circular linear prediction (CLP) modeling that estimates the linear-prediction filter from a single cycle of a periodic signal. The quasi-periodic nature of the voiced speech signal is very well matched to this model. Experiments on synthetic speech signal proved that this method can find the linear-prediction filter from fully-voiced and noisy speech signals as reliably as the autocorrelation method. A multicycle generalization of the same method also improves the spectral estimation performance when the length of the cycles used in the analysis varies. In addition, this multicycle generalization also improves the reliability of the linear-prediction filter estimation when the signal is partially-voiced and the cycle lengths are short. As pitch-cycle lengths are rarely an integer in real speech signals, the CLP method was extended to use pitch cycles with fractional cycle lengths. In experiments on real speech signal, it was observed that this method fits the frequency response of the linear-prediction filter to the speech harmonics at transition segments better than the autocorrelation method. In addition, the estimation performance of this method at stationary segments is similar to the autocorrelation method. The circular processing of the speech signal was also used in another linear-prediction technique called as single pulse excited-circular linear prediction (SPE-CLP) modeling. This new model also takes into account the impulses with large amplitudes in the excitation signal. This new technique usually results in a linear-prediction filter whose frequency response better fits

to the speech harmonics even when the cycle length is very short. However, the synthetic speech experiments also revealed that this method is not suitable for partially-voiced speech signals. The multi-cycle generalization of this method was also presented in this thesis.

The constant pitch transformation (CPT) was also re-introduced in this thesis. This transformation normalizes the length of pitch cycles in the speech signal to a constant length to generate an alternative representation of the residual signal. As the harmonics of the resulting signal are always at fixed locations, this new signal is suitable for quantization purposes. In addition, it is also possible to invent simple and effective algorithms for processing the speech signal using individual pitch cycles. The zero-phase equalization filter is a good example of such an algorithm. It is also possible to use this method to synthesize pitch cycles with fractional cycle lengths from the Fourier series of the signal without the requirement of very high computational complexity.

A series of new speech coders that use these proposed techniques was also presented in this thesis. For low bit rate speech coding, instead of designing a new coder from scratch, the proposed methods were integrated into the 2.4 kb/s U.S. military standard MELP (MS-MELP) coder to improve its quality. This new coder is called as the improved-MELP (I-MELP) coder. The listening test results proved that the new methods improve the quality of this state-of-the-art speech coder for female speakers while preserving it for male speakers. It was shown that the overall quality of this new coder is statistically better than the MS-MELP coder. In addition, although it was also observed that the CLP method is not better than the autocorrelation method for low bit rate speech coding, it was also shown that it is indeed possible to reliably estimate the linear-prediction filter from a single cycle of speech signal.

In addition to a low bit rate implementation, a new class of parametric/hybrid coders based on the new I-MELP coder was also introduced in this thesis. For this purpose, a new pitch-cycle modification algorithm was developed that modifies the input signal without changing its perceptual characteristics such that it becomes time-synchronous with the synthesized signal and the waveform of both processed and I-MELP encoded signals are similar. This new algorithm allows a speech coder design that generates an excitation signal

from a fully-parametric representation at voiced segments and transmits the waveform of the residual signal at unvoiced and transition segments without introducing any audible artifacts. This new technique is used in an experimental I-MELP/PCM and I-MELP/MP coders. Both of these coders use the I-MELP coder to encode and synthesize the voiced speech segments. During transition and unvoiced segments, the modified-residual signal is transmitted to the decoder in the I-MELP/PCM coder and is encoded using a MP algorithm in the I-MELP/MP coder. The listening tests proved that the I-MELP/PCM coder slightly improves the quality of the I-MELP coder. However because of the simple design of the MP coder, this improvement was shadowed with other types of minor distortions that resulted in an overall similar quality with the 2.4 kb/s I-MELP coder. Overall, the improvements in I-MELP/PCM coder prove that it is indeed possible to improve the quality of the current low bit rate coder by efficiently encoding the transition and unvoiced segments.

7.1 *Future Work*

The pitch-synchronous processing of speech signal is one of the least explored areas in speech coding research. This thesis proves that it is possible to capture the perceptually important characteristics of the speech signal reliably from individual pitch cycles and improve the quality of the fully parametric speech coders by synthesizing individual pitch cycles with fractional cycle lengths. However, the work presented in this thesis can still be improved in a number of ways.

First of all, every pitch-synchronous speech analysis/synthesis system always depend on the correct pitch-cycle segmentation of the speech signal and that is only possible when the pitch period is estimated correctly. The proposed pitch-period estimation algorithm in this thesis finds the pitch-period accurately most of the time even at onsets and in transition regions. However, it was also observed that the algorithm still occasionally makes mistakes, especially for male speakers at onsets. The blind segmentation method used in the pitch-cycle segmentation algorithm to find the correct segmentation locations at onsets can also be used as a pitch-period estimation algorithm. This idea can be used to improve the accuracy of the current pitch-period estimation algorithm, especially for male speakers.

One of the current constraints in the design of the algorithms is preserving the delay of the MS-MELP coder. Although the delay of the 2.4 kb/s I-MELP coders are increased because of the accurate calculation of the residual signal, this extra delay is not used in either the pitch-period estimation or the pitch-cycle segmentation algorithms. By allowing an additional small amount of delay, future pitch-period values can be estimated and pitch smoothing techniques can be used to reduce the pitch-estimation errors, especially at onsets for male speakers. In addition, this additional delay can also be used in the pitch-cycle segmentation algorithm so that the initial segmentation location can be placed in stationary segments and the segment locations in the transition frames can be obtained by processing the speech signal backwards in time. This method not only improves the segmentation accuracy, but also provides better segmentation locations at the onsets that will reduce the audible artifacts in the output of the pitch-cycle modification algorithm resulting from the badly segmented speech signal.

Another area that can be further explored is the pulse excited-circular linear prediction (PE-CLP) modeling. The new SPE-CLP method introduced in this thesis takes into account the impulses with large amplitudes in the excitation signal. However, the assumption of the presence of only a single pulse in the excitation signal is one of the main shortcomings of this new method. Perhaps, allowing multiple pulses in the excitation signal would improve the performance of the SPE-CLP method.

One of the other areas that requires further exploration is the encoding of the female speech with a MELP model. The results of the subjective listening tests proved that the quality of the I-MELP/PCM encoded speech for male speakers is similar to the modified-input speech. On the other hand, there is a significant quality difference between these two signals for female speakers. An exploration of what may cause this difference, and a solution to this problem may improve the overall quality of current coders.

The proposed techniques in this thesis can also be used in a true variable bit rate speech coder that encodes the speech signal with the I-MELP coder between 1.6 and 4.0 kb/s while encodes the waveform of the excitation signal above 4.0 kb/s using well-known techniques such as CELP. Even a super-frame approach can be used to compress the speech signal at

and below 1.2 kb/s within the same variable rate speech coder.

Finally, the pitch-synchronous processing of speech signal is also suitable for other speech processing applications. These techniques can be used to make time-scale or pitch-scale modification of the speech signal without affecting the quality. Furthermore, they can be used in natural-sounding high-quality text-to-speech synthesis applications as well.

APPENDIX A

PITCH-PERIOD ESTIMATION ALGORITHM

APPENDICES

This appendix presents the proof for the equivalence of the pitch-prediction error minimization and the frame's pitch-track's normalized correlation maximization. Also, the details about the decision logic used in the pitch-estimation algorithm are given in this appendix.

A.1 The Proof for the Equivalence of Pitch-Prediction Error Minimization and Frame's Normalized Correlation Maximization

In this section, the equivalence of minimization of ε_Γ and maximization of ρ_Γ will be proved.

In Chapter 3.1, the track's pitch-prediction residual energy, ε_Γ , is defined as

$$\begin{aligned}
 \varepsilon_\Gamma &= \sum_{s=1}^K \varepsilon_{\tau_s} \\
 &= \sum_{s=1}^K \left[\langle x_0, x_0 \rangle_s - \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \\
 &= \sum_{s=1}^K \left[P_s - \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right], \tag{130}
 \end{aligned}$$

where P_s is s^{th} subframe's energy and equal to $\langle x_0, x_0 \rangle_s$, τ_s is the pitch lag of the s^{th} subframe, and $\langle x_k, x_l \rangle_s$ is defined in (57). The same equation can be written as

$$\begin{aligned}
 \varepsilon_\Gamma &= \sum_{s=1}^K P_s - \sum_{s=1}^K \left[\frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \\
 &= P - \sum_{s=1}^K \left[\frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \\
 &= P \left[1 - \frac{1}{P} \sum_{s=1}^K \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \\
 &= P \hat{\varepsilon}_\Gamma, \tag{131}
 \end{aligned}$$

where P is the energy of the whole frame, and $\hat{\varepsilon}_\Gamma$ is defined as

$$\hat{\varepsilon}_\Gamma = \left[1 - \frac{1}{P} \sum_{s=1}^K \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \quad (132)$$

Since P is constant regardless of τ_s , minimization of $\hat{\varepsilon}_\Gamma$ is the same as minimization of ε_Γ .

Note that $\hat{\varepsilon}_\Gamma$ can be written as

$$\begin{aligned} \hat{\varepsilon}_\Gamma &= \left[1 - \frac{1}{P} \sum_{s=1}^K \langle x_0, x_0 \rangle_s \frac{\langle x_0, x_{\tau_s} \rangle_s^2}{\langle x_0, x_0 \rangle_s \langle x_{\tau_s}, x_{\tau_s} \rangle_s} \right] \\ &= \left[1 - \frac{\sum_{s=1}^K P_s \rho_{\tau_s}^2}{P} \right] \\ &= 1 - \rho_\Gamma^2. \end{aligned} \quad (133)$$

By definition, ρ_Γ^2 is bounded by zero and one, hence $\hat{\varepsilon}_\Gamma$ is also bounded by zero and one. Furthermore, $\hat{\varepsilon}_\Gamma$ decreases as ρ_Γ^2 increases. For these reasons, maximization of ρ_Γ^2 is the same as minimization of $\hat{\varepsilon}_\Gamma$, and also, minimization of ε_Γ , which concludes the proof.

A.2 The Decision Logic of the Pitch Estimation Algorithm

In this section, the details of the decision logic used in the pitch-estimation algorithm are given. This logic is used to find the frame's pitch period and correlation level by using a unique algorithm for each combination of correlation level of the speech and residual signals. These levels are assigned according to the correlation degree of the primary pitch candidate of each signal and given in Table 2 in Chapter 3.1.

