

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A197 093

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
REPORT NUMBER AFIT/CI/NR 88-178	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
TITLE (and Subtitle) ROBUST RECOGNITION OF LOUD AND LOMBARD SPEECH IN THE FIGHTER COCKPIT ENVIRONMENT		5. TYPE OF REPORT & PERIOD COVERED PHD MS THESIS
AUTHOR(s) BILL J. STANTON Jr		6. PERFORMING ORG. REPORT NUMBER
PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: PURDUE UNIVERSITY		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) AFIT/NR Wright-Patterson AFB OH 45433-6583		12. REPORT DATE 1988
		13. NUMBER OF PAGES 413
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) DISTRIBUTED UNLIMITED: APPROVED FOR PUBLIC RELEASE		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) SAME AS REPORT		
18. SUPPLEMENTARY NOTES Approved for Public Release: IAW AFR 190-1 LYNN E. WOLAVER Dean for Research and Professional Development Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) ATTACHED		

DTIC

AUG 18 1988

H

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

88 8 16 023

**ROBUST RECOGNITION OF LOUD AND LOMBARD SPEECH
IN THE FIGHTER COCKPIT ENVIRONMENT**

A Thesis

Submitted to the Faculty

of

Purdue University

by

Bill J. Stanton Jr., Major, USAF

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

August 1988

To God be the glory, great things he hath done;
So loved he the world that he gave us his Son,
Who yielded his life an atonement for sin,
And opened the lifegate that all may go in.

Fanny J. Crosby, 1875

In loving memory of my mother

ACKNOWLEDGMENTS

I would like to thank Professor Leah Jamieson for her technical expertise and professional guidance. She provided a quality of academic excellence that was truly respected and appreciated. I would also like to thank Professor George Allen for his insights into the science of speech as well as his editorial comments on the draft manuscript. Thanks are also due to Professor Clare McGillem for his flexibility and support, to Professor Avinash Kak for his enthusiasm and love of flying, and to Professor Edward Delp for his generosity with equipment and facilities. This research would not have been possible without the suggestions and data provided by the Biological Acoustics Branch of the Armstrong Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base, Ohio. Special thanks go to Mark Ericson, Thomas Moore, Timothy Anderson, and Richard McKinley for their assistance in defining the problem and acquiring the data. Also essential to this research was the tremendous support provided by Purdue's Engineering Computer Network staff. The generous assistance of Curt Freeland, George Goble, and Kent De La Croix is sincerely appreciated. I also owe much to my colleagues at the U. S. Air Force Academy, specifically to Robert Phelps and Harold Bare for their encouragement, and to Erlind Royer whose singular efforts made my attendance at Purdue possible. In addition, I want to thank Legand Burge and James Mosko for cultivating my interests in the field of speech recognition. As an officer in the U. S. Air Force, I am indebted to the American taxpayers for the financial support provided through the Air Force Institute of Technology. I also gratefully acknowledge my parents, who provided discipline, direction, and love that have served me well as an adult. Finally, I would like to express my sincere thanks to my wife, Donna, and to my sons, Spencer and Stuart. Their love, understanding, and encouragement were blessings beyond measure.



DTIC TAB	☒
Unannounced	☐
Justification	☐
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	xii
ABSTRACT	xviii
1. INTRODUCTION	1
2. SPEECH PRODUCTION.....	5
2.1 Physiology	5
2.2 Acoustic Phonetics	6
2.2.1 Excitation	6
2.2.2 Sounds of Speech	8
2.3 Modeling the Acoustic System	9
2.4 Summary	12
3. SPEECH RECOGNITION.....	13
3.1 Factors Determining Complexity	14
3.1.1 Recognition Units	14
3.1.2 Vocabulary	16
3.1.3 Syntactic Constraints	16
3.1.4 Speaker Variability	17
3.1.5 Speaker Dependence.....	17
3.1.6 Target Tasks	18
3.1.7 Environment	18
3.2 Front-End Design Approaches	19
3.3 Knowledge Sources	20
3.4 Summary	21

	Page
4. THE FIGHTER COCKPIT ENVIRONMENT	23
4.1 Why the Fighter Cockpit.....	23
4.2 Typical Mission Profiles.....	25
4.3 Pilot Tasks	27
4.4 Summary	30
5. THE CHALLENGES OF RECOGNIZING COCKPIT SPEECH.....	31
5.1 Personal Equipment	32
5.2 Pilot Workload and Stress.....	33
5.3 Pilot Acceptance.....	35
5.4 Related Research	36
5.5 Summary	37
6. RESEARCH OBJECTIVE AND BACKGROUND	38
6.1 Background	39
6.2 Description of Database.....	40
6.3 Summary	46
7. ANALYSIS OF ABNORMAL SPEECH.....	47
7.1 Previous Research.....	47
7.2 Analysis Procedures.....	49
7.3 Results.....	53
7.3.1 Spectral Energy Distributions	53
7.3.2 Spectral Energy Attributes.....	57
7.3.3 Pitch	59
7.3.4 Formants.....	59
7.3.5 Duration.....	60
7.4 Summary	61
8. BASELINE RECOGNITION SYSTEM.....	69
8.1 System Description	69
8.2 Recognition Assessment	70
8.3 Baseline Metric	72
8.4 Baseline Performance	73
8.5 Figure of Merit	75
8.6 Summary	77

	Page
9. EXPERIMENTS AND RESULTS	79
9.1 Cepstral Measure of Euclidean Distance	79
9.2 Likelihood Ratio	82
9.3 Spectral Slope.....	89
9.3.1 Computation from Templates	89
9.3.2 Root Power Sums.....	93
9.4 Slope-Dependent Weighting.....	96
9.4.1 Non-Linear Weighting Function.....	96
9.4.2 Performance.....	98
9.5 Smallest Cumulative Distance	99
9.5.1 Performance with Other Metrics	101
9.5.2 Performance with Slope-Dependent Weighting.....	102
9.6 Performance by Phoneme Categories.....	109
9.7 Performance in Terms of Error Rate	117
9.8 Summary	119
10. CONCLUSIONS.....	121
LIST OF REFERENCES.....	125
APPENDICES	
Appendix A: Applications of Voice Interaction in the AFTI F-16	131
Appendix B: Lisp Code for Symbolics 3670	132
Appendix C: AAMRL Database Vocabulary.....	153
Appendix D: Complete AAMRL List of Enrollment Utterances.....	154
Appendix E: List of Speech Sessions.....	164
Appendix F: Vocabulary Transcriptions.....	165
Appendix G: List of Phonemes	169
Appendix H: Vocabulary Word - Phoneme Cross Reference.....	170
Appendix I: Algorithm for Selection of Utterance Set	174
Appendix J: Fortran Code.....	177
Appendix K: Utterance Subset for Research.....	326
Appendix L: Analyses of Normal, Loud, and Lombard Speech	327
Appendix M: Performance Curves for the Baseline System	379
Appendix N: Figure of Merit Comparisons.....	393
Appendix O: Performance Curves for SDW-SCD	399
VITA.....	413

LIST OF TABLES

Table	Page
1. Phoneme occurrences in utterances A001 - A020 for speaker #2, loud speech	44
2. Total phonemes processed for eight speakers.....	45
3. Features for analysis.....	50
4. Set of 40 phonemes grouped by category	50
5. Changes in energy (dB) from normal to loud speech across all eight speakers.....	58
6. Changes in energy (dB) from normal to Lombard speech across all eight speakers.....	59
7. Changes in center of gravity and spectral tilt from normal to loud and Lombard speech across all eight speakers.....	60
8. Changes in pitch from normal to loud and Lombard speech across all eight speakers.....	61
9. Changes in formant frequencies from normal to loud and Lombard speech across all eight speakers (Hz)	65
10. Changes in duration from normal to loud and Lombard speech across all eight speakers.....	66
11. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #1. Level of significance: 0.01	67
12. Shifts in phoneme features that are common to speakers #1 and #3, level of significance: 0.01.....	68
13. Shifts in phoneme features that are common to speakers #1, #3, #6, #7, and #8, level of significance: 0.01	68

Table	Page
14. Figures of merit of the baseline recognizer for all eight speakers	78
15. Performance of cepstral distance compared to baseline system	81
16. Key for summary section of all figure of merit comparison tables	84
17. Performance of the likelihood ratio compared to baseline system	86
18. Performance of spectral slope estimate compared to baseline system	91
19. Performance of root power sums compared to baseline system	94
20. Performance of slope-dependent weighting compared to baseline system, $s_T = 0.5$	100
21. Comparison of smallest cumulative distance (SCD), versus raw nearest neighbor (RNN), for various metrics	102
22. Performance of slope-dependent weighting with smallest cumulative distance compared to baseline system, $s_T = 1.0$	104
23. Phoneme categories for performance comparisons	109
24. Figures of merit for all recognition methods, all speakers, broken down by phoneme category	110
25. Error rates for the case, $M = 5$, for all recognition methods, all speakers, broken down by phoneme category	118
26. Rank order of recognition methods from best to worst, according to the overall figure of merit	123
Appendix	
Table	
27. Identification data for speech sessions	164
28. Average differences in phoneme features between Loud and normal speech, all speakers	328

Appendix Table	Page
29. Average differences in phoneme features between Lombard and normal speech, all speakers.....	329
30. Average differences in phoneme features between Lombard and loud speech, all speakers.....	330
31. Average differences in phoneme features between Loud and normal speech, speaker #1	331
32. Average differences in phoneme features between Lombard and normal speech, speaker #1	332
33. Average differences in phoneme features between Lombard and loud speech, speaker #1	333
34. Average differences in phoneme features between Loud and normal speech, speaker #2	334
35. Average differences in phoneme features between Lombard and normal speech, speaker #2	335
36. Average differences in phoneme features between Lombard and loud speech, speaker #2	336
37. Average differences in phoneme features between Loud and normal speech, speaker #3	337
38. Average differences in phoneme features between Lombard and normal speech, speaker #3	338
39. Average differences in phoneme features between Lombard and loud speech, speaker #3	339
40. Average differences in phoneme features between Loud and normal speech, speaker #4	340
41. Average differences in phoneme features between Lombard and normal speech, speaker #4	341
42. Average differences in phoneme features between Lombard and loud speech, speaker #4	342
43. Average differences in phoneme features between Loud and normal speech, speaker #5	343

Appendix
Table

Page

44. Average differences in phoneme features between Lombard and normal speech, speaker #5	344
45. Average differences in phoneme features between Lombard and loud speech, speaker #5	345
46. Average differences in phoneme features between Loud and normal speech, speaker #6	346
47. Average differences in phoneme features between Lombard and normal speech, speaker #6	347
48. Average differences in phoneme features between Lombard and loud speech, speaker #6	348
49. Average differences in phoneme features between Loud and normal speech, speaker #7	349
50. Average differences in phoneme features between Lombard and normal speech, speaker #7	350
51. Average differences in phoneme features between Lombard and loud speech, speaker #7	351
52. Average differences in phoneme features between Loud and normal speech, speaker #8	352
53. Average differences in phoneme features between Lombard and normal speech, speaker #8	353
54. Average differences in phoneme features between Lombard and loud speech, speaker #8	354
55. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #1. Level of significance: 0.01.....	371
56. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #2, level of significance: 0.01	372
57. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #3, level of significance: 0.01	373
58. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #4, level of significance: 0.01	374

Appendix
Table

Page

59. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #5, level of significance: 0.01	375
60. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #6, level of significance: 0.01	376
61. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #7, level of significance: 0.01	377
62. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #8, level of significance: 0.01	378
63. Performance of baseline system with smallest cumulative distance compared to baseline system.....	394
64. Performance of the cepstral measure with smallest cumulative distance compared to baseline system.....	395
65. Performance of the likelihood ratio with smallest cumulative distance compared to baseline system.....	396
66. Performance of the spectral slope estimate with smallest cumulative distance compared to baseline system	397
67. Performance of root power sums with smallest cumulative distance compared to baseline system.....	398

LIST OF FIGURES

Figure	Page
1. Generalized model of speech production	10
2. Division of time required to preprocess analog speech data	43
3. Differences in energy from normal to loud for the vowels of all eight speakers	54
4. Differences in energy from normal to Lombard for the vowels of all eight speakers	55
5. Differences in energy from normal to Lombard for the vowels of speaker 2	56
6. Differences in energy from normal to Lombard for the vowels of speaker 5	57
7. Differences in energy from normal to Lombard for the vowels of speaker 6	58
8. Average shifts of the first and second formants for selected vowels of speaker #3	62
9. Average shifts of the first and second formants for selected vowels of speaker #6	63
10. Average shifts of the first and second formants for selected vowels of speaker #7	64
11. Baseline recognizer diagram	70
12. Phoneme lattice for utterance caution test	71
13. Baseline recognizer performance for utterance caution test	72
14. Baseline recognition system performance on all eight speakers	74

Figure	Page
15. Baseline recognition system performance all eight speakers averaged, and broken down by phoneme category	75
16. Recognition performance for baseline system, speaker #2	76
17. Recognition performance for baseline system, speaker #7	83
18. Recognition performance for cepstral distance, speaker #7	83
19. Recognition performance for baseline system, speaker #6	87
20. Recognition performance for likelihood ratio, speaker #6	87
21. Comparison of LPC spectra for normal, loud, and Lombard for phoneme AA of speaker #2	88
22. Recognition performance for baseline system, speaker #2	92
23. Recognition performance for spectral slope estimate, speaker #2	92
24. Recognition performance for baseline system, speaker #5	95
25. Recognition performance for root power sums, speaker #5	95
26. Non-linear weighting function.....	9
27. Distribution of values for magnitude difference in spectral slope, session 53.....	98
28. Overall performance of slope-dependent weighting for different threshold values.....	98
29. Recognition performance for baseline system, speaker #3	101
30. Recognition performance for slope-dependent weighting $s_T = 0.5$, speaker #3.....	101
31. Comparison of SCD and RNN with the method of slope-dependent weighting.....	103
32. Recognition performance for baseline system, speaker #3	105
33. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #3	105
34. Recognition performance for baseline system, speaker #1	106

Figure	Page
35. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #1	106
36. Recognition performance for baseline system, speaker #2	107
37. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #2	107
38. Recognition performance for baseline system, speaker #7	108
39. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #7	108
40. Recognition performance for baseline system, all phonemes, all speakers	111
41. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, all phonemes, all speakers	111
42. Recognition performance for baseline system, stops, all speakers	112
43. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, stops, all speakers.....	112
44. Recognition performance for baseline system, nasals, all speakers	113
45. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, nasals, all speakers	113
46. Recognition performance for baseline system, fricatives, all speakers	114
47. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, fricatives, all speakers.....	114
48. Recognition performance for baseline system, liquids, all speakers	115
49. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, liquids, all speakers.....	115
50. Recognition performance for baseline system, vowels, all speakers	116

Figure	Page
51. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, vowels, all speakers.....	116
Appendix	
Figure	
52. Utterance selection algorithm.....	176
53. Average shifts of the first and second formants for selected vowels of Speaker #1	355
54. Average shifts of the first and second formants for selected vowels of Speaker #2	356
55. Average shifts of the first and second formants for selected vowels of Speaker #3	357
56. Average shifts of the first and second formants for selected vowels of Speaker #4	358
57. Average shifts of the first and second formants for selected vowels of Speaker #5	359
58. Average shifts of the first and second formants for selected vowels of Speaker #6	360
59. Average shifts of the first and second formants for selected vowels of Speaker #7	361
60. Average shifts of the first and second formants for selected vowels of Speaker #8	362
61. Average shifts of the first and third formants for selected vowels of Speaker #1	363
62. Average shifts of the first and third formants for selected vowels of Speaker #2	364
63. Average shifts of the first and third formants for selected vowels of Speaker #3	365
64. Average shifts of the first and third formants for selected vowels of Speaker #4	366
65. Average shifts of the first and third formants for selected vowels of Speaker #5	367

Appendix Figure	Page
66. Average shifts of the first and third formants for selected vowels of Speaker #6	368
67. Average shifts of the first and third formants for selected vowels of Speaker #7	369
68. Average shifts of the first and third formants for selected vowels of Speaker #8	370
69. Recognition performance for baseline system, Speaker #1	379
70. Recognition performance for baseline system, Speaker #2	380
71. Recognition performance for baseline system, Speaker #3	381
72. Recognition performance for baseline system, Speaker #4	382
73. Recognition performance for baseline system, Speaker #5	383
74. Recognition performance for baseline system, Speaker #6	384
75. Recognition performance for baseline system, Speaker #7	385
76. Recognition performance for baseline system, Speaker #8	386
77. Recognition performance for baseline system, all phonemes, all speakers	387
78. Recognition performance for baseline system, stops, all speakers	388
79. Recognition performance for baseline system, nasals, all speakers	389
80. Recognition performance for baseline system, fricatives, all speakers	390
81. Recognition performance for baseline system, liquids, all speakers	391
82. Recognition performance for baseline system, vowels, all speakers	392
83. Recognition performance for SDW-SCD, Speaker #1	399

Appendix Figure	Page
84. Recognition performance for SDW-SCD, Speaker #2	400
85. Recognition performance for SDW-SCD, Speaker #3	401
86. Recognition performance for SDW-SCD, Speaker #4	402
87. Recognition performance for SDW-SCD, Speaker #5	403
88. Recognition performance for SDW-SCD, Speaker #6	404
89. Recognition performance for SDW-SCD, Speaker #7	405
90. Recognition performance for SDW-SCD, Speaker #8	406
91. Recognition performance for SDW-SCD, all phonemes, all speakers	407
92. Recognition performance for SDW-SCD, stops, all speakers	408
93. Recognition performance for SDW-SCD, nasals, all speakers	409
94. Recognition performance for SDW-SCD, fricatives, all speakers	410
95. Recognition performance for SDW-SCD, liquids, all speakers	411
96. Recognition performance for SDW-SCD, vowels, all speakers	412

ABSTRACT

Stanton, Bill J., Jr., Major, USAF, Ph.D., Purdue University, August 1988.
Robust Recognition of Loud and Lombard Speech in the Fighter Cockpit
Environment. Major Professor: Leah H. Jamieson.

— There are a number of challenges associated with incorporating speech recognition technology into the fighter cockpit. One of the major problems is the wide range of variability in the pilot's voice. The dependency of current recognition technology on the speech data that is used for training suggests that the pilot minimize the variability in his voice to optimize recognition performance. However, restrictions such as these are counterproductive to the premier goals of cockpit speech recognition: reducing the pilot's workload and improving the overall man-machine interface. To be truly effective in the cockpit, a speech recognition system must be capable of handling the wide range of variability in input speech that can result from changing levels of stress and workload. Increasing the training set to include abnormal speech is not an attractive option because of the innumerable conditions that would have to be represented and the inordinate amount of time to collect such a training set. A more promising approach is to study subsets of abnormal speech that have been produced under controlled cockpit conditions with the purpose of characterizing reliable shifts that occur relative to normal speech. Such was the initiative of this research. Acoustic phonetic deviations were carefully examined for two types of abnormal speech: loud (nominally 10 dB above normal) and Lombard

(speech produced when 90 dB of pink noise is injected into the speaker's ears through headphones). Analyses were conducted for 18 features on 17671 phoneme tokens across eight speakers for normal, loud, and Lombard speech. The most reliable differences were found to be in the spectral energies of the various frequency bands. Specifically, it was discovered that there was a consistent migration of energy in the sources out of the 0-500Hz and 4k-8kHz ranges, and into the 500-4kHz range. This discovery of reliable energy shifts led to the development of a method to reduce or eliminate these shifts in the Euclidean distance between LPC log magnitude spectra. The method, called Slope-Dependent Weighting, was used with a Smallest Cumulative Distance selection process. This combination significantly improved recognition performance of loud and Lombard speech. Discrepancies in recognition error rates between normal and abnormal speech were reduced by approximately 50% for all eight speakers combined.

1. INTRODUCTION

The ability for machines to react to human speech conjures a myriad of potential applications in the imaginations of people. At the extreme, ideas of sustaining an intelligent dialogue with a computer are brought to life with science fiction entertainment media. But in a more realistic sense, the field of speech recognition is now at the point that thirty years ago would have been considered to be fantasy. The methods now available in signal processing, the computing power of new architectures, advances in artificial intelligence, and the ability to deliver hardware and software in extremely small packages have combined synergistically to pave the way for significant advances in speech recognition. The outlook is promising, but as speech recognition research evolves from its infancy of limited vocabulary, isolated words, and benign environment, the research goals become more complex and diffuse. Such is the case when faced with the challenge of using voice interaction to improve the interface between the pilot and the fighter aircraft. The cockpit environment presents a number of new considerations that must be addressed in order to employ speech recognition successfully in this arena. The most important considerations are the noise present in the cockpit, the personal equipment the pilot must wear, especially the oxygen mask, the high workloads to which the pilot is subjected, and the range of stress experienced by the pilot under different flight conditions.

The first question that comes to mind for people unfamiliar with fighter operations is whether or not the fighter cockpit is an appropriate place for voice interaction. Can the fighter pilot benefit from such a system, or would it be a hindrance to the performance of his duties? The answer lies in the implementation. It is true that today's single-seat fighter aircraft possesses more capability than ever before and confronts the pilot with over 300 switches, buttons, and knobs requiring tactile manipulation. With his left hand controlling the throttle(s) and his right hand flying the aircraft with the control stick, the pilot must momentarily sacrifice control whenever he is required to activate a switch that is not directly mounted on the throttle or

control stick. The most classic case where the fighter pilot cannot sacrifice control is when he is required to fly close formation (approximately three feet of wingtip clearance) due to weather penetration (flying through cloud layers) or night conditions. Control inputs to maintain position must be continuous and immediate. His eyes must remain fixed on the references of the lead aircraft that he is using to maintain position. Moving away from the other aircraft in order to reduce the level of effort and attention is not an option because visual contact would be lost. Yet in this situation it is not uncommon to have to select a new radio frequency for communication with ground control agencies or other aircraft. There is no approved method for dealing with this dilemma; pilots handle it however they can. While this example alone would be justification for incorporating voice interaction in fighter aircraft, there are a number of analogous situations the pilot routinely faces that require him to sacrifice control during a critical phase of flight in order to reach a button, switch, or knob elsewhere in the cockpit. For illustration, Appendix A contains a list of essential tasks that are being tested in the Advanced Fighter Technology Integrator (AFTI) F-16 as applications of voice interaction. So it would seem that voice interaction is an obvious necessity in order to provide the pilot with an additional control channel. But to serve as an aid to the pilot in controlling his aircraft, the voice interaction system should be reliable and should not impose unnatural or unrealistic restrictions on the pilot.

With the many types of limited recognition systems commercially available, it is tempting to take an *off-the-shelf model and try to plug it into the cockpit with minor* modifications. The major flaw with this line of reasoning is the assumption that limitations such as a highly constrained syntax or minimal variability in speaking style or rate are factors that the pilot can easily accept. It is true that the pilot could be asked to adhere to strict procedures when using a voice recognition system, but in this case, the overall objective of reducing his workload and improving the man-machine interface has in reality been sacrificed. Instead, the pilot has been given yet another system whose idiosyncrasies must be committed to memory. To reiterate, the utility of voice interaction in the cockpit lies in the implementation. To be of benefit, the system must accommodate the pilot in *his* environment. It must be capable of performing in the presence of noise, tolerant to the range of variability in the pilot's voice under all flight conditions, and flexible to the grammar used.

It is in this setting that this research was motivated. The thrust was to study a controlled subset of voice variability to characterize reliable differences

from normal speech. Having quantified these distinctions, the aim was to compare and develop methods of signal processing that allowed the acoustic front end of speech recognition systems to perform more robustly on abnormal speech, given that training was accomplished only with normal speech. The intuitive notion that speech recognizers perform best when trained in conditions that resemble the operating environment has been confirmed in recent work (see section 6.1), and for most applications, this presents no problem. But for application to the cockpit environment, it is not plausible to collect an adequate training set of speech that would represent the range of variability caused by stress and high workloads because of the amount of time this would entail for the operational pilot. Instead, a first step in attacking the wide range of variability was to carefully study a subset of abnormal speech with the purpose of finding sufficiently reliable shifts that could be exploited to improve the front-end processing of a speech recognition system.

The subset of abnormal speech chosen for this research is divided into two categories: loud and Lombard speech. Loud speech is where the speaker has been instructed to talk louder than normal (nominally 10dB above normal conversation levels). Lombard speech is where 90dB of pink noise is injected into the speaker's ears with headphones, and he is instructed to talk normally. These two conditions were chosen because they closely resemble the changes that take place in speech under stress [Pa86], and because recent research has shown them to have a pernicious effect on speech recognizers designed for the cockpit [Ra86].

The discussion of this research is organized in the following manner. Chapter 2 covers the essential elements of speech production as background to the acoustic phonetic analysis that was performed in this research. The physiology of the vocal tract is reviewed along with the types of excitation. The sounds of speech are broken down into vowels, nasals, fricatives, stops, diphthongs, affricates, and semivowels. Models of the acoustic system are related to the dominant methods of speech processing: linear predictive coding, short-time Fourier transform, and cepstral analysis. In Chapter 3, the field of speech recognition is reviewed with an emphasis on clarifying terms. The factors that determine the complexity of speech recognition systems are examined in terms of recognition units, vocabulary, syntactic constraints, speaker variability, speaker dependence, target tasks, and environment. In addition, the various approaches to designing front-end acoustic processors are discussed along with the concept of knowledge sources for processing beyond the acoustic level. The knowledge sources can be broadly categorized as task

dependent and task independent. Chapter 4 provides an overview of the fighter cockpit environment. First a more detailed rationale is offered for incorporating voice interaction into the cockpit. Next, the types of fighter mission profiles are summarized. Aspects of air-to-air and air-to-surface missions are used to illustrate the variety of tasks required of the pilot. Then specific pilot tasks are explained from a standpoint of essential skills and training. Chapter 5 uses the context of Chapter 4 to identify the specific challenges of incorporating voice interaction into the fighter cockpit environment. The personal equipment worn by the pilot is detailed with specific emphasis on the oxygen mask. Issues of workload and stress are presented as causes for variability in the pilot's speech. Pilot acceptance is shown to be a prime consideration in the implementation of cockpit speech recognition. And in terms of Air Force efforts, it is shown how this research dovetails with other projects. Chapter 6 contains an expanded discussion of the objectives of this research. The motivation of the research is presented, and the database from the Armstrong Aerospace Medical Research Laboratory is described in detail. Chapter 7 reviews previous work concerning the analyses of loud and Lombard speech along with recent work at AAMRL. The analysis procedures of this research are then described, and the actual findings are presented in terms of differences in energy bands, spectral center of gravity, spectral tilt, pitch frequency, formant frequencies, and phoneme durations. Chapter 8 discusses the baseline recognition system used in this research, along with the rationale for the choices of its various features. The performance of the baseline system is presented for the eight speakers in the database, and a figure of merit is developed as a basis of comparison. Chapter 9 presents the actual experiments performed in this research. Based on the characteristics of the abnormal speech, a new distance metric is proposed, and its performance is documented for the speakers in the database. Lastly, Chapter 10 compiles the major conclusions of this research and provides suggestions for future research in this area.

2. SPEECH PRODUCTION

Speech production can be thought of as one half of a larger process, that of speech communication [He83]. It is the generation of an acoustic signal that carries information. The receipt of the speech signal, extraction, and interpretation of the encoded information constitute speech perception. In turn, speech is one of the various forms of human communication. Those who study communication as a science are well aware of the other forms such as written and visual. Indeed, there is more than one active channel of communication when two people carry on a conversation face to face. But nonverbal communication such as facial expressions and body movements are not at issue. The only concern of this work is the speech waveform and the information that can ultimately be recovered. In this respect, speech production would not seem to be a central issue either. The reality, however, is that regardless of how parochial the signal processing might be, the speech signal possesses such diversity that knowledge of the production lends considerable insight in selecting the most reasonable processing methods. This chapter provides a brief overview of speech production from three standpoints. First, the basic physiology is discussed along with the sounds of speech. Then the common models of speech production are reviewed. This compendium is compiled from works by Flanagan [F172], Rabiner and Schafer [RS78], Markel and Gray [MG76], and Fant [F60], [F73].

2.1 Physiology

As Markel and Gray put it, "speech physiology is the springboard for many different areas which are relevant to a better understanding of speech." Therefore, it is appropriate to initially consider the apparatus directly involved in speech production. This includes the vocal cords (sometimes called the vocal folds), vocal tract, and nasal tract. Ancillary to the speech apparatus is the subglottal system which consists of the lungs, bronchi, trachea, rib cage, and diaphragm. The subglottal system serves as a source of energy for speech production by providing the necessary air flow through steady contraction of the rib cage and diaphragm.

The vocal cords and the muscles controlling them are contained in an anatomical component called the larynx. The cords are actually two lips of ligament and muscle. The narrow opening between the two vocal cords is called the glottis. In the relaxed state, the glottis is fully open and air freely passes through as in normal breathing.

The vocal tract is a non-uniform tube that extends from the vocal cords to the lips, and has an average length of 17 cm in adult males. It can be broken down into the mouth or oral cavity, and the pharynx. Sometimes the mouth is referred to as the front cavity and the pharynx as the back cavity, but this is not uniform in the literature. Another common reference for dividing the front and back cavity is the point in the vocal tract where the cross-sectional area is minimized. The geometry of the vocal tract is controlled by movement of the articulators: lips, jaw, tongue, and velum.

The nasal tract is a primary air path for respiration but is only supplementary in terms of speech production. It is about 12 cm long in adult males and extends from the velum to the nostrils. Acoustic coupling between the nasal and vocal tract is controlled by the size and position of the velopharyngeal port. For a majority of speech sounds, the velum is in the raised position preventing significant air flow through the nasal tract. When the velum is lowered, the nasal passage is coupled to the vocal tract, causing a change in the characteristics of the speech sound.

2.2 Acoustic Phonetics

2.2.1 Excitation

Excitation of the vocal tract can be divided into three categories: voiced, unvoiced, and mixed. Voiced excitation originates at the vocal cords and is quasi-periodic in nature. Unvoiced excitation can occur anywhere in the vocal tract where sufficient narrowing occurs in conjunction with adequate air flow to produce turbulence. Mixed excitation is simply where voiced and unvoiced excitation occur simultaneously. The excitation source(s) produce what can be thought of as a raw signal. This signal is then altered by the characteristics of the vocal tract and possibly the nasal tract to form the speech signal.

Voiced excitation is a phenomenon that results from air being forced through a closed glottis. Sub-glottal pressure builds to the point that the vocal folds are pushed apart allowing air to flow through the glottis. Due to the release of pressure and the fact that there is an inverse relationship between air velocity and pressure (Bernoulli relation), the air rushing through

the glottis causes the vocal folds to slam shut thereby completing one cycle of the process. The output is a series of air pulses that excite the acoustic system above the glottis. The waveform of these air pulses is roughly triangular in shape and exhibits a duty factor of 0.3 to 0.7. The exact shape of the waveform can vary widely and depends on sound pitch and intensity. The glottal waveform is only quasi-periodic because of the small variations that are inherent in its production. Its spectrum contains harmonics whose amplitudes fall off at approximately 12dB per octave in normal voicing. Excitation of this type is sometimes referred to as phonation.

Unvoiced excitation is more noise-like in nature, possessing a relatively broad distribution of frequencies. Its aperiodicity results from the turbulent air flow produced at a constriction in the vocal tract. The turbulence in turn is produced by the separation of smooth air flow from the contours in the vocal tract. If the pressure distribution in the air flow is relatively uniform, then the air flow will be laminar and smoothly follow the surrounding surface. For a change in the velocity of the air flow, there will be a corresponding pressure gradient because of the inverse relationship between fluid pressure and velocity. When the pressure gradient is positive and sufficiently large, flow separation occurs and turbulent air flow is produced. At a point of constriction in the vocal tract, the air velocity increases due to the reduction in cross-sectional area of the passage. The air pressure at this point is lowered. Once the constriction is passed, cross-sectional area increases thereby reducing velocity and increasing air pressure. The resulting positive pressure gradient breaks down the laminar flow producing turbulence and noise.

Unvoiced excitation can take any of the following forms: frication, aspiration, whisper, and plosion. Frication is continuant in nature and results from the more extreme strictures found in the vocal tract. Aspiration is associated with greater laryngeal opening than fricative constrictions and tends to be distributed along the vocal tract rather than concentrated at a specific point. It also involves sub-laryngeal or tracheal resonances because of the opening in the larynx. Whisper is frication that occurs due to the closure of the glottis short of the point of phonation. Plosion is the noise made by the release of built-up pressure at some point of closure. Turbulent air again produces the noise, but the abrupt release provides only a transient excitation of the vocal tract, initially behaving as frication, but quickly degrading to aspiration as the stricture widens and air velocity decreases.