The following parameters are used in the decision block:

- τ_{PR}^{Sp} : The primary pitch candidate obtained from the speech signal.
- τ_{PR}^{Rs} : The primary pitch candidate obtained from the residual signal.
- τ_k^{Sp} : The pitch candidate obtained in the k^{th} octave region from the speech signal.
- τ_k^{Rs} : The pitch candidate obtained in the k^{th} octave region from the residual signal.
- ρ_{PR}^{Sp} : The correlation degree at the primary pitch candidate obtained from the speech signal.

- ρ_{PR}^{Rs} : The correlation degree at the primary pitch candidate obtained from the residual signal.
- ρ_k^{Sp} : The correlation degree at the pitch candidate obtained in the k^{th} octave region from the speech signal.
- ρ_k^{Rs} : The correlation degree at the pitch candidate obtained in the k^{th} octave region from the residual signal.
- τ^{Avg} : Average pitch-period obtained from strongly correlated and high energy frames.
- K_{Sp} : The octave region of the primary pitch candidate obtained from the speech signal.
- K_{Rs} : The octave region of the primary pitch candidate obtained from the residual signal.

The decision block generates the following parameters:

- τ_{FR} : The frame's pitch-period
- ρ_{FR} : The frame's correlation degree

A simple pitch-doubling elimination algorithm is also used to eliminate the remaining pitch-doubling errors. The pseudo-code of this algorithm is as follows

```

Inputs:  $\tau_k^{Sig}$  and  $\rho_k^{Sig}$  for all octave regions,  $K_{Sig}$ , and dblcoeff
Outputs:  $\tau_{out}$ ,  $\rho_{out}$ 

if  $K_{Sig} \neq 1$ , % If  $K_{Sig}$  is not the first octave region
    if  $\rho_{K-1}^{Sig} > \text{dblcoeff} * \rho_K^{Sig}$ 
         $\tau_{out} = \tau_{K-1}^{Sig}$ 
         $\rho_{out} = \rho_{K-1}^{Sig}$ 
    else
         $\tau_{out} = \tau_K^{Sig}$ 
         $\rho_{out} = \rho_K^{Sig}$ 
    endif
else
     $\tau_{out} = \tau_K^{Sig}$ 
     $\rho_{out} = \rho_K^{Sig}$ 
endif

```


The decision algorithm for each correlation level of the signals is as follows:

- *Speech Signal: SC, Residual Signal: SC/MC :*

```

if  $K_{Sp} \neq 1$ , % If  $K_{Sp}$  is not the first octave region
    if  $|\tau_{K_{Sp}-1}^{Sp} - \tau_{K_{Rs}-1}^{Rs}| < 3$  AND  $\rho_{K_{Rs}-1}^{Rs} > 0.8$ 
         $\tau_{FR}, \tau_{FR} = \text{DblCheck}(\tau_k^{Sp}, \rho_k^{Sp}, K_{Sp}, 0.9)$ 
    endif
else
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
endif

```

- *Speech Signal: SC, Residual Signal: WC/NC :*

```

 $\text{DblCheck}(\tau_k^{Sp}, \rho_k^{Sp}, K_{Sp}, 0.9)$ 

```

- *Speech Signal: MC, Residual Signal: SC :*

```

if  $|\tau_{K_{Sp}}^{Sp} - \tau_{K_{Rs}}^{Rs}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\tau_{FR} = \rho_{PR}^{Sp}$ 
else
     $\text{DblCheck}(\tau_k^{Rs}, \rho_k^{Rs}, K_{Rs}, 0.9)$ 
endif

```

- *Speech Signal: MC, Residual Signal: MC :*

```

if  $K_{Sp} \neq 1$ , % If  $K_{Sp}$  is not the first octave region
    if  $\rho_{K_{Rs}-1}^{Rs} > 0.8 * \rho_{K_{Rs}}^{Rs}$ 
         $\tau_{FR}, \tau_{FR} = \text{DblCheck}(\tau_k^{Sp}, \rho_k^{Sp}, K_{Sp}, 0.9)$ 
    endif
else
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
endif

```


- *Speech Signal: MC, Residual Signal: WC :*

```

if  $|\tau_{K_{Sp}}^{Sp} - \tau_{K_{Rs}}^{Rs}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
else if  $|\tau_{K_{Rs}}^{Rs} - \tau^{Avg}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Rs}$ 
     $\rho_{FR} = \rho_{PR}^{Rs}$ 
else
     $\tau_{FR}, \tau_{FR} = \text{DblCheck}(\tau_k^{Sp}, \rho_k^{Sp}, K_{Sp}, 0.85)$ 
endif

```

- *Speech Signal: MC, Residual Signal: NC :*

```

DblCheck( $\tau_k^{Sp}, \rho_k^{Sp}, K_{Sp}, 0.9$ )

```

- *Speech Signal: WC/NC, Residual Signal: SC/MC :*

```

if  $|\tau_{K_{Sp}}^{Sp} - \tau_{K_{Rs}}^{Rs}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
else
     $\tau_{FR}, \tau_{FR} = \text{DblCheck}(\tau_k^{Rs}, \rho_k^{Rs}, K_{Rs}, 0.9)$ 
endif

```

- *Speech Signal: WC, Residual Signal: WC/NC :*

```

if  $|\tau_{K_{Sp}}^{Sp} - \tau^{Avg}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
else if  $|\tau_{K_{Rs}}^{Rs} - \tau^{Avg}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Rs}$ 
     $\rho_{FR} = \rho_{PR}^{Rs}$ 
else
     $\tau_{FR} = \tau^{Avg}$ 
     $\rho_{FR} = 0.0$ 
endif

```


- *Speech Signal: NC, Residual Signal: WC :*

```

if  $|\tau_{K_{Sp}}^{Sp} - \tau^{Avg}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Sp}$ 
     $\rho_{FR} = \rho_{PR}^{Sp}$ 
else if  $|\tau_{K_{Rs}}^{Rs} - \tau^{Avg}| < 2$ 
     $\tau_{FR} = \tau_{PR}^{Rs}$ 
     $\rho_{FR} = \rho_{PR}^{Rs}$ 
else
     $\tau_{FR} = \tau^{Avg}$ 
     $\rho_{FR} = 0.0$ 
endif

```

- *Speech Signal: NC, Residual Signal: NC :*

```

if  $\rho_{K_{Sp}}^{Sp} > 0.45$  AND  $\rho_{K_{Rs}}^{Rs} > 0.45$ 
    if  $|\tau_{K_{Sp}}^{Sp} - \tau_{K_{Rs}}^{Rs}| < 2$ 
         $\tau_{FR} = \tau_{PR}^{Sp}$ 
         $\rho_{FR} = \rho_{PR}^{Sp}$ 
    else
         $\tau_{FR} = \tau^{Avg}$ 
         $\rho_{FR} = 0.0$ 
    endif
else
     $\tau_{FR} = \tau^{Avg}$ 
     $\rho_{FR} = 0.0$ 
endif

```


APPENDIX B

LEVINSON-DURBIN RECURSION

The Levinson-Durbin recursion is an efficient technique for solving a set of linear equations by eliminating the necessity of a matrix inversion. To use this recursion, the weight matrix of the matrix-vector representation of the linear equation set (e.g. (11)) has to be a Toeplitz matrix (e.g (17)) and the weighted sum of the unknown coefficients in all equations must be zero except the first equation. When these constraints are satisfied, this recursion can be used to compute a reflection coefficient and a new set of predictor coefficients in each iteration.

This method can be used to solve the linear equations obtained by the autocorrelation and the CLP methods by adding the sum of squared prediction error to linear equation set given in (17):

$$\begin{aligned}
 \varepsilon_{LP} &= \sum_{n=-\infty}^{\infty} (x[n] - \hat{x}[n])^2 \\
 &= \sum_{n=-\infty}^{\infty} x[n]^2 - 2 \sum_{k=1}^p a_k \sum_{n=-\infty}^{\infty} x[n]x[n-k] + \sum_{k=1}^p \sum_{l=1}^p a_k a_l \sum_{n=-\infty}^{\infty} x[n-k]x[n-l] \\
 &= r(0) - 2 \sum_{k=1}^p a_k r(k) + \sum_{k=1}^p \sum_{l=1}^p a_k a_l r(|k-l|) \\
 &= r(0) - 2\mathbf{a}^T \mathbf{r} + \mathbf{a}^T \mathbf{R} \mathbf{a}, \\
 &= r(0) - \mathbf{a}^T \mathbf{r}
 \end{aligned} \tag{134}$$

where \mathbf{R} is defined in (17), and \mathbf{a} and \mathbf{r} are defined in (12). Note that, when the optimum \mathbf{a} is used, $\mathbf{R} \mathbf{a}$ is equal to \mathbf{r} , and hence, $\mathbf{a}^T \mathbf{R} \mathbf{a}$ is equal to $\mathbf{a}^T \mathbf{r}$. Before adding this new

equation, the matrix representation of the equations with Toeplitz \mathbf{R} can be written as

$$\begin{bmatrix} r(1) & r(0) & r(1) & \dots & \dots & r(p-1) \\ r(2) & r(1) & r(0) & \dots & \dots & r(p-2) \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ r(p) & r(p-1) & r(p-2) & \dots & \dots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ \vdots \\ -a_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}. \quad (135)$$

The sum of the squared prediction error, ε_{LP} , can be added to this representation to generate the following equation set

$$\begin{bmatrix} r(0) & r(1) & r(2) & \dots & \dots & r(p) \\ r(1) & r(0) & r(1) & \dots & \dots & r(p-1) \\ r(2) & r(1) & r(0) & \dots & \dots & r(p-2) \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ r(p) & r(p-1) & r(p-2) & \dots & \dots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ \vdots \\ -a_p \end{bmatrix} = \begin{bmatrix} \varepsilon_{LP} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}. \quad (136)$$

The new equation set in (136) satisfies the requirements of the Levinson-Durbin recursion.

The Levinson-Durbin recursion assumes that the optimum prediction coefficients and the associated prediction error are already computed for a given order and the problem is to find a new set of prediction filter coefficients whose order is one larger than the computed one. Assume that, the coefficients in the vector, $[-a_1^{j-1} \dots -a_{j-1}^{j-1}]$, and the sum of squared prediction error, ε_{LP}^{j-1} , have already been calculated. A new set of linear equation can be written as

$$\begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(j-1) & r(j) \\ r(1) & r(0) & r(1) & \dots & r(j-2) & r(j-1) \\ r(2) & r(1) & r(0) & \dots & r(j-3) & r(j-2) \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ r(j-1) & r(j-2) & r(j-3) & \dots & r(0) & r(1) \\ r(j) & r(j-1) & r(j-2) & \dots & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_1 \\ -a_2 \\ \vdots \\ \vdots \\ -a_{j-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \varepsilon_{LP}^{j-1} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \psi_j \end{bmatrix}. \quad (137)$$

Futhermore, same set of equations can also be written as

$$\begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(j-1) & r(j) \\ r(1) & r(0) & r(1) & \dots & r(j-2) & r(j-1) \\ r(2) & r(1) & r(0) & \dots & r(j-3) & r(j-2) \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ r(j-1) & r(j-2) & r(j-3) & \dots & r(0) & r(1) \\ r(j) & r(j-1) & r(j-2) & \dots & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 0 \\ -a_{j-1} \\ -a_{j-2} \\ \vdots \\ \vdots \\ -a_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \psi_j \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \varepsilon_{LP}^{j-1} \end{bmatrix}. \quad (138)$$

Note that, since \mathbf{R} is a symmetric Toeplitz matrix, the weight matrices in both (137) and (138) are the same. Therefore, these two equations can be combined as

$$\begin{bmatrix} r(0) & r(1) & \dots & r(j) \\ r(1) & r(0) & \dots & r(j-1) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ r(j) & r(j-1) & \dots & r(0) \end{bmatrix} \hat{\mathbf{a}}_{j-1} = \left(\begin{bmatrix} \varepsilon_{LP}^{j-1} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \psi_j \end{bmatrix} + \Gamma_j \begin{bmatrix} \psi_j \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \varepsilon_{LP}^{j-1} \end{bmatrix} \right), \quad (139)$$

where $\hat{\mathbf{a}}_{j=1}$ is defined as

$$\hat{\mathbf{a}}_{j-1} = \left(\begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ \vdots \\ -a_{j-1} \\ 0 \end{bmatrix} + \Gamma_j \begin{bmatrix} 0 \\ -a_{j-1} \\ \vdots \\ \vdots \\ -a_1 \\ 1 \end{bmatrix} \right) \quad (140)$$

This new set of equations would satisfy the linear equation set for a j^{th} order prediction filter only when

$$\psi_j + \Gamma_j \varepsilon_{LP}^{j-1} = 0. \quad (141)$$

Since ψ_j is equal to $r(j) - \sum_{k=1}^{j-1} a_k^{j-1} r(j-k)$, the reflection coefficient, Γ_j , for the j^{th}

iteration is obtained as

$$\Gamma_j = -\frac{r(j) - \sum_{k=1}^{j-1} a_k^{j-1} r(j-k)}{\varepsilon_{LP}^{j-1}}, \quad (142)$$

where ε_{LP}^{j-1} is equal to $r(0) - \sum_{k=1}^{j-1} a_k^{j-1} r(k)$. One of the interesting side product of this recursion is that ε_{LP}^j can also be calculated recursively:

$$\begin{aligned} \varepsilon_{LP}^j &= \varepsilon_{LP}^{j-1} + \Gamma_j \psi_j \\ &= \varepsilon_{LP}^{j-1} (1 - \Gamma_j^2). \end{aligned} \quad (143)$$

Finally, the new set of predictor coefficients for the j^{th} step can be found as

$$\begin{aligned} a_k^j &= a_k^{j-1} + \Gamma_j a_{j-k}^{j-1} \quad k=1..j-1 \\ a_j^j &= -\Gamma_j. \end{aligned} \quad (144)$$

The equations (142), (143) and (144) can be used recursively until the predictor coefficients with desired order is obtained. In the initialization stage, ε_{LP}^0 is set to $r(0)$ and Γ_1 is computed as

$$\Gamma_1 = -\frac{r(1)}{r(0)}. \quad (145)$$

Solving the linear equations of the autocorrelation method and the CLP method by Levinson-Durbin recursion has many advantages over solving them by matrix inversion. While the computational complexity of Gaussian elimination is on the order of $O(n^3)$, the computational complexity of Levinson-Durbin recursion is on the order of $O(n^2)$. Furthermore, as a side product, Levinson-Durbin recursion also computes the reflection coefficients which can be used directly in a lattice filter implementation of the same all-pole model. Finally, although the stability of the filter is always guaranteed in autocorrelation method, the filter may become unstable resulting from the rounding errors in a digital signal processing application. The Levinson-Durbin recursion makes it possible to make a simple test to check the stability of the filter in each iteration, as absolute value of all reflection coefficients of a stable filter must be less than one. Currently, almost all speech coding applications use the Levinson-Durbin recursion in conjunction with the autocorrelation method to obtain the linear-prediction coefficients because of the low computational complexity and guaranteed filter stability properties.

APPENDIX C

EQUIVALENCE OF THE CLP ANALYSIS TO COMMON LINEAR-PREDICTION ESTIMATION ALGORITHMS

In this section, the equivalence of the CLP analysis to the common LP estimation methods are proved. This equivalence holds only when the estimation method are used to model an infinitely periodic signal or a single pitch-cycle of this signal. These methods include:

- Modified covariance method
- Forward covariance method
- Backward covariance method
- Burg's method
- Discrete-spectra linear-prediction modeling

C.1 Equivalence of the CLP Method to the Modified Covariance Method

In this section, the equivalence of modified covariance method to the CLP estimation method when applied to an infinitely periodic signal is proved. Similar to the CLP method, modeling an infinitely periodic signal is also the same as modeling only single pitch cycle of this infinite signal in the modified covariance method. As a result, for this special case, the modeling region for the modified covariance is set to $\tau + p$ samples so that only one cycle of the infinite signal is used in modeling. In this case, the sum of the squared error, ε_{MC}^2 , is defined as

$$\varepsilon_{MC}^2 = \sum_{n=0}^{\tau-1} \left[x[n] - \sum_{k=1}^p a_k x[n-k] \right]^2 + \sum_{n=0}^{\tau-1} \left[x[n-p] - \sum_{k=1}^p a_k x[n-p+k] \right]^2. \quad (146)$$

When this error is minimized with respect to predictor coefficients, a set of linear equations given in (21) is obtained. However, as the modeling region contains only one cycle, any

offset added to both k and l in (18) does not change the result. Hence, $r(k, l)$ can be written as $r(|k - l|)$, and (21) reduces to

$$2 \sum_{k=1}^p a_k r(|k - l|) = 2r(l) \quad l = 1..p, \quad (147)$$

where $r(|k - l|)$ is computed as

$$r(|k - l|) = r(i) = \sum_{n=0}^{\tau-1} x[n]x[n \pm i], \quad (148)$$

which is the same as (72). Therefore, the prediction coefficients obtained by the modified covariance method is the same as the ones obtained by the CLP method when only $\tau + p$ samples of this infinitely periodic signal is modeled.

C.2 Equivalence of the CLP Method to the Forward Covariance Method

In the next three sections, the equivalence of the CLP estimation method with three all-pole lattice-filter design techniques is proved, when these techniques are used to model an infinitely periodic signal. The proof makes use of Levinson-Durbin recursion described in Appendix 2.2.1 as it is also a recursive technique and finds an optimum reflection coefficient in one step at a time as in the case for these lattice-filter design methods.

In this proof, it is assumed that the first $j - 1$ partial correlation coefficients have already been calculated, and they are equal to the reflection coefficients obtained by Levinson-Durbin recursion using the correlation coefficients obtained by CLP method. It is also assumed that since the analyzed signal is an infinitely periodic signal, the sum of squared forward error prediction signal, ε_j^+ , is modified as

$$\varepsilon_j^+ = \sum_{n=-\infty}^{\infty} |e_j^+[n]|^2. \quad (149)$$

The optimum partial correlation coefficient, Γ_j^+ , can be found by setting the partial derivative of the new ε_j^+ with respect to Γ_j^+ to zero and solving the linear equations. The partial derivative is computed as

$$\frac{\partial \varepsilon_j^+}{\partial \Gamma_j^+} = 2 \sum_{n=-\infty}^{\infty} e_{j-1}^-[n-1] \left[e_{j-1}^+[n] + \Gamma_j^+ e_{j-1}^-[n-1] \right] = 0, \quad (150)$$

where $e_j^+[n]$ and $e_j^-[n]$ are defined in (22) and (23), respectively. Since, the both $e_j^+[n]$ and $e_j^-[n]$ are infinitely periodic signals with the same period length, (150) can be written as

$$\frac{\partial \varepsilon_j^+}{\partial \Gamma_j^+} = \lim_{N \rightarrow \infty} N \sum_{n=0}^{\tau-1} e_{j-1}^-[n-1] \left[e_{j-1}^+[n] + \Gamma_j^+ e_{j-1}^-[n-1] \right] = 0. \quad (151)$$

This equation can only be satisfied when

$$\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1] \left[e_{j-1}^+[n] + \Gamma_j^+ e_{j-1}^-[n-1] \right] = 0. \quad (152)$$

Γ_j^+ can be obtained by solving (152) as

$$\Gamma_j^+ = - \frac{\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1] e_{j-1}^+[n]}{\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1]^2}. \quad (153)$$

Note that, (152) also proves that the same Γ_j^+ can be computed when ε_j^+ is defined as

$$\varepsilon_j^+ = \sum_{n=0}^{\tau-1} |e_j^+[n]|^2, \quad (154)$$

which means that modeling of this infinite signal is the same as modeling of a single pitch cycle of this signal. As it is going to be used in the proof, the sum of squared backward prediction error, ε_j^- , can be defined similarly as

$$\begin{aligned} \varepsilon_j^- &= \sum_{n=-\infty}^{\infty} |e_j^-[n-1]|^2 \\ &= \lim_{N \rightarrow \infty} N \sum_{n=0}^{\tau-1} |e_j^-[n-1]|^2. \end{aligned} \quad (155)$$

The equivalence of forward covariance analysis with the new ε_j^+ definition and CLP analysis can be shown by proving the equivalence of Γ_j^+ obtained by (153) and Γ_j obtained by the Levinson-Durbin recursion. To show this equivalence, it is necessary to show the equivalence of both denominator and numerator parts of both equations ((153) and (142)).