2.2.2 Sounds of Speech

The basic linguistic units used to describe the distinctive sounds of speech are called phonemes. Phonemes are the smallest units of speech that serve to distinguish one utterance from another in a language or dialect. For example, the phonemes /P/ and /B/ serve to distinguish the words *pig* and *big*. Variations of a given phoneme that are manifested in different contexts and pronunciations are called allophones. Generally, phonemes can be categorized by the manner and place of production in the vocal tract. The broadest categories are not unique. Rabiner and Schafer divide all the phonemes of American English into continuants and non-continuants while Flanagan divides them into vowels and consonants. Both divisions are equally valid and tend to emphasize different but overlapping characteristics of the phonemes. Another equally valid division would be according to the type of excitation. For this discussion, the more specific classes of phonemes will be reviewed: vowels, nasals, fricatives, stops, diphthongs, affricates, and semivowels.

Vowels are those sounds produced by phonation only and a relatively stable vocal tract configuration. Vowels are distinguished by their formant structure. Strictly speaking, formants are the peaks exhibited in the magnitude spectrum of the speech signal, but they are highly correlated to, and in practice, identified with the resonances of the vocal tract. The resonant frequencies in the vocal tract are determined by the way in which the cross-sectional area of the vocal tract varies with distance. This dependence of cross-sectional area upon distance along the tract is referred to as the *area function* of the vocal tract [RS78]. In turn, the area function is most predominantly affected by the position of the hump of the tongue and the degree of constriction it produces. Consequently, vowels can be distinguished as front, middle, or back, referring to location of narrowing in the vocal tract by the tongue; and as high, medium, or low, referring to the degree of constriction.

Nasals are those sounds produced by phonation but with the velum in the lower position and the oral vocal tract completely blocked at some point, acoustically coupling the nasal tract to the excitation. Radiation of sound is through the nostrils, neck, and other vocal tract walls. The oral vocal tract, while constricted, still influences the characteristics of nasal sounds by acting as a resonator cavity. Depending on the location of the constriction, and hence the size of the resonator cavity, the sound of the nasal will vary accordingly.

Fricatives are continuant phonemes produced, as the name implies, by a narrow constriction in the vocal tract. They are distinguished by the place of constriction, with the forward cavity performing the major spectral shaping and the rear cavity trapping energy and therefore introducing anti-resonances. Fricatives occur in cognate pairs, meaning that phonation can either be absent (unvoiced) or present (voiced).

Stops are those phonemes where a temporary closure occurs in the vocal tract during air flow causing a buildup of pressure. The closure is then abruptly relaxed, releasing the pressure and producing transient friction followed by aspiration. The place of closure distinguishes the stops, with the forward cavity affecting the signal the most. Stops also occur in cognate pairs. In the case of voiced stops, phonation exists throughout the closure and release. For unvoiced stops, the period of closure is manifested as complete silence in the speech signal. The allophonic variations of the stops exhibit a wide range of possibilities, depending on the surrounding phonemes and the position in the utterance. A case in point where there is a strong departure from the typical closure, burst, and aspiration sequence is when the stops /T/ or /D/ are preceded and followed by a vowel in the middle of a word. In this situation, the tongue can momentarily touch the alveolar ridge causing only a slight dip in amplitude as the articulators transition from the first vowel to the second. This is called a *flap* or *tap* and is sometimes added to the set of phonemes for convenience.

Some phonemes are actually concatenations of others already discussed. A diphthong is a phoneme that begins as one vowel and smoothly transitions to another vowel. Phonation is continuous and the transition is easily observed in the formant structure. An affricate is the combination of a stop with a fricative and can be voiced or unvoiced. The other dynamic phonemes are called semivowels. They are characteristically similar to vowels and diphthongs, and they provide smooth transitions between adjacent phonemes. Semivowels are also known as *liquids* and *glides*.

2.3 Modeling the Acoustic System

To do anything meaningful with current computer technology requires a quantitative approach and hence a model of a process. In the most general sense, models are artifacts developed by humans for the purpose of better understanding a physical process. If the process is simple, then the model can be extremely accurate. As the process becomes more complex, the goal of modeling becomes one of trying to approximate as closely as possible the

salient features of the process while retaining a reasonable amount of computational simplicity. The modeling of human speech is no exception. The acoustic signal is the product of a very complex physical process, to which researchers have little access. A tractable model of speech production has emerged, however, and its popularity has steadily grown in the recent years.

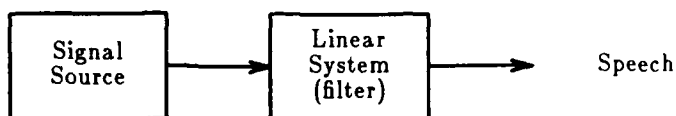


Figure 1. Generalized model of speech production

In the late 1950's, Gunnar Fant developed a linear model of speech production [MG76]. The general idea, which has persisted to the present, was to model speech as the output of some linear system excited by a signal source, as depicted in Figure 1. The basis is to first consider the vocal tract as an acoustic tube whose wave propagation characteristics can be mathematically expressed given simplifying assumptions such as plane wave propagation along the axis of the tube. Rather than replicating the exact shape of the vocal tract, it is approximated as connected sections of right-circular geometry. Concatenated uniform cylinders provide the easiest formulation. Circuit theory is then exploited by drawing the following analogies: sound pressure to voltage, volume velocity to current, inertance of the air to inductance, compliance of the air to capacitance, and viscous and heat conduction losses to impedance. Modeling each cylinder section as a two-port circuit, the resulting cascaded circuit resembles the more familiar configurations used to represent transmission lines. Using Fourier transforms, an expression for the transfer function of the vocal tract can be obtained.

The vocal tract transfer function can have either poles only or both poles and zeros, depending on the sound of speech being modeled. When the signal source is phonation only and when radiation is primarily from the lips, the transfer function contains poles only. The values of the poles represent the resonant frequencies of the linear system. Zeros are introduced when there is an additional cavity in the acoustic system besides the primary one from which radiation takes place. This occurs for nasals or when there is a signal source other than the glottis. In the case of nasals, the nasal tract is the primary flow path with the vocal tract acting as a side branch resonator. The vocal tract will sink or trap certain resonant frequencies, thereby extracting

them from the speech signal. When there is a constriction in the vocal tract for frication or plosion, the front cavity of the vocal tract performs spectral shaping and the back cavity acts as a side branch resonator.

In the array of digital signal processing techniques that have been applied to speech, the most predominant have been the short-time Fourier transform (STFT), linear predictive coding (LPC), and cepstral analysis. Of the three, the STFT does not have a direct relation to acoustic models. The Fourier transform simply treats the speech signal as a linear combination of complex exponentials (i.e. sines and cosines). By Parseval's theorem, the energy expressed in the time domain is preserved in the frequency domain. Therefore the coefficients of the complex exponentials are faithful representations of the energy at any particular frequency in the spectrum of the speech signal. The short-time aspect of the STFT serves two purposes. First, it is one way to insure the existence of the Fourier Transform. Second, it isolates vocal tract configurations and thus preserves the temporal information of the speech signal by the observed differences in successive analysis frames. Normally, each frame of speech is multiplied by a window that tapers the signal at the beginning and end. This reduces the extraneous frequency information introduced by abrupt edges. To prevent loss of information by windowing, the successive frames can be overlapped.

Cepstral analysis is a variant of the STFT and belongs to a class called homomorphic signal processing [OS75]. This method assumes the linear model of speech production with a signal source and filter. In the frequency domain the spectrum of the speech signal would then represent the product of the transforms of the source and filter. By taking the logarithm of the Fourier spectrum, the multiplicative combination becomes additive or linear. This means that linear filtering (called liftering because it occurs in the quefrequency domain! [Bog63]) can be used to separate the vocal tract filter characteristics from the excitation. In reality, the rapidly varying portion of the Fourier spectrum comes primarily from voiced excitation, and the more general contours represent the resonances of the vocal tract. It is these two components of the spectrum that cepstral analysis separates successfully.

Linear predictive coding is the other widely used method that incorporates the linear model of speech production. The efficiency with which LPC coefficients can be computed has made it a preferred choice in a number of speech processing applications. The simple concept of finding coefficients that most closely express a sample of the speech signal as a linear combination of the P previous samples provides an all-pole transfer function from which

any of a number of features of the speech signal are readily derived. The minor deficiency of not directly modeling nasal or fricative zeros is compensated by adding extra poles to the model (i.e. increasing the order of the predictor, P). The LPC spectrum has the characteristic of faithfully representing the resonances of the vocal tract system without the superposition of the source spectrum as does the STFT.

2.4 Summary

The process of speech production is undeniably intricate, with a number of variables contributing to the final output. From a linguistic standpoint, the physiology can be understood, and the various sounds of speech can be categorized and related to types of excitation and configuration of the vocal tract. From a signal processing standpoint, the acoustic signal can be modeled as the output of a linear, all-pole filter, extracting the pertinent resonance characteristics of the vocal tract. The model has worked well for applications involving normal speech, but has been found to be sensitive to the changes in speech that occur under different environmental conditions. The variability in loud and Lombard speech, as the central issue of this research, will be discussed in Chapter 6.

3. SPEECH RECOGNITION

With the large diversity of efforts pursued by governmental, commercial, and educational institutions, the term *speech recognition* takes on a number of different meanings. What is actually meant by speech recognition cannot be determined until specific options along various dimensions have been specified. Newell et al. [N75] has listed the following dimensions for defining the speech recognition problem: (1) type of speech, (2) number of speakers, (3) type of speakers, (4) speaking environment, (5) transmission system, (6) type and amount of training, (7) vocabulary size, and (8) spoken input format. While this list is not unique, it does demonstrate that many factors must be considered. The recognition systems possible with the variety of combinations range from the trivial to the impractical.

The terms *recognition* versus *understanding* are generally used to distinguish between the simpler versus the more intricate speech systems, but there can be confusion in what differences are actually implied. Rabiner and Schafer [RS78] suggest a fairly simple distinction. In speech recognition, they say, the goal is to transcribe the entire spoken utterance exactly. This could be in the form of a string of phoneme symbols (phonetic) or a set of words (orthographic). On the other hand, speech understanding implies an ability to provide the correct response or action to what was spoken. Reddy [R76] distinguishes speech understanding from speech recognition in that speech understanding must have the ability to respond correctly even when the utterance does not adhere to good grammatical rules and when the utterance is contaminated with speech-like noise such as babble, mumble, and cough. The stringency of these requirements is somewhat mitigated if speech understanding is viewed as seeking the intent of the message rather than a literal translation of every sound in the utterance. Still, the distinction between speech recognition and speech understanding is not sharp or well defined. A speech understanding system clearly needs as accurate a transcription as possible in order to maximize its probability of correctly determining the intent of the spoken utterance. But it can also be argued

that for a completely accurate transcription, a speech recognition system cannot neglect the meaning of an utterance. In light of these ambiguities, the purpose of this chapter is to present the dimensions and options that determine the nature of a speech recognition system. Unless otherwise specified, the term *recognition* will be used in the broader sense of seeking to produce the correct action or response to an input utterance, with correct transcription being a subtask.

3.1 Factors Determining Complexity

To view machine recognition from a standpoint of complexity requires consideration of a number of factors, many of which are highly interrelated. The task of machine recognition can be thought of as an attempt to correctly extract the message that has been encoded into the acoustic signal. From the vast amount of information present in the signal, only a certain subset is significant to the task at hand. The type of information sought in effect defines the goal and helps to determine the complexity of the task. At one extreme, the goal might be to correctly identify isolated words that are part of a limited vocabulary. This represents the simplest type of speech recognition, one that has been achieved by a number of different approaches with very good accuracy (recognition rates above 95%). At the other extreme, the goal could be to understand complete or partial sentences regardless of the context or task and generate an appropriate response or action to what was said. This is representative of human cognition and is beyond the present technology. However there is a continuum between these two extremes wherein lie a number of achievable tasks with varying degrees of difficulty inversely related to performance accuracy. The factors that determine positions along the continuum are discussed below.

3.1.1 Recognition Units

Recognition units refer to the temporal divisions of the speech signal and can actually be discussed on two different levels. On the upper level, the recognition units consist of isolated words (words separated by nominally 200 msec or more of silence), short phrases consisting of a limited number of words (sometimes referred to as connected speech), or complete sentences (continuous speech). The distinction between isolated words and continuous speech could very well be the most dominating feature in characterizing the complexity of a speech recognition task [R76]. Note however that words can also be the recognition units even though the speech is continuous. Such is the case in a particular form of speech recognition where the objective is to locate the occurrences of a relatively small set of key words. Word spotting, as it is

called, is simpler than continuous speech recognition because of the ability to discard portions of the speech signal that definitely do not have promise, but it is more complex than isolated speech because of the lack of distinct word boundaries.

The lower level recognition units are those used by the front end acoustic processor of a recognition system. They are the entities to which labels are first assigned. In isolated-word and some short phrase recognition systems, the units are the entire utterances. Otherwise, the units consist of shorter segments, the boundaries of which can be determined by different criteria [Le80]. One possibility is to use phonemes or allophones as recognition units. Another way is to segment on subphonemic units such as silence, burst, and aspiration for stop consonants. A method that attempts to capture coarticulation effects places the segment boundaries at the steady-state centers of phonemes. The resulting units are sometimes called diphones or transeemes. It is also possible to let the system make independent determinations of boundaries that emerge naturally from the analysis, rather than attempting to enforce phonetic constraints. On the other hand, IBM has found it more fruitful to view the acoustic processor as a data compressor rather than a mechanical phonetician [J82]. They use a time-synchronous segmentation which produces parameter vectors from fixed-length, successive frames of the speech signal. Similar to this was the Dragon system [Ba75] which used 10-msec speech segments as the basic unit of recognition.

The process of finding appropriate boundaries and classifying the individual units (or segmenting and labeling as it is normally called) is associated with an acoustic phonetic approach to speech recognition. This is in contrast to dealing with the entire utterance as an entity having a distinguishable pattern of features that can be classified by comparison to a set of stored utterance templates. There are advantages and disadvantages to each approach. In isolated word recognition, end points are easily detected, and there is no requirement to determine additional boundaries within the utterance. Therefore for a given utterance, the recognition process consists only of extracting a set of features by conventional signal processing techniques and systematically comparing them to all stored templates by using a distance metric. A drawback to isolated word recognition is that the methodology does not readily extend to longer utterances or more involved tasks. Another disadvantage is that processing time is directly proportional to the number of templates (size of the vocabulary). An acoustic phonetic approach has the potential of eliminating this problem because phonetic or

subphonetic label sets typically have on the order of 50 to 100 entries from which any word in the language can be derived. This does not hold true for diphones however. Because of the large number of phonemic combinations possible, diphone inventories can easily number in the thousands [Sh80]. The disadvantages of acoustic phonetic processing are mainly related to limitations in current speech technology for dealing with the ambiguities in phonetic boundaries and with the significant amount of variability in the speech signal.

3.1.2 Vocabulary

The individual words or utterances that make up the vocabulary have a significant effect on the complexity of a recognition system. As discussed in the previous section, increasing the size of the vocabulary makes template matching less and less tractable because of the sequential search requirements. Unless other constraints are applied, the size of the vocabulary will at some point make a template matching approach prohibitive, thereby forcing the use of acoustic phonetics and substantially increasing the overall complexity. Confusability is another issue. If the entries in the lexicon are distinctly different in phonetic pattern and makeup, this separability simplifies the amount of processing required to extract features for recognition. On the other hand, words that are highly confusable (e.g. similar vowel-consonant patterns but differing in a single phoneme) must be processed more rigorously to provide correct recognition. Vocabulary size and confusability are linked in that as size increases, the probability of occurrence of confusable word sets also increases.

3.1.3 Syntactic Constraints

The inclusion of syntactic constraints is a method of controlling the complexity of a recognition system. It provides the capability of accommodating a large vocabulary by breaking it up into a series of relatively small subsets. Given an appropriate scenario where the human operator is providing command, control, or data inputs by voice interaction, the number of syntactically feasible words at any one node of the input string can be severely limited. Thus the problems of confusability and searching long template lists can be significantly reduced. There is a tradeoff however. Reducing system complexity by syntactic constraints imposes a workload on the speaker by requiring him to adhere to the specific syntax built in to the system. It is a shift of burden whereby the speaker must compensate for limitations in the recognition system. Syntactic constraints therefore reduce system complexity while increasing usage complexity.

3.1.4 Speaker Variability

The inescapable differences that exist between repetitions of the same utterance by one speaker are normally referred to as speaker variability, although the same term can also refer to differences from one speaker to the next. To distinguish between the two meanings the former can be referred to as intra-speaker variability and the latter as inter-speaker variability. According to Zue [Z85], inter-speaker variabilities can be attributed to sociolinguistic background, dialect, and vocal tract size and shape. Intra-speaker variabilities can result from changes in the speaker's physiological or psychological state, speaking rate, or voice quality.

Regardless of the amount of conscious effort, a person cannot eliminate the inherent variability in his speech. The subtle differences in prosodics and pronunciation will always be present. Aside from the normal variability, the speech signal can be altered by any of a number of factors influencing the individual such as stress, workload, fatigue, and ambient noise. The Lombard effect, discussed in Chapter 1, is a classic example of induced variability in the speech. Another induced variability that is unique to the aviation environment is that of acceleration or G loading. Positive G loading is acceleration in the downward direction of the vertical axis of the pilot's body; it is the most common type encountered in aggressive flight maneuvers. The primary physiological effect of positive G loading is pooling of the blood in the lower extremities. To counter this effect, the pilot performs what is called an M-1 maneuver. This is where he tenses the muscles in his legs and increases pressure in the intrathoracic cavity by forcibly exhaling against a partially or completely closed glottis [Bu74]. Clearly the M-1 maneuver alone will have a severe impact on normal speech production. Initial research on the variability of speech under G loading has been done by Bond et al. [Bo86], and related work has been done by Davis et al. [Da84] and Sharp et al. [Sh78].

3.1.5 Speaker Dependence

Speaker dependence refers to the scope of the population of users of a recognition system. If the system is totally speaker independent then it has the ability to recognize the speech of any casual user regardless of age, sex, or dialect background. This capability implies a high degree of complexity because of the need of the recognizer to account for all possible inter-speaker as well as intra-speaker variations. Making a recognizer dependent on one person's voice patterns is a very common way of reducing system complexity. Usage complexity increases in this case because of the need for each new user to train the system by uttering a set of tokens from which reference templates

can be obtained. The system is also inherently sensitive to the intra-speaker variations subject to the conditions of the training. In other words, speaker dependent recognition systems work best on speech uttered in conditions similar to those trained on. This is because recognition algorithms for the most part use traditional pattern matching techniques without any attempt to model or compensate for intra-speaker variation.

3.1.6 Target Tasks

Proper consideration of the end application for a recognition system can help to control overall complexity. Examples of end applications offered by Lea [Le80] include: package sorting, quality control and inspection, programming of numerically controlled machines, voice-actuated wheelchair, banking and credit card transactions, security and access control, cartography in defense mapping, training air traffic controllers, command and control in defense applications, and cockpit communications. An expanded list of potential military applications is provided in the paper by Beek et al. [B77]. In fact most types of man-machine interaction, especially those where the person is already using his hands for other functions, are fertile areas that can stand to gain from speech recognition. The issue of how the target task affects system complexity can be broken down according to the constituent requirements of the task such as vocabulary, user population, environment, etc. It will also have a direct effect on the pragmatics or contextual constraints of the recognition problem.

3.1.7 Environment

The environment in which speech recognition is to be applied will affect complexity most commonly by the amount and type of noise that is added to the speech signal. Optimum conditions that produce a speech signal possessing a high signal to noise ratio (e.g. a sound booth or recording studio) are virtually non-existent in actual applications. The more likely environments would be those such as an office, computer room, factory, or warehouse. The additive noise could come from equipment fans, air conditioning, typewriters, line printers, other conversations, or heavy machinery. While a majority of additive noise in these environments can be reduced by closely-mounted, noise-cancelling microphones, the reduced signal to noise ratio in the low-energy portions of speech can still cause problems. An aspect of the environment that is less commonly considered is the effect it has on the speaker himself. The environment can be a major factor in the variability of a person's speech. This issue is discussed in the context of the fighter cockpit in Chapters 4 and 5.

3.2 Front-End Design Approaches

The methodology for designing the signal processing, feature extraction, and initial classification sections of speech recognition systems is not clear cut or well established. Approaches used tend to draw heavily from the established areas of pattern recognition and statistics, or they might attempt to exploit one or more models of the speech communication process. Lea [Le80] attempts to organize the approaches into the following four viewpoints: (1) acoustic signal, (2) speech production, (3) sensory reception, and (4) speech perception. In the acoustic signal viewpoint, the speech signal is treated just like any other waveform with no regard for its origin. Therefore a host of general signal analysis techniques are available for assigning the input to a particular class. In the speech production viewpoint, it is important to consider the origin of the signal and attempt to identify the elements and configuration of the human vocal system. This approach focuses on features such as formant frequencies, rate of vibration of the vocal cords, manner of articulation, place of articulation, and coarticulatory movements. The sensory reception viewpoint is concerned with exploiting the methods of the auditory apparatus in the human. This is done by extracting features and classifying patterns in a manner similar to the processes found in the ear, auditory nerve, and sensory feature detectors. For example, Seneff [Se84] [Se85] uses a model of the peripheral auditory system that incorporates a bank of 40 critical band filters covering 130 to 6400 Hz. The output of each filter is further processed to account for nonlinearities found in the cochlea such as rectification, adaptation, and saturation. The speech perception viewpoint uses features and classification techniques that have been experimentally shown to be significant to human perception. An example is the use of voice onset times and formant transitions as cues to determine whether or not a stop consonant is voiced.

Several classification techniques have emerged for assigning labels to input speech. One method with limited potential uses a decision tree based on specific characteristics of a very limited vocabulary. By exploiting reliable differences between members of the lexicon, decisions are made to assign the speech to increasingly restrictive categories until a single label is produced. Generally used for speaker independent, isolated word recognition, the decision tree must be tailored around the vocabulary, and it is not easily adaptable to changes in the vocabulary. Another method that has been widely used is pattern matching. Relatively independent of the particular units of recognition, templates are constructed from training speech to form a

reference set. The input speech is then segmented into test templates, and they are individually compared to each member of the reference set to determine the best match. A method that has been gaining in popularity uses Markov chains to statistically capture the temporal variations in speech. The method is called *hidden Markov models* (HMM) because it models speech as a doubly stochastic process. There is an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce a sequence of observed features derived from the speech [RJ86]. Since there is no way to solve for a maximum likelihood model analytically, iterative or gradient techniques must be used for training.

3.3 Knowledge Sources

The concept of using knowledge sources for analyzing the speech signal evolved from the ARPA speech understanding project in the 1970s. Due to the complexity of the speech communication process, looking at any one aspect of the acoustic signal yielded incomplete information. It was therefore determined that multiple sources of knowledge needed to be integrated in order to accomplish restricted levels of understanding. The method of integration, called a control strategy, was in itself a source of diversity. Bottom-up and top-down strategies indicated the hierarchical nature that could be assigned to the different knowledge sources. Interaction among the knowledge sources was limited, and artificial intelligence techniques were used to help limit the search space of alternatives. The Hearsay-II system developed at Carnegie-Mellon University used a more innovative strategy known as a blackboard model [Er80]. The blackboard model required knowledge sources to be independent yet cooperative. The existence and functioning of each knowledge source could not be necessary or crucial to the others. The blackboard was actually a dynamic global data structure where hypotheses from any knowledge source could be accessed and altered by any other knowledge source. In contrast, the bottom-up and top-down strategies arranged the knowledge sources in line where hypotheses propagated from one knowledge source to the next with minor provisions for backtracking or looping.

Reddy [R76] divides the knowledge sources into two general categories: task-dependent and task independent. The task independent knowledge sources are lower level in nature and are associated with features directly available in the speech signal: segmental phonetics and prosodics. Phonetics, as discussed in Chapter 2, is the systematic classification of sounds made in spoken utterances. Acoustic phonetics is the broader term, whereby the

particulars of production are also emphasized. This knowledge source is associated with the job of initial segmentation and labeling of the speech signal. Prosodics describes the suprasegmental information such as stress, intonation, and rhythm patterns of speech. Stress patterns can be used to provide "islands of reliability" whereby phonemic segments are more likely to exhibit predictable features [Le80-2]. Intonations or pitch contours can provide essential insight to the structure of sentences, as can the timing of phrases and the pauses between them.

The task dependent knowledge sources are derived from the lexicon, syntax, semantics, and pragmatics of the communication process. The lexicon, or vocabulary, was discussed in section 3.1.2 and is self explanatory. Syntax or grammar is the set of rules that prescribes how the words fit together to form meaningful utterances. Since only a certain subset of words can serve as alternatives at a given point in an utterance, syntactical knowledge can significantly prune the search space, as was noted in section 3.1.3. Semantics deals with the meaning and interrelationships of words within a sentence. The principal technique used to represent this knowledge source is a semantic network [R76]. Semantic networks give a simple structural picture of a body of facts [Ni80], which in this case are the relations among the words of the lexicon. Pragmatics models the contextual significance of the utterance. In man-machine communication, it would represent the overall scope of the target task as well as the body of knowledge gathered thus far in an ongoing dialogue. For example, the meaning of a pronoun could be resolved with this knowledge source by referencing the entities currently under discussion.

3.4 Summary

The term speech recognition eludes a simple or concise definition. Instead, choices along several dimensions must be clearly specified. In turn, these choices help determine the overall system complexity. The factors that determine complexity are the choice of recognition units, vocabulary, syntactic constraints, speaker variability, speaker dependence, target tasks, and environment. All of these factors are interrelated to some degree. The design of the acoustic front end of a recognition system can be approached from a signal processing standpoint exclusively, or knowledge of speech production, sensory reception, and speech perception can be incorporated. This will affect the types of features and cues sought for in the speech signal. To account for the deficiencies of analyzing the speech signal from only one perspective, multiple knowledge sources can be utilized to verify, reject, or augment initial hypotheses. The knowledge sources share information and collectively work

toward a conclusion under an executive control strategy.

4. THE FIGHTER COCKPIT ENVIRONMENT

4.1 Why the Fighter Cockpit

The modern fighter aircraft of today pose a formidable challenge as designers and engineers attempt to achieve the most effective man-machine interface, where man-machine interface simply refers to the flow of information and control inputs that occur between the man and the machine. The goal of human factors engineering is to optimize this interface such that the operator can control the machine effectively based not only on the information from the machine but also on his perception of the external environment. In accomplishing a task, the machine should augment the operator's basic abilities; it should become an extension of the human being itself. The information provided by the machine should be readily available, easily understood, and not distractive to activities underway. The operator should be able to provide control inputs to the machine in a natural and timely manner, again without hampering current activities. This in fact constitutes the classical feedback loop that is germane to any control system. If the man and machine are viewed together as a single system, with man being the essence of the feedback loop, it is obvious that the machine's performance will be a function of the effectiveness of the feedback loop. Notwithstanding varying abilities and skill levels of individual operators, the effectiveness of communication between man and machine will be a major factor in the overall performance of the man-machine system.

While man-machine interface is a generic term and as such can apply to situations such as a secretary and a typewriter or a nuclear power plant operator and the master console, the thrust of this research is centered around single-seat fighter aircraft. It is with modern high performance aircraft that the man-machine interface is indeed a crucial issue. In the early days of aviation, the primary consideration was not the design of a flying machine around a pilot; the pioneers were most concerned about building a vehicle that could become airborne and sustain flight. Once this was mastered, the

emphasis shifted to the achievement of particular missions such as transporting cargo or protecting certain airspaces. The man-machine interface was very simple. It consisted of a control *stick* or *yoke* whereby aircraft attitude was controlled, throttle(s) to control power or thrust, and a few buttons, switches, and levers to select functions, configurations, or information. Flight instruments to indicate aircraft attitude, speed, and altitude, and gauges to monitor engine performance, fuel status, etc evolved with time. As technology progressed in fighter aviation, aircraft became more sophisticated and capable of handling a wider variety of missions, but there was minimal modification in the man-machine interface. The stick and throttle remained intact, but the number of switches and buttons increased in direct proportion to the complexity of the aircraft and its systems. The most critical switches and buttons were mounted directly on the stick and throttle to facilitate easy access. To accommodate additional systems, more displays had to be incorporated into the instrument and other forward panels of the cockpit to provide all available information to the pilot. As this trend continued, the fighter cockpit became more and more crowded to the extent that displays, buttons, and switches took on multiple functions. Almost every finger and thumb had its own switch or set of switches to attend to. As an example, the throttles in the F-15 are sometimes sarcastically referred to as the *piccolo*. Reflecting this complexity, the term *switchology* has been coined which is essentially the study of all the cockpit switches, buttons, and displays, and the interaction among them. The fighter aircraft, more accurately referred to as a weapon system, can now challenge the pilot to the limits of his abilities to control all the subsystems available to him. Lovesey [Lo76] claims there has been a four-fold increase since 1950 in information sources for a typical single-seat aircraft. Lane [L80], in describing the problem, states, "The luxury of depending on operators to function in spite of system deficiencies is rapidly vanishing. Modern systems have crossed a critical threshold of operator workload. There is too much information for the operator to use, presented too rapidly, and in the wrong forms."

It must be emphasized that the primary function of a pilot's hands is to control the throttle and stick; in other words, to fly the aircraft. Even with major buttons and switches literally at his fingertips, considerable dexterity is required to select the proper switch setting for the desired result. Complicating the pilot's workload is the necessity for him to take one of his hands off a control (most likely the throttle) in order to manipulate a switch that is located elsewhere in the cockpit. Efficient and acceptable performance becomes questionable when the need to relinquish control in favor of switch

actuation occurs in a critical phase of flight (e.g. formation flight, low altitude navigation, takeoff, landing, air-refueling, etc). Further complicating the issue is the fact that costly visual inspection is often required either to locate the desired switch or to confirm its position. As will be discussed in the following sections, the types of missions and the pilot tasks required for these missions will further demonstrate the need for significant improvements in the man-machine interface. The proliferation of tactile switchology has created a type of bottleneck in the ability of the pilot to exercise complete control of the aircraft. It is in this context that speech recognition is being actively pursued to decrease the number of tasks required of the pilot's hands, digits, and eyes. The effort being spent to overcome the difficulties associated with speech recognition in the cockpit is well justified by the benefits to be gained in pilot workload reduction.

4.2 Typical Mission Profiles

Fighter missions can be divided into two broad categories: air-to-surface missions and air-to-air missions. Air-to-surface missions can be further divided into strike, interdiction, and close air support, while air-to-air missions can be divided into combat air patrol, escort, and intercept.

An air-to-surface mission is one where the objective is neutralization of enemy targets on land or sea. The ordnance can be either conventional or nuclear. In a nuclear strike mission, there is a single aircraft involved, whereas in a conventional strike mission, a formation of two or more aircraft is involved. The mission profile can include any of the following: high altitude flight for fuel economy; low altitude, high speed ingress and egress for detection avoidance; level, loft, or pop-up deliveries; re-attacks; evasive maneuvers for enemy ground and air threats; and air-refueling. There is little distinction between strike and interdiction in terms of pilot workload. On the other hand, close air support (CAS) is inherently a more complex mission. CAS describes the situation where aircraft are providing firepower in direct support of friendly troops engaged in ground combat. Information from the ground commander is often relayed through a forward air controller (FAC) who could be either airborne in a slow-moving observation aircraft or situated at a secure vantage point on the ground where he could survey a major portion of the battle area. The responsibility of the FAC is to coordinate the needs of the ground battle with the available airborne resources. In this scenario, the fighter pilot workload can become extreme. Not only is he required to maneuver at low altitudes and high speeds to avoid the battlefield threats, he is also required to visually acquire targets in unfamiliar and

dynamic settings based on verbal information from the FAC.

It is in a situation such as CAS that switchology becomes one of the weakest links in the chain. A large portion of switchology is learned by establishing regular habit patterns. Mistakes are minimized by following a relatively invariant set of procedures. But in the case of CAS, circumstances can change in seconds. In attempting to adapt to continually updated information, the pilot can become task saturated. In a sense, the pilot can be thought of as a system with multiple input and output channels [L80], each of which possessing a finite capacity. When the channel capacity is exceeded, the human behaves much like any other overloaded system. Sudden performance decrements occur, far out of proportion to the increase in message rate [Sha49]. Mistakes in switchology are inevitable; the consequence can be failure to expend ordnance, or even worse, the inadvertent expenditure of weapons on friendly troop positions.