The denominator part of (153) is equal to ε_{j-1}^- . When extended, it can be written as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \sum_{n=0}^{\tau-1} (e_{j-1}^-[n-1])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1] + \Gamma_{j-1}^+ e_{j-2}^+[n])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 + 2\Gamma_{j-1}^+ \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n] + (\Gamma_{j-1}^+)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 \\
&= \varepsilon_{j-2}^- + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^+ \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n].
\end{aligned} \tag{156}$$

Note that (153) for the previous iteration can be written as

$$-\Gamma_{j-1}^+ \varepsilon_{j-2}^- = \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n]. \tag{157}$$

As a result, it is possible to extend (156) as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \varepsilon_{j-2}^- + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^+ (-\Gamma_{j-1}^+ \varepsilon_{j-2}^-) \\
&= \varepsilon_{j-2}^- + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^+ - 2\Gamma_{j-1}^{+2} \varepsilon_{j-2}^- \\
&= \varepsilon_{j-2}^- (1 - (\Gamma_{j-1}^+)^2) + (\Gamma_{j-1}^+)^2 (\varepsilon_{j-2}^+ - \varepsilon_{j-2}^-)
\end{aligned} \tag{158}$$

Similarly ε_{j-1}^+ can be extended as

$$\begin{aligned}
\varepsilon_{j-1}^+ &= \sum_{n=0}^{\tau-1} (e_{j-1}^+[n])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n] + \Gamma_{j-1}^+ e_{j-2}^-[n-1])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 + 2\Gamma_{j-1}^+ \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1] + (\Gamma_{j-1}^+)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 \\
&= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^+ \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1].
\end{aligned} \tag{159}$$

Using (157), (159) can be further extended as

$$\begin{aligned}
\varepsilon_{j-1}^+ &= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^+ (-\Gamma_{j-1}^+ \varepsilon_{j-2}^-) \\
&= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^- - 2(\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^- \\
&= \varepsilon_{j-2}^+ - (\Gamma_{j-1}^+)^2 \varepsilon_{j-2}^-
\end{aligned} \tag{160}$$

If ε_{j-2}^+ and ε_{j-2}^- is assumed equal to each other, ε_{j-1}^- and ε_{j-1}^+ can be written as

$$\begin{aligned}\varepsilon_{j-1}^- &= \varepsilon_{j-2}^-(1 - (\Gamma_{j-2}^+)^2) \\ \varepsilon_{j-1}^+ &= \varepsilon_{j-2}^+(1 - (\Gamma_{j-2}^+)^2),\end{aligned}\tag{161}$$

which makes ε_{j-1}^+ and ε_{j-1}^- equal also. This assumption holds if and only if ε_0^+ and ε_0^- are equal to each other. Since both forward and backward prediction error signals are set to the input signal in the initialization, both ε_0^+ and ε_0^- are equal to the energy of the signal, which makes the initial assumption correct. (161) proves that the denominator part of (153) can be calculated recursively as in the case of Levinson-Durbin recursion. In fact, since the ε_0 is also set to the energy of the input signal in Levinson-Durbin recursion and it is assumed that first $j-1$ partial correlation coefficients calculated by the forward covariance method and the Levinson-Durbin recursion are equal, ε_{j-1}^- is equal to ε_{j-1} found by the Levinson-Durbin recursion.

The numerator part of (153) can be extended as

$$\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1]e_{j-1}^+[n] = \sum_{n=0}^{\tau-1} \left[x[n] - \sum_{k=1}^{j-1} a_k^{j-1} x[n-k] \right] \left[x[n-j] - \sum_{l=1}^{j-1} a_l^{j-1} x[n-j+l] \right],\tag{162}$$

using the definition of forward and backward error prediction. The extension of this equations results into

$$\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1]e_{j-1}^+[n] = r(j) - \sum_{l=1}^{j-1} a_l^{j-1} r(j-l) - \sum_{k=1}^{j-1} a_k^{j-1} r(j-k) + \sum_{l=1}^{j-1} \sum_{k=1}^{j-1} a_l^{j-1} a_k^{j-1} r(j-k-l).\tag{163}$$

However, when the first $j-1$ partial correlation coefficients calculated by forward covariance method are equal to the first $j-1$ reflection coefficients calculated by Levinson-Durbin recursion, the following equation is always satisfied:

$$\sum_{k=1}^{j-1} a_k^{j-1} r(j-k) = \sum_{l=1}^{j-1} \sum_{k=1}^{j-1} a_l^{j-1} a_k^{j-1} r(j-k-l)\tag{164}$$

Using (164), it is possible to simplify (163) as

$$\sum_{n=0}^{\tau-1} e_{j-1}^-[n-1]e_{j-1}^+[n] = r(j) - \sum_{l=1}^{j-1} a_l^{j-1} r(j-l).\tag{165}$$

Equation (165) proves that the numerator of (153) is identical to numerator of (142). This proves that the partial correlation coefficient obtained by the forward covariance method in an iteration is identical to the reflection coefficient obtained by the Levinson-Durbin recursion in the same iteration if and only if all of the previous partial correlation coefficients and the reflection coefficients are the same. To conclude the proof, it is sufficient to show that the first partial correlation coefficient is equal to the first reflection coefficient.

First reflection coefficient can be computed by Levinson-Durbin recursion as

$$\Gamma_1 = -\frac{r(1)}{r(0)}, \quad (166)$$

where $r(i)$ is computed with (72). First partial correlation coefficient can be computed with the forward covariance method as

$$\begin{aligned} \Gamma_1^+ &= -\frac{\sum_{n=0}^{\tau-1} e_0^-[n-1]e_0^+[n]}{\sum_{n=0}^{\tau-1} e_0^-[n-1]^2} \\ &= -\frac{\sum_{n=0}^{\tau-1} x[n-1]x[n]}{\sum_{n=0}^{\tau-1} x[n-1]^2} \\ &= -\frac{r(1)}{r(0)}. \end{aligned} \quad (167)$$

Since Γ_1 is equal to Γ_1^+ , the prediction coefficients obtained by the forward covariance method is identical to the one obtained by CLP analysis method, which concludes the proof.

C.3 Equivalence of the CLP Method to the Backward Covariance Method

In this proof, it is also assumed that the first $j - 1$ partial correlation coefficients have already been calculated and they are equal to the reflection coefficients obtained by the Levinson-Durbin recursion using the correlation coefficients obtained by the CLP method. It is also assumed that the analyzed signal is an infinitely periodic signal, therefore, ε_j^+ and ε_j^- are defined as (154) and (155), respectively.

When ε_j^- is minimized with respect to Γ_j^- and the resulting equation is solved, Γ_j^- is obtained as

$$\Gamma_j^- = -\frac{\sum_{n=0}^{\tau-1} e_{j-1}^+[n]e_{j-1}^-[n-1]}{\sum_{n=0}^{\tau-1} |e_{j-1}^+[n]|^2}. \quad (168)$$

The equivalence of backward covariance analysis with the new ε_j^- definition and CLP analysis can be shown by proving the equivalence of Γ_j^- obtained by (168) and Γ_j obtained in the Levinson-Durbin recursion. Since numerator parts of (168) and (153) are the same and the equivalence of Γ_j^+ and Γ_j has already been proved, it is sufficient to show only the equivalence of the denominator parts of (168) and (142).

The denominator part of (168) is equal to ε_{j-1}^+ . When it is extended, it can be written as

$$\begin{aligned}
\varepsilon_{j-1}^+ &= \sum_{n=0}^{\tau-1} (e_{j-1}^+[n])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n] + \Gamma_{j-1}^- e_{j-2}^-[n-1])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 + 2\Gamma_{j-1}^- \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1] + (\Gamma_{j-1}^-)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 \\
&= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^- \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1].
\end{aligned} \tag{169}$$

Note that (168) for the previous iteration can be written as

$$-\Gamma_{j-1}^- \varepsilon_{j-2}^+ = \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1]. \tag{170}$$

Using (170), (169) can be further extended as

$$\begin{aligned}
\varepsilon_{j-1}^+ &= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^- (-\Gamma_{j-1}^- \varepsilon_{j-2}^+) \\
&= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^- - 2(\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+ \\
&= \varepsilon_{j-2}^+ (1 - (\Gamma_{j-1}^-)^2) + (\Gamma_{j-1}^-)^2 (\varepsilon_{j-2}^- - \varepsilon_{j-2}^+)
\end{aligned} \tag{171}$$

Similarly ε_{j-1}^- can be extended as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \sum_{n=0}^{\tau-1} (e_{j-1}^-[n-1])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1] + \Gamma_{j-1}^- e_{j-2}^+[n])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 + 2\Gamma_{j-1}^- \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n] + (\Gamma_{j-1}^-)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 \\
&= \varepsilon_{j-2}^- + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^- \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n].
\end{aligned} \tag{172}$$

Using (170), (172) can be further extended as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \varepsilon_{j-2}^- + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^- (-\Gamma_{j-1}^- \varepsilon_{j-2}^+) \\
&= \varepsilon_{j-2}^- + (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+ - 2(\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+ \\
&= \varepsilon_{j-2}^- - (\Gamma_{j-1}^-)^2 \varepsilon_{j-2}^+.
\end{aligned} \tag{173}$$

As in the case of the proof in the previous section, if it is assumed that ε_{j-2}^+ and ε_{j-2}^- are equal to each other, ε_{j-1}^- and ε_{j-1}^+ can be written as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \varepsilon_{j-2}^- (1 - (\Gamma_{j-2}^-)^2) \\
\varepsilon_{j-1}^+ &= \varepsilon_{j-2}^+ (1 - (\Gamma_{j-2}^-)^2),
\end{aligned} \tag{174}$$

which makes ε_{j-1}^+ and ε_{j-1}^- equal also. This assumption hold if and only if ε_0^+ and ε_0^- are equal to each other. Since both forward and backward prediction error signals are set to input signal in the initialization, both ε_0^+ and ε_0^- are equal to the energy of the signal, which makes the initial assumption correct. (174) proves that the denominator part of (168) can be calculated recursively as in the case of the Levinson-Durbin recursion. In fact, since ε_0 is also set to the energy of the input signal in the Levinson-Durbin recursion and it is assumed that first $j - 1$ partial correlation coefficients calculated by backward covariance method are equal to the first $j - 1$ reflection coefficients calculated by the Levinson-Durbin recursion, ε_{j-1}^+ is equal to ε_{j-1} found by the Levinson-Durbin recursion. As a result, the denominator parts of (168) and (142) are the same, which concludes the proof.