An air-to-air mission is one where the objective is to gain or maintain air superiority. Air superiority is simply control of the airspace in and around the battle area. It is a state whereby air operations of all types can be conducted with freedom from enemy threat. The mission profile can include high altitude cruise, air refueling, and threat evasion, but otherwise has little in common with an air-to-surface mission. The most challenging (and perhaps the most glamorous and sensationalized) aspect of an air-to-air mission is the close-in engagement or *dogfight*. It is in the close-in engagement that the situation evolves most rapidly. A position of advantage where the pilot is on offense and a kill is assured can deteriorate to a defensive situation in seconds due to a single error in judgement, timing, or again, switchology.

Air-to-air maneuvering is three-dimensional, and the pilot must be intimately aware of his total energy state at all times. From classical physics, the total energy is made up of potential (altitude) and kinetic (airspeed). One can be traded for another as does a roller coaster in the simple analogy of slowing down while going uphill and speeding up while going downhill. Energy is dissipated from the effects of parasitic and induced drag, and is restored by the engine thrust. Energy management is one of the primary elements to winning or losing an air-to-air engagement. The pilot must make instant judgements on when to gain, conserve, or dissipate energy, and he must continually strive to seek the maneuvering envelope that puts his aircraft at an energy advantage relative to his opponent. The energy state for maximum performance is called *corner velocity*. This is the minimum speed at which the aircraft can generate the most G forces structurally allowed. Said another

way, it is the speed that gives the maximum turn rate for the minimum turn radius. Being able to out-turn the opponent provides the ability to attain a position of advantage from which weapons can be employed.

The aspects of air-to-air missions receiving less publicity are those events leading to an actual engagement. Combat air patrol (CAP) is where a number of fighters systematically search a specific airspace for enemy aircraft. The goal is to insure the airspace is clear of enemy threats so other air operations can be freely conducted. Escort is an air-to-air mission where the goal is to protect another flight of aircraft that are incapable of protecting themselves. These aircraft could be transporting personnel or supplies, or they could be involved with air-to-surface operations. In both CAP and escort, the actual tactics used will attempt to optimize a combination of radar and visual coverage to locate enemy threats. An intercept mission differs in that a specific target has been located by surveillance and the fighters' mission is to close on the target's position and engage, in the event the target is confirmed hostile, or identify as friend or foe in the event the target's status is unknown.

4.3 Pilot Tasks

While the above brief overview of mission profiles lends insight into understanding the complexity of pilot tasks, the picture is incomplete without consideration of the more generic pilot activities associated with fighter aircraft. These activities include basic aircraft control, aircraft systems monitoring, radio communication, navigation, visual lookout, system mode selection, formation flight, instrument flight, air refueling, handling inflight emergencies, and weapons employment. This list represents the types of tasks typical to any fighter mission.

Basic aircraft control is the term used for the fundamental aspects of flying. This consists of take-off, climb, cruise, turns, descent, traffic pattern, and landing. It involves continuous monitoring of the aircraft's attitude, speed, and altitude by observing either the flight instruments on the front panel or the information on the heads-up display (HUD). Control inputs are via the stick and throttle(s). Configuration changes (extension and retraction of landing gear, wing flaps, speed brake, etc) are accomplished by moving handles, levers, or switches to the proper position. Closely associated with basic aircraft control is the monitoring of all aircraft systems. The most critical of these is the fuel system. Even though most of the fuel system is autonomous and safeguards are incorporated, pilots have literally run out of gas in the heat of battle due to the diversion of their attention. Other

systems that the pilot is responsible for are the electrical system, hydraulic system, air conditioning and pressurization system, oxygen system, and of course, the engine(s).

The electronic equipment in an aircraft is referred to as the *avionics* and generally includes all communication and navigation equipment as well as radar, radar warning, specialized sensors, electronic countermeasures, and weapons computers. The pilot is required to maintain radio contact with appropriate ground controllers as well as other aircraft, and this may require cycling through a number of different frequencies in a short period of time. The pilot is also responsible for navigation. He generally will use either transmitted information from ground stations called TACANs (abbreviation for Tactical Air Navigation), inertial guidance onboard, pilotage (associating features on the ground with charts of the area), radar ground-mapping, or a combination depending on the aircraft altitude. As can be expected, low level operations complicate the task of navigation because of the increase in pilot workload.

Visual lookout is important in all phases of flight. This is simply the term given to the pilot's task of continually searching the airspace around him for other aircraft of any type. To obtain visual contact on an aircraft that is on the outer fringe of acquisition range requires considerable effort and methodical movement of the eyes. Moreover, a visual contact that is on the acquisition fringe puts a pilot in a condition called *padlocked*. This means that the pilot cannot take his eyes off the contact without risk of losing it entirely. If the contact is unfriendly, then the pilot definitely does not want to lose visual contact. Therefore a padlocked condition can severely restrict what the pilot is able to do. Switches that are not located on the stick or throttle must often be neglected.

Formation flying is when two or more aircraft maintain a specific spatial relationship with each other. The type of formation most familiar to the general public is called close formation or *fingertip*. In close formation, the aircraft maintain a separation of three feet between wingtips. The practical uses of close formation are to launch and recover large numbers of aircraft in minimum time, and to maintain visual contact with other formation members while penetrating difficult weather conditions. The pilot workload during close formation is at a maximum. Not only is the wingman unable to look away from the aircraft on which he is flying formation, but he is also unable to relinquish control of either the throttle or stick because of the continuous flow of instantaneous corrections required to maintain proper position. To relieve

pilot workload, the alternative to close formation is route formation where the planes spread apart the distance of a few wingspans. This gives the wingmen the opportunity to attend to cockpit tasks and look around somewhat.

For combat missions, any of a variety of tactical formations are used which in general are characterized as being much more widely spaced. Tactical formations can be optimized for either offensive or defensive maneuvering, but most often reflect a compromise of the two postures. Aircraft are positioned to take advantage of the mutual support available from both visual and radar coverage, and to provide maneuvering room in the event one of the members of the formation is attacked. Because of the distances among aircraft, pilot workload will increase due to the effort required to stay in position and follow the leader's movements.

Air refueling can be thought of as a specialized form of formation flight. The basic principles of maintaining position relative to another aircraft remain the same, but the physical coupling imposes a specific maneuvering envelope. Exceeding the envelope can result in aircraft damage in the worst case and a simple disconnect in the best case. During air refueling, the pilot must continuously adjust his technique as the flight characteristics of his aircraft change due to its increasing gross weight.

Instrument flying is required while penetrating cloud layers, flying in areas of reduced visibility, and flying at night. The pilot maintains proper aircraft attitude by continually crosschecking the displays available in the cockpit. Instrument flight conditions present an additional hazard to the pilot because of the potential for spatial disorientation. Since the vestibular apparatus of the inner ear senses acceleration rather than velocity or displacement, and since there is a lower threshold below which acceleration cannot be detected, it is quite possible for the pilot to feel as though he is upright when he is actually in a climb, descent, or turn. Spatial disorientation causes an increase in the workload due to the conflict between the instrument readings and the pilot's senses. Pilots are trained to rely on the instruments rather than the inner senses, but to do so in reality requires considerable concentration and results in increased stress.

All pilots are required to be thoroughly familiar with emergency procedures in the event of a malfunction or battle damage during flight. Procedures for matters that require immediate attention must be committed to memory while the remaining procedures are contained in an abbreviated checklist that the pilot has immediately available during flight. In addition,

the pilot must exercise sound judgement based on his training and experience when applying the procedures to a particular situation. It is obvious that inflight emergencies can maximize the pilot's workload and therefore create considerable amounts of stress.

The final pilot task that merits discussion is weapon employment. This is in fact the very essence of the fighter pilot's mission: to bring ordnance to bear on enemy targets. In air-to-surface missions, the weapons could be general or special purpose unguided bombs, an array of guided munitions, rockets, missiles, or cannon. Each weapon has unique ballistic characteristics and requires a specific combination of airspeed, altitude, and dive angle for proper delivery. Air-to-air missions use missiles or cannon as primary weapons. Weapon employment requires the pilot to first acquire the target visually and/or on radar. The pilot must also have the proper weapon selected and armed. After flying the aircraft to a position that insures optimum weapon performance, the pilot then releases or fires the ordnance and then commences any necessary evasive maneuvers.

4.4 Summary

The purpose of this chapter was to give a brief overview of the fighter cockpit environment. To understand the environment, one must have an appreciation for the array of missions fighter aircraft are required to perform as well as the complexity of the pilot tasks involved in accomplishing any given mission. Compared with the rapid advances in technology, the abilities of the human body are relatively invariant. This reality must be accepted in light of the fact that we can build machines that flood the operator with more information than he can absorb and provide him with more capability than he can exploit. The challenge of designing the aircraft around the human being was well summarized by Lt. Gen. Thomas H. McMullen, Commander of Aeronautical Systems Division, Air Force Systems Command, while speaking at the Air Force Association's Tactical Airpower Symposium in Orlando, Florida, January 1986 [U86]. He pointed out that the number of cockpit controls has proliferated since World War II to a point where there are more than 300 in the F-15. "We have got to take a giant step forward to help the driver, because his aircraft will be so much more capable." He stated that the imperative is to work on man-machine integration prudently to keep "the airplane from outflying the pilot." The goal is to "integrate man and machine to an unprecedented extent -- pilot, airframe, engines, weapons, fire controls, and sensors, all working together."

5. THE CHALLENGES OF RECOGNIZING COCKPIT SPEECH

In view of current speech recognition technology, the fighter cockpit is not an environment that can readily accept all the restrictions required for high recognition rates. On the positive side, there is no pressing need to consider speaker independence. But the other conditions that offer the best recognition performance with current technology (i.e. isolated words, limited vocabulary, highly constrained grammar, and minimal variability in speaking style) are not conducive to reducing the pilot's workload. It has been shown that a reasonable size for a cockpit vocabulary lies somewhere in the range of 200 to 700 words [D85, Hv88]. Indeed, the need to be cognizant of a precise speaking style, minimal vocabulary, and grammar can actually increase total workload.

While a noise-free environment also improves recognition performance, the problem of additive noise is not as significant as one might immediately suspect when considering jet aircraft. According to work done by Rajasekaran and Doddington [Ra85, Ra86], the noise level in the F-16 cockpit varies from 85dBA to 112 dBA, but this is significantly attenuated by the oxygen mask and helmet. They found the difficulty in recognition to lie more in the variability in the pilot's speech due to the factors affecting him rather than in noise being added to the speech. In their experiments they used an isolated-word system based on template matching with dynamic time warping. They compared recognition performance of speech with additive noise to speech produced under the Lombard condition. (Refer to Chapter 6 for a discussion of Lombard speech.) They found the substitution rate (i.e. the rate at which words were incorrectly recognized) for Lombard speech to be roughly an order of magnitude greater than that for speech with additive noise.

To clarify the challenges involved with recognizing cockpit speech, this chapter will first discuss the personal equipment worn by the fighter pilot and how it affects his speech. Next workload and stress will be addressed along with main factors that produce stress. Another issue worthy of discussion is pilot acceptance of speech recognition systems. No system will perform to

expectations if the user has already predetermined its performance to be lacking. And finally a brief review will be given of the current research being done in cockpit speech recognition.

5.1 Personal Equipment

A fighter pilot who is going on a combat mission will normally climb into the cockpit wearing the following items: flight suit, flight boots, flight jacket, gloves, anti-g suit, parachute harness, survival vest, helmet, and oxygen mask. Of these, the items that can have a direct effect on speech production are the helmet and oxygen mask. The helmet is a piece of protective gear that is custom fit to the pilot's head. It consists of a shell with hard foam on the interior and lined with leather. Earphones are mounted inside that allow him to hear radio communications as well as sidetone for his own voice. The helmet fits in such a way that a majority of cockpit noise is blocked, giving the pilot a relatively quiet listening environment.

The oxygen mask connects to each side of the helmet using adjustable bayonet fittings. The mask consists of soft rubber supported by a hard plastic outer shell. A length of hose connects the mask to a pressure-demand oxygen regulator. When properly fitted, the mask provides an airtight seal around the nose and mouth. An M101 microphone is mounted inside the mask directly in front of the pilot's mouth. Behind the microphone is the inhalation/exhalation valve. When the pilot inhales, an oxygen mixture is provided from the regulator; on exhalation, the valve vents to the cockpit. Depending on the cabin altitude, the oxygen regulator will deliver the appropriate mixture of oxygen by demand flow. If for some reason cabin pressurization fails and altitude is approximately 28,000 ft or above, the regulator will deliver 100% oxygen under positive pressure to the face mask. In this case the pilot must pressure breathe, a procedure whereby he passively allows his lungs to be inflated and then forcefully exhales against the pressure. The pilot's speech is significantly affected during pressure breathing due to the effort required to overcome the positive pressure in the mask. If the pressure is great enough, speech will be virtually impossible. For normal situations, however, speech production is fairly natural. Depending on facial features and the actual fit, the oxygen mask may partially block the nasal passages of some pilots. In addition, if the oxygen mask is not snug and secure, it can shift downward under high G loads increasing the probability of blocking the nasal passage. On the other hand, the snug fit of the mask can restrict the mandible, causing some resistance to jaw lowering which in turn could produce articulatory variations [Mo87]. Due to the structure and placement of the

mask, it can be viewed as an invariant extension of the vocal tract, but no studies have been done to date on this perspective.

Measurements have been made to assess the amount of noise attenuation provided by the oxygen mask in the cockpit environment. Rajasekaran and Doddington [R86] experimented with five subjects wearing an Air Force oxygen mask, M101 microphone, and subjected to four levels of F-16 noise ranging from 85dBA to 112dBA. Their results showed signal to noise ratios from 38dB to 16dB with the average being 26.9dB. In earlier work [R85] they reported that breath noise due to inhalation and exhalation to be a more significant problem than the ambient cockpit noise. While measuring signal to noise ratios in excess of 20dB, the signal to breath ratio was shown to be as low as 10dB. A significant component of the breath noise was found to be at about 4.5kHz which is understandable considering that this noise results largely from air passing through the narrow slits formed by the flaps of the inhalation/exhalation valve. It is in fact nothing more than friction with the constriction occurring at the valve.

5.2 Pilot Workload and Stress

The variabilities in the pilot's speech due to the cockpit environment elude straightforward quantification. In attempting to define the cause for a pilot's speech changes, a blanket answer might be the stress to which he is subjected. But the term stress is fairly ambiguous. It has only been in the last two decades that stress has become a central concept in psychological thinking, according to Hockey [Ho83]. He states that originally, stress was discussed in relation to disease and illness, but now has become "the most generally accepted term for those aspects of behavior which relate to bodily states, environmental changes, and the like." In a similar vein Welford [W74], in discussing the concept of stress, says "the relationships between man and his environment . . . are both the causes and results of physiological and psychological processes in the individual and of his social interactions with others. . . . Studies of stress have generated a vast literature in physiology, biochemistry, medicine, psychiatry, psychology, and sociology." He goes on to describe stress as a response that arises when an organism cannot correct or has difficulty correcting conditions that depart from optimum. Likewise, McGrath [M70] contends that stress is the result of an imbalance between demand and the organism's capacity.

In this context, pilot workload can be thought of as a direct contributor to stress. Noting that man performs best when a moderate demand is placed

upon him [W74], the relationship of stress to workload can be thought of as a downwardly concave function. At one extreme is minimum workload where an increased level of stress is most commonly referred to as boredom. This can be thought of as a form of sensory deprivation. Airline pilots are often the victims of this kind of stress because of the long periods of straight-and-level flight involved in their work, and the majority of a flight being controlled by autopilot. At the other extreme is maximum workload where the pilot becomes overwhelmed with the demands placed on him. This state is sometimes called task saturation, and performance falls while stress rises due to an inability to keep up. This situation is most likely to occur in combat situations, but can also happen in the most routine training missions, given the right conditions (e.g. an inflight emergency). Somewhere between the two extremes lies an optimum workload where stress will be minimized.

Other sources of stress besides workload must also be considered. Sheridan [S74] likes to correlate environmental stressors to the many different forms of energy that affect man: heat, noise, glare, vibration, acceleration, radiation, pressure, drugs and chemicals, etc. These can all be present in the fighter cockpit in one form or another. The term *self-induced stress* is often used to describe any type of stress over which the individual has direct control. The hypoxic¹ effects of smoking and drinking are examples of self-induced stress as is the failure to get the proper amount of sleep. And finally, a significant source of stress is the level of danger or threat perceived by the individual. Although a pilot can appear calm and collected over a wide variety of situations, every individual will experience a certain amount of stress in a life-threatening predicament. But regardless of the factors contributing to stress, the amount of stress actually produced will be a function of the individual, and can vary considerably from one person to the next. Because of the difficulty in quantifying stress and the large variation in responses of individuals to different levels of stress, very little is known about the exact changes of speech under stress beyond the intuitive notions of

1. Hypoxia is a condition where the body is deprived of oxygen. In milder degrees, the symptoms can take the form of drowsiness, inability to concentrate, irritability, increased reaction time, headache, nausea, or dizziness. The more severe forms can lead to loss of consciousness and death. The after-effects of smoking and drinking cause a reduction in the blood's ability to carry oxygen to the body cells, thereby inducing a mild degree of hypoxia.

increased loudness, increased pitch, and the like.

5.3 Pilot Acceptance

As is true with any new technology, the time and effort required for development can be a wasted investment if the needs and capabilities of the end user are not kept in focus. While there are cases when the application directs the development of technology, and vice versa, extreme caution must be exercised to insure compromises are not made based on limitations in the current technology. According to Lane [L80], "application of any new technology to solve one system problem can cause more severe difficulties if it is not considered in a total system and mission context." In the case of providing voice interaction capability to the pilot, the technology must be furthered in order to make speech recognition a viable enhancement to the man-machine interface in the fighter cockpit. Confining the pilot to speaking a certain way or highly constraining the grammar is tantamount to adding yet another burden to his workload.

Given a speech recognition system that performs acceptably in the cockpit, it is essential for the pilot to be intimately comfortable with the system before it can be useful to him. Experienced pilots, who are introduced to such a system as an add-on feature to a cockpit with which they are already familiar, may find it difficult to integrate voice interaction into their habit patterns. It is even possible for it to evoke a somewhat emotional response. (E.g. "I have been flying for years now without talking to my airplane and have gotten along just fine. Why should I start now?") The answer to this problem is simple: incorporate the technology into the Air Force training aircraft such that it becomes an integral feature expected by all Air Force pilots. Acknowledging that training the pilot in voice interaction is an essential step toward the success of cockpit speech recognition, the best time for this training is while he is developing habit patterns in basic flight skills. It should become second nature to the pilot to verbalize desired cockpit tasks or mode selections rather than reaching for a switch, button, or knob. To receive important information such as fuel status when he cannot glance at the gauge directly, the pilot should automatically query the airplane as if it were another crew member.

Fundamental to pilot acceptance of voice interaction systems is the reliability of such systems. A pilot who works with a system that is prone to errors will quickly lose confidence in the usefulness of the system. He will be conditioned to doubt whether or not the system will work in a crisis. And if

this occurs, the whole purpose of the system will have been negated. In a critical situation where the workload and stress are maximized, the pilot will elect to use the conventional switchology rather than risk any delay that would be induced by an unsuccessful recognition attempt. Verbally *arguing* with a black box is the last thing a pilot wants when he is being shot at. If this situation occurs, then the utility of voice interaction is lost. The system becomes little more than a novelty, a system the pilot can play with during non-critical phases of flight, but otherwise shuns.

5.4 Related Research

While virtually all speech recognition research is aimed at improving the man-machine interface in the generic sense, the Air Force's efforts that directly address the issues of cockpit speech recognition are centered at the Wright Aeronautical Laboratory (AFWAL) and the Armstrong Aerospace Medical Research Laboratory (AAMRL) at Wright-Patterson Air Force Base, Ohio. AFWAL has examined the performance of current-technology isolated word systems in the Advanced Fighter Technology Integrator F-16 (AFTI F-16), and AAMRL has collected speech data and is managing basic research in exploring methods of improving the robustness of recognition systems for military applications. In turn, the efforts of these laboratories are portions of larger programs. The AFTI F-16 program is exploring a wide array of new technologies for incorporation into the next-generation fighter scheduled for the latter 1990s. Besides voice interaction, other systems being considered include digital flight controls, automated maneuvering attack system with redundant ground/aerial target-collision avoidance, G-induced loss-of-consciousness recovery system, conformal infrared sensor/tracker, digital terrain management and display system with autonavigation function, automatic real-time weapon fuzing, and a helmet sight [AFA87]. (This list alone can stand as adequate testimony to the ever increasing complexity of the fighter cockpit.) AAMRL is participating in the Defense Advanced Research Projects Agency (DARPA) program on Robust Speech Recognition. Other work is being done by researchers at Carnegie-Mellon University, Massachusetts Institute of Technology, BBN Laboratories, Texas Instruments, MIT-Lincoln Laboratory, National Bureau of Standards, SRI International, and Schlumberger Palo Alto Research. Because of similar interests, the Navy is also working the problem at the Naval Air Development Center.

5.5 Summary

The fighter cockpit is indeed a severe environment in which to attempt automatic speech recognition. It is noisy, physically restraining, and demands the constant attention and concentration of the pilot. However, the primary difficulty of voice interactive systems is not the additive noise in the speech signal, but rather the variability induced in the pilot's speech, as reported by Rajasekaran and Doddington. The diversity of cockpit tasks and the continually evolving nature of combat missions produce challenging workloads and evoke considerable stress in the pilot, thus affecting his speech. Of all the personal equipment that the pilot wears, the oxygen mask is of singular concern in cockpit speech recognition. It provides good attenuation of the aircraft noise in the cockpit, but introduces turbulence noise from air passing through the inhalation/exhalation valve. And finally, the willingness of the pilot to use voice interaction must be addressed in the development of any cockpit recognition system. The pilot's confidence in the system must be to the extent that he will use it naturally and not avoid it during stressful or critical situations.

6. RESEARCH OBJECTIVE AND BACKGROUND

From the discussion in Chapters 4 and 5, voice interaction technology is a viable method of improving the interface between the pilot and the fighter aircraft. It can provide an additional channel of communication from the pilot to the aircraft, thereby reducing the number of tasks required of his hands. But there are a number of challenges to overcome to make cockpit recognition a workable and effective system, not the least of which is to account for the variability in the pilot's voice throughout the regime of flight conditions. While it is virtually impossible to quantify the exact changes that will take place in a pilot's speech under different degrees of stress, it is possible to induce certain types of variation that are likely to be present in speech under stress. By studying these variations under controlled conditions, it is possible to gain insight into the more general effects of stress on speech. The primary objective of this research was to consider two of these controlled conditions and learn as much as possible about the variations produced in the speech signal in order to develop compensation methods that would reduce the errors in recognition systems caused by these variations. The two controlled conditions selected for this research were loud speech and Lombard speech. Loud speech, as used in this research, is defined as speech that is produced by instructing the speaker to simply talk louder than normal (nominally 10dB). Lombard speech refers to the speech produced by subjecting the speaker to 90dB of pink noise via earphones and instructing him to talk normally.

The essence of this research was to discover a method of signal processing that would reduce the errors in speech recognition that are caused when systems are trained on normal speech and then required to recognize loud or Lombard speech. Recent studies in assessing the performance of recognizers under such conditions have verified that increased errors do result when the training and testing conditions differ [Ba86 Pa86 Ra86, Ro83, Co82, Ke82]. It has also been suggested that training be accomplished under conditions similar to those under which the speech recognizer will be required to operate [Pa86, Ra85, Ke82]. A methodology such as this could prove to be

self-defeating in the context of fighter cockpit recognition. Not only would this represent a significant training effort for each pilot to provide voice samples for all possible conditions in the cockpit, but it would also be virtually impossible to recreate the entire range of variabilities that can be present in cockpit speech. Instead, the aim of this research was to provide algorithms and signal processing techniques that would allow successful recognition of loud and Lombard speech, given that training had been accomplished only with normal speech.

6.1 Background

Motivation for this research originated with work at the Armstrong Aerospace Medical Research Laboratory (AAMRL) at Wright-Patterson Air Force Base, Ohio. As mentioned in Chapter 5, AAMRL collected speech data in cockpit and simulated cockpit environments, and under a number of different conditions designed to elicit changes in subjects' voices. The purpose of the database was to support the research being conducted by the DARPA robust speech recognition program. A subset of this database consisting of normal, loud, and Lombard speech from eight different speakers was acquired in analog form and served as the database for this research. The database is described in detail in the following section. Through conversations in the spring of 1986 with Timothy Anderson, Lt Mark Ericson, and Dr Thomas Moore of AAMRL, and Prof Leah Jamieson of Purdue University, it was determined that investigation of loud and Lombard cockpit speech was a timely topic that was vital to the robust recognition program and appropriate for doctoral research. Hence the objective of the research was not only to satisfy doctoral requirements but also to serve current Air Force research needs.

For reference, AAMRL uses Symbolics LISP machines running the Speech and Phonetics Interactive Research Environment (SPIRE) developed at MIT as primary speech workstations. Because of this, it was suggested that SPIRE be used to whatever extent possible at Purdue in order to simplify the transfer of data and intermediate results. Accordingly, SPIRE was acquired from MIT and successfully installed on the electrical engineering department's Symbolics 3670 in September 1986. Software was then written that provided the ability to transfer raw speech data as well as analysis information between SPIRE and other machines on Purdue's Engineering Computer Network (ECN). Source code for this software is contained in Appendix B. Details of the SPIRE workstation and possible configurations are described by Cyphers et al. [Cy86]. SPIRE was used to hand-label over 17500 phonemes as well as to

directly compute pitch and formant frequencies.

Recent research on the recognition of Lombard and similar types of stressed speech has been reported by Paul et al. [Pa86] at MIT Lincoln Laboratory. Their work is focused on the use of hidden Markov Models (HMMs) as an experimental isolated word recognition system. The system computes the first 12 mel-frequency cepstral coefficients every 10 msec as observation parameters. They were able to improve on the baseline performance of this system using a number of different techniques, most of which exploited the characteristics of HMM recognition. The more successful techniques were variance limiting (i.e. placing a lower bound on the variances to correct for occasional gross underestimation due to a small training set), adding temporal difference parameters to the baseline observation parameter set, and using multiple types of speech for training. By incorporating these three techniques simultaneously, they were able to reduce the baseline error rate by an order of magnitude. They also investigated the feasibility of eliminating the need to train on multiple types of speech by applying corrections to the statistics of the cepstral coefficients for normal speech. They experimented with two methods of obtaining the compensation to be applied to the coefficients. In what they called *single-model compensation*, a set of cepstral mean differences observed in multi-style models were applied as compensation in recognition on the following styles: fast, loud, Lombard, soft, and shout. The other method, *multi-model compensation*, developed four word models for each word: normal speech, low vocal effort, high vocal effort, and shout. From a baseline error rate of 13.0%, single-model compensation was able to reduce errors to 9.7%, and multi-model compensation produced an error rate of 4.5%, although the computational effort was increased by a factor of four. In general, they found the lower order and higher order cepstral coefficients required the most compensation, which is consistent with Pisoni's energy measurements in the low and high frequency bands [Pi85]. The method of cepstral compensation was expanded in a more recent paper by Chen [Ch88] that discussed an hypothesis-driven approach to correcting the cepstral coefficients.

6.2 Description of Database

The contents of the database being collected by AAMRL was determined by researchers from Texas Instruments, AAMRL, and MIT Lincoln Laboratories in April 1985, and is described by Doddington [D85, RD86]. The vocabulary and grammar were tailored to the task of the fighter pilot under all flight conditions. A finite-state grammar developed by Texas Instruments

was used to generate utterances for data collection. The vocabulary consists of the 207 words contained in Appendix C. The utterance set available for this research is listed in Appendix D. It was generated with the requirement that each word in the vocabulary be represented at least five times.

The complete AAMRL database is divided into two major portions: enrollment and test. This research worked with part of the data from the enrollment portion. Enrollment was conducted with the subjects wearing complete flight gear, including M101 microphone, oxygen mask, oxygen regulator, and helmet. The subject was seated in an anechoic chamber and prompted with utterances displayed on a video monitor. A technician monitored each recording session and initiated re-prompts for any mistakes or incorrect pronunciations. The five conditions under which enrollment data was collected were (1) normal, (2) loud, (3) Lombard, (4) fast, and (5) without flight gear. In the normal condition, the subject was given no special instructions on how to speak other than to use his natural voice. In the loud condition, the subject was instructed to speak loudly to simulate some of the effects that could occur during high stress conditions. For the Lombard condition, 90dB of pink noise was played through the headset to evoke the changes in the speech signal attributable to the Lombard effect. For the fast condition, the subject was instructed to speak rapidly to simulate some of the effects that could occur during high workload and stress conditions. For the last condition, the helmet and oxygen mask were not worn. Instead, the speech was collected from two microphones simultaneously. One microphone was a B&K 4165 placed in the far field and the other was an M162 close-talking noise-cancelling microphone on a head-mounted boom.

For this research, copies of the normal, loud, and Lombard enrollment sessions for each of ten speakers were obtained from AAMRL, for a total of 30 sessions, as listed in Appendix E. Each session contains the 539 utterances listed in Appendix D, for a total of 16170 utterances. The original plan was to use all ten speakers in this research, but upon close examination two speakers were found to have contaminated data. Speaker #9 recorded a significant portion of the utterance set with the oxygen mask not correctly seated (that is, one or both bayonets of the mask not securely fastened to the helmet), thus increasing the distance between the speaker's mouth and microphone. Speaker #10 experienced positive pressure at times due to a faulty oxygen regulator, thereby significantly restricting the normal flow of air from his mouth. Therefore these two speakers were excluded from this research, leaving the first eight speakers listed in Appendix E.

Because of the limited growth potential of isolated word recognition systems, this work focused on an acoustic phonetic approach to analysis and compensation of loud and Lombard speech. With this in mind, a preliminary analysis of the phonetic content of the database acquired from AAMRL was conducted to insure it could provide sufficient phonetic variation for experimentation. A first-order estimate of the phonetic content of the database was obtained by transcribing each of the 207 words in the vocabulary as depicted in Appendix F. The majority of American English phonemes was well represented in the vocabulary, with the following exceptions. The two phonemes /DH/ and /ZH/ were not represented at all. Additionally, the phoneme /UH/ existed only in the word ENDURANCE and was found to be dependent on the speaker's pronunciation and dialect. The remaining set of phonemes that were adequately represented in the database are listed in Appendix G and were used as the basic units of analysis and recognition in this research. A cross reference of the words in which each phoneme appears is given in Appendix H.