C.4 Equivalence of the CLP Method to the Burg's Method

In the proof of the equivalence of the Burg's method with the CLP method, it is also assumed that the first $j - 1$ partial correlation coefficients have already been calculated and they are equal to the reflection coefficients obtained by the Levinson-Durbin recursion using the correlation coefficients obtained by the CLP method. It is also assumed that the analyzed signal is an infinitely periodic signal, therefore, ε_j^+ and ε_j^- are defined as (154) and (155), respectively.

As discussed in Section 2.2.2, the Burg's method minimizes ε_j^B , which is the summation of ε_j^- and ε_j^+ . Γ_j^B can be calculated by setting the partial derivative of ε_j^B with respect to

Γ_j^B to zero and solving the resulting equation. In this case, Γ_j^B is obtained as

$$\Gamma_j^- = -\frac{2 \sum_{n=0}^{\tau-1} e_{j-1}^+[n] e_{j-1}^-[n-1]}{\sum_{n=0}^{\tau-1} |e_{j-1}^+[n]|^2 + \sum_{n=0}^{\tau-1} |e_{j-1}^-[n-1]|^2}. \quad (175)$$

The equivalence of the Burg's method and CLP analysis can be proved by showing the equivalence of Γ_j^B obtained by (175) and Γ_j obtained in the Levinson-Durbin recursion. Since the numerator part of (175) is twice the numerator part of (153) and the equivalence of Γ_j^+ and Γ_j has already been proved, it is sufficient to show only the denominator parts of (175) is twice that of (142).

The denominator part of (175) is equal to the summation of ε_{j-1}^+ and ε_{j-1}^- . Before extending the denominator, it is beneficial to show how ε_{j-1}^+ and ε_{j-1}^- can be calculated recursively from the ε_{j-2}^+ , ε_{j-2}^- and Γ_{j-1}^B . The squared sum of the forward prediction error in the $(j-1)^{th}$ step can be extended as

$$\begin{aligned} \varepsilon_{j-1}^+ &= \sum_{n=0}^{\tau-1} (e_{j-1}^+[n])^2 \\ &= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n] + \Gamma_{j-1}^B e_{j-2}^-[n-1])^2 \\ &= \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 + 2\Gamma_{j-1}^B \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1] + (\Gamma_{j-1}^B)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 \\ &= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^B \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1]. \end{aligned} \quad (176)$$

Note that, (175) for the previous iteration can be written as

$$-\Gamma_{j-1}^B (\varepsilon_{j-2}^+ + \varepsilon_{j-2}^-) = 2 \sum_{n=0}^{\tau-1} e_{j-2}^+[n] e_{j-2}^-[n-1]. \quad (177)$$

Using (177), (176) can be further extended as

$$\begin{aligned} \varepsilon_{j-1}^+ &= \varepsilon_{j-2}^+ + (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^- + 2\Gamma_{j-1}^B \left(-\frac{\Gamma_{j-1}^B}{2} (\varepsilon_{j-2}^+ + \varepsilon_{j-2}^-)\right) \\ &= \varepsilon_{j-2}^+ - (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^+ \\ &= \varepsilon_{j-2}^+ (1 - (\Gamma_{j-1}^B)^2) \end{aligned} \quad (178)$$

Similarly ε_{j-1}^- can be extended as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \sum_{n=0}^{\tau-1} (e_{j-1}^-[n-1])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1] + \Gamma_{j-1}^B e_{j-2}^+[n])^2 \\
&= \sum_{n=0}^{\tau-1} (e_{j-2}^-[n-1])^2 + 2\Gamma_{j-1}^B \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n] + (\Gamma_{j-1}^B)^2 \sum_{n=0}^{\tau-1} (e_{j-2}^+[n])^2 \\
&= \varepsilon_{j-2}^- + (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^B \sum_{n=0}^{\tau-1} e_{j-2}^-[n-1] e_{j-2}^+[n].
\end{aligned} \tag{179}$$

Using (177), (179) can be further extended as

$$\begin{aligned}
\varepsilon_{j-1}^- &= \varepsilon_{j-2}^- + (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^+ + 2\Gamma_{j-1}^B \left(-\frac{\Gamma_{j-1}^B}{2} (\varepsilon_{j-2}^+ + \varepsilon_{j-2}^-)\right) \\
&= \varepsilon_{j-2}^- - (\Gamma_{j-1}^B)^2 \varepsilon_{j-2}^- \\
&= \varepsilon_{j-2}^- (1 - (\Gamma_{j-1}^B)^2).
\end{aligned} \tag{180}$$

Since ε_{j-1}^B is equal to summation of ε_{j-1}^+ and ε_{j-1}^- , it can be calculated as

$$\begin{aligned}
\varepsilon_{j-1}^B &= \varepsilon_{j-1}^+ + \varepsilon_{j-1}^- \\
&= (\varepsilon_{j-2}^+ + \varepsilon_{j-2}^-) (1 - (\Gamma_{j-1}^B)^2) \\
&= \varepsilon_{j-2}^B (1 - (\Gamma_{j-1}^B)^2)
\end{aligned} \tag{181}$$

(181) proves that the denominator part can also be obtained recursively as in the case of the Levinson-Durbin recursion. Furthermore, since it is assumed that the first $j-1$ partial correlation coefficients calculated by the Burg's method is the same as the ones calculated by the Levinson-Durbin recursion, ε_{j-1}^B is a scaled version of ε_{j-1} obtained in Levinson-Durbin recursion. Since this scaling factor is constant in all iterations, it can be found from ε_0^B and ε_0 . As discussed before, ε_0 is set to the energy of the input signal. Furthermore, since both forward and backward prediction error signals are set to the input signal, both ε_0^+ and ε_0^- are equal to the energy of the input signal. As a result, ε_0^B is equal to the twice the energy of the input signal. Since this scale factor does not change, ε_j^B can be written as

$$\varepsilon_j^B = 2\varepsilon_j \quad j = 1, \dots, p, \tag{182}$$

which makes that the denominator part of (175) is twice the denominator part of (142) in all iterations. This also proves the equivalence of the Burg's method to the CLP method when the input signal is infinitely periodic.

C.5 Equivalence of the CLP Method to the Discrete-Spectra Linear Prediction Modeling

In this section, the equivalence of the discrete-spectra linear-prediction modeling to CLP analysis is shown. As in the previous proofs, it is assumed that the analyzed signal is an infinitely periodic signal with period, τ , which makes the fundamental frequency of the same signal, ω_0 . In this case, (39) reduces to

$$\varepsilon_{LP} = \frac{G^2}{N} \sum_{n=0}^{N-1} \frac{P_x(e^{jn\omega_0})}{\hat{P}_x(e^{jn\omega_0})}, \quad (183)$$

and the correlation coefficient can be obtained as

$$R(i) = \frac{1}{N} \sum_{m=0}^{N-1} P(e^{m\omega_0}) \cos(mi\omega_0). \quad (184)$$

$R(i)$ can also be found as follows

$$\begin{aligned} R(i) &= \frac{1}{\tau} \sum_{m=0}^{\tau-1} X(e^{jm\omega_0}) X^*(e^{jm\omega_0}) \cos(im\omega_0) \\ &= \frac{1}{\tau} \sum_{m=0}^{\tau-1} \left(\sum_{n=0}^{\tau-1} x[n] e^{-jn\omega_0} \right) \left(\sum_{l=0}^{\tau-1} x[l] e^{jlm\omega_0} \right) \frac{(e^{jim\omega_0} + e^{-jim\omega_0})}{2} \\ &= \frac{1}{2\tau} \sum_{m=0}^{\tau-1} \left(\sum_{n=0}^{\tau-1} \sum_{l=0}^{\tau-1} x[n] x[l] e^{-jn\omega_0} e^{jlm\omega_0} \right) (e^{jim\omega_0} + e^{-jim\omega_0}) \\ &= \frac{1}{2\tau} \sum_{m=0}^{\tau-1} \left(\sum_{n=0}^{\tau-1} \sum_{l=0}^{\tau-1} x[n] x[l] e^{-j(n-i)m\omega_0} e^{jlm\omega_0} + \sum_{n=0}^{\tau-1} \sum_{l=0}^{\tau-1} x[n] x[l] e^{-j(n+i)m\omega_0} e^{jlm\omega_0} \right) \\ &= \frac{1}{2\tau} \sum_{n=0}^{\tau-1} \sum_{l=0}^{\tau-1} x[n] x[l] \left(\sum_{m=0}^{\tau-1} e^{-j(n-i)m\omega_0} e^{jlm\omega_0} + \sum_{m=0}^{\tau-1} e^{-j(n+i)m\omega_0} e^{jlm\omega_0} \right) \end{aligned} \quad (185)$$

Note that

$$\begin{aligned} \sum_{m=0}^{\tau-1} e^{-j(n-i)m\omega_0} e^{jlm\omega_0} &= 1 & n - i = l \\ &= 0 & \text{elsewhere,} \end{aligned} \quad (186)$$

and

$$\begin{aligned} \sum_{m=0}^{\tau-1} e^{-j(n+i)m\omega_0} e^{jlm\omega_0} &= 1 & n+i=l \\ &= 0 & \text{elsewhere.} \end{aligned} \quad (187)$$

Using (186) and (187), (185) can be written as

$$R(i) = \frac{1}{2\tau} \left(\sum_{n=0}^{\tau-1} x[n]x[n-i] + \sum_{n=0}^{\tau-1} x[n]x[n+i] \right) \quad (188)$$

Since the signal is infinitely periodic signal with period τ , (188) can be written as

$$\begin{aligned} R(i) &= \frac{1}{2\tau} \left(\sum_{n=0}^{\tau-1} x[n]x[((n-i))_{\tau}] + \sum_{n=0}^{\tau-1} x[n]x[((n+i))_{\tau}] \right) \\ &= \frac{2}{2\tau} \sum_{n=0}^{\tau-1} x[n]x[((n \pm i))_{\tau}] \\ &= \frac{1}{\tau} r(i), \end{aligned} \quad (189)$$

where $r(i)$ is defined in (71). Since $R(i)$ is the scaled $r(i)$, the prediction coefficients obtained by the discrete-spectra linear-prediction modeling is the same as the ones obtained by the CLP method, which concludes the proof.

APPENDIX D

C++ RESEARCH AND DEVELOPMENT ENVIRONMENT FOR SPEECH CODING APPLICATIONS

In this appendix, a programming framework that can be used to implement analysis and synthesis systems is described. The programming techniques used in this framework allows a researcher to concentrate on innovations and fine-tuning the existing algorithms rather than dealing with programming problems. Furthermore, this framework also allows rapid and efficient implementation of algorithms for real-time speech coding applications [16]. All speech coding algorithms described in this thesis are implemented using this framework.

The traditional way of implementing a speech coding application is to write the algorithm in C programming language because of its portability among various platforms. However, serious difficulties also arise because of this choice:

1. Code readability: This problem becomes more severe when many people with different programming styles involved in the same project.
2. Code integration and extraction: Usually the codes are tightly integrated into each other in C implementation. Furthermore, the programmer has to know the nature of the extracted data and how to deal with it to ensure the proper operation of algorithms.
3. Buffer management: The buffer sizes in C style programming are usually hand computed, and any requirement for changing the buffer sizes in the algorithm requires re-computation of these array sizes.
4. Data collecting and analysis: The data collection requires additional code in the application, which is usually distributed throughout the source code and makes its

readability worse.

Besides these problems, the bitstream generation for coding applications is also another tedious task, which has to be implemented separately in a function and changed each time when the final bitstream of the algorithm changes. This makes experimenting with the bitstream a frustrating task.

The problems stated above are addressed in this framework. To solve the problems with the C programming style, C++ was chosen as the programming language. This selection not only solves most of the problems given above, but also allows the researcher to port the final algorithm to C programming language without much difficulty. The object-oriented nature of the C++ allows the framework to make use of class derivations to force programmers to write the code in a pre-determined way, which improves the readability of the code [75]. Furthermore, the same idea is used to group related functions under same class and unrelated functions to different classes to solve the code integration and extraction problem. The rest of the stated problems are solved by an automation system using an execution algorithm working on objects common in speech coding applications. The rest of this section will describe this system.

A speech coding application can be seen as execution of a flowchart. Each basic block of this flowchart has a task to be executed to generate a feature or a signal. This algorithm execution of this framework is based on a similar logic as shown in Figure 69: Each basic block is implemented as a separate node, containing an object, a task executed to generate this object and a node dependency list. The collection of these objects forms the object layer of the system. These nodes are attached each other with a linked list, also named as the *execution script*, which defines the flowchart of the algorithm. The modules in the execution layer shown in Figure 69 is used to execute either the flowchart or handle the bitstream generation and data collection. The communication between these two layers is handled via the communication system, implemented as a C++ class. The purpose of this class is to fetch and deliver the data stored in the objects requested by the tasks and modules in the execution layer. Therefore, this class has functions to check the identification and a structure to store the data and related information. Lastly, since a task may request

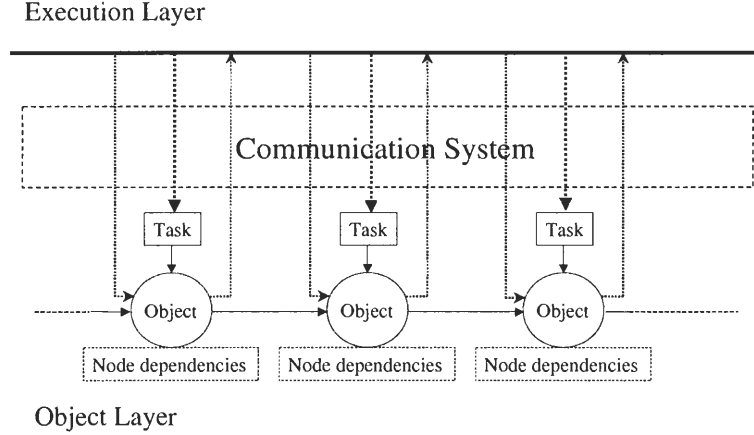


Figure 69: The system overview.

multiple types of data, this class can also deliver unlimited number of different data types by using linked-lists.

D.1 Object Layer

A typical speech coding algorithm has two kinds of objects: *features* and *signals*. Features are defined based on dynamics of the signal such as fundamental period or linear prediction coefficients. On the other hand, filtering operations result in several different types of signals within an application such as the residual signal. Both objects are defined to handle these data types.

The features are designed as a base-class for a hierarchical class system. This class provides data structures and functions common to all features and defines a template consisting of pure virtual functions which all derived classes must implement [75]. To add a new feature to an application, a researcher must write a new class derived from this base class and write a function to fill the informative structure, including type, size, dimension and name of the feature. The feature class system is also capable of handling multiple features under the same class so that related features can be grouped under the same class.

One of the most important design aspects of this framework is the isolation of the extraction and quantization methods realization from the feature itself. This is a logical choice since a feature may have multiple methods for both extraction and quantization. In this framework, both methods are implemented as two abstract classes that define a template

all extraction and quantization classes must implement. An example of this concept is given in Figure 70 for the line spectrum frequencies (LSF). In this example, the LSF class is derived from the base feature class, and the base LSF extractor and LSF quantizer classes are derived from the feature extractor and feature quantization classes, respectively. On the implementation side, the Chebyshev polynomial expansion method and DCT methods are implemented in two separate classes for LSF extraction. Both of these classes are derived from the LSF extraction class. Different quantization methods are also implemented similarly as shown in Figure 70. In an application, two C++ objects, one for extraction and one for quantization, are declared among these methods and then attached to an LSF object upon declaration. To test another method, only an object from that method should be declared and attached to the feature class instead of the previously attached method. This method simplifies testing of different methods and provides modularity to the system.

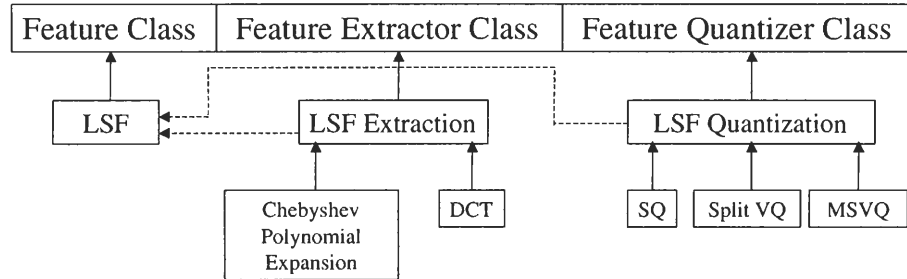


Figure 70: Implementation illustration of LSF and its extraction and quantization methods.

The signals in the system can be generated by two methods. In the first method, a generic signal class is used to apply fixed FIR/IIR filter continuously to each frame of the selected input signal. The second method is used to implement adaptive and/or non-linear filters. In this method, a base class, similar to the feature extractor, is defined to implement the common functions for the signal handling. This base class also defines a template for the filtering function which all derived classes must implement. It is possible to use any previously extracted features or filtered signals within this filtering method. This base class also has functions to make FIR/IIR filtering with and without memory. Both methods also support rate conversion after or before filtering depending on the nature of the rate conversation. Finally, both methods support automatic generation of FIR filter by applying

a Kaiser window to a proper sinc function.

Buffer management in signal classes is handled in conjunction with the execution layer at run-time. Upon initialization of the application, the execution layer computes the required array size for the extraction/quantization of all features and then requests the signal classes to allocate this amount of array size for their buffers. Upon data request from signal classes, the boundary of the required array are computed according to the center of the analysis frame, requested subframe number and requested data length. It is also possible to request a specific portion of the buffer or the current frame. An example for a buffer which can handle three subframe of data is shown in Figure 71.

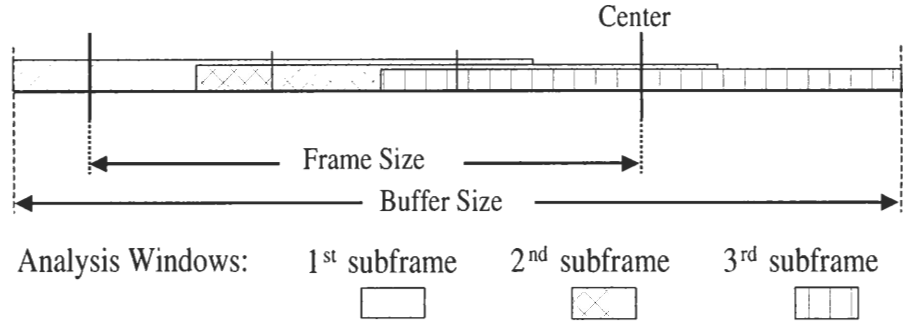


Figure 71: Memory management for three subframe analysis.

D.2 Execution Layer

The execution layer consists of four different modules. The first two modules, analyzer (encoder) and synthesizer (decoder), are the basis of all speech coding applications. The last two modules are used for automatic bitstream processing and data collection for storage and analysis.

The primary purpose of the analyzer module is to extract information from an input signal by executing the analysis script - the flowchart of the analysis algorithm. In the initialization stage, the dependencies between the nodes in the script are resolved and signal buffer sizes are calculated. In the execution stage, the incoming input signal is first processed with a user defined pre-processor. Then, the task in each node is executed sequentially: first the nature of the task is resolved, and the data requested by the task

is fetched from other nodes and delivered to the task. The task is then executed and the analyzer module proceeds to the next node until all nodes are processed. At the end of each frame, the features and signals are generated and some of the features are quantized as well.

The synthesizer module is similar to the analyzer module. As in the analyzer module, a script is processed to obtain the final synthesized signal. To achieve this, the object in the final node in the synthesis script must be a signal. Apart from this difference, the synthesizer module also requires another script that has the required extracted and/or quantized features obtained in analysis module. Finally, a marker class that specifies the parameter interpolation locations in the frame is required in this module. This would allow an algorithm to make pitch-synchronous or subframe based interpolation in the synthesizer.