The acoustic phonetic approach of this research limited the amount of data that could be thoroughly analyzed. To obtain a preliminary estimate of the amount of time required to prepare data for analysis, the first 20 utterances in one session were processed in the following manner. First the utterances were digitized in the One-Dimensional Signal Processing Laboratory (ODSP) using a PDP 11/40. Then they were transferred to a VAX 11/780 on ECN where DC bias was removed. Each utterance was stored in its own file. Next the utterance files were transferred to the Symbolics 3670 and converted to a format compatible with SPIRE. Finally, SPIRE was used to produce phonetic and orthographic transcriptions of each utterance by hand-labelling phoneme and word boundaries. The 105 words in the 20 utterances contained a total of 388 phonemes. The time spent actually working with this subset of data was recorded with a stopwatch and found to be almost 11 hours. Figure 2 shows the breakdown of the amounts of time required by each individual activity. Note that hand-labelling required over 70% of the total time. Because of the amount of concentration required for hand-labelling, it was found that this task could not be pursued for more than one or two hours at a time without a break or change in activity. Hence, the total elapsed time to complete the processing of the 20 utterances was approximately one week. At this rate, the processing of all 539 utterances (2151 words) for only one session (i.e. one condition for one speaker) would take almost six months! This is not only infeasible but also unnecessary for an adequate acoustic phonetic analysis. Consistent with the design of the baseline recognition system

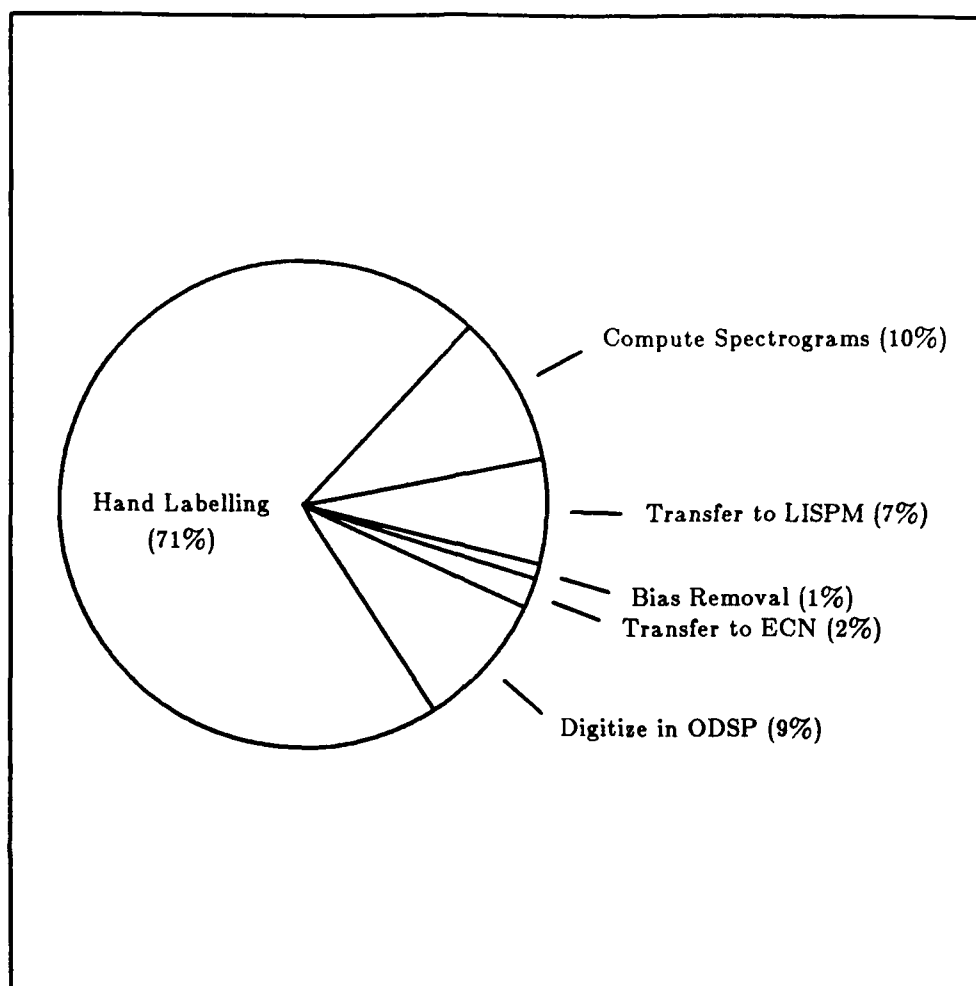


Figure 2. Division of time required to preprocess analog speech data

(described in Chapter 8), six tokens per phoneme were chosen as the minimum number required for analysis and experimentation. This allowed the pruning of each session to a reasonable amount of data.

The need to carefully select a subset of utterances to provide at least minimum coverage for each phoneme was borne out by the lack of adequate coverage in the previous situation where the first 20 utterances of a session for speaker #2 were processed. The phoneme occurrences for this set of 20 utterances are listed in Table 1. Note that 14 of the 40 phonemes have fewer than five tokens in this set. In light of the amount of time required for preliminary processing, the goal then became one of selecting the set of utterances that provided adequate coverage of all 40 phonemes while minimizing the total number of phonemes that had to be labelled. An

Table 1. Phoneme occurrences in utterances A001 - A020 for speaker #2, loud speech

Phoneme	Number of Occurrences
P	11
T	18
K	5
B	4
D	6
G	0
DX	8
M	2
N	48
NX	0
S	26
Z	4
CH	1
TH	6
F	14
SH	2
JH	5
V	8
L	12
R	24
Y	0
HH	2
EL	4
W	8
EH	14
AO	3
AA	6
UW	6
ER	2
AY	20
EY	9
AW	5
AX	24
IH	22
AE	6
AH	5
OY	0
IY	15
OW	9
AXR	4

algorithm was therefore developed to select such an utterance set. The algorithm is detailed in Appendix I, and the source code implementing the algorithm is listed in Appendix J. With the minimum phoneme coverage set to six, the algorithm selected a set of 43 utterances. Because this was a first order selection based on the phonetic transcription of the isolated words in the vocabulary, it did not account for the inevitability of missing segments that

Table 2. Total phonemes processed for eight speakers

Phoneme	Number of Tokens
P	356
T	1003
K	410
B	190
D	262
G	170
DX	477
M	168
N	1367
NX	188
S	990
Z	332
CH	157
TH	265
F	546
SH	169
JH	272
V	272
L	268
R	616
Y	166
HH	264
EL	329
W	240
EH	764
AO	348
AA	453
UW	344
ER	714
AY	747
EY	335
AW	180
AX	594
IH	1233
AE	222
AH	235
OY	168
IY	543
OW	876
AXR	438
Total	17671

often occur in connected speech. To remedy this, phoneme deficiencies were counted after labelling the 43 utterances for speaker #1, and additional utterances were then selected manually to eliminate the deficiencies while minimizing the total excess phoneme count. This resulted in the set of 56 utterances listed in Appendix K. After all labelling was complete, it was found that an average of 1275 labels were assigned to each session of which

736 were usable phonemes. The difference is attributed to labels being assigned to word boundaries, silence regions, and phonemes excluded from the list in Appendix G. Note also that achieving a minimum coverage of six for each of the 40 phonemes (240 phoneme tokens) required the labelling of over three times the minimum total tokens (i.e. 736 tokens). This was a consequence of working with the predetermined vocabulary and utterance set from AAMRL. While the excess phonemes were not used directly in the recognition experiments, they did contribute to the task of analyzing and comparing normal, loud, and Lombard speech. The differing sample sizes for each phoneme were taken into account in the analysis of variance calculations. The actual breakdown of usable phonemes across sessions is summarized in Table 2.

6.3 Summary

The purpose of this research is to search for methods to improve the performance of cockpit speech recognition by developing signal processing techniques to overcome the errors induced by loud and Lombard speech. This research is part of an ongoing effort distributed among the Defense Advanced Projects Research Agency (DARPA), Armstrong Aerospace Medical Research Laboratory (AAMRL), and the Air Force Wright Aeronautical Laboratory (AFWAL). The database used in this research is a subset of the robust speech recognition database collected at AAMRL. There were a total of 24 sessions of data, representing normal, loud, and Lombard speech from eight speakers. Fifty-six utterances were digitized and hand-labelled from each session. Out of the 30,608 labels assigned, 17,671 were members of the 40-phoneme set.

7. ANALYSIS OF ABNORMAL SPEECH

In order to compensate for a phenomenon, one must first learn as much as possible about the phenomenon. Consequently the first phase of this research was devoted to a thorough analysis of loud and Lombard speech in order to characterize the deviations that occur relative to normal speech. The analysis was at the phonetic level and looked at features such as energy distributions along several frequency bands, gross spectral shape, formants, and pitch. Since all data was hand-labelled, detailed results on the variations of these features were obtained for every phoneme listed in Appendix G. Earlier work, as detailed in the next section, discovered general trends and provided some quantitative results. As will be shown, the research in this thesis clarifies and more completely quantifies some of the findings by other researchers.

7.1 Previous Research

The fact that a speaker changes his voice level dependent on the amount of ambient noise and the level at which he hears his own voice (i.e. his sidetone) was first observed in 1911 by the French otorhinolaryngologist Etienne Lombard and reported in his paper, *Le Signe de L'elevation de la Voix* [Lom11]. Since then, the phenomenon has come to be known as the *Lombard sign* or *reflex*, although the term *reflex* could be misleading since the phenomenon encompasses a range of voice levels rather than a single reflexive shift. A number of studies, as listed by Lane and Tranel [La71], have shown there to be reliable changes in speech characteristics when the speaker is subjected to controlled levels of ambient noise. In the years immediately following Lombard's discovery, the major emphasis in research on the Lombard reflex was in using it to help diagnose hearing disorders. Gradually, issues such as intelligibility and communication were also raised for better understanding the Lombard reflex. As early as 1949, Hanley and Steer [Ha49] found that when speakers were subjected to increasing levels of noise through headphones, they tended to speak at successively slower rates (i.e. words per minute), to increase syllable duration, and to speak with greater intensity.

Having hypothesized that reduced speaking rate, syllable prolongation, and increased voice level increased intelligibility, they concluded that speakers subjected to high noise levels naturally invoked the measures necessary to be better understood.

In 1970 Lane, Tranel, and Sisson [La70] reported on the codification of the Lombard reflex. By relating the sone scale (sound pressure from an external source must be more than tripled for a subject to perceive a doubling in loudness) to the autophonic scale (sound pressure from the subject's own voice must be less than doubled for him to perceive a doubling in loudness) they found the slope of the voice compensation function to be 0.5 in log-log coordinates (decibels). Said another way, for a given dB increase in noise level, a speaker would increase his voice level by half the dB increase in the noise. His perception of the noise and his own voice would indicate that he had provided adequate compensation. Furthermore, they found the same slope to exist in two other situations, providing additional substantiation in the relation between the sone and autophonic scales. One situation was where subjects were instructed to match the level of a sound with their own voice. Their voice level would increase by half for a particular increase in the given sound. The other situation was in sidetone compensation. As the level of sidetone through headphones was increased, subjects would reduce their voice level by one-half the amount of sidetone increase.

In following work, Lane and Tranel [La71] showed that the slope of 0.5 for the noise compensation function could be taken as an upper bound on the amount a person raises his voice in the presence of noise. Slopes of less than 0.5 were obtained when varying emphasis was placed on the intelligibility of what was spoken. They concluded that the magnitude of the Lombard reflex was also governed by the premium placed on intelligible communication in addition to the obvious effect of the ambient noise level. At one extreme, where no premium was placed on intelligibility, they found speakers to show very little reaction to changes in the ambient noise level. This extreme was discovered when speakers were asked to simply read a list of utterances into a microphone. Dreher and O'Neill [Dr58] reported a slope of 0.11 when performing such an experiment. At the other extreme, a high premium was assigned to intelligibility by placing speakers in two-way conversation situations. Experiments subjecting pairs of speakers to ambient noise while conversing were conducted by Webster and Klump [We62] and yielded a slope of 0.5.

Recent work that had similar motivations to the research of this thesis was conducted by Pisoni et al. [Pi85]. They studied the speech of two male talkers uttering the digits zero through nine in both a quiet and a noisy (90dB) environment. In addition to the reliable changes that had been previously reported in the literature (i.e. shifts in prosodic features such as amplitude, duration, and pitch), they were concerned with examining the spectral properties of Lombard speech. Their findings included observed changes in formant frequencies and changes in generalized slope or *tilt* of the short-term spectra. By obtaining the formant frequencies at three different points in the voiced portion of each word, they found a general tendency for F1 to shift upward and F2 to shift downward in Lombard speech. Unfortunately they could not be more specific because of the limitation in the syllabic environments of the vocabulary. The only reported observation in the nonvocalic consonants was a significant decrease in the first reflection coefficient of the LPC analysis.

Pisoni et al. used two different methods to assess spectral tilt. One method fit a least-squares regression line to the power spectrum of vocalic segments across all ten digits, and the slope of this line was computed. This method indicated a 1.5dB/octave increase in the spectral tilt for both speakers. The other method divided the spectrum into three frequency bands using the 9th and 20th harmonics of the speaker's mean fundamental frequency as boundaries. Then the peaks in the LPC spectrum for each band were averaged across tokens for a given word and across vocalic segments for all words. They found a decrease in the low frequency band of over one dB for both talkers in the Lombard speech relative to normal speech. There was no significant shift in the middle band, but the high band exhibited an increase in energy of over two dB for Lombard speech.

Research conducted at AAMRL by Moore and Bond [Mo87] used four male speakers uttering two repetitions of ten spondaic words under normal and Lombard conditions. Analysis was limited to the effects observed in the nine vowels: /IY/, /IH/, /EY/, /EH/, /AE/, /UW/, /OW/, /AA/, and /AH/. For the Lombard speech, they observed an increase in pitch frequency of up to 40 Hz, increases in the first formant of 20 to 70 Hz, and decreases in the second formant of 20 to 100 Hz.

7.2 Analysis Procedures

Each of the 56 utterances in Appendix K were digitized at a rate of 16k samples/second with an accuracy of 12 bits/sample, as described in Chapter 6.

Table 3. Features for analysis

1	Energy Band 1	0 - 250Hz
2	Energy Band 2	250 - 500Hz
3	Energy Band 3	500 - 1kHz
4	Energy Band 4	1k - 2kHz
5	Energy Band 5	2k - 3kHz
6	Energy Band 6	3k - 4kHz
7	Energy Band 7	4k - 5kHz
8	Energy Band 8	5k - 6kHz
9	Energy Band 9	6k - 7kHz
10	Energy Band 10	7k - 8kHz
11	Spectral Center of Gravity	
12	Low-band Spectral Tilt	0 - 3kHz
13	High-band Spectral Tilt	3 - 8kHz
14	Pitch Frequency	
15	First Formant	
16	Second Formant	
17	Third Formant	
18	Duration	

Table 4. Set of 40 phonemes grouped by category

Stops	Nasals	Fricatives	Liquids	Vowels	
1 P	8 M	11 S	19 L	25 EH	33 AX
2 T	9 N	12 Z	20 R	26 AO	34 IH
3 K	10 NX	13 CH	21 Y	27 AA	35 AE
4 B		14 TH	22 HH	28 UW	36 AH
5 D		15 F	23 EL	29 ER	37 OY
6 G		16 SH	24 W	30 AY	38 IY
7 DX		17 JH		31 EY	39 OW
		18 V		32 AW	40 AXR

Each utterance was then linearly amplitude normalized and stored in its own data file. The files were then transferred to the Symbolics 3670 via ethernet where SPIRE was used to hand-label all the phonemes for each utterance. Statistics were compiled on the 18 features listed in Table 3 for each of the 40 phonemes listed in Appendix G. The 40-phoneme set is listed in Table 4 with reference indices for each phoneme. Linear predictive coding was used to provide an estimate of the speech spectrum for computing feature numbers 1 through 13. Routines available in SPIRE were used for computing feature numbers 14 through 17, and feature number 18 was derived directly from the phonetic labeling. A 24th order LPC was used because the oxygen mask acts as an extension of the vocal tract. Assuming the average length of the male vocal tract to be 17 cm and the effective increase due to the oxygen mask to

be 3 cm, the memory of the predictor, τ , is [At71]:

$$\tau = \frac{2 \cdot l}{c} = 1.31 \text{ msec} \quad (1)$$

where l is the effective length of the acoustic tube, and c is the speed of sound. A sampling rate of 16kHz then gives a predictor order of 21 to which a factor of 3 is added to account for glottal volume flow and radiation effects. The validity of the predictor order was checked by noting the performance of the baseline recognition (described in Chapter 8) system using 14th, 24th, and 34th order LPC computations. Performance improved significantly when going from 14th to 24th order, but not when going from 24th to 34th order.

To compute the energy in the various frequency bands, each phoneme token was divided into 50 overlapping frames, each 16msec (256 samples) long. The degree of overlap was dependent on the duration of the phoneme, but provided at least 50% overlap for phonemes less than 400 msec in duration. For each frame, a 128-point log magnitude spectrum was computed using the 24th order LPC of that frame. The points in the spectrum representing a particular energy band were averaged together and normalized for the bandwidth, and then averaged across the 50 frames for one phoneme token. These values were combined for all tokens of a given phoneme to obtain a sample mean and sample variance.

More specifically, the LPC coefficients, $\{a_k\}_{k=1}^P$, approximate an all-pole model of the vocal tract where the transfer function, $H(z)$, is expressed as

$$H(z) = \frac{1}{1 - \sum_{k=1}^P a_k z^{-k}} \quad (2)$$

and P is the order of the predictor. The magnitude spectrum is then obtained by evaluating $H(z)$ in the interval $0 \leq \theta \leq \pi$ along the unit circle. Individual samples of the log magnitude spectrum can then be written as

$$A_l = \ln \left| H \left(e^{j \frac{2\pi l}{N}} \right) \right| = \ln \left| \frac{1}{1 - \sum_{k=1}^P a_k e^{-j \frac{2\pi k l}{N}}} \right| \quad (3)$$

where N was chosen as 256, and $0 \leq l < \frac{N}{2}$. This choice of N divided the LPC spectrum into 128 samples. The frequency spacing of the samples, Δf , is simply

$$\Delta f = \frac{2f_n}{N} \quad (4)$$

where f_n is the nyquist frequency of 8kHz in this research. Now for a given frequency band, b , with lower cutoff frequency, f_{bl} , and upper cutoff frequency, f_{bh} , the average energy, E , measured for a particular phoneme token, t , for phoneme p , of speaker s , and condition c , is defined by

$$E_{bpsct} = \frac{1}{N_F(l_{bh}-l_{bl})} \sum_{i=1}^{N_F} \sum_{l=l_{bl}+1}^{l_{bh}} A_{lisct} \quad (5)$$

where i is the individual frame index, N_F is the total number of frames (50 per phoneme token), and the indices of summation for the frequency band are

$$l_{bl} = \text{int} \left(\frac{f_{bl}}{\Delta f} \right) \quad (6)$$

$$l_{bh} = \text{int} \left(\frac{f_{bh}}{\Delta f} \right) \quad (7)$$

When combining all T tokens of a given phoneme, the energy, E , for a given frequency band, phoneme, speaker, and condition becomes

$$\begin{aligned} E_{bpsc} &= \frac{1}{T} \sum_{t=1}^T E_{bpsct} \\ &= \frac{1}{N_F T (l_{bh}-l_{bl})} \sum_{t=1}^T \sum_{i=1}^{N_F} \sum_{l=l_{bl}+1}^{l_{bh}} A_{lisct} \end{aligned} \quad (8)$$

The spectral center of gravity, *COG*, was computed by viewing the 128 samples of the LPC log magnitude spectrum as point masses on a line. A positive constant, ξ , was added to each sample to eliminate negative sample values. The result was expressed as a frequency where the spectrum would *balance*. The formula for center of gravity is expressed as

$$COG = \frac{\Delta f \sum_{l=0}^{\frac{N}{2}-1} l(A_l + \xi)}{\sum_{l=0}^{\frac{N}{2}-1} (A_l + \xi)} \quad (9)$$

The sample statistics for each phoneme were derived as for the energy bands. Spectral tilt for a given frame was obtained by using energy band features as coarse samples of the spectrum and computing a linear regression by the method of least squares. The slope of the line was then the estimated value of spectral tilt. The low-band spectral tilt was computed with the energy bands one through five, and the high-band spectral tilt used energy bands six through ten. Sample statistics for each phoneme were then derived as above.

For pitch and formant frequencies, SPIRE provided estimates with 5 msec spacing. After a simple smoothing operation, the same procedure was applied to derive the sample statistics for these features. Duration statistics were obtained by combining the individual token durations for each of the 40 phonemes. Significant differences in the sample statistics of a given speaker were found by applying a three-way analysis of variance [Do74, Bl80] (normal, loud, and Lombard conditions) on every feature of every phoneme. The level of significance was set to 0.01.

7.3 Results

Given that 720 three-way analyses (18 features $\times$ 40 phonemes) were performed on each of the eight speakers for a total of 5760 analyses, it is easy to become overwhelmed with all the possible combinations and comparisons. Rather than providing an exhaustive discussion of all of these analyses, it is more beneficial to discuss significant trends and commonalities as well as striking differences. The information in its entirety is contained in Appendix L for reference. Unless otherwise noted, observations are based on grouping both the loud and Lombard speech as *abnormal* and comparing it to the normal speech. Findings are discussed in terms of significant differences in spectral energy distribution, spectral center of gravity, spectral tilt, pitch, formants, and phoneme duration.

7.3.1 Spectral Energy Distributions

Spectral energies for the ten frequency bands in Table 3 were compiled by the method discussed in the previous section. These bands provide a relatively smooth summary of spectral changes when comparing normal to

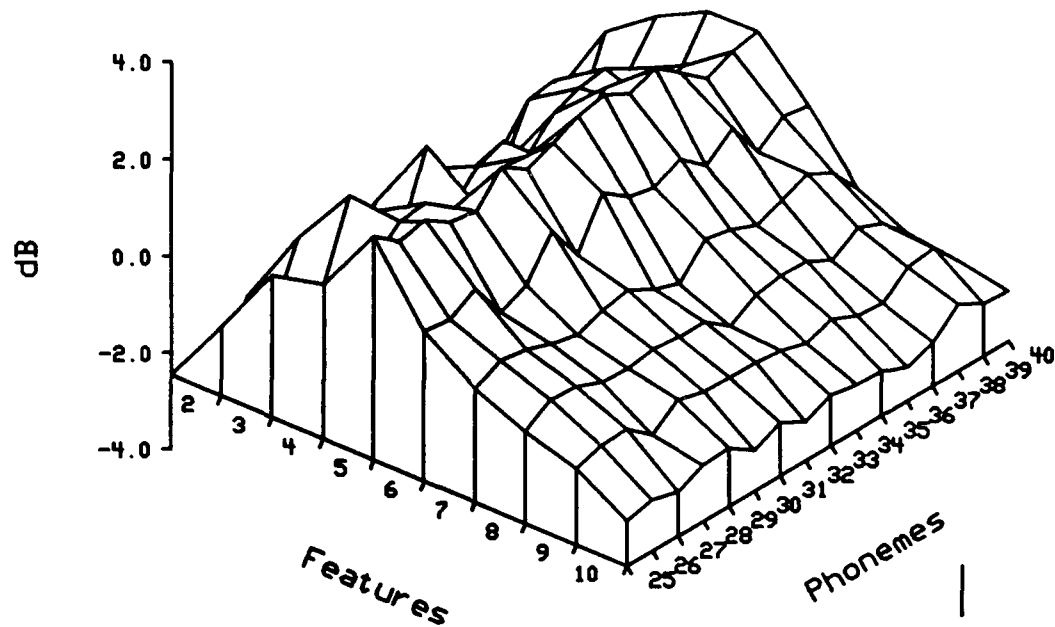


Figure 3. Differences in energy from normal to loud for the vowels of all eight speakers

abnormal conditions. When viewing these energy distributions averaged across all eight speakers, there are interesting qualities that can be noted. The most prominent characteristic was observed in the sonorants. There was a significant decrease of energy in the 0-500Hz and the 4k-8kHz ranges with a corresponding increase in the 500-4kHz range. This concentration of energy in the mid bands at the expense of the low and high bands could be viewed as an energy migration directly correlated to the change in vocal effort. This phenomenon can be easily seen by referring to Figures 3 and 4. These figures illustrate the change in energies from normal to abnormal for the vowels. Note that the feature indices refer to those listed in Table 3, and the phoneme indices refer to those listed in Table 4. The average loss of energy in band 1 (0-250Hz) was 2.41 dB for loud speech and 1.23 dB for Lombard speech; for band 10 (7k-8kHz) the average loss was 1.45 dB for loud speech and 1.36 dB

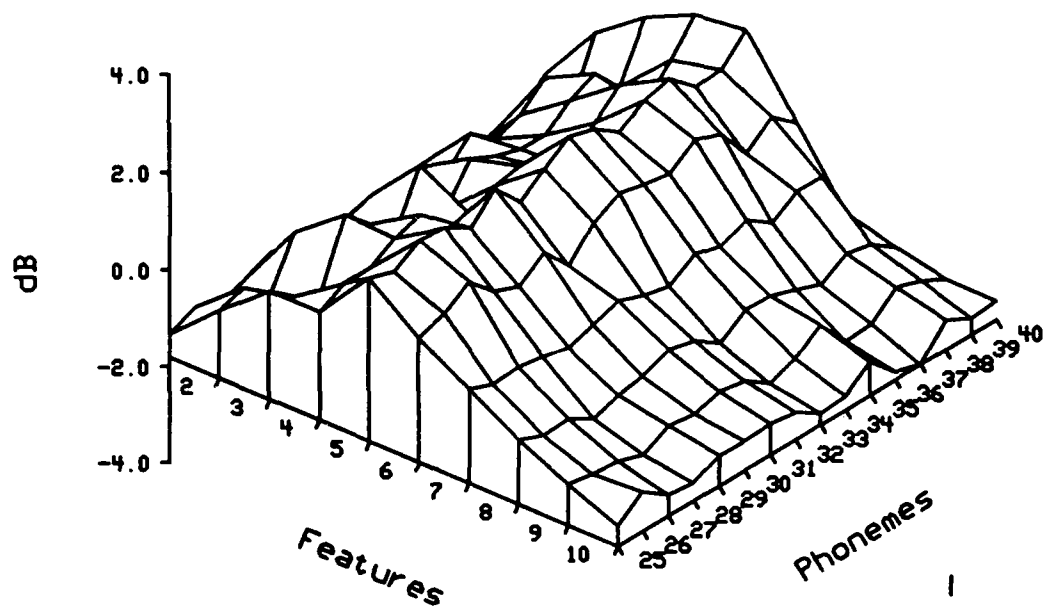


Figure 4. Differences in energy from normal to Lombard for the vowels of all eight speakers

for Lombard speech. For both loud and Lombard speech, the largest increases were in band 5 (2k-3kHz) and ranged from 1.3 dB to 2.3 dB across the vowels. Refer to the tables in Appendix L for a complete breakdown of the values.

When comparing these overall results to those of individual speakers, the energy migration trend was fairly consistent for the loud speech of all eight subjects. On the other hand, the Lombard speech exhibited more variability across speakers, attributable to each speaker's perceived premium on intelligibility [La71]. For speakers #2, #5, and #6, there were increases in the energy of the lower frequency bands. Figure 5 illustrates how the Lombard speech of speaker #2 exhibited an increase in energy in the 0-4kHz range, and a decrease in energy in the 4k-8kHz range. It can also be seen from Figure 5 that the changes were more monotonic in nature rather than the more characteristic maximum in the 2k-3kHz band observed for the other sessions.

In Figure 6, the energy in the Lombard speech for speaker #5 increased in the 0-250Hz and 1k-5KHz ranges while decreasing in the 250-1kHz and 5kH-8kHz ranges. This is likely attributable to smaller bandwidths in the formants. In this case, LPC should predict poles closer to the unit circle, and this proximity of poles to the unit circle sharpens the peaks and deepens the troughs of the LPC spectrum. In Figure 7, there is a general lack of consistency in the Lombard speech of speaker #6 although there are trends similar to speaker #5.

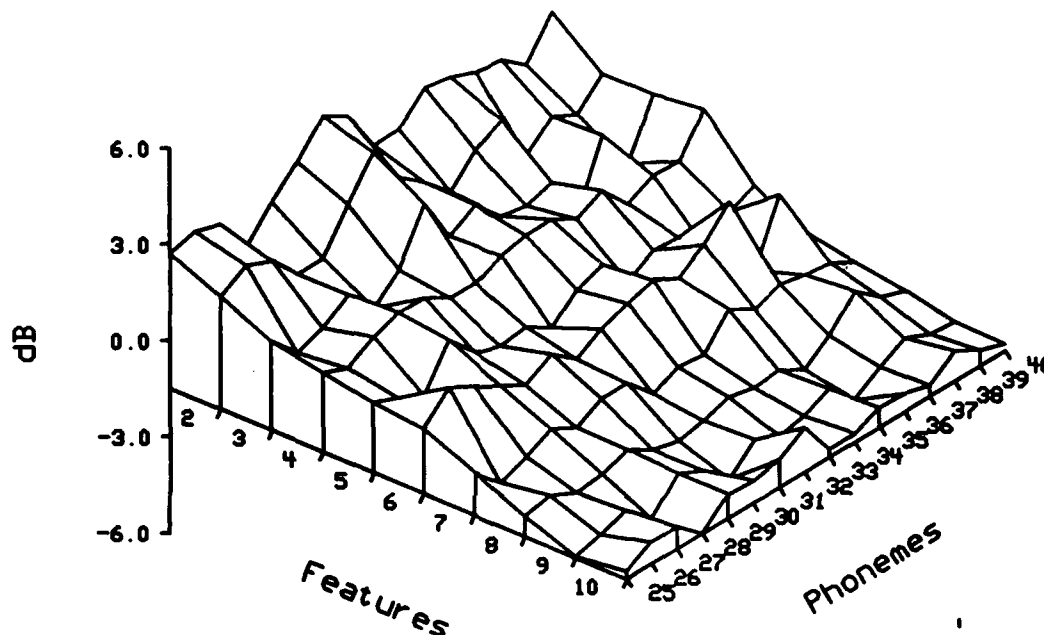


Figure 5. Differences in energy from normal to Lombard for the vowels of speaker 2

Although not as dramatic, there were some overall trends observed in the voiceless fricatives. Note in Tables 5 and 6 that the energy migration was from the lower frequency bands to the upper frequency bands with the crossover around 4kHz. Losses in the 0-250Hz range were as low as 2 dB and gains in the 7k-8kHz range were just over 1 dB. This trend reflects the added

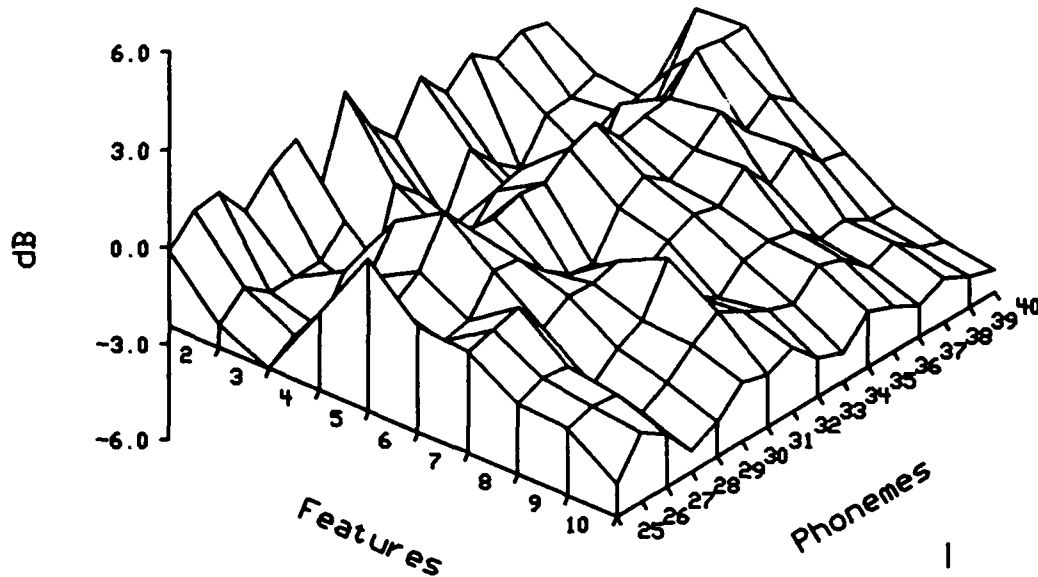


Figure 6. Differences in energy from normal to Lombard for the vowels of speaker 5

high frequency noise generated when there is an increase in airflow at the point of constriction in the vocal tract.

7.3.2 Spectral Energy Attributes

The spectral center of gravity, *COG*, and spectral tilt provide additional ways of viewing the changes that occur in the energy distribution across frequency. Table 7 summarizes these shifts for loud and Lombard speech across the eight speakers. Note that for loud speech, there is a tendency for *COG* to shift upward, with the most dramatic shifts understandably being in the voiceless fricatives, /S/, /CH/, and /SH/. For Lombard speech, *COG* has a much smaller upward shift, especially in the vowels, which is due to the differing responses of speakers as discussed in previous sections. The voiceless fricatives, however, are more consistent with the shifts in loud speech. Spectral tilt was assessed individually in both the 0-3kHz range and the 3k-

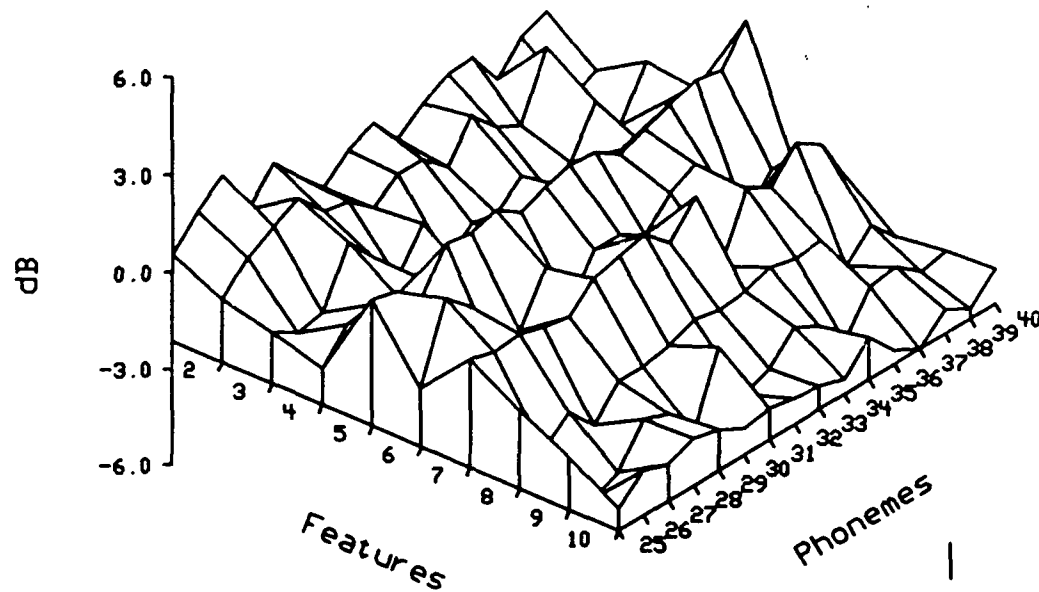


Figure 7. Differences in energy from normal to Lombard for the vowels of speaker 6

Table 5. Changes in energy (dB) from normal to loud speech across all eight speakers

Phonemes	Energy in Frequency Bands (kHz)									
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8
S	-0.7	-0.3	-0.3	-0.2	-0.2	-1.6	0.0	0.1	1.0	1.2
CH	-1.2	-0.6	-0.4	-0.8	-0.8	-0.7	1.0	0.6	0.6	0.6
TH	-1.5	-0.4	-0.4	0.0	0.6	-0.1	0.1	0.1	0.1	-0.1
F	-0.8	0.0	-0.2	-0.5	0.0	-0.3	0.3	0.4	0.4	0.0
SH	-2.1	-1.8	-1.8	-1.9	-0.9	0.2	1.7	1.4	0.7	0.6

8kHz range. The general trend for sonorants was for tilt to increase in the low band and decrease in the high band. Increase in the low band was about 1 dB/octave, and decrease in the high band was about 2 dB/octave. The other phenomenon of interest is the increase of almost 3 dB/octave in the high band

Table 6. Changes in energy (dB) from normal to Lombard speech across all eight speakers

Phonemes	Energy in Frequency Bands (kHz)									
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8
S	-0.7	-0.3	-0.1	-0.4	-0.6	-1.8	0.6	0.2	1.0	1.3
CH	-0.9	-0.4	-0.4	-1.0	-0.7	-0.6	1.2	0.6	0.6	0.5
TH	-0.5	-0.3	-0.4	-0.1	0.0	-0.5	0.4	0.3	0.3	-0.1
F	-0.4	0.0	0.2	-0.2	0.1	-0.6	0.3	0.3	0.2	-0.2
SH	-1.9	-1.3	-0.9	-1.8	-0.8	0.1	1.6	1.1	0.7	0.4

for the phonemes /S/ and /Z/ in both loud and Lombard speech.

7.3.3 Pitch

As expected, pitch had the most reliable shift, increasing for all voiced phonemes in both the loud and Lombard conditions. These results are summarized for all eight speakers in Table 8. For 31 phonemes, the overall average increase in pitch was 50 Hz for loud speech and 30 Hz for Lombard speech. These increases follow naturally from the increase in vocal effort.

7.3.4 Formants

Formant behavior exhibited a wide range of variability across speakers. This is best illustrated by sample formant trajectories from normal to loud to Lombard of speakers #3, #6, and #7. Figures 8, 9, and 10 plot the average values of the first and second formants with the individual points labelled by speech condition and phoneme. The speech condition is indicated with the conventional indices: 1=normal, 2=loud, and 3=Lombard. Phonemes are indicated by ARPABET symbols. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech. Note in Figure 8 that the first formant of speaker #3 increases moderately while the second formant decreases for most vowels. Speaker #6 in Figure 9 has moderate increases in the first formant with a mixture of increases and decreases in the second formant. And in Figure 10, speaker #7 tends to show more consistent increases in the second formant. A complete set of formant trajectory graphs is included in Appendix L. Table 9 shows overall averages in formants across the eight speakers. The most consistent result is the increase in the first formant frequency, up an average 45 Hz for loud speech and 35 Hz for Lombard speech. The second and third formants are much less reliable, tending to increase for some speaker/vowel combinations and decrease for others.

Table 7. Changes in center of gravity and spectral tilt from normal to loud and Lombard speech across all eight speakers

Phonemes	Normal to Loud			Normal to Lombard		
	COG (Hz)	Tilt (dB/octave)		COG (Hz)	Tilt (dB/octave)	
		Lo	Hi		Lo	Hi
P	140	0.4	-1.9	154	0.7	-3.4
T	176	0.2	0.3	190	0.0	0.6
K	-41	0.2	-0.4	3	-0.1	-0.4
B	-16	-0.2	-0.1	-80	-0.4	0.1
D	14	0.0	0.5	30	-0.3	0.2
G	-23	0.2	0.3	-10	0.1	0.0
DX	-47	0.6	0.1	-83	0.3	0.0
M	57	0.2	-0.4	-47	-0.1	-0.2
N	1	0.4	0.1	-96	0.1	0.2
NX	-14	0.4	0.1	118	0.0	0.0
S	165	0.1	2.8	204	-0.1	2.7
Z	49	-0.3	2.8	122	-0.4	2.8
CH	171	0.2	0.9	180	0.2	0.8
TH	42	0.4	-0.4	64	0.2	-0.2
F	44	0.1	0.4	29	0.0	0.3
SH	282	0.1	-0.4	224	0.2	0.3
JH	101	0.1	-0.1	105	0.3	0.0
V	13	0.6	-0.4	-31	0.1	-0.3
L	32	0.5	-1.6	3	0.6	-1.2
R	30	0.8	-1.1	35	0.9	-1.7
Y	29	0.4	-1.2	-56	0.2	-0.6
HH	31	0.0	0.3	2	-0.3	0.2
EL	49	0.6	-1.8	-13	0.5	-1.1
W	-32	0.1	-0.6	-63	0.0	-0.8
EH	113	1.2	-1.5	86	1.0	-1.8
AO	121	1.1	-1.6	41	0.6	-1.6
AA	160	1.1	-2.1	92	0.8	-2.1
UW	31	0.6	-1.6	3	0.7	-1.7
ER	93	0.8	-1.2	31	0.6	-1.4
AY	143	1.1	-1.9	56	0.8	-1.8
EY	70	1.2	-2.3	34	0.7	-1.5
AW	147	1.0	-1.3	72	0.8	-1.4
AX	76	1.3	-1.8	41	1.2	-2.2
IH	114	1.5	-2.3	84	1.1	-2.2
AE	130	1.1	-1.8	54	0.7	-1.8
AH	119	1.2	-1.8	93	0.8	-2.5
OY	95	1.0	-2.2	74	0.8	-2.4
IY	47	0.8	-1.8	16	0.6	-1.7
OW	119	0.9	-1.7	87	0.7	-1.9
AXR	9	0.7	-0.9	-27	0.9	-1.0

7.3.5 Duration

The average durations are summarized in Table 10. For loud speech there is a fairly clear distinction between continuant and obstruent phonemes. The continuants tend to increase in duration with the vowels lengthening an average 18 msec. The obstruents such as stops and fricatives tend to become

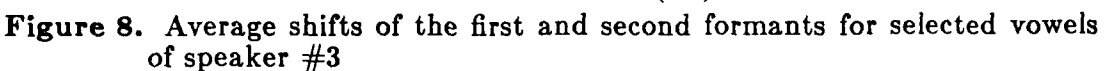
Table 8. Changes in pitch from normal to loud and Lombard speech across all eight speakers

Phonemes	Pitch changes from Normal (Hz)	
	Loud	Lombard
B	47	34
D	34	24
G	18	13
DX	60	36
M	51	31
N	52	29
NX	41	29
Z	37	25
V	48	31
L	54	32
R	61	31
Y	45	29
HH	31	33
EL	43	27
W	45	25
EH	58	33
AO	55	28
AA	64	34
UW	54	31
ER	63	35
AY	55	31
EY	57	31
AW	55	32
AX	55	34
IH	63	30
AE	56	33
AH	62	31
OY	53	32
IY	56	36
OW	58	33
AXR	32	18

shorter. Again, the effect is not as pronounced with Lombard speech, owing to the increased variation in reactions among speakers to noise injection in the ear.

7.4 Summary

With the enormous amount of information represented by the features calculated for each phoneme, it is helpful to view a summary of the changes from a qualitative standpoint. Table 11 attempts to display all the *significant* changes in the phonemes and features for one speaker. This table was derived using analysis of variance [Do74, Bl80] and setting the level of significance to 0.01. Symbols are printed in the table only if the changes from normal to loud or Lombard were found to be significant at this stringent level. There is a



logical key to understanding the symbology. For any given symbol, the left side refers to the comparison of loud to normal speech, and the right side refers to the comparison of Lombard to normal speech. If a side of the symbol is open (white), then the feature for that abnormal condition was significantly higher than normal. Filled (black), on the other hand, means that the feature for that abnormal condition was significantly lower than normal. With this interpretation, the table clearly depicts the energy migration in the vowels from the lower and higher frequency bands into the 500-4kHz range for both loud and Lombard speech for speaker #1. Significance tables for all eight speakers are included in Appendix L. In addition, Table 12 illustrates common significances in the vowel features of speakers #1 and #3, and Table 13 shows common significances in the vowel features for five out of the eight speakers.

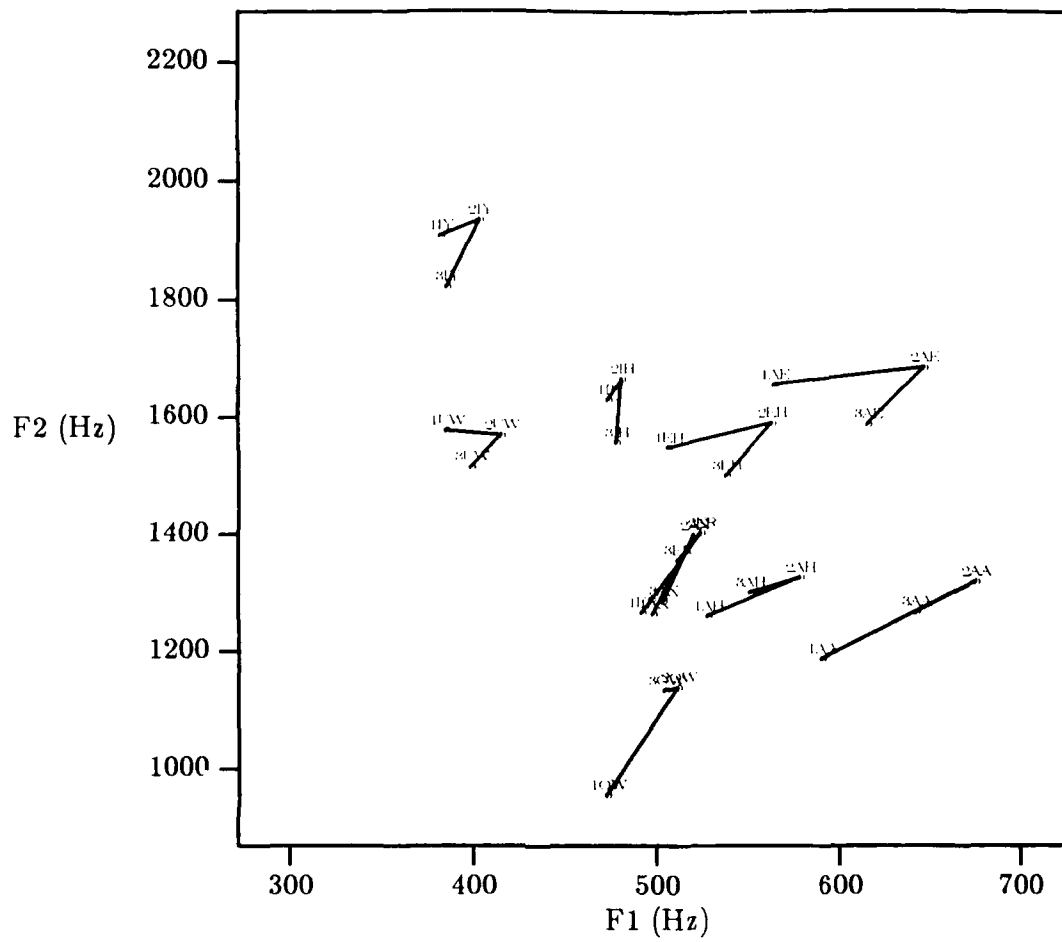


Figure 9. Average shifts of the first and second formants for selected vowels of speaker #6

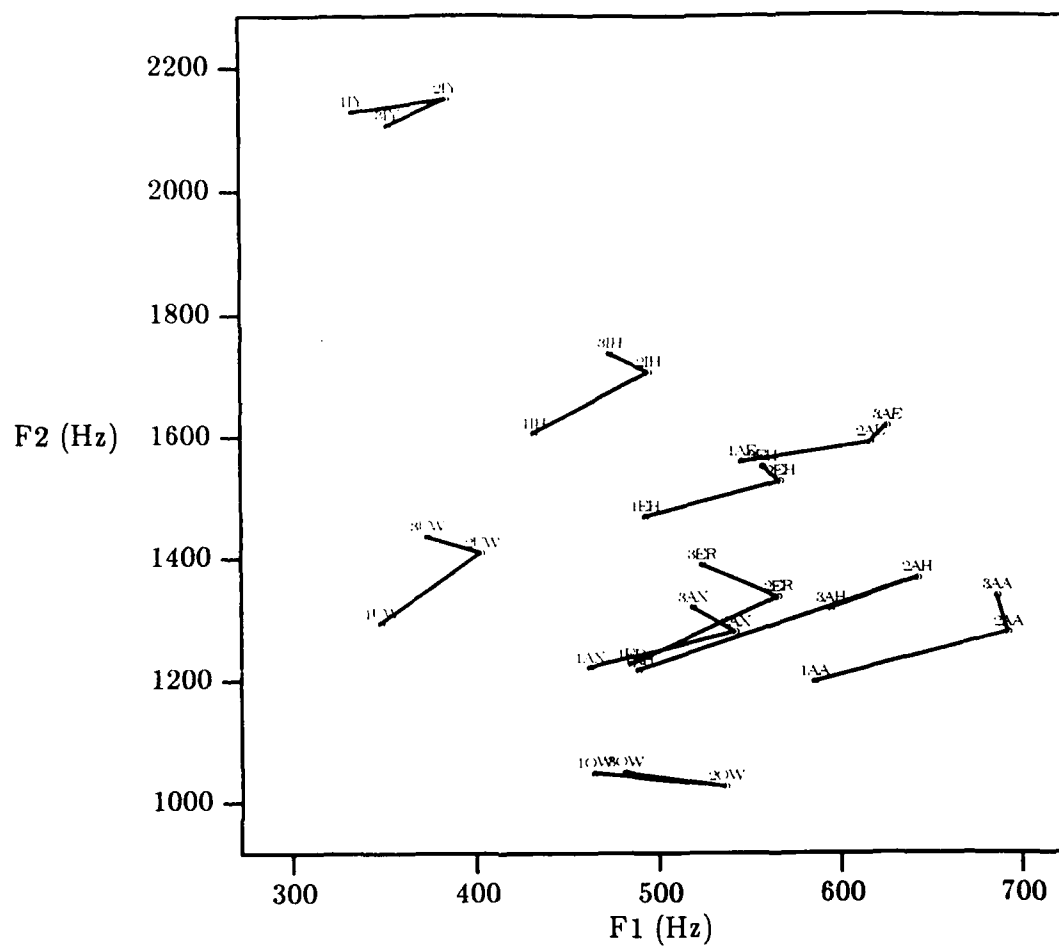


Figure 10. Average shifts of the first and second formants for selected vowels of speaker #7

Table 9. Changes in formant frequencies from normal to loud and Lombard speech across all eight speakers (Hz)

Phonemes	Normal to Loud Formants			Normal to Lombard Formants		
	1	2	3	1	2	3
EH	50	39	26	34	-41	-10
AO	67	-11	62	53	23	47
AA	59	37	12	45	24	-23
UW	31	6	-21	26	-18	-19
ER	47	74	-93	29	29	-71
AY	65	21	57	48	9	40
EY	37	12	-40	33	-32	-121
AW	62	45	49	43	41	11
AX	25	45	48	21	-34	40
IH	28	56	23	21	-23	-33
AE	56	40	29	39	-9	30
AH	67	50	0	51	37	-45
OY	42	97	-4	37	59	-109
IY	18	24	-76	23	-56	-241
OW	54	55	84	46	70	27
AXR	16	-29	-206	15	-55	-227

Table 10. Changes in duration from normal to loud and Lombard speech across all eight speakers

Phonemes	Duration changes from Normal (sec)	
	Loud	Lombard
P	-0.012	-0.013
T	-0.005	-0.002
K	-0.009	-0.009
B	-0.002	0.002
D	0.000	0.001
G	-0.004	0.001
DX	-0.003	-0.001
M	-0.005	-0.003
N	-0.003	0.000
NX	-0.015	0.001
S	-0.008	0.003
Z	-0.006	-0.006
CH	-0.012	-0.006
TH	-0.014	-0.008
F	-0.014	-0.007
SH	-0.008	-0.003
JH	-0.010	0.013
V	-0.002	0.000
L	0.002	0.002
R	0.006	0.005
Y	0.013	0.005
HH	-0.009	-0.007
EL	0.008	0.013
W	0.017	0.004
EH	0.021	0.018
AO	0.024	0.015
AA	0.023	0.021
UW	0.022	0.006
ER	0.007	0.005
AY	0.020	0.016
EY	0.026	0.025
AW	0.022	0.014
AX	0.001	0.002
IH	0.011	0.009
AE	0.024	0.022
AH	0.020	0.011
OY	0.019	0.019
IY	0.008	0.011
OW	0.014	0.013
AXR	0.034	0.024

Table 11. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #1. Level of significance: 0.01

KEY:																		
Δ indicates feature was higher than normal for both loud and Lombard																		
▼ indicates feature was lower than normal for both loud and Lombard																		
● indicates feature was higher for loud and lower for Lombard																		
■ indicates feature was lower for loud and higher for Lombard																		
Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Tilt-h	Formants			Dur
	0-25	25-50	50-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P		●				■							●					▼
T	▼			▼	▼	●		Δ	Δ	Δ	Δ					●	Δ	▼
K							■								▼			▼
B		Δ																
D																		
G																		
DX				Δ		▼	▼				▼						▼	
M	▼	Δ			Δ				▼	▼		Δ		Δ	Δ			
N		Δ	Δ			▼			▼			Δ		Δ				
NX		Δ												Δ				
S		▼	▼	▼	▼	▼	Δ	Δ	Δ	Δ	Δ	▼	Δ	Δ	Δ	Δ	Δ	
Z					▼	▼							Δ					
CH	▼	▼	▼	▼	▼	Δ	Δ	Δ			Δ					▼		
TH							Δ				Δ							
F									●			●						
SH			▼	▼		Δ	Δ	Δ									Δ	
JH				▼	●	Δ	Δ		■								Δ	
V				Δ						▼		▼	Δ					
L	▼				Δ	■			▼	▼		Δ	▼	Δ				
R	▼	▼	Δ	Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
Y	▼				Δ		■		▼	▼		Δ	▼	Δ				Δ
HH						▼												
EL					Δ	■												
W	▼		Δ									Δ		Δ				
EH	▼	▼	Δ	Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		Δ
AO	▼	▼		Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ			Δ
AA	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			Δ
UW	▼				Δ	Δ	■	▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
ER	▼		Δ	Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
AY	▼	▼	Δ	Δ	Δ	■	▼	▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
EY	▼		Δ	Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
AW	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
AX	▼		Δ	Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ				
IH	▼			Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ				Δ
AE	▼	▼		Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		Δ
AH	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
OY	▼		Δ	Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ	▼	▼	Δ
OY	▼	▼		Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
IY	▼	▼		Δ	Δ	Δ	Δ	▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ	▼	
OW	▼	▼			Δ	Δ	Δ	▼	▼	▼	Δ	Δ	▼	Δ	Δ	Δ		
AXR	▼		Δ	Δ	Δ	■	▼	▼	▼	▼	Δ	Δ	▼				▼	

Table 12. Shifts in phoneme features that are common to speakers #1 and #3, level of significance: 0.01

Phonemes	Energy in Frequency Bands (kHz)										CXX	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
EH	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼	Δ				
AO	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼					
AA	▼			Δ	Δ			▼	▼	▼		Δ	▼	Δ				
UW	▼				Δ	Δ		▼	▼	▼		Δ	▼	Δ	Δ			
ER	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼	Δ				
AY	▼		Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ				
EY	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼	Δ	Δ			
AW	▼			Δ	Δ	Δ		▼	▼	▼		Δ	▼	Δ				
AX	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼					
IH	▼			Δ	Δ			▼	▼	▼		Δ	▼					
AE	▼			Δ	Δ			▼	▼	▼		Δ	▼	Δ				
AH	▼			Δ	Δ			▼	▼	▼		Δ	▼	Δ				
OY	▼		Δ	Δ	Δ			▼	▼	▼		Δ	▼	Δ		▼		
IY	▼			Δ	Δ			▼	▼	▼		Δ	▼	Δ				
OW	▼				Δ	Δ		▼	▼	▼		Δ	▼	Δ	Δ			
AXR	▼		Δ	Δ	Δ	■		▼	▼	▼							▼	

Table 13. Shifts in phoneme features that are common to speakers #1, #3, #6, #7, and #8, level of significance: 0.01

Phonemes	Energy in Frequency Bands (kHz)										CXX	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
EH					Δ					▼				Δ				
AO					Δ					▼								
AA					Δ					▼				Δ				
UW														Δ				
ER														Δ				
AY						Δ				▼				Δ				
EY										▼				Δ				
AW						Δ				▼				Δ				
AX						Δ				▼								
IH	▼				Δ					▼		Δ	▼					
AE					Δ					▼				Δ				
AH					Δ					▼				Δ				
OY										▼				Δ				
IY										▼				Δ				
OW						Δ				▼				Δ	Δ			
AXR					Δ					▼								

8. BASELINE RECOGNITION SYSTEM

The primary purpose of the recognition system for this research was to assess the effectiveness of signal processing techniques on loud and Lombard speech, given the fact that all training was performed with normal speech. Rather than being designed as an end product, the recognition system was intended to simulate the front end of a more complete speech understanding system. In this light, hand-labelled phonemes from the analyses of Chapter 7 were the basic units of recognition in a conventional template-matching algorithm utilizing dynamic time warping. This recognition system employed no higher knowledge sources and thus yielded results that directly reflected the performance effectiveness of the acoustic processing under test.

8.1 System Description

The operation of the baseline system is illustrated in Figure 11. The basic units of recognition were hand-labelled phoneme tokens with the lexicon consisting of the 40 most common English phonemes in the AAMRL database, as listed in Appendix G. For each speaker and condition, templates were produced for N_T tokens of each of the 40 phonemes in the following manner. Each phoneme token was divided into 50 overlapping 16msec frames, as described in Section 7.2. Each frame was multiplied by a Hamming window, and then a 128-point log magnitude spectrum was computed from the 24th order LPC. The templates were then divided into reference and test categories based on the token occurrence number. In a *leaving-one-out* method [Fu72], one token of each phoneme was reserved as the unknown or *test* token while the other $(N_T - 1)$ tokens of the phoneme were held as *reference* tokens. Each test template was then compared to $40 \times (N_T - 1)$ reference tokens using a dynamic time warp with symmetric weighting and slope constraint condition, $P=1$, according to Sakoe and Chiba [Sa78]. A recognition experiment on one session (i.e. one speaker and one condition) then consisted of testing $40 \times N_T$ tokens. When testing abnormal speech against normal reference templates, the same scheme was used in order to standardize the handling of data even though there

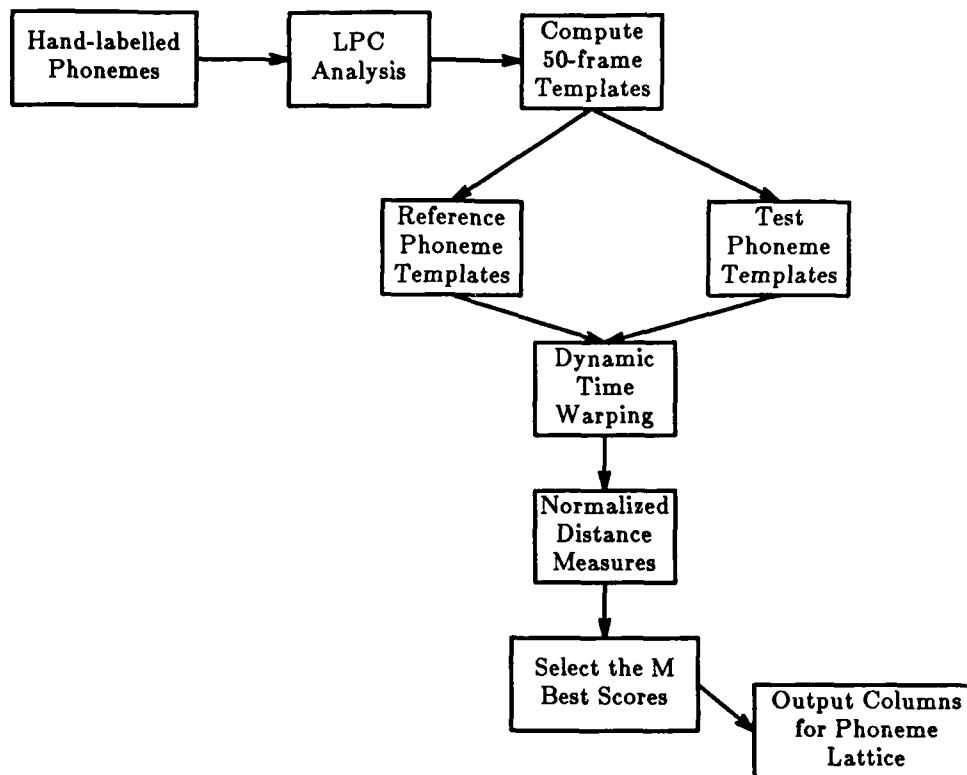


Figure 11. Baseline recognizer diagram

was no explicit need to use the leaving-one-out method.

The number of tokens, N_T , to use in recognition experiments was determined by preliminary testing on speaker #1. The number of reference tokens for recognizing normal, loud, and Lombard speech was varied between one and seven. In this case, it was found that increasing the reference tokens beyond five produced negligible changes in the performance of the baseline system. Thus $N_T = 6$ was chosen for this research, giving five reference templates for all experiments.

8.2 Recognition Assessment

The recognition output of a single test template was obtained in the following way. The warping of test template i to reference template j produced a normalized distance measure, D_{ij} , where $D_{ii} = 0$. The set $\{D_{ij}\}_{j=1}^{40 \times (N_T - 1)}$ was then sorted in ascending order. The M best scores were then selected to form a *phoneme candidate vector*, $\bar{p}_i$. Recognition was successful if any reference phoneme in $\bar{p}_i$ matched the test phoneme. This type of measure captured only

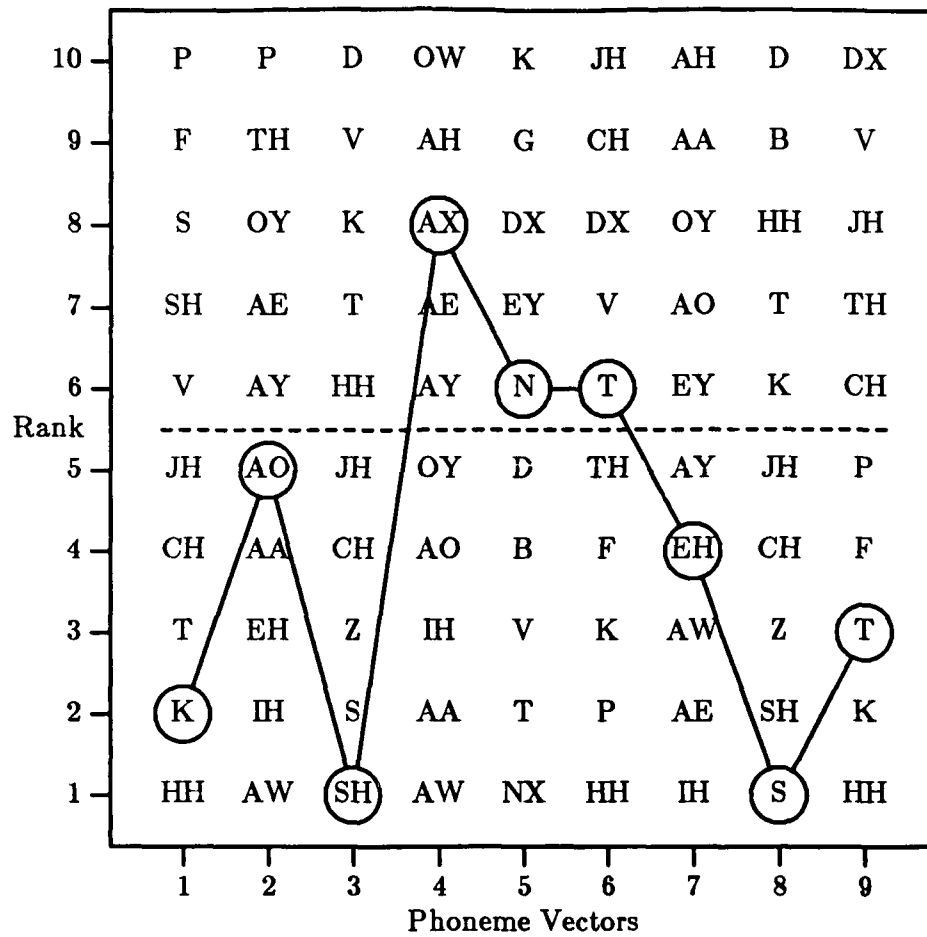


Figure 12. Phoneme lattice for utterance **caution test**

the success of the front-end processing, as indeed it should have. For the hypothetical speech understanding system, the vectors, $\bar{p}_i$, would have then been combined as columns to a phoneme lattice, as illustrated in Figure 12. The circled phonemes depict the utterance, **caution test**. To better visualize the assessment of performance of the baseline recognition system, note the horizontal dashed line that serves as a boundary between rank 5 and 6. With the dashed line positioned at $M = 5$ for $\bar{p}_i$, the recognition performance would be calculated as $\frac{6}{9}$, or 67%. Since performance is dependent on the choice of M , it gives a more complete picture of performance if the percentage is graphed as a function of M as in Figure 13. Traveling from left to right on the x-axis of Figure 13 is equivalent to the dashed boundary shifting upward in Figure 12. In theory, M would be fixed at some value that rendered a reasonable probability

of the correct phoneme being included while limiting the size of the lattice for processing by higher knowledge sources. For this research, however, a more complete evaluation of recognition performance is provided by considering the range $1 \leq M \leq 10$.