The bitstream processor module is used in conjunction with the analyzer and synthesizer module. This module allows a researcher to generate a bitstream from quantized features without writing a single line of C/C++ code. To achieve this, a simple bitstream generation language is designed. This language contains commands for moving bits between main bitstream and bitstream of the objects and conditional execution based on the values of objects and bitstream. To generate a bitstream, the researcher must write a small program in a text file and pass the name of the file together with the script containing the quantized features to the bitstream processor module. In initialization, the module converts the text file to bitstream generation op-codes suitable for fast execution. This generated code is similar to a computer's central processing unit's (CPU) assembler program that allows much faster execution speed compared to the execution of an application written in high level language with an interpreter. The bitstream processor has two execution modes: (1) bitstream generation from the quantized and index assigned features, (2) bitstream decoding and generation of the appropriate quantized values from the tables. It is also possible to apply a custom function to the final bitstream after encoding and before decoding stages. Error correction coding and encryption algorithms are perfect candidates for such functions. Finally, the bitstream processor packs the bits in multiples of eight bits (bytes) for storage purpose. To simplify direct decoding across all platforms, the packing can be performed on

both big endian and little endian byte formats for multi-byte packing.

The last module of the execution layer is the data processor module. This module is also used in conjunction with the analyzer and synthesizer modules. The purpose of this module is to collect data from desired feature and signal objects for a specified number of frames, and save this data to a file in a pre-determined format or invoke Matlab and execute a tool to analyze the data using the Matlab Engine API [48, 49]. By writing only a few lines of C++ code, it is possible to analyze all data used in analysis and synthesis modules.

D.3 Programming Libraries and Tools

In addition to the object and execution layers, the framework also supports numerous tools to simplify programming tasks:

- Vector operations library
- Windowing functions library
- Audio file I/O library
- Interpolation methods library
- Macros for decreasing programming workload
- Matlab communication library for graphics command (e.g. `plot()`, `plot3()`)

Besides these libraries, a visualization and analysis tool in Matlab was also developed to analyze the data produced by the data processor module. Furthermore, this tool may be invoked directly within the C++ application without manually launching Matlab. Figure 72 is generated by only eight lines of C++ code. The visualization tool supports following operations:

- Zoom/unzoom on the time axis
- Zoom/unzoom on the data axis
- Center adjustment on the data axis

- Sound play
- Possible to attach a narrowband/wideband spectrogram to any signal
- Synchronized feature-data display
- Extracted and quantized data display in the same window
- Multidimensional data display in a separate window
- FFT/LPC spectrum magnitude display in a separate window with optional LSF display

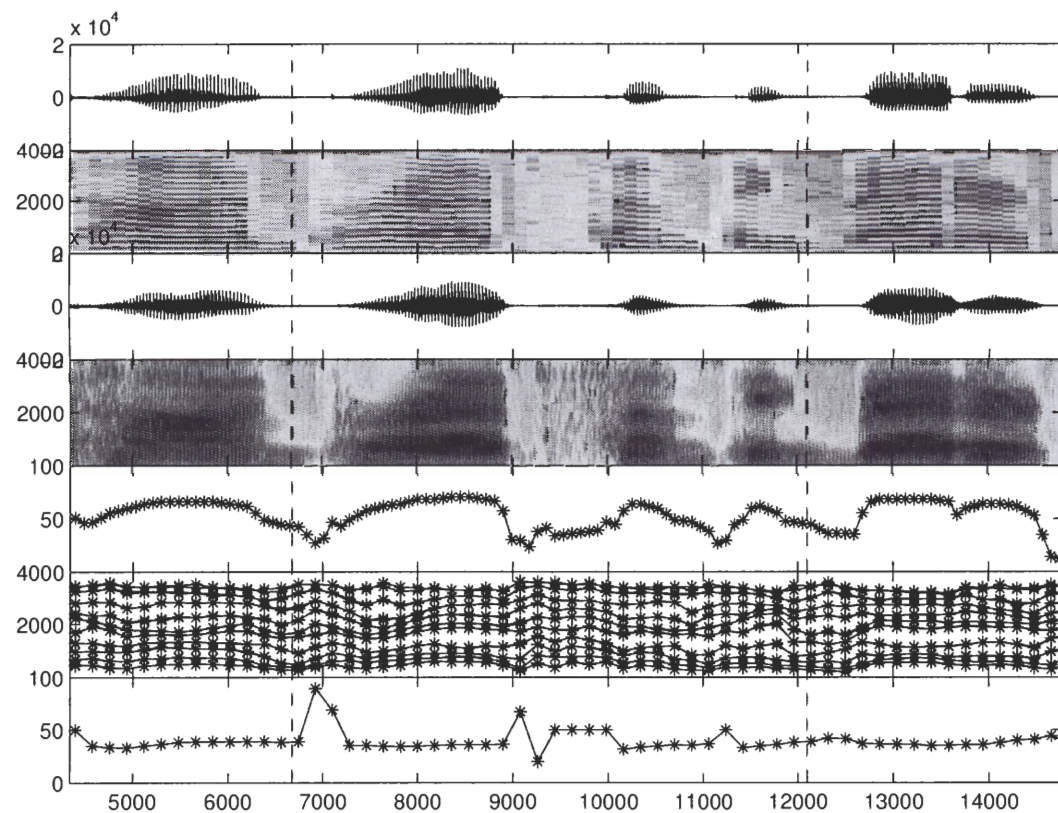


Figure 72: Snapshot of the visualization tool.

REFERENCES

- [1] ATAL, B. S., CUPERMAN, V., and GERSHO, A., *Advances in speech coding*. Boston : Kluwer Academic Publishers, 1991.
- [2] ATAL, B. and REMDE, J., "A new model of LPC excitation for producing natural sounding speech at low bit rates," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 614–617, 1982.
- [3] ATAL, B. and SCHROEDER, J., "Stochastic coding of speech at very low bit rates," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1611–1613, 1984.
- [4] ATAL, B. and SCHROEDER, M., "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 27, pp. 247–254, 1979.
- [5] BARNWELL III, T., "Analysis-synthesis systems for speech coding based on nonrecursive and recursive filters," *Bell Labs. Technical Memorandum*, 1970.
- [6] BARNWELL III, T., "Circular correlation and the LPC," *Proc. IEEE Int. Conf. on Communications*, pp. 31/5–31/10, 1976.
- [7] BARNWELL III, T., "Windowless techniques for LPC analysis," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 28, pp. 421–427, 1980.
- [8] BENVENUTO, N., BERTOCCI, G., and DAUMER, W., "The 32 kb/s ADPCM coding standard," *AT&T Technical Journal*, vol. 65, pp. 12–22, 1986.
- [9] BRANDSTEIN, M., MONTA, P., HARDWICK, J., and LIM, J., "A real-time implementation of the improved MBE speech coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. 1, pp. 5–8, 1990.
- [10] CHEN, J., "LD-CELP: A high quality 16 kb/s speech coder with low delay," *Proc. IEEE Global Telecommunications Conference*, pp. 528–532, 1990.
- [11] COX, R., *Speech Coding and Synthesis*, ch. 2: Speech Coding Standards, pp. 63–65. Elsevier, 1995.
- [12] DAS, A., RAO, A., and GERSHO, A., "Enhanced multi-band excitation coding of speech at 2.4 kb/s with discrete all-pole spectral modeling," *Proc. IEEE Global Telecommunications Conference*, pp. 863–866, 1994.
- [13] DELLER, J., PROAKIS, J., and HANSEN, J., *Discrete Time Processing of Speech Signals*, ch. 7: Speech Coding and Synthesis. Prentice Hall, 1993.
- [14] EL-JAROUDI, A. and MAKHOUL, J., "Discrete all-pole modeling," *IEEE Trans. Signal Processing*, vol. 39, pp. 411–423, 1991.

- [15] ERKELENS, J., *Autoregressive Modeling of Speech Coding: Estimation, Interpolation and Quantization*. Delft University Press, 1996.
- [16] ERTAN, A. and BARNWELL III, T., "A C++ research and development environment for speech and audio processing applications," *Proc. of 34th Asilomar Conference on Signals, Systems and Computers*, pp. 1449–1453, 2000.
- [17] ERTAN, A. and BARNWELL III, T., "Circular LPC modeling and constant pitch transformation for scalable bit-rate, scalable quality speech coding," *Proceedings of the Speech Coding Workshop*, pp. 50–52, 2002.
- [18] FANT, G., *Acoustic Theory of Speech Production*. Mouton and Co.'s Gravenhage, 1960.
- [19] GERSHO, A., "Advances in speech coding and audio compression," *Proceedings of IEEE*, vol. 82, 1994.
- [20] GRIFFIN, D. and LIM, J., "Multiband excitation vocoder," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 36, pp. 1223–1235, 1988.
- [21] HARDWICK, J. and LIM, J., "A 4.8 kbs multi-band excitation speech coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 374–377, 1988.
- [22] HASIB, A. and HACIOGLU, K., "Source combined linear predictive analysis in pulse-based speech coders," *IEE Proc. Vision, Image and Signal Processing*, vol. 143, pp. 143–148, 1996.
- [23] HAYES, M. H., *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., 1996.
- [24] HELLWIG, K., VARY, P., MASSALOUX, D., and PETIT, J., "Speech codec for european mobile radio system," *Proc. IEEE Global Telecommunications Conference*, pp. 1065–1069, 1989.
- [25] HERMANSKY, H., FUJISAKI, H., and SATO, Y., "Analysis and synthesis of speech based on spectral transform linear predictive method," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 777–780, 1983.
- [26] HERMANSKY, H., FUJISAKI, H., and SATO, Y., "Spectral envelope sampling and interpolation in linear predictive analysis of speech," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 2.2.1–2.2.4, 1984.
- [27] HESS, W., *Pitch Determination of Speech Signals*. Springer, 1983.
- [28] ITAKURA, F. and SAITO, S., "Analysis synthesis telephony based on maximum likelihood method," *Rep. 6th Int. Congr. Acoustics*, pp. C17–C20, 1968.
- [29] ITU, "Recommendations p.80 methods of subjective determination of transmission quality," 1993.
- [30] JAYANT, N., "Digital coding of speech waveforms: PCM, DPCM and DM quantizers," *Proceedings of IEEE*, vol. 62, pp. 611–632, 1974.
- [31] KABAL, P. and KLEIJN, B., "All-pole modeling of mixed excitation signals," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 97–100, 2001.