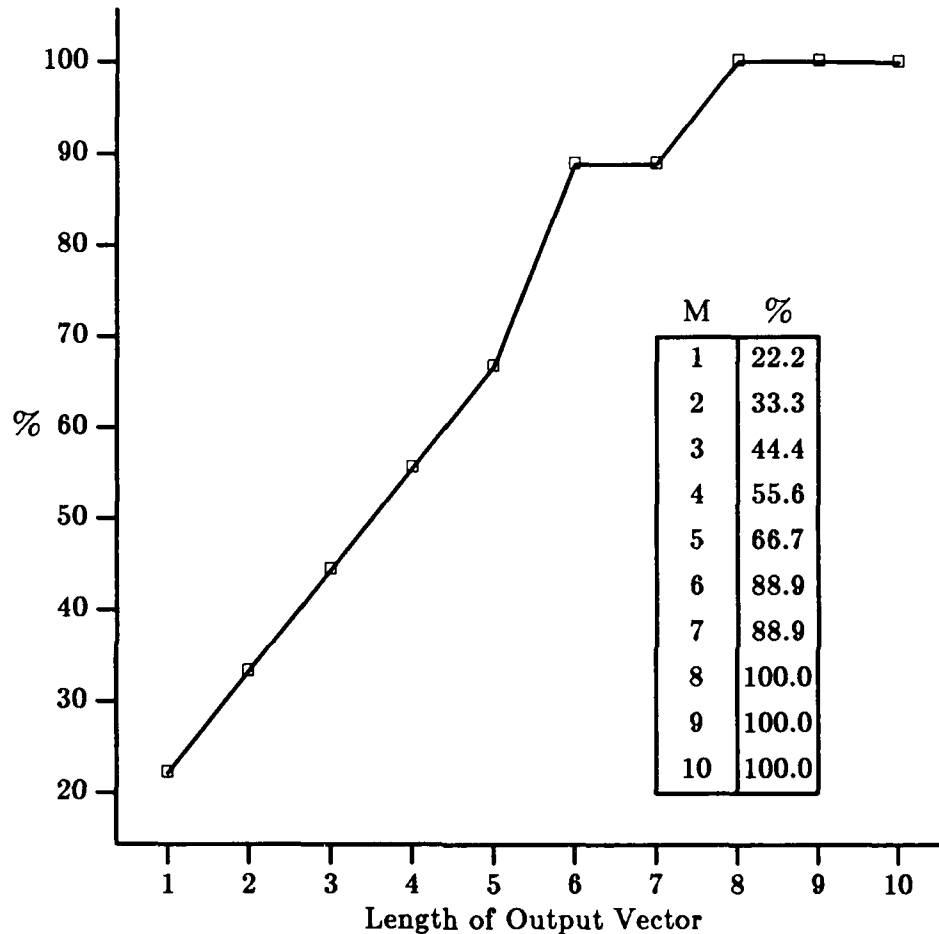


Figure 13. Baseline recognizer performance for utterance caution test

8.3 Baseline Metric

A standard Euclidean measure was used to determine the distance between the individual frames of different templates. Recall first that an individual frame in a template consisted of samples of the LPC log magnitude spectrum. If A_i , as defined by Equation (3) is one such sample, then $\{A_{lmi}\}_{l=0}^{N-1}$ comprises frame m of template i . Now if the test data is indicated by a prime, then the distance between frame m of test phoneme token i and frame k of reference phoneme token j is calculated as

$$d_{mkij} = \left\{ \sum_{l=0}^{\frac{N-1}{2}} (A'_{lmi} - A_{lkj})^2 \right\}^{\frac{1}{2}} \quad (10)$$

The set $\{d_{mkij}\}$, where the ranges for m and k are determined by the slope and boundary constraints of the dynamic time warping algorithm, is then used to derive the total normalized distance, D_{ij} .

8.4 Baseline Performance

For each speaker, the baseline recognizer was trained by loading reference templates of normal speech, and then tested by performing recognition of normal, loud, and Lombard speech. The overall results of each speaker are graphed individually in Appendix M. In addition, a reduced version is contained in Figure 14, with the miniature graphs from left to right, top to bottom, referring to speakers #1 through #8 respectively. For each speaker, the top curve represents recognition performance of normal speech, and the lower curves represent abnormal speech. The *gaps* between the normal and abnormal curves are clear examples of the degradation in performance due to the differences in the normal reference phonemes and the abnormal test phonemes. The ultimate goal of this research, in essence, then is to reduce the size of the gaps between the normal and abnormal performance curves while minimizing the degradation in overall performance.

The degree of variability among speakers is easily seen by comparing the graphs in Figure 14. Speaker #3 clearly exhibited the most degradation from normal to abnormal speech, with Lombard speech also being quite distinct from loud speech. Speaker #6 also caused significant degradation in the baseline system, but with very little distinction between loud and Lombard. Speaker #4 displayed a large gap between normal and loud speech, but very little difference between normal and Lombard. In fact, the noise injection into the ears of speaker #4 caused the least noticeable differences in his overall speech patterns as compared to the other speakers. The remaining speakers had similar patterns on the baseline system where both loud and Lombard performance was clearly separated from normal performance, but with very little performance distinction between loud and Lombard.

Figure 15 shows a different breakdown of the baseline recognizer performance. In this series of graphs, the individual results of all eight speakers have been combined, and then separated according to phoneme category.

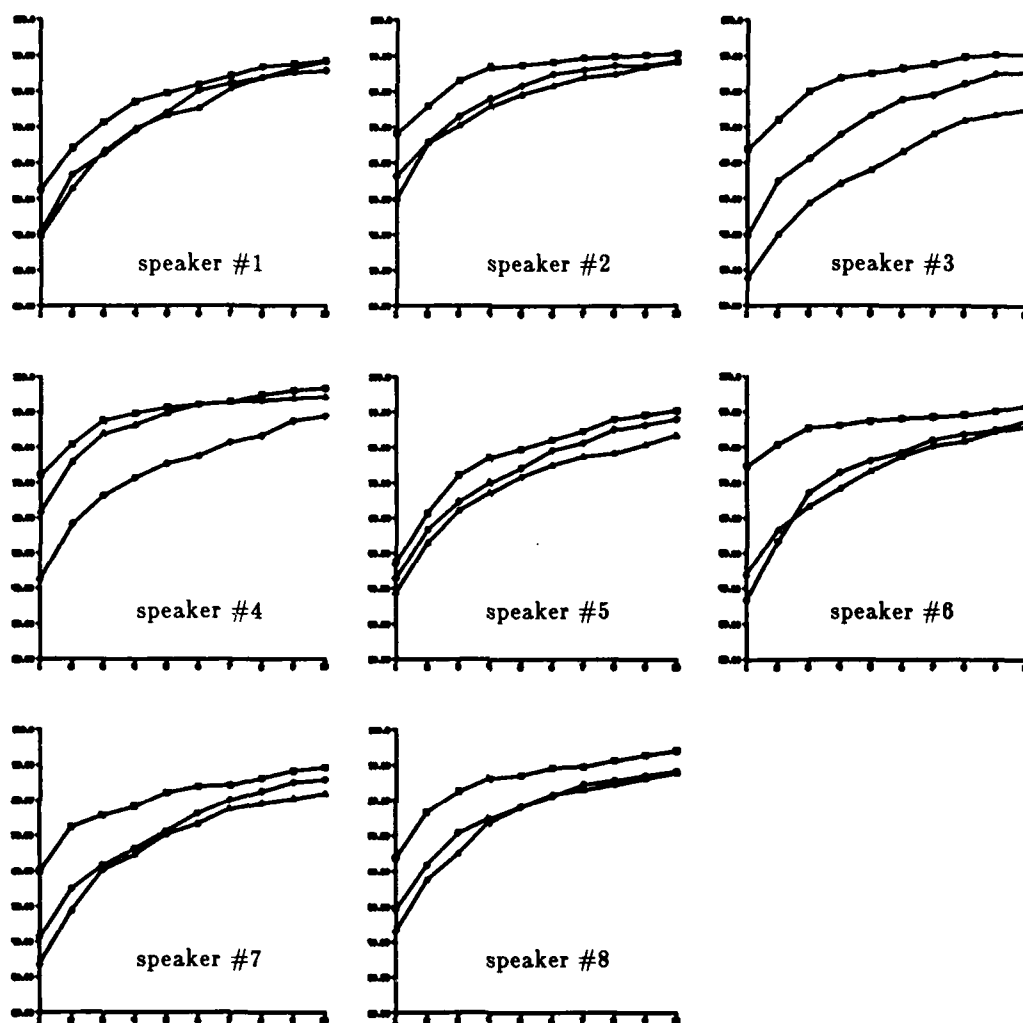


Figure 14. Baseline recognition system performance on all eight speakers

Starting with the graph in the upper left corner and moving right, the performance for all phonemes is depicted followed by the performance of stops and nasals. The second row graphs show the results for fricatives, liquids, and vowels. These graphs are also printed individually in Appendix M. Note the similarity of performance in vowels and fricatives. There is an obvious gap between normal and abnormal speech, but minimal distinction between loud and Lombard speech. Since the vowels and fricatives combined represent 24 out of the 40 phonemes, these results naturally dominate the overall performance for all phonemes in the upper left graph of Figure 15. The most distinction between loud and Lombard speech occurs in the stops and nasals. Note

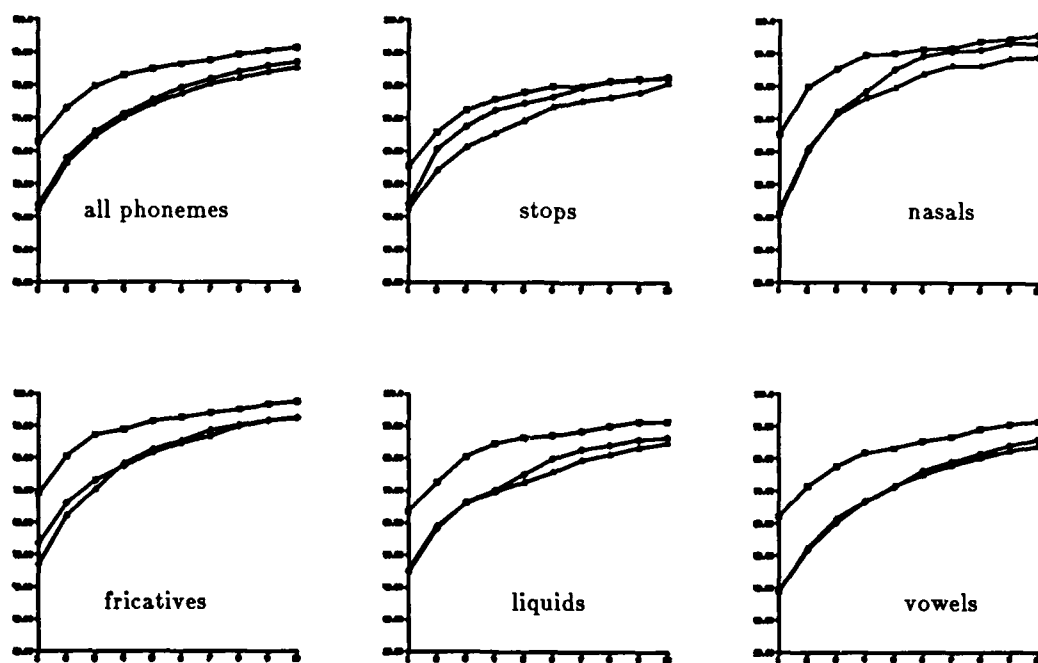


Figure 15. Baseline recognition system performance all eight speakers averaged, and broken down by phoneme category

particularly that for nasals, performance for loud and Lombard is essentially indistinguishable for $M \leq 3$, and then separates for $M \geq 4$. Note also that for the stops, there is a merge of normal and loud performance for $M \geq 7$.

8.5 Figure of Merit

A simple and effective way of quantifying the overall differences in recognition performance is necessary to be able to accurately measure and compare curves such as those in Figures 14 and 15. This can be accomplished by designating the percentage correct (vertical axis) as a score, such that the score for a particular phoneme vector length, M , would be S_M . Now define a figure of merit, F_{sc} , for speaker s and speech condition c as the average of these scores.

$$F_{sc} = \frac{1}{10} \sum_{M=1}^{10} S_M \quad (11)$$

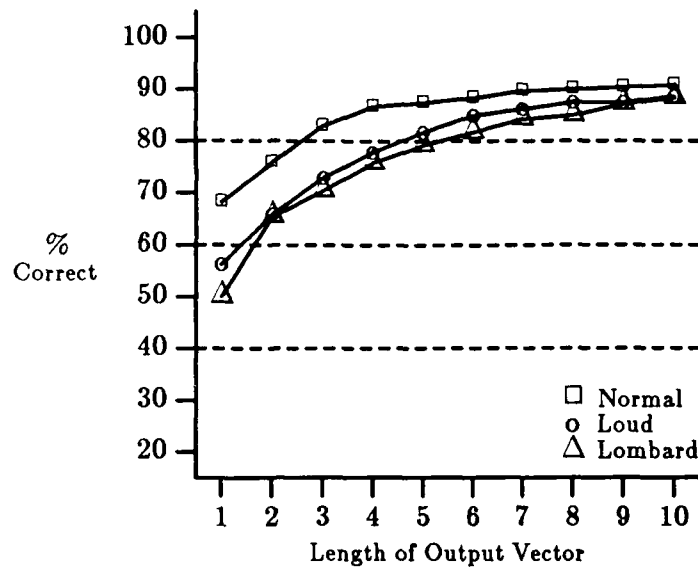


Figure 16. Recognition performance for baseline system, speaker #2

For example, recall that the actual points of the performance curve in Figure 13 are listed in the inset table. By averaging these percentages, we obtain $F = 70.00$. To gain an appreciation for the range of F , note that if all correct phonemes appeared in the number one position of the phoneme candidate vector, $\bar{p}_i$, for a given experiment, then this would give a perfect score of 100 for F . Conversely, if there were *no* correct phonemes in any of the positions one through ten of $\bar{p}_i$, then this would result in $F = 0$. As can be seen, F can be used to quantify the area under the performance curve, such as the example in Figure 13, and can be easily used to measure the gaps between the performance curves for normal and abnormal speech. For example, Figure 16 graphs the performance curves of the baseline recognizer for the normal, loud, and Lombard speech of speaker #2. (The horizontal dashed lines are included in the graph for the convenience of the reader.) The figures of merit for the various speech conditions¹ in Figure 16 are

$$F_{21} = 85.03$$

$$F_{22} = 78.95$$

1. Throughout this research, the three speech conditions are indexed as 1 = normal speech, 2 = loud speech, and 3 = Lombard speech.

$$F_{23} = 76.71$$

The gap in performance between loud and normal speech can then be quantified as

$$F_{22} - F_{21} = -6.08$$

and the gap between Lombard and normal as

$$F_{23} - F_{21} = -8.32$$

The sign convention for the values of the gaps (i.e. the area between normal and abnormal curves) preserves the intuitive notion that the performance for loud and Lombard speech was worse than the performance for normal speech. The figure of merit, F , then provides a concise representation of performance curves and faithfully preserves the relationship between curves. It is important that the reader establish the relationship between the performance curves and values of F . Especially important is the ability to relate differences in F values to the size of the gap between performance curves.

Because of its ability to provide an overall comparison of relative merit, F will be used extensively to assess the effectiveness of algorithms in Chapter 9. For reference, Table 14 provides the figures of merit of the baseline recognizer for all 24 sessions, where the tens digit of the session number designates the speaker, and the units digit designates the speech condition (i.e. 1 = normal, 2 = loud, and 3 = Lombard). Listed also are the gap measures between normal and abnormal speech for each of the speakers. Note that the values in Table 14 are derived from the curves in Figure 14.

8.6 Summary

A baseline recognition system was developed to assess the relative performance of normal, loud, and Lombard speech. Training was restricted to normal speech. Template matching of phonemes was accomplished with dynamic time warping methods. Five reference templates were stored for each phoneme, and Euclidean measure between LPC log magnitude spectra was used to determine distance between test and reference templates. The output of the baseline system was a rank-ordered vector of the best scoring phonemes. If the correct phoneme was included in the vector, the recognition was considered successful. Absence of the correct phoneme indicated an error. Recognition rate was dependent on the length, M , of the phoneme vector. Performance of the

Table 14. Figures of merit of the baseline recognizer for all eight speakers

Session	<i>F</i>	Comparison to Normal
11	77.51	
12	72.17	-5.34
13	71.41	-6.10
21	85.03	
22	78.95	-6.08
23	76.71	-8.32
31	83.09	
32	70.79	-12.30
33	58.16	-24.93
41	89.42	
42	73.22	-16.20
43	86.35	-3.07
51	77.18	
52	72.81	-4.37
53	68.75	-8.43
61	86.28	
62	72.32	-13.96
63	71.58	-14.70
71	80.05	
72	70.52	-9.53
73	67.01	-13.04
81	85.36	
82	76.21	-9.15
83	74.13	-11.23

system could be displayed with a curve of recognition rate as a function of M . To permit the rapid comparison of large numbers of performance curves, a figure of merit, F , was defined as a single-number description for a given performance curve.

9. EXPERIMENTS AND RESULTS

The work described in this chapter embodied the primary thrust of this research, namely to reduce the gap in recognition performance between normal and abnormal speech, given training on only normal speech. As a first step, it was important to compare the performance of the baseline recognizer to other established methods used in template-based recognition, specifically cepstral measurement and the likelihood ratio. This was designed to give a measure of validity to the baseline system as well as test the performance of these established methods on the database. Then the central issue of improving recognition performance on abnormal speech was addressed. For reasons that will be explained in a subsequent section, effort was focused on exploiting information contained in the slope of the LPC spectrum.

9.1 Cepstral Measure of Euclidean Distance

Recall in Chapter 7 that the all-pole model of the vocal tract provided by LPC was expressed as $H(z)$ in Equation (2). The log magnitude of this transfer function on the unit circle can be expressed with a Fourier series expansion as [G76]

$$\ln |H(e^{j\theta})|^2 = \sum_{k=-\infty}^{\infty} c_k e^{-jk\theta} \quad (12)$$

where c_k are known as *cepstral* coefficients. The Euclidean distance, d , between test frame, H' , and reference frame, H , can be expressed in terms of the cepstral coefficients by using Parseval's relation

$$\begin{aligned} d^2 &= \frac{1}{2\pi} \int_0^{2\pi} \left[\ln |H'(e^{j\theta})|^2 - \ln |H(e^{j\theta})|^2 \right]^2 d\theta \\ &= \sum_{k=-\infty}^{\infty} (c'_k - c_k)^2 \end{aligned} \quad (13)$$

Since $c_{-k} = c_k$, Equation 13 can also be written as

$$d^2 = (c'_0 - c_0)^2 + 2 \sum_{k=1}^{\infty} (c'_k - c_k)^2 \quad (14)$$

The cepstral coefficients can be computed directly from the LPC coefficients using the recurrence relations [O68, At74]

$$c_k = \begin{cases} -a_k - \frac{1}{k} \sum_{i=1}^{k-1} i c_i a_{k-i} & , \quad 1 \leq k \leq P \\ -\frac{1}{k} \sum_{i=1}^P (k-i) c_{k-i} a_i & , \quad k > P \end{cases} \quad (15)$$

If the series is truncated to L coefficients, then the resulting distance, d_L^2 , can be interpreted as the rms distance between the log spectra after each log spectrum has been cepstrally smoothed to L coefficients. Clearly, d_L^2 approaches d^2 from below with $\lim_{L \rightarrow \infty} d_L^2 = d^2$. For this research, $L=24$ was chosen since $L=P$ provides a reasonable estimate of the Euclidean distance. (For 800 frames of data, Gray and Markel [G76] found a correlation of 0.98 between d^2 and d_L^2 for $L=P=10$ over the distance range from 0-6 dB.)

The performance of cepstral distance is compared with the baseline recognition system in Table 15. This table uses the figure of merit, F , that was defined in Equation (11), Chapter 8. Note that there are different types of entries in the table as annotated in the key at the top. Absolute values of F are listed in the *baseline* and *test* columns for each of the 24 sessions (8 speakers $\times$ 3 conditions). Each session is annotated as a two-digit number with the tens digit referring to the speaker, and the units digit referring to the condition (1=normal, 2=loud, and 3=Lombard). Also in these two columns are differences between loud and Lombard sessions versus normal sessions for each speaker. (These are indicated in the session column as: test session - reference session. For example, 52-51 indicates the difference (gap) between loud and normal speech recognition for speaker #5.) This entry provides a measure of the performance gaps between normal and abnormal speech. The column on the far right shows the difference between the *baseline* and *test* column entries for each row, with the sign of the difference indicating whether the test performed better (+) or worse (-) than the baseline. For each session, the difference value directly indicates how much better or worse the test method performed

Table 15. Performance of cepstral distance compared to baseline system

Key of table entries for speaker 1						
Session (speaker, condition)	Baseline F		Test F		Difference F_{Δ}	
11	F_{11}		F_{11}		F_{11}	F_{11}
12	F_{12}		F_{12}		F_{12}	F_{12}
13	F_{13}		F_{13}		F_{13}	F_{13}
12-11	F_{12}	F_{11}	F_{12}	F_{11}	F_{12}	$(F_{12} - F_{11})$
13-11	F_{13}	F_{11}	F_{13}	F_{11}	F_{13}	$(F_{13} - F_{11})$

Session (speaker, condition)	Baseline F	Test F	Difference F_{Δ}
11	77.51	77.09	0.42
12	72.17	69.50	2.67
13	71.41	69.77	1.64
12-11	5.34	7.59	2.25
13-11	8.10	7.22	1.22
21	65.03	64.91	0.12
22	76.95	77.65	-1.10
23	76.71	75.95	0.76
22-21	6.08	7.08	-0.90
23-21	8.22	8.38	-0.84
31	63.09	63.77	-0.13
32	70.79	69.55	1.24
33	58.16	57.70	0.46
32-31	12.20	13.67	-1.27
33-31	24.53	25.32	-0.59
41	69.47	67.69	1.73
42	73.77	72.17	1.05
43	66.25	65.77	0.58
42-41	16.70	15.57	0.68
43-41	2.07	1.92	0.15
51	77.16	75.79	1.89
52	72.81	71.00	1.81
53	68.75	67.18	1.27
52-51	4.37	4.79	-0.08
53-51	8.42	7.91	0.52
61	66.29	65.87	0.41
62	72.32	71.71	1.11
63	71.58	71.79	-0.22
62-61	13.96	14.06	-0.70
63-61	14.70	14.81	-0.09
71	60.05	79.11	0.94
72	70.52	70.22	0.20
73	67.01	65.15	1.86
72-71	9.53	8.89	0.64
73-71	12.04	12.98	-0.92
81	85.36	85.27	0.09
82	76.21	74.43	1.78
83	74.13	73.28	0.75
82-81	9.15	10.84	-1.69
83-81	11.23	11.89	-0.66
Total Normal	82.99	82.31	0.68
Total Loud	73.37	71.99	1.28
Total Lombard	71.78	70.79	0.97
Overall	76.04	75.03	1.01

compared to the baseline. For instance, speaker #7, Lombard speech (session 73), exhibited a degradation in performance of cepstral distance compared to the baseline with the difference in figure of merit, $F_{\Delta 73} = -1.86$. Each abnormal-normal comparison line in Table 15 (e.g. 83-81) has a difference value, F_{Δ} , that

indicates whether the normal-abnormal gap became larger (−) or smaller (+), compared to the baseline. As an example, the gap between Lombard and normal speech for speaker #7 (73–71) became worse because the gap as measured by F went from −13.04 in the baseline system to −13.96 in the test system. $F_{\Delta(73-71)} = -0.92$ indicates that the gap *worsened* by that amount in the figure of merit.

The usefulness of the figure of merit, F , becomes more apparent when one tries to make accurate comparisons from the original performance graphs. Figures 17 and 18 illustrate the performance of the baseline and cepstral measures respectively for speaker #7. Performance is indeed quite similar, and it is difficult to discern from visual inspection exactly how the cepstral measure compares to the baseline system. On the other hand, the figure of merit, F , indicates that performance for speaker #7 worsened by −0.94 for normal speech ($F_{\Delta 71}$), −0.30 for loud speech ($F_{\Delta 72}$), and −1.86 for Lombard speech ($F_{\Delta 73}$). While these are indeed small differences, they indicate reliable trends when gathered for all eight speakers.

The bottom rows of Table 15 contain averages for all eight speakers for each condition, and the very last row is a grand average across all three conditions. Table 16 clarifies the entries in the bottom section of Table 15. Hence, the bottom right number, −1.01, is a final score of the overall performance of the test system compared to the baseline system. This number will be referred to as F_G , and will serve as a gross summary of the comparison of any recognition system to the baseline system. Interpreted for this experiment, the use of the cepstral distance measure caused an overall degradation in recognition performance compared to the baseline system. This is understandable since the cepstral sequence was truncated to 24 terms and therefore provided an estimate of the Euclidean distance. Note in the summary section of Table 15 that when the final score is broken down by condition, performance degraded most for loud recognition ($F_{\Delta loud} = -1.38$) and least for normal recognition ($F_{\Delta normal} = -0.68$), with Lombard recognition scoring in between ($F_{\Delta Lombard} = -0.97$). In other words, the degradation caused from using cepstral distance was aggravated by abnormal speech, with loud speech causing more problem than Lombard speech.

9.2 Likelihood Ratio

The likelihood ratio, as applied to linear predictive coding, is a method of comparing two different sets of LPC coefficients by determining the probability that the two sets came from the same speech waveform. It was first proposed by

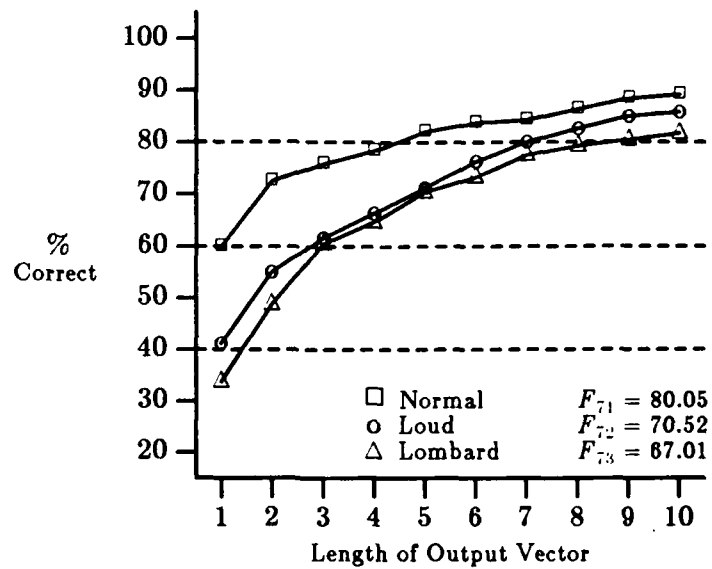


Figure 17. Recognition performance for baseline system, speaker #7

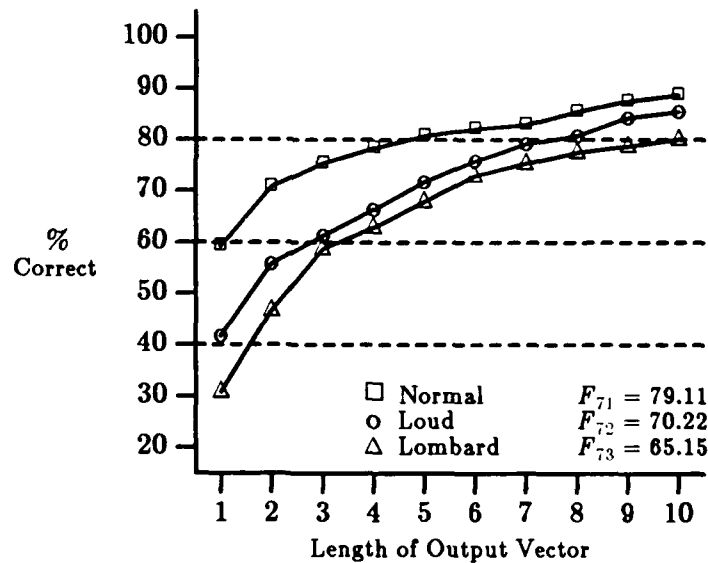


Figure 18. Recognition performance for cepstral distance, speaker #7

Itakura [I75]. As explained by Gray and Markel [G76], the ratios are actually derived from the residual or prediction error of the LPC filter, and are shown to be likelihood ratios when the data is assumed to be Gaussian and the analysis window is much greater than the filter length. Specifically, for a set of LPC coefficients, $\{a_i\}_{i=1}^P$, derived from speech samples $\{x_n\}_{n=0}^{N-1}$, the prediction error can be expressed as

Table 16. Key for summary section of all figure of merit comparison tables

Key of table entries for speaker <i>i</i>			
	Baseline F	Test F	Difference, F_{Δ}
Total Normal	$\frac{1}{8} \sum_{j=1}^8 F_{i,1}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,1}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,1} - F_{i,1}$
Total Loud	$\frac{1}{8} \sum_{j=1}^8 F_{i,2}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,2}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,2} - F_{i,2}$
Total Lombard	$\frac{1}{8} \sum_{j=1}^8 F_{i,3}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,3}$	$\frac{1}{8} \sum_{j=1}^8 F'_{i,3} - F_{i,3}$
Overall	$\frac{1}{8} \sum_{j=1}^8 \frac{1}{3} \sum_{i=1}^3 F_{ij}$	$\frac{1}{8} \sum_{j=1}^8 \frac{1}{3} \sum_{i=1}^3 F'_{ij}$	$\frac{1}{8} \sum_{j=1}^8 \frac{1}{3} \sum_{i=1}^3 (F'_{ij} - F_{ij})$

$$e_n = x_n - \sum_{i=1}^P a_i x_{n-i} \quad (16)$$

For an individual frame of speech, $\{x_n\}_{n=0}^{N-1}$, the autocorrelation method of LPC assumes the speech signal to be zero outside the frame such that the total squared error will be a function only of the frame of interest. The total squared error, α , is given as

$$\alpha = \sum_{n=-\infty}^{\infty} e_n^2 \quad (17)$$

Now for a test set of LPC coefficients, $\{a'_i\}_{i=1}^P$, derived from test frame, $\{x'_n\}_{n=0}^{N-1}$, define another total squared error term

$$\delta = \sum_{n=-\infty}^{\infty} \left[x_n - \sum_{i=1}^P a'_i x_{n-i} \right]^2 \geq \alpha \quad (18)$$

where equality is guaranteed when $\{x'_n\} = \{x_n\}$. The ratio $\frac{\delta}{\alpha}$ is then used as a measure of difference between $\{x'_n\}$ and $\{x_n\}$. These ratios can be efficiently computed using the autocorrelation sequence of the speech data, $\{r_{xn}\}$, and the autocorrelation sequence of the LPC coefficients, $\{r_{an}\}$ by

$$\alpha = \sum_{n=-P}^P r_{an} r_{xn} \quad (19)$$

$$\delta = \sum_{n=-P}^P r'_{an} r_{zn} \quad (20)$$

The ratio, $\frac{\delta}{\alpha}$, is asymmetrical, depending on which frames are considered reference and test. To obtain a symmetrical measure [G76], the other combinations of autocorrelations for the speech data and LPC coefficients are also calculated

$$\alpha' = \sum_{n=-P}^P r'_{an} r'_{zn} \quad (21)$$

$$\delta' = \sum_{n=-P}^P r_{an} r'_{zn} \quad (22)$$

and then combined

$$\Omega = \frac{\frac{\delta}{\alpha} + \frac{\delta'}{\alpha'}}{2} - 1 \quad (23)$$

Finally, Ω is related to a decibel scale by

$$d_{\Omega} = \ln \left[1 + \Omega + \sqrt{\Omega(2+\Omega)} \right] \quad (24)$$

The performance of the likelihood ratio is compared to the baseline system in Table 17. In general, the likelihood ratio performance was remarkably equivalent to that of the baseline system for normal recognition with a score of $F_{\Delta normal} = -0.09$. Half of the speakers scored worse while the other half scored better in the normal recognition tests. However there was minor degradation in the abnormal speech from the baseline system with loud speech scoring $F_{\Delta loud} = -0.34$ and Lombard speech scoring $F_{\Delta Lombard} = -0.26$. The overall score for all three speech conditions was $F_G = -0.23$, or about one fourth of the degradation caused by using cepstral coefficients. (Recall from Table 15 that for the cepstral measure, $F_G = -1.01$.) In other words, F_G was about four times worse for the cepstral measure as compared to the likelihood ratio. Although the likelihood ratio performed considerably better than the cepstral measure, it was still not quite as good as the baseline system. Individually, speaker #6 was the main contributor to degraded performance for abnormal speech, with $F_{\Delta 62} = -3.28$ and $F_{\Delta 63} = -3.89$, for loud and Lombard speech respectively.

Table 17. Performance of the likelihood ratio compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	76.45	-1.06
12	72.17	71.82	-0.35
13	71.41	71.18	-0.23
12-11	-5.34	-4.63	0.71
13-11	-6.10	-5.27	0.83
21	85.03	85.34	0.31
22	78.95	78.11	-0.84
23	76.71	76.35	-0.36
22-21	-6.08	-7.23	-1.15
23-21	-8.32	-8.99	-0.67
31	83.09	82.62	-0.47
32	70.79	72.36	1.57
33	58.16	62.55	4.39
32-31	-12.30	-10.26	2.04
33-31	-24.93	-20.07	4.86
41	89.42	89.43	0.01
42	73.22	72.75	-0.47
43	86.35	86.33	-0.02
42-41	-16.20	-16.68	-0.48
43-41	-3.07	-3.10	-0.03
51	77.18	77.50	0.32
52	72.81	73.33	0.52
53	68.75	67.83	-0.92
52-51	-4.37	-4.17	0.20
53-51	-8.43	-9.67	-1.24
61	86.28	87.01	0.73
62	72.32	69.04	-3.28
63	71.58	67.69	-3.89
62-61	-13.96	-17.97	-4.01
63-61	-14.70	-19.32	-4.62
71	80.05	79.79	-0.26
72	70.52	70.80	0.28
73	67.01	67.53	0.52
72-71	-9.53	-8.99	0.54
73-71	-13.04	-12.26	0.78
81	85.36	85.05	-0.31
82	76.21	76.07	-0.14
83	74.13	72.54	-1.59
82-81	-9.15	-8.98	0.17
83-81	-11.23	-12.51	-1.28
Total Normal	82.99	82.90	-0.09
Total Loud	73.37	73.04	-0.34
Total Lombard	71.76	71.50	-0.26
Overall	76.04	75.81	-0.23

Accordingly, his loud-normal gap, $F_{\Delta(62-61)}$, worsened by -4.01 , and his Lombard-normal gap, $F_{\Delta(63-61)}$, worsened by -4.62 . Figures 19 and 20 show the performance graphs of speaker #6 for the baseline and likelihood ratio, respectively. On the other hand, speaker #3 exhibited improved performance

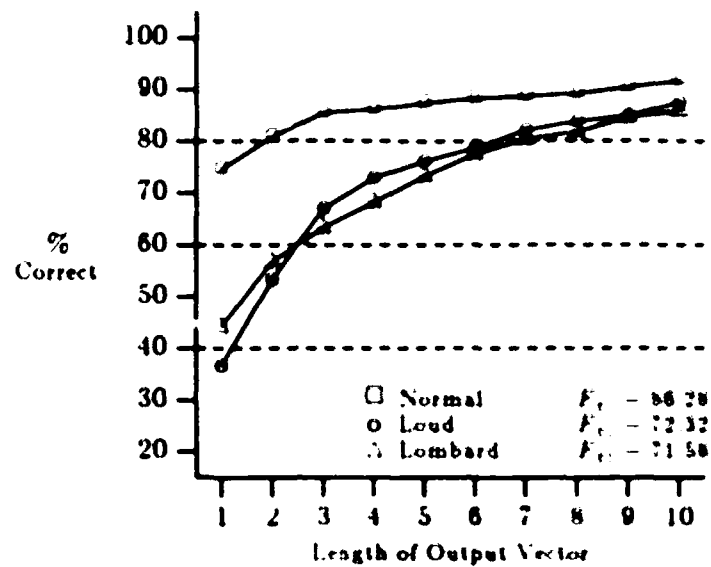


Figure 19. Recognition performance for baseline system, speaker #6

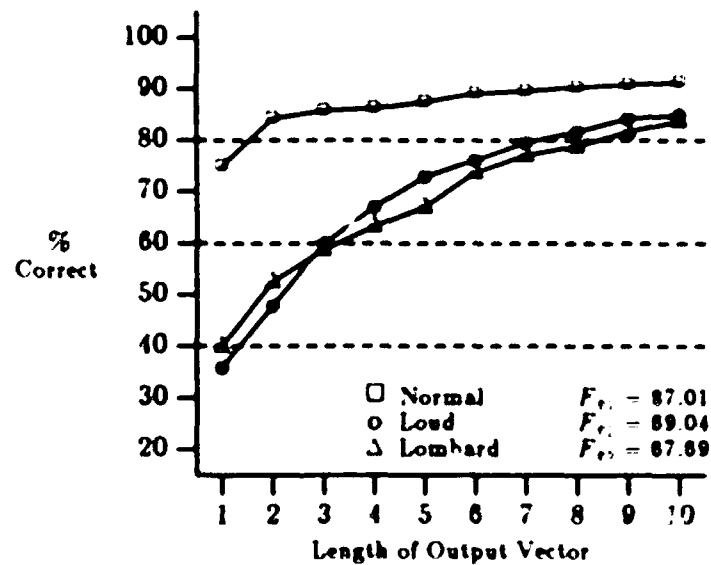


Figure 20. Recognition performance for likelihood ratio, speaker #6

for abnormal speech, scoring increases for loud speech ($F_{\Delta 32} = 1.57$), and Lombard speech ($F_{\Delta 33} = 4.39$). This resulted in improvements in the loud-normal gap, $F_{\Delta (32-31)}$, of 2.04, and in the Lombard-normal gap, $F_{\Delta (33-31)}$, of 4.86.

Two conclusions can be drawn from examining cepstral distances and the likelihood ratio. First, the baseline recognition system with its direct computation of Euclidean distance between log spectra performed favorably compared to popular established methods of distance measures. It could therefore be used as a reliable measure of recognition performance degradation for loud and Lombard speech. Second, cepstral distances and the likelihood ratio provided no immediate promise of improving recognition performance for loud and Lombard speech. It is important to note however that the likelihood ratio outperformed the cepstral measure in all three speech conditions, normal, loud, and Lombard.

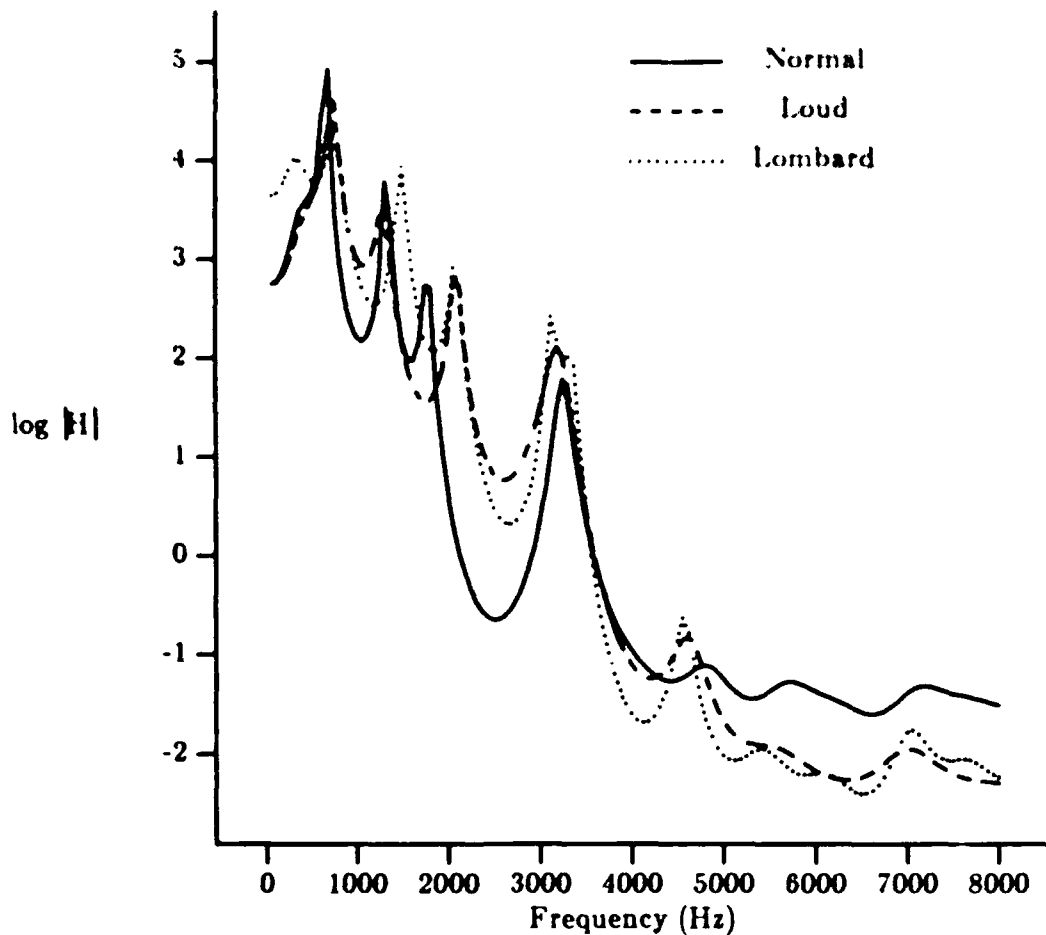


Figure 21. Comparison of LPC spectra for normal, loud, and Lombard for phoneme AA of speaker #2

9.3 Spectral Slope

From the analyses of abnormal speech in Chapter 7, the most reliable shifts in abnormal speech appeared in the sonorants in the form of energy migrations. It was therefore logical to use this phenomenon as the basis for designing a compensatory method to improve the recognition performance of abnormal speech. A major objective was to develop a method that could be used across all speakers without any direct knowledge of their individual abnormal speech characteristics. Not to be confused with speaker independence, which is a completely separate issue, the objective sought a universal compensatory method within the context of speaker dependence. In other words, it was a given conclusion that each speaker would be required to individually train the recognition system with normal speech. The goal, however, was to avoid any training on abnormal speech. Adaptation methods were excluded from consideration because of the infeasibility of providing the necessary feedback on recognition performance in the cockpit environment. A pilot in the heat of battle would not have the opportunity (or inclination) to correct the errors committed by a speech recognition system. He would rather elect to bypass the system in the interest of expediency. An approach using a fixed weighting scheme was not considered practical for exploiting the energy migrations due to the wide range of variability observed across speakers. Instead, an attribute was sought that would provide resistance to the energy migrations while preserving the distinction among different phonemes. By studying spectral overlays of the same phoneme for normal, loud, and Lombard speech, such as the typical example in Figure 21, it was noted that the overall shape of all three spectra were very similar, and the energy migrations were manifested as vertical shifts in the abnormal spectra. It was therefore hypothesized that the first derivative of the log magnitude spectrum with respect to frequency, $\frac{d \log |H|}{d \theta}$, would be a viable attribute for capturing the similarity in shape among spectra without being adversely affected by energy shifts.

9.3.1 Computation from Templates

The first experiment with spectral slope used the LPC spectrum templates to derive a first order estimate, $S1$, of $\frac{d \log |H|}{d \theta}$

$$S1_l = \begin{cases} 0 & , A_l > A_{l-1} \text{ and } A_l > A_{l+1} \\ \frac{A_{l+1} - A_{l-1}}{2\Delta f} & , \text{ otherwise} \end{cases} \quad (25)$$

where $0 < l < \frac{N}{2}-1$, and A_l , as defined in Equation (3), Chapter 7, is

$$A_l = \ln \left| H \left(e^{j \frac{2\pi l}{n}} \right) \right| = \ln \left| \frac{1}{1 - \sum_{k=1}^P a_k e^{-j k \frac{2\pi l}{n}}} \right|$$

The estimate, $S1$, uses the difference between the following and preceding samples of the LPC spectrum divided by the frequency spanning those points. In the event that point A_l is greater than the point preceding and following, (i.e. the first case in Equation 25) then it is one of the peaks of the LPC spectrum and is therefore assigned zero slope. The distance between frame m of test phoneme token i and frame k of reference phoneme token j is then given as

$$d_{S1 \ mki j} = \left\{ \sum_{l=1}^{\frac{N}{2}-2} (S1'_{lmi} - S1_{lkj})^2 \right\}^{\frac{1}{2}} \quad (26)$$

Table 18 compares the performance of this spectral slope measurement to the baseline system. The overall score, $F_G = -11.30$ clearly indicates extremely poor performance with the most serious degradations occurring in the recognition of loud ($F_{\Delta loud} = -12.40$) and Lombard ($F_{\Delta Lombard} = -16.42$) speech. Individually, sessions 23, 63, 82, and 83 (i.e. the Lombard sessions for speakers #2, #6, and #8, and the loud session for speaker #8) gave the worst performances.

Figure 22 shows the performance of the baseline system, and Figure 23 shows the performance of the spectral slope estimate, both for speaker #2. The degradation is dramatic when comparing these two plots. Relating the graphs to the figures of merit for speaker #2, note how the Lombard curve in Figure 22 ($F_{23} = 76.71$) contrasts with the Lombard curve in Figure 23 ($F'_{23} = 55.88$). The difference between these two curves, $F_{\Delta 23} = -20.83$, speaks for itself. The Lombard-normal gap (23-21) is also easily compared in the two figures. For the baseline, $F_{(23-21)} = -8.32$, whereas for the spectral slope estimate, $F'_{(23-21)} = -26.59$, giving a widening (worsening) of the Lombard-normal gap of $F_{\Delta(23-21)} = -18.27$.

Table 18. Performance of spectral slope estimate compared to baseline system

Key of table entries for speaker i				
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}	
i1	F_{i1}	F'_{i1}	F'_{i1}	F_{i1}
i2	F_{i2}	F'_{i2}	F'_{i2}	F_{i2}
i3	F_{i3}	F'_{i3}	F'_{i3}	F_{i3}
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1}$	$(F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1}$	$(F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	74.03	-3.48
12	72.17	66.71	-5.46
13	71.41	65.40	-6.01
12-11	-5.34	-7.32	-1.98
13-11	-6.10	-8.63	-2.53
21	85.03	82.47	-2.56
22	78.95	72.83	-6.12
23	76.71	55.88	-20.83
22-21	-6.08	-9.64	-3.56
23-21	-8.32	-26.59	-18.27
31	83.09	75.68	-7.41
32	70.79	58.84	-11.95
33	58.16	55.59	-2.57
32-31	-12.30	-16.84	-4.54
33-31	-24.93	-20.09	4.84
41	89.42	79.16	-10.26
42	73.22	59.36	-13.86
43	86.35	72.56	-13.79
42-41	-16.20	-19.80	-3.60
43-41	-3.07	-6.80	-3.53
51	77.18	73.62	-3.56
52	72.81	62.59	-10.22
53	68.75	49.63	-19.12
52-51	-4.37	-11.03	-6.66
53-51	-8.43	-23.99	-15.56
61	86.28	83.47	-2.81
62	72.32	52.53	-19.79
63	71.58	48.09	-23.49
62-61	-13.96	-30.94	-16.98
63-61	-14.70	-35.38	-20.68
71	80.05	75.52	-4.53
72	70.52	65.08	-5.44
73	67.01	52.04	-14.97
72-71	-9.53	-10.44	-0.91
73-71	-13.04	-23.48	-10.44
81	85.36	79.28	-6.08
82	76.21	49.87	-26.34
83	74.13	43.55	-30.58
82-81	-9.15	-29.41	-20.26
83-81	-11.23	-35.73	-24.50
Total Normal	82.99	77.90	-5.09
Total Loud	73.37	60.98	-12.40
Total Lombard	71.76	55.34	-16.42
Overall	76.04	64.74	-11.30

The spectral slope estimate obviously did not perform as expected, and in fact caused much larger gaps between normal and abnormal recognition. What was not clear, however, was whether the poorer performance was due to spectral slope not providing the distinction among phonemes as anticipated or whether

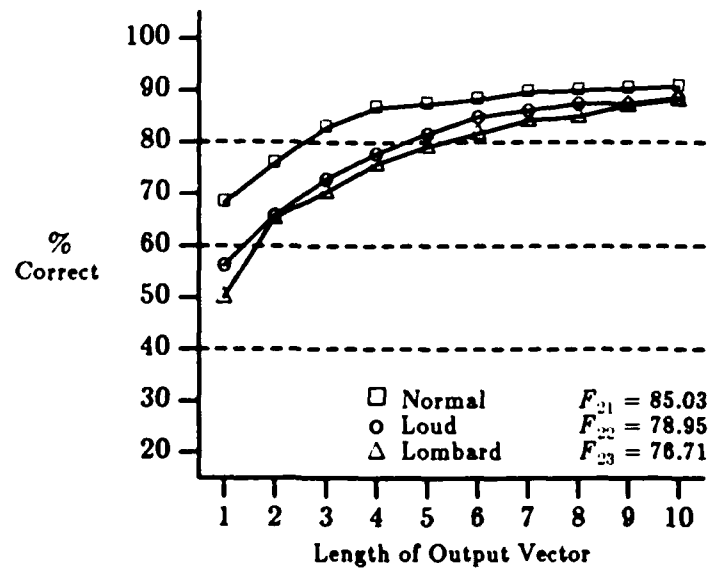


Figure 22. Recognition performance for baseline system, speaker #2

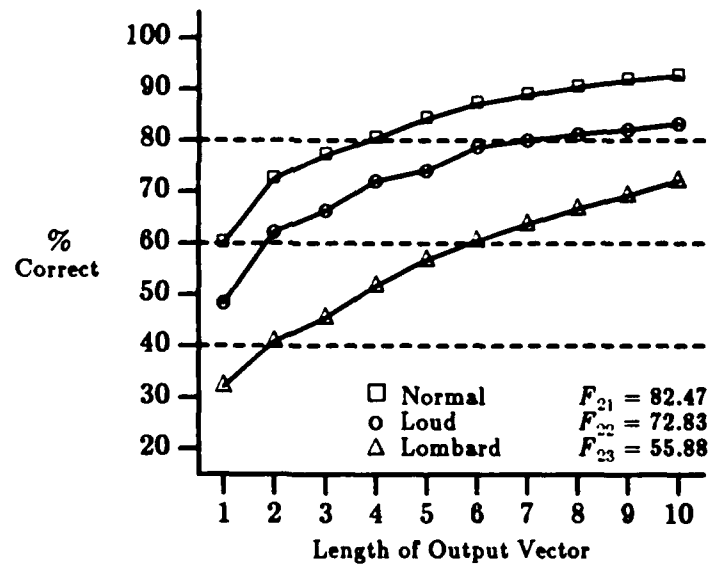


Figure 23. Recognition performance for spectral slope estimate, speaker #2

the estimate, S_1 , was inducing a majority of the increased error. Recall from Equation (25) that the slope was estimated over a range of $2\Delta f$. In light of the inherent error with this estimate, a more accurate method of calculating spectral slope was therefore tested to resolve the question, and is discussed in the following section.

9.3.2 Root Power Sums

In previous discussion, Equation (13) described how the Euclidean distance between log magnitude LPC spectra was efficiently calculated with cepstral coefficients. When the derivatives of the log magnitude spectra are taken with respect to angular frequency, a similar expression is obtained for the distances between spectral slopes [Han87].

$$\begin{aligned}
 d_{SLOPE}^2 &= \frac{1}{2\pi} \int_0^{2\pi} \left[\frac{d}{d\theta} \ln |H'(e^{j\theta})|^2 - \frac{d}{d\theta} \ln |H(e^{j\theta})|^2 \right]^2 d\theta \\
 &= \sum_{k=-\infty}^{\infty} (k(c'_k - c_k))^2 \\
 &= \lim_{L \rightarrow \infty} 2 \sum_{k=1}^L (k(c'_k - c_k))^2
 \end{aligned} \tag{27}$$

The weighted cepstral coefficients, kc_k , in Equation (27) are called *root power sums* by Schroeder, [Sc81], and have also been termed as *frequency-weighted cepstral coefficients* by Paliwal [Pal82]. For this implementation of spectral slope measure, L was set to 24. As discussed in Section 9.1, this provided a degree of cepstral smoothing which, in the case of root power sums, yielded control over the sensitivity of d_{SLOPE}^2 to spectral peaks [Han87].

The test results for root power sums are contained in Table 19. Performance was again degraded in all three speech conditions with $F_{\Delta normal} = -3.64$, $F_{\Delta loud} = -7.21$, and $F_{\Delta Lombard} = -10.48$. The overall score, $F_G = -7.11$ was somewhat of an improvement over the spectral slope estimate method. Individually, sessions 53, 62, 63, 82, and 83 (i.e. the loud sessions for speakers #6 and #8, and the Lombard sessions for speakers #5, #6, and #8) yielded the worst scores. As an example, the recognition performance graphs for the baseline and root power sums of speaker #5 are shown in Figures 24 and 25 respectively. The widening of the Lombard-normal gap is obvious with $F_{\Delta(53-51)} = -11.14$. Note also how normal recognition was degraded ($F_{\Delta 51} = -4.30$) with root power sums.

In general, the overall score, $F_G = -7.11$, for root power sums displayed considerable improvement over the spectral slope estimate, $S1$, where $F_G = -11.30$, owing to the increased accuracy in measurement of the differences in spectral slope provided by root power sums. But even with this improvement, the message was still clear. Instead of improving the recognition of abnormal speech, spectral slope measures alone actually widened the

Table 19. Performance of root power sums compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	75.76	-1.75
12	72.17	68.27	-3.90
13	71.41	68.96	-2.45
12-11	-5.34	-7.49	-2.15
13-11	-6.10	-6.80	-0.70
21	85.03	83.96	-1.07
22	78.95	79.40	0.45
23	76.71	66.80	-9.91
22-21	-6.08	-4.56	1.52
23-21	-8.32	-17.16	-8.84
31	83.09	77.75	-5.34
32	70.79	62.01	-8.78
33	58.16	58.50	0.34
32-31	-12.30	-15.74	-3.44
33-31	-24.93	-19.25	5.68
41	89.42	82.17	-7.25
42	73.22	66.93	-6.29
43	86.35	77.30	-9.05
42-41	-16.20	-15.24	0.96
43-41	-3.07	-4.87	-1.80
51	77.18	72.88	-4.30
52	72.81	67.28	-5.53
53	68.75	53.31	-15.44
52-51	-4.37	-5.60	-1.23
53-51	-8.43	-19.57	-11.14
61	86.28	85.34	-0.94
62	72.32	57.23	-15.09
63	71.58	52.71	-18.87
62-61	-13.96	-28.11	-14.15
63-61	-14.70	-32.63	-17.93
71	80.05	76.10	-3.95
72	70.52	66.73	-3.79
73	67.01	58.20	-8.81
72-71	-9.53	-9.37	0.16
73-71	-13.04	-17.90	-4.86
81	85.86	80.83	-4.53
82	76.21	61.48	-14.73
83	74.13	54.50	-19.63
82-81	-9.15	-19.35	-10.20
83-81	-11.23	-26.33	-15.10
Total Normal	82.99	79.35	-3.64
Total Loud	73.37	66.17	-7.21
Total Lombard	71.76	61.29	-10.48
Overall	76.04	68.93	-7.11

performance gaps between normal and abnormal speech. This is likely attributable to the extreme sensitivity of spectral slope distance to frequency shifts of narrow bandwidth spectral peaks, as reported by Hanson and Wakita [Han87]. With minor shifts in the formant frequencies, the spectral slope

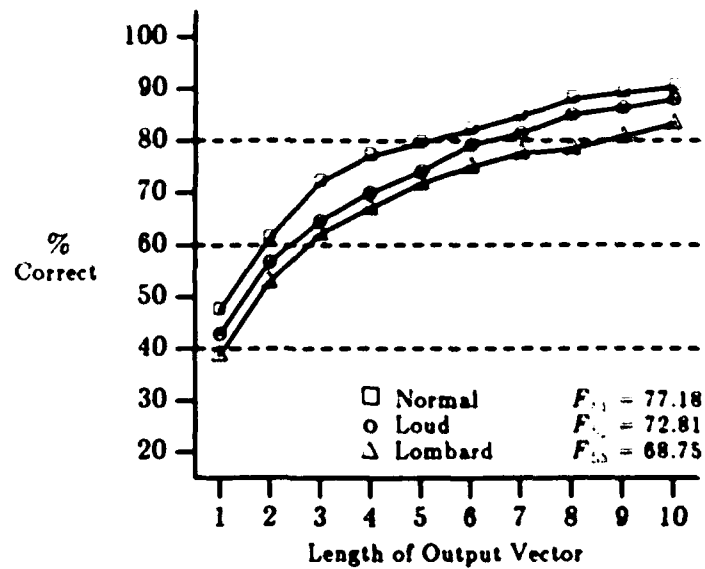


Figure 24. Recognition performance for baseline system, speaker #5

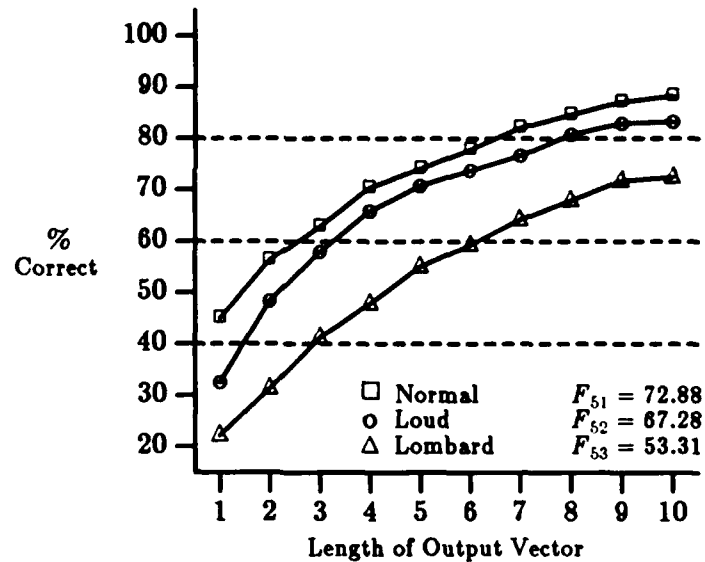


Figure 25. Recognition performance for root power sums, speaker #5

distance can be significantly affected. Since the analyses of Chapter 7 found significant differences in formant frequencies for the abnormal speech of a number of speakers, this can explain why the performance gaps between normal and abnormal speech actually widened when using spectral slope measures alone.

9.4 Slope-Dependent Weighting

In spite of the poor performance of the direct use of spectral slopes for assessing distances, this researcher was still encouraged that information contained in the spectral slope could be used to improve recognition performance for abnormal speech. This opinion was nurtured by the visual inspection of overlays of normal, loud, and Lombard spectra from the same phoneme, such as in Figure 21. It was felt that the similarity in the shape of the spectra should be useful in some way to reduce the errors of the baseline system. The following line of thinking was then pursued. The shift in energy between normal and abnormal spectra was intuitively a major contributor to the performance gaps exhibited by the baseline recognition system. This was deduced by the fact that Euclidean distance between two spectra is simply a measure of the *area* bounded by the two spectra. When the abnormal spectra contain an energy shift, the Euclidean distance between these spectra and the normal (reference) spectra necessarily increases, thus adding to the confusability when the spectral distances are sorted for the top candidates. If, on the other hand, the spectral distances due to energy shift could be reduced or eliminated while preserving spectral distances between dissimilar phonemes, then the performance gaps between normal and abnormal recognition should also be reduced. In this vein, it was hypothesized that similarity in spectral slope could be used as an indicator of energy shift due to abnormal speech. The Euclidean distance between spectra at a given frequency could be weighted by the dissimilarity in spectral slope at that frequency. If spectral slope was dissimilar, then the energy difference between spectra would be fully weighted, and if the spectral slope was similar then the energy difference would be reduced or eliminated under the premise that energy differences in regions of similar slope were due to the energy shifts in abnormal speech.

9.4.1 Non-Linear Weighting Function

To accomplish the necessary weighting, a non-linear function, $w(s)$, was proposed, where s_{lmkij} is the magnitude difference in spectral slope at a particular frequency index l , between frame m of test phoneme i and frame k of reference phoneme j .

$$s_{lmkij} = \left| S1'_{lmi} - S1_{lkj} \right| \quad (28)$$

The function, $w(s)$, is depicted in Figure 26. This weighting function was then incorporated into the baseline metric of Equation (10), Chapter 8 to obtain the weighted metric, d_w , expressed as

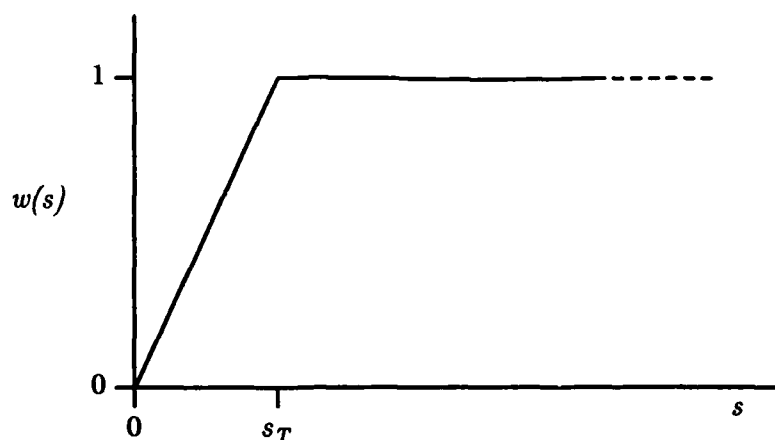


Figure 26. Non-linear weighting function

$$d_{W \text{ } mki j} = \left\{ \sum_{l=0}^{\frac{N}{2}-1} w(s_{lmki j}) (A'_{lmi} - A_{lkj})^2 \right\}^{\frac{1}{2}} \quad (29)$$

The knee-point, s_T , in $w(s)$ serves as a threshold such that for $s \geq s_T$, unity weight is assigned, and for $0 \leq s \leq s_T$, $w(s) = s/s_T$. In essence, the energy between two spectra at a particular frequency is reduced only when the magnitude difference in spectral slope is less than the threshold, s_T . For frequencies where the magnitude difference in spectral slope exceeds s_T , the Euclidean distance is preserved. Note that when $s_T = 0$, there is no reduced weighting for regions of similar slope, and the baseline Euclidean metric is preserved.

To get an idea of the values of the magnitude difference in spectral slope, session 53 (Lombard speech from speaker #5) was selected arbitrarily to evaluate the distribution of s . This provided 3.07×10^8 ($128 \times 50 \times 240 \times 200$) samples. The distribution of these samples is shown in Figure 27. Note that the solid curve represents slope differences for the case where reference and test templates were individual tokens of the *same* phoneme, and the dotted curve illustrates the case where reference and test templates represented different phonemes. For $0 \leq s \leq 0.3$, there is a higher density of samples for templates of like phonemes. Interpreted, this indicates that there is more slope similarity for spectra of like phonemes than for spectra of different phonemes.

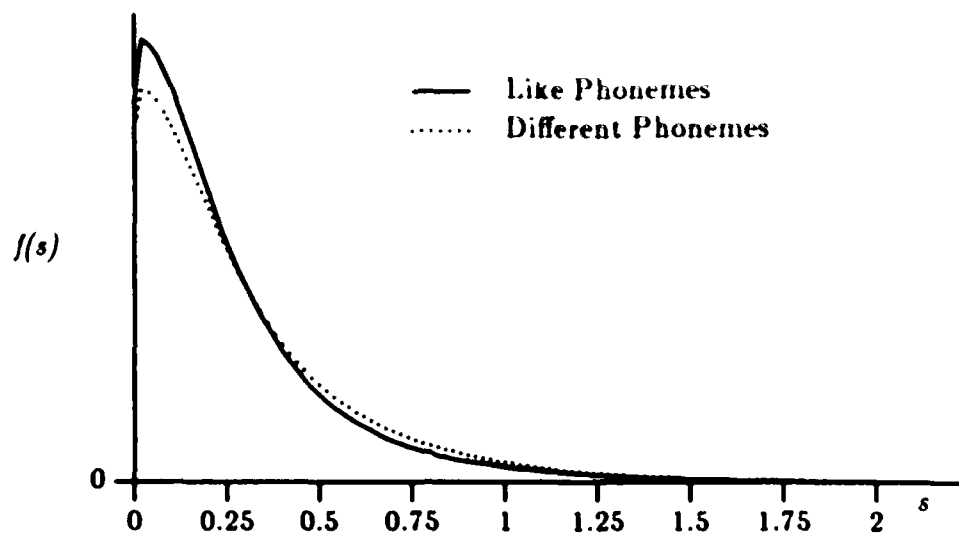


Figure 27. Distribution of values for magnitude difference in spectral slope, session 53

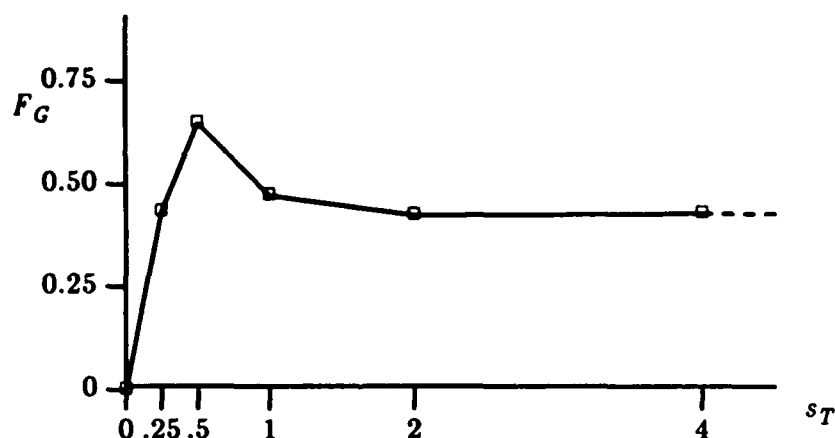


Figure 28. Overall performance of slope-dependent weighting for different threshold values

9.4.2 Performance

The method of slope-dependent weighting was tested with several values of s_T against the baseline system in order to determine optimum performance. The overall scores, F_G , as a function of s_T are graphed in Figure 28. Note that for $s_T = 0.0$, there is no slope-dependent weighting, and the distance measure defaults to the Euclidean baseline. The overall score, F_G , is zero because in effect the baseline system is compared with itself. As s_T increases, F_G increases

rapidly to a maximum of 0.65 for $s_T = 0.5$, and then falls to a constant value of 0.43 for $s_T > 2$. ($s_T = 8.0$ also yielded this value.) This correlates quite well with the distribution in Figure 27 that shows negligible occurrences of $s > 2$. When $s_T > 2$, the non-linear cap of unity in the weighting function, $w(s)$ (see Figure 26), is never invoked. Hence the optimum performance appears to occur when the threshold, s_T , is set such that slope-dependent weighting is selected for values of s where like phonemes have a higher distribution than different phonemes (i.e. the region in Figure 27 where the solid curve has higher value than the dotted curve).