- [32] KABAL, P. and RAMACHANDRAN, R., "Joint optimization of linear predictors in speech coders," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, pp. 642–650, 1989.
- [33] KANG, G. and EVERETT, S., "Improvement of the excitation source in the narrowband linear prediction vocoder," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 33, pp. 377–386, 1985.
- [34] KANG, H. and SEN, D., "Phase adjustment in waveform interpolation," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 261–264, 1999.
- [35] KATUGAMPALA, N. and KONDOZ, A., "A hybrid coder based on a new phase model for synchronization between harmonic and waveform coded segments," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 685–688, 2001.
- [36] KLEIJN, B., *Analysis-by-Synthesis Speech Coding Based on Relaxed Waveform Matching Constraints*. PhD thesis, Delft University of Technology, 1991.
- [37] KLEIJN, W., "Encoding speech using prototype waveforms," *IEEE Trans. Speech, Audio and Signal Processing*, vol. 1, pp. 386–399, 1993.
- [38] KLEIJN, W. and HAAGEN, J., "A speech coder based on decomposition of characteristic waveform," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 508–511, 1995.
- [39] KLEIJN, W., KRASINKI, D., and KETCHUM, R., "Improved speech quality and efficient vector quantization in SELP," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 155–158, 1988.
- [40] KLEIJN, W., SHOHAM, Y., SEN, D., and HAGEN, R., "A low-complexity waveform interpolation coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 212–215, 1996.
- [41] KOHLER, M., "A comparison of the new 2400 bps MELP federal standard with other standard coders," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1587–1590, 1997.
- [42] KROON, P., DEPRETTERE, E., and SLUYTER, R., "Regular-pulse excitation - a novel approach to effective and efficient multipulse coding of speech," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 34, pp. 1054–1063, 1986.
- [43] LEE, C., "On robust linear prediction of speech," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 36, pp. 642–650, 1988.
- [44] MAKHOUL, J., "Linear prediction: A tutorial review," *Proceedings of IEEE*, vol. 63, pp. 561–580, 1975.
- [45] MAKHOUL, J., "Spectral linear prediction: Properties and applications," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 23, pp. 283–296, 1975.
- [46] MAKHOUL, J., VISWANATHAN, R., SCHWARTZ, R., and HUGGINS, A., "A mixed-source model for speech compression and synthesis," *Journal of Acoustical Society of America*, vol. 64, pp. 1577–1581, 1978.

- [47] MARKEL, J. and GRAY, A., *Linear Prediction of Speech*. Springer-Verlag Berlin Heidelberg, 1976.
- [48] MATHWORKS, "Matlab application programming interface reference," 1999.
- [49] MATHWORKS, "Matlab application programming user's guide," 1999.
- [50] MCAULAY, R. and QUATIERI, T., "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 34, pp. 744–754, 1986.
- [51] MCAULAY, R. and QUATIERI, T., "Multirate sinusoidal transform coding at rates from 2.4 kbps to 8 kbps," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1645–1648, 1987.
- [52] MCAULAY, R. and QUATIERI, T., "Sine-wave phase coding at low data rates," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 577–580, 1991.
- [53] MCAULAY, R. and QUATIERI, T., "The application of subband coding to improve the quality and robustness of the sinusoidal transform coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 439–442, 1993.
- [54] MCCREE, A., UNNO, T., ANANDKUMAR, A., BERNARD, A., and PAKSOY, E., "An embedded adaptive multi-rate wideband speech coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 761–764, 2001.
- [55] MCCREE, A. V., *A New LPC Vocoder Model for Low Bit Rate Speech Coder*. PhD thesis, Georgia Institute of Technology, 1992.
- [56] MCCREE, A. and BARNWELL III, T., "A mixed excitation lpc vocoder model for low bit-rate speech coding," *IEEE Trans. Speech, Audio and Signal Processing*, vol. 3, pp. 242–250, 1995.
- [57] MCCREE, A. and MARTIN, J. D., "A 1.7 kb/s MELP coder with improved analysis and quantization," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 593–596, 1998.
- [58] MIYOSHI, Y., YAMATO, K., MIZOGUCHI, R., YANAGIDA, M., and KAKUSHO, O., "Analysis of speech signals of short pitch period by a sample selective linear prediction," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 35, pp. 1233–1240, 1987.
- [59] MIZOGUCHI, R., YANAGIDA, M., and KAKUSHO, O., "Speech analysis by selective linear prediction in the time domain," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1573–1576, 1982.
- [60] NOLL, P., "MPEG digital audio coding," *IEEE Signal Processing Magazine*, vol. 14, pp. 59–81, 1997.
- [61] RABINER, L., ATAL, B., and SAMBUR, M., "LPC prediction error - analysis of its variation with the position of the analysis frame," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 25, pp. 434–442, 1977.

- [62] RAMACHANDRAN, R. and KABAL, P., "Stability and performance analysis of pitch filters in speech coders," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 35, pp. 937–946, 1987.
- [63] ROSE, R. and BARNWELL III, T., "The self-excited vocoder - an alternative approach to toll quality at 4.8 kbit/s," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. 1, pp. 453–456, 1986.
- [64] SALAMI, R., LAFLAMME, C., ADOUL, J., KATAOKA, A., HAYASHI, S., MORIYA, T., LAMBLIN, C., MASSALOUX, D., PROUST, S., KROON, P., and SHOHAM, Y., "Design and description of CS-ACELP: a toll quality 8 kb/s speech coder," *IEEE Trans. Speech, Audio and Signal Processing*, vol. 6, pp. 116–130, 1998.
- [65] SAMBUR, M., ROSENBERG, A., RABINER, L., and MCGONEGAL, C., "On reducing the buzz in lpc synthesizer," *Journal of Acoustical Society of America*, vol. 63, pp. 918–924, 1978.
- [66] SCHAFER, R. and MARKEL, J., *Speech Analysis*. John Wiley & Sons, 1979.
- [67] SERIZAWA, M. and GERSHO, A., "Joint optimization of LPC and closed-loop pitch parameters in CELP coders," *IEEE Signal Process. Letters*, vol. 6, pp. 52–54, 1999.
- [68] SHESKIN, D., *Handbook of Parametrical and Non-Parametrical Statistical Procedures*. CRC Press LLC., 1997.
- [69] SHLOMOT, E., CUPERMAN, V., and GERSHO, A., "Hybrid coding: combined harmonic and waveform coding of speech at 4 kb/s," *IEEE Trans. Speech and Audio Processing*, vol. 9, pp. 632–646, 2001.
- [70] SHUKLA, S., ERTAN, A., and BARNWELL III, T., "Circular LPC modeling and constant pitch transform for accurate speech analysis and high quality speech synthesis," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 269–272, 2002.
- [71] SINGHAL, S. and ATAL, B., "Optimizing LPC filter parameters for multi-pulse excitation," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 781–784, 1983.
- [72] SINGHAL, S. and ATAL, B., "Amplitude optimization and pitch prediction in multi-pulse coders," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, pp. 317–327, 1989.
- [73] STACHURSKI, J. and MCCREE, A., "A 4 kb/s hybrid MELP/CELP coder with alignment phase encoding and zero-phase equalization," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1379–1382, 2000.
- [74] STACHURSKI, J., MCCREE, A., and VISWANATHAN, V., "High quality MELP coding at bit rates around 4 kb/s," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 485–488, 1999.
- [75] STROUSTRUP, B., *The C++ Programming Language*. Addison-Wesley Publishing Company, 1991.
- [76] TREMAIN, T., "The government standard linear predictive coding algorithm:LPC-10," *Speech Technology*, pp. 40–49, 1980.

- [77] UNNO, T., BARNWELL III, T., and CLEMENTS, M., "The multimode multipulse excitation vocoder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1683–1686, 1997.
- [78] UNNO, T., BARNWELL III, T., and TRUONG, K., "An improved mixed excitation linear prediction (MELP) coder," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 245–248, 1999.
- [79] U.S. DEPARTMENT OF DEFENSE, "Specifications for the analog to digital conversion of voice by 2,400 bit/second mixed excitation linear prediction," 1998.
- [80] WANG, T., KOISHIDA, K., CUPERMAN, V., GERSHO, A., and COLLURA, J., "A 1200/2400 bps coding suite based on MELP," *Proceedings of the Speech Coding Workshop*, pp. 90–92, 2002.
- [81] WANG, T., KOISHIDA, S., CUPERMAN, V., GERSHO, A., and COLLURA, J., "A 1200 BPS speech coder based on MELP," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1375–1378, 2000.
- [82] WARREN, R. M., *Auditory perception: a new synthesis*. Cambridge University Press, 1999.
- [83] YANG, H., KLEIJN, W., DEPRETTERE, E., and CHEN, H., "Pitch synchronous modulated lapped transform of the linear prediction residual of speech," *Proc. of 4th Int. Conf. on Signal Process.*, pp. 591–594, 1998.
- [84] YELDENER, S., KONDOZ, A., and EVANS, B., "Multiband linear predictive coding at very low bit rates," *IEE Proc. Vision, Image and Signal Processing*, pp. 289–296, 1994.
- [85] ZELINSKI, R. and NOLL, P., "Adaptive transform coding of speech signals," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 25, pp. 299–309, 1977.

VITA

Ali Erdem Ertan was born in Ankara, Turkey on July 17, 1974. In 1992, he graduated from Ankara Atatürk Anadolu High School, Ankara, Turkey. He received his bachelor of science degree in electrical and electronics engineering from Ortadoğu Teknik Üniversitesi, Ankara, Turkey, in 1996. He graduated from Bilkent Üniversitesi, Ankara, Turkey, with a master of science degree also in electrical and electronics engineering in 1998.

From 1994 through 1996, he worked part-time in the multimedia group of Tübitak-Bilten, Ankara, Turkey, as a software engineer. He developed a real-time software-only video decoder and a hardware abstraction layer for a PC computer game at this time. Between 1996 and 1998, he joined the speech processing group of the same institution and participated in the development of a speech analysis software and a fixed-point real-time implementation of the 2.4 kb/s U.S. military standard MELP coder on a DSP hardware.

In 1999, he enrolled in Georgia Institute of Technology as a Ph.D. student. Dr. Thomas P. Barnwell III offered him a graduate research assistantship for speech coding research in the fall of 1999. His Ph.D. studies were funded by Texas Instruments. During his Ph.D. study, he also made an internship in Texas Instruments, Dallas, Texas, in the summer of 2001. He expects to receive his Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, Georgia, in the spring of 2004.