The performance figures for the case $s_T = 0.5$ are contained in Table 20. The overall score, $F_G = 0.65$, indicates definite improvement over the baseline system. There was a small overall gain in normal recognition with $F_{\Delta normal} = 0.25$. Individually, speakers #1, #2, #5, and #6 posted improvements in normal recognition while the remaining speakers posted minor degradations, with the worst being speaker #8 ($F_{\Delta 81} = -0.67$). Loud recognition improved for seven out of the eight speakers with $F_{\Delta loud} = 0.75$. Speaker #6 was the only speaker to post a degradation for loud speech ($F_{\Delta 62} = -1.44$). Lombard recognition displayed the most improvement of the three conditions ($F_{\Delta Lombard} = 0.94$) even though speakers #2, #5, #6, and #8 posted degradations for Lombard speech.

The most significant improvements occurred for speaker #3, whose performance curves for the baseline system and slope-dependent weighting are graphed in Figures 29 and 30. Note in Figure 30 how the performance curves are visibly closer together than those in Figure 29. This was precisely the type of improvement sought in this research, but it was hoped that the abnormal-normal performance gaps could be reduced even further. Additional success in gap reduction was indeed achieved with the modification described in the next section.

9.5 Smallest Cumulative Distance

Up to this point, recognition was performed on a test template by finding its distance from each of 200 reference templates. Even though there were five reference templates for each of the 40 phonemes, the performance of the baseline recognition system (and its variations) was determined by the *single* best scoring reference template that matched the test template. This could be termed as the *raw nearest neighbor* or *RNN* approach. In an effort to assess the *collective* performance of all five tokens of a given phoneme, experiments were conducted whereby the distances of like tokens were combined to form a cumulative

Table 20. Performance of slope-dependent weighting compared to baseline system, $s_T = 0.5$

Key of table entries for speaker i				
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}	
i1	$F_{i,1}$	$F'_{i,1}$	$F'_{i,1} - F_{i,1}$	$(F_{i,1} - F_{i,1})$
i2	$F_{i,2}$	$F'_{i,2}$	$F'_{i,2} - F_{i,2}$	$(F_{i,2} - F_{i,2})$
i3	$F_{i,3}$	$F'_{i,3}$	$F'_{i,3} - F_{i,3}$	$(F_{i,3} - F_{i,3})$
i2-i1	$F_{i,2} - F_{i,1}$	$F'_{i,2} - F'_{i,1}$	$F'_{i,2} - F'_{i,1}$	$(F_{i,2} - F_{i,1})$
i3-i1	$F_{i,3} - F_{i,1}$	$F'_{i,3} - F'_{i,1}$	$F'_{i,3} - F'_{i,1}$	$(F_{i,3} - F_{i,1})$

Session (speaker, condition)	Baseline F	Test F	Difference F_{Δ}
11	77.51	78.41	0.90
12	72.17	72.58	0.41
13	71.41	74.17	2.76
12-11	-5.34	-5.83	0.49
13-11	-6.10	-4.24	1.86
21	85.03	86.22	1.19
22	78.95	80.54	1.59
23	76.71	76.37	-0.34
22-21	-6.08	-5.68	0.40
23-21	-8.32	-9.85	-1.53
31	83.09	83.03	-0.06
32	70.79	73.27	2.48
33	58.16	64.88	6.72
32-31	-12.30	-9.76	2.54
33-31	-24.93	-18.15	6.78
41	89.42	89.33	-0.09
42	73.22	73.29	0.07
43	86.35	86.75	0.40
42-41	-16.20	-16.04	0.16
43-41	-3.07	-2.58	0.49
51	77.18	77.48	0.30
52	72.81	73.67	0.86
53	68.75	68.42	-0.33
52-51	-4.37	-3.81	0.56
53-51	-8.43	-9.06	-0.63
61	86.28	87.30	1.02
62	72.32	70.88	-1.44
63	71.58	69.64	-1.94
62-61	-13.96	-16.42	-2.46
63-61	-14.70	-17.66	-2.96
71	80.05	79.46	-0.59
72	70.52	71.90	1.38
73	67.01	68.43	1.42
72-71	-9.53	-7.56	1.97
73-71	-13.04	-11.03	2.01
81	85.36	84.69	-0.67
82	76.21	76.89	0.68
83	74.13	72.97	-1.16
82-81	-9.15	-7.80	1.35
83-81	-11.23	-11.72	-0.49
Total Normal	82.99	83.24	0.25
Total Loud	73.37	74.13	0.75
Total Lombard	71.76	72.70	0.94
Overall	76.04	76.69	0.65

distance. The 40 cumulative distances for the phonemes in the lexicon were then sorted, and the M best scores were again selected to form the phoneme candidate vector, $\bar{p}_i$, as described in Chapter 8. This five-nearest neighbor

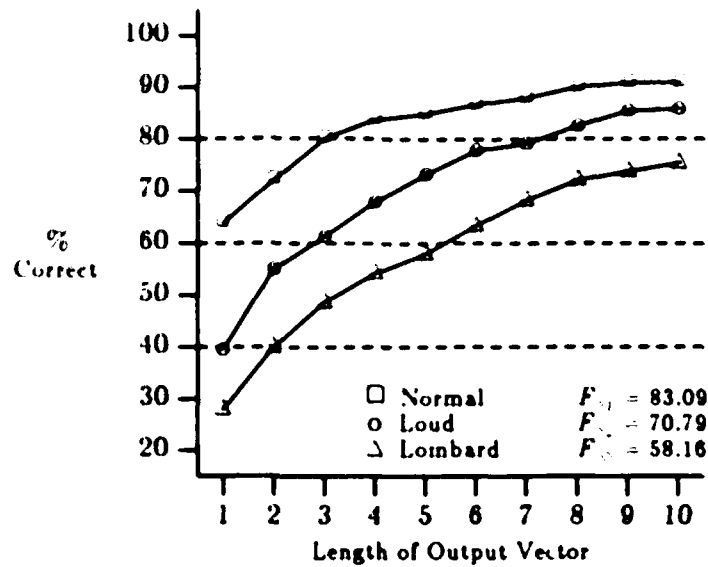


Figure 29. Recognition performance for baseline system, speaker #3

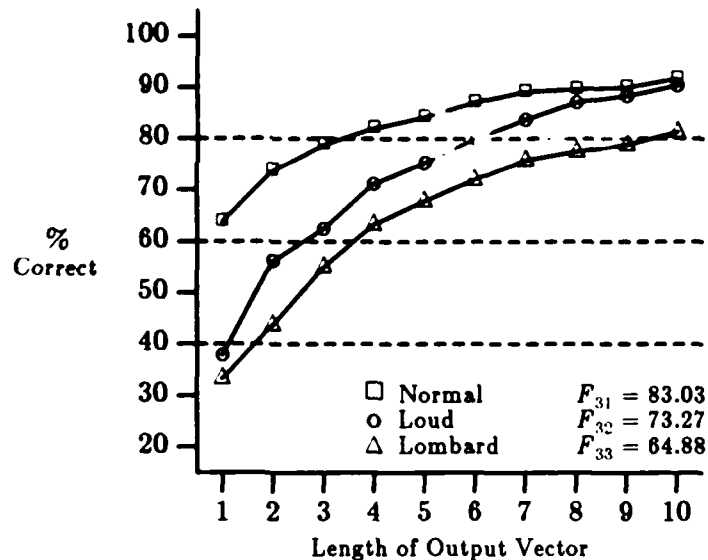


Figure 30. Recognition performance for slope-dependent weighting $s_T = 0.5$, speaker #3

method could also be called the *smallest cumulative distance* or *SCD* approach.

9.5.1 Performance with Other Metrics

The overall performance scores of the smallest cumulative distance method are compared to the raw nearest neighbor method in Table 21 for the metrics previously discussed. The complete figure of merit tables for these comparisons

Table 21. Comparison of smallest cumulative distance (SCD), versus raw nearest neighbor (RNN), for various metrics

Metric	F_G for RNN	F_G for SCD
Baseline Euclidean	0.0	0.68
Cepstral Measure	-1.01	-1.89
Likelihood Ratio	-0.23	0.65
Spectral Slope Estimate	-11.30	-9.03
Root Power Sums	-7.11	-4.99

are contained in Appendix N. Note that SCD improved performance for all except the cepstral measure. For this case, a majority of the degradation occurred in the recognition of normal speech ($F_{\Delta normal} = -4.80$). Lombard recognition was hardly affected ($F_{\Delta Lombard} = -0.06$), and loud recognition was only moderately affected ($F_{\Delta loud} = -0.81$). The baseline Euclidean measure was improved by an overall score of 0.68, but this is somewhat misleading. The recognition of abnormal speech was significantly improved ($F_{\Delta loud} = 1.94$; $F_{\Delta Lombard} = 2.42$) at the expense of a degradation for normal speech ($F_{\Delta normal} = -2.32$). The overall improvement in the case of the likelihood ratio, $F_G = 0.65$, breaks down similarly with abnormal speech performing higher ($F_{\Delta loud} = 1.99$; $F_{\Delta Lombard} = 2.51$), and normal speech performing lower ($F_{\Delta normal} = -2.55$). In fact this trend was observed in all the SCD tests. SCD produced considerable gains in the recognition of abnormal speech and moderate losses in the recognition of normal speech.

9.5.2 Performance with Slope-Dependent Weighting

Smallest cumulative distance was incorporated into the method of slope-dependent weighting, and the threshold, s_T , was again varied to determine the effect on overall performance. These scores are graphed as a function of s_T in Figure 31. Note in this figure that SCD is clearly superior to RNN. It is also interesting that the RNN and SCD curves are maximized at different values of threshold, s_T . While RNN achieves a maximum of $F_G = 0.65$ for $s_T = 0.5$, SCD achieves a maximum of $F_G = 2.24$ for $s_T = 1.0$. As s_T increases beyond 2.0, F_G levels at 2.22 for SCD.

The figures of merit of slope-dependent weighting ($s_T = 1.0$) using SCD (abbreviated as SDW-SCD) are contained in Table 22. (Note that unless specifically stated otherwise, reference to the baseline system implies the use of RNN.) Clearly the combination of slope-dependent weighting and SCD

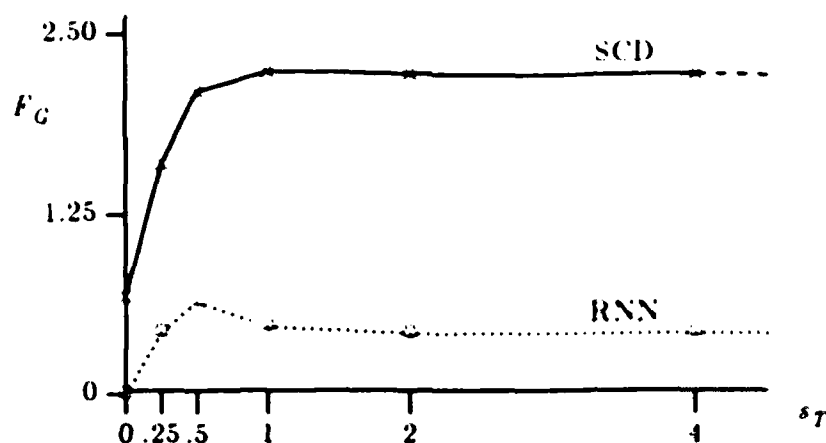


Figure 31. Comparison of SCD and RNN with the method of slope-dependent weighting

provided a synergistic improvement since slope-dependent weighting alone resulted in $F_G = 0.65$, and SCD alone resulted in $F_G = 0.68$, but together the overall score was $F_G = 2.24$. The improvements in the recognition of abnormal speech were the most significant for this research with $F_{\Delta\text{loud}} = 3.67$ and $F_{\Delta\text{Lombard}} = 4.12$. These gains were offset somewhat by the degradation of normal recognition in six out of the eight speakers, resulting in $F_{\Delta\text{normal}} = -1.09$. Still, the improvements in abnormal recognition more than compensate for this degradation.

The effectiveness of SDW-SCD can be seen more clearly by inspecting the performance graphs for individual speakers. For instance, Figures 32 and 33 compare the baseline system to slope-dependent weighting with SCD for speaker #3. Referring also to the data in Table 22, note that speaker #3 experienced significant improvements in abnormal recognition ($F_{\Delta 32} = 5.89$; $F_{\Delta 33} = 13.65$) with only a slight degradation in normal recognition ($F_{\Delta 31} = -1.65$). The gaps between normal and abnormal recognition were markedly reduced, as is easily seen in Figure 33. Note that while the Lombard curve in Figure 33 shows only slightly improved performance for $M = 1$, the performance rises much more rapidly than the baseline Lombard performance for $M \geq 2$, which considerably closes the gap between Lombard and normal speech for SDW-SCD. Quantitatively, the Lombard-normal gap for the baseline measures $F_{(33-31)} = -24.93$, and for SDW-SCD measures $F'_{(33-31)} = -9.63$, for a total gap reduction of $F_{\Delta(33-31)} = 15.30$. In view of the fact that speaker #3 had the largest baseline gaps between abnormal and normal recognition of all eight speakers, these improvements are considered to be quite significant.

Table 22. Performance of slope-dependent weighting with smallest cumulative distance compared to baseline system, $s_T = 1.0$

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_Δ
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_Δ
11	77.51	79.12	1.61
12	72.17	75.22	3.05
13	71.41	76.13	4.72
12-11	-5.34	-3.90	1.44
13-11	-6.10	-2.99	3.11
21	85.03	83.99	-1.04
22	78.95	81.12	2.17
23	76.71	79.77	3.06
22-21	-6.08	-2.87	3.21
23-21	-8.32	-4.22	4.10
31	83.09	81.44	-1.65
32	70.79	76.68	5.89
33	58.16	71.81	13.65
32-31	-12.30	-4.76	7.54
33-31	-24.93	-9.63	15.30
41	89.42	86.92	-2.50
42	73.22	79.58	6.36
43	86.35	82.62	-3.73
42-41	-16.20	-7.34	8.86
43-41	-3.07	-4.30	-1.23
51	77.18	78.92	1.74
52	72.81	76.21	3.40
53	68.75	71.71	2.96
52-51	-4.37	-2.71	1.66
53-51	-8.43	-7.21	1.22
61	86.28	85.69	-0.59
62	72.32	74.62	2.30
63	71.58	74.10	2.52
62-61	-13.96	-11.07	2.89
63-61	-14.70	-11.59	3.11
71	80.05	75.55	-4.50
72	70.52	74.90	4.38
73	67.01	74.24	7.23
72-71	-9.53	-0.65	8.88
73-71	-13.04	-1.31	11.73
81	85.36	83.58	-1.78
82	76.21	78.05	1.84
83	74.13	76.71	2.58
82-81	-9.15	-5.53	3.62
83-81	-11.23	-6.87	4.36
Total Normal	82.99	81.90	-1.09
Total Loud	73.37	77.05	3.67
Total Lombard	71.76	75.89	4.12
Overall	76.04	78.28	2.24

Figures 34 and 35 compare the baseline system and SDW-SCD for speaker #1. There is little distinction in the performance curves of Figure 35 because the abnormal-normal gaps are quite small: $F'_{(12-11)} = -3.90$, and

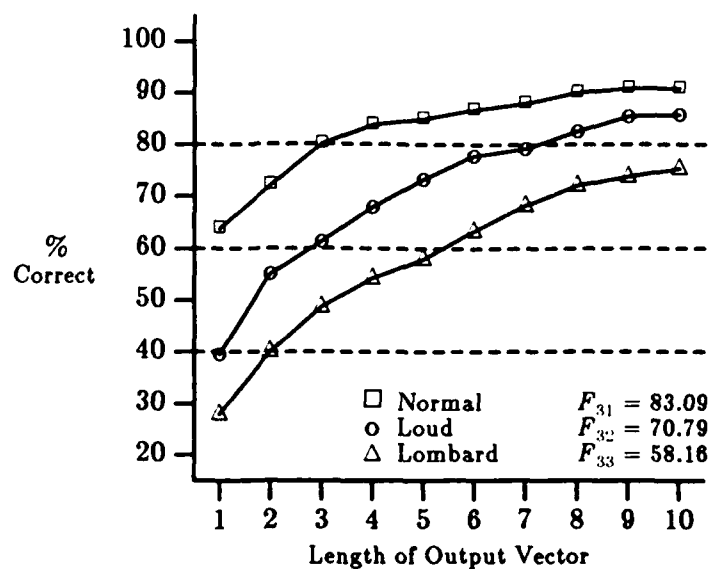


Figure 32. Recognition performance for baseline system, speaker #3

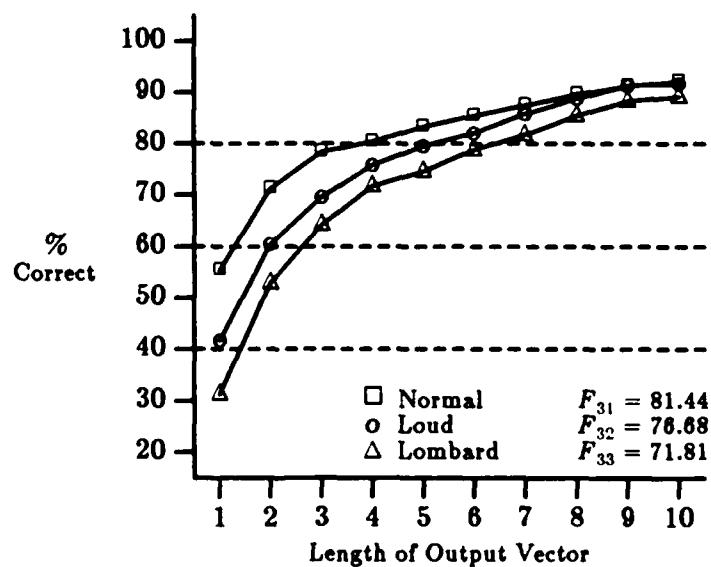


Figure 33. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #3

$F'_{(13-11)} = -2.99$. Obviously this is quite desirable since the objective is to reduce the size of the abnormal-normal gaps to the smallest possible, with the perfect gap size being zero. Note that the performance curve for normal recognition with SDW-SCD again is lower than the corresponding performance curve for the baseline system for $M = 1$, but rises more rapidly for $M \geq 2$, such

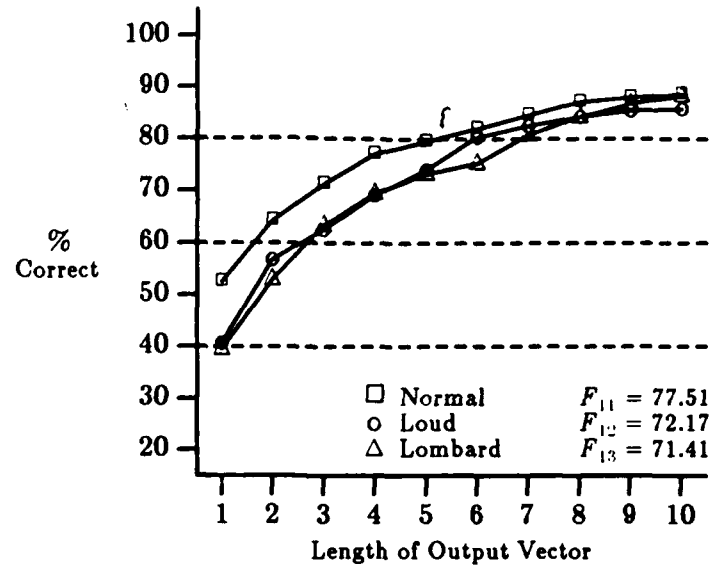


Figure 34. Recognition performance for baseline system, speaker #1

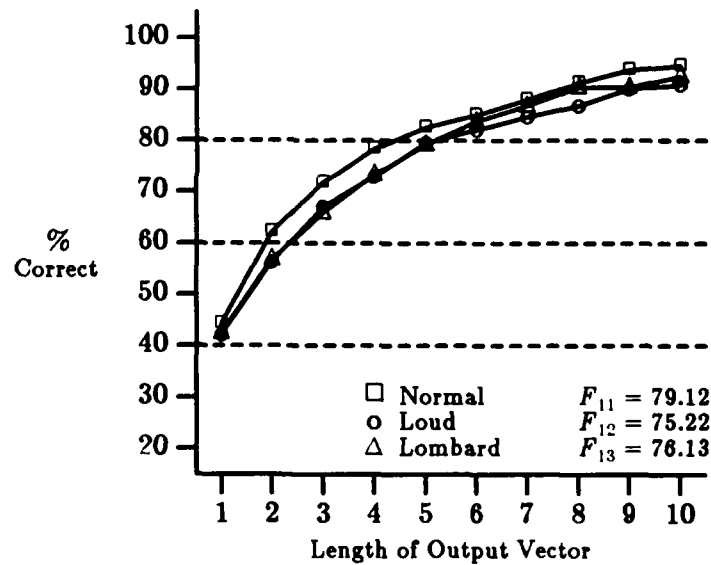


Figure 35. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #1

that the figure of merit shows overall improvement ($F_{\Delta 11} = 1.61$). This phenomenon of tending to reduce performance for low value(s) of M in normal recognition was found to be characteristic of the SCD method across speakers.

Figures 36 and 37 show the baseline system and SDW-SCD, respectively, for speaker #2. Normal recognition was degraded slightly ($F_{\Delta normal} = -1.04$),

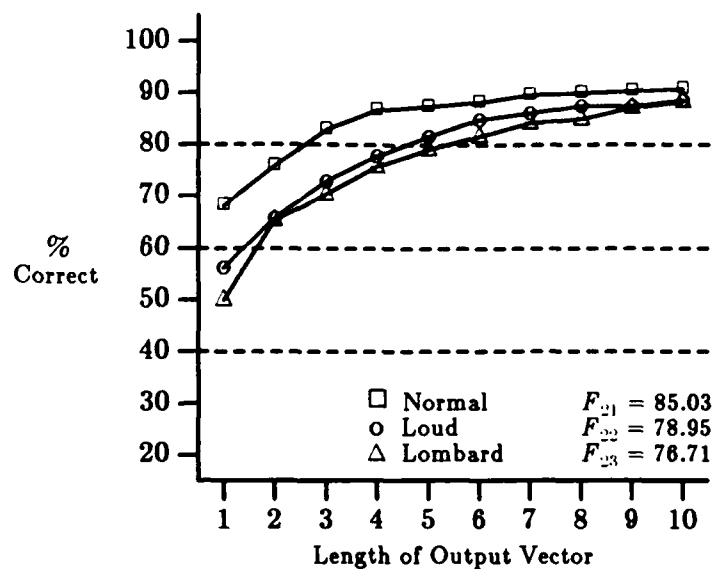


Figure 36. Recognition performance for baseline system, speaker #2

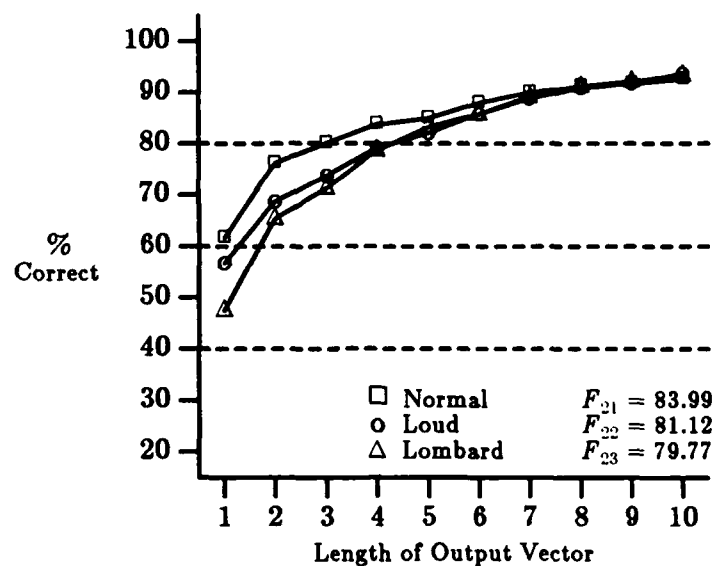


Figure 37. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #2

but the abnormal-normal gaps were reduced to very small values ($F'_{(22-21)} = -2.87$; $F'_{(23-21)} = -4.22$). Most notable is the mergence of the normal, loud, and Lombard performance curves for $M \geq 7$ with the SDW-SCD method. In other words, there is *no* distinction in recognition performance for normal, loud, or Lombard speech for phoneme vector lengths of seven or greater

with speaker #2.

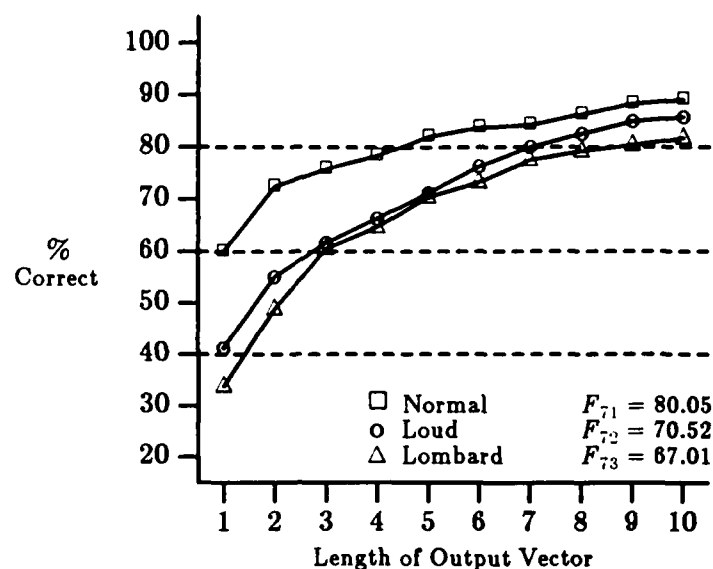


Figure 38. Recognition performance for baseline system, speaker #7

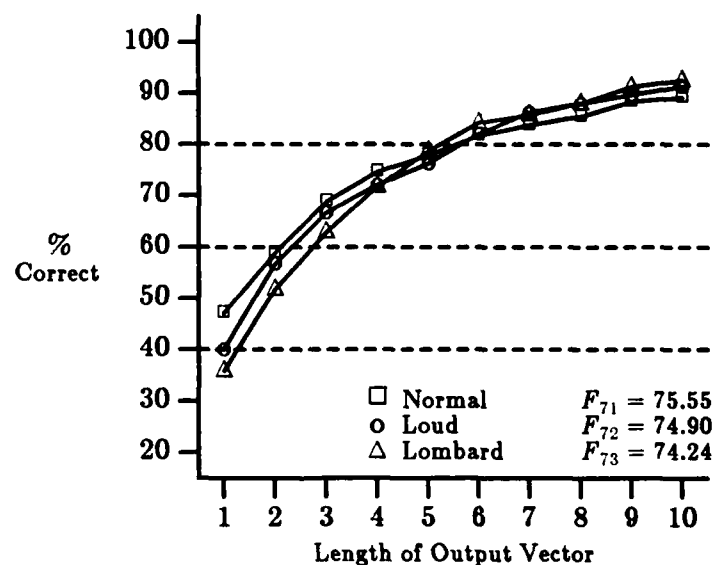


Figure 39. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, speaker #7

Finally, Figures 38 and 39 compare the baseline system and SDW-SCD for speaker #7. The normal recognition was degraded somewhat ($F_{\Delta 71} = -4.50$), but loud and Lombard recognition improved by equal or greater amounts ($F_{\Delta 72} = 4.38$; $F_{\Delta 73} = 7.23$). Of all the speakers, speaker #7 displayed the

smallest gaps between abnormal and normal recognition when the method of SDW-SCD was applied ($F'_{(72-71)} = -0.65$; $F'_{(73-71)} = -1.31$).

Table 23. Phoneme categories for performance comparisons

Vowels		Liquids	Fricatives	Nasals	Stops
EH	AX	L	S	M	P
AO	IH	R	Z	N	T
AA	AE	Y	CH	NX	K
UW	AH	HH	TH		B
ER	OY	EL	F		D
AY	IY	W	SH		G
EY	OW		JH		DX
AW	AXR		V		

9.6 Performance by Phoneme Categories

The performance of the various recognition methods was assessed for the different phoneme categories by grouping all the speakers together, and then separating results by the categories listed in Table 23. The figures of merit, F (absolute only), for each of the recognition methods are listed by phoneme category in Table 24 with the baseline figures of merit in boldface for easy reference. Note that the far right RNN and SCD columns result from combining all phoneme categories. Key comparisons are made in the figures following Table 24.

Figure 40, first of all, shows the overall performance of the baseline recognition system for all speakers and phoneme categories. The abnormal-normal recognition gaps clearly illustrate the degradation caused by loud and Lombard speech in the baseline recognition system. Using the figure of merit, these gaps measure $F_{\Delta(loud-normal)} = -9.63$, and $F_{\Delta(Lombard-normal)} = -11.23$. It is interesting to note that there is a remarkably small difference between loud and Lombard: $F_{\Delta(Lombard-loud)} = -1.60$. Figure 41 is the same as Figure 40, except using SDW-SCD. Note how the abnormal-normal gaps have been visibly reduced, measuring $F_{\Delta(loud-normal)} = -4.86$, and $F_{\Delta(Lombard-normal)} = -6.03$. In other words, the abnormal-normal gaps with SDW-SCD are about *half* the size of the gaps that result in the baseline system.

Figures 42 and 43 compare the performance of the baseline system to SDW-SCD for the stops. While the normal, loud, and Lombard curves merged quite nicely for SDW-SCD, the overall performance dropped for all three conditions compared to the baseline. Clearly, SDW-SCD did not function well

Table 24. Figures of merit for all recognition methods, all speakers, broken down by phoneme category

	Figure of Merit, F					
	stops	nasals	fricatives	liquids	vowels	overall
	RNN SCD	RNN SCD	RNN SCD	RNN SCD	RNN SCD	RNN SCD
Euclidean Measure						
Normal	75.47 64.27	88.05 89.70	89.26 89.00	83.55 76.38	81.95 83.60	83.00 80.67
Loud	72.25 62.01	79.72 83.52	79.66 82.87	73.33 72.02	69.53 77.05	73.37 75.31
Lombard	67.81 62.75	76.66 80.56	78.89 81.73	71.73 68.47	69.03 76.36	71.77 74.19
Cepstral Measure						
Normal	75.48 63.94	88.05 89.64	89.09 88.61	82.46 71.35	80.76 79.62	82.30 78.20
Loud	71.87 61.60	79.16 83.94	79.37 82.52	70.45 65.58	67.57 72.88	71.97 72.55
Lombard	67.30 62.69	75.33 80.15	78.40 81.60	71.22 63.80	67.53 72.07	70.80 71.70
Likelihood Ratio						
Normal	75.80 64.27	88.59 87.56	89.91 89.06	82.79 76.60	81.46 83.32	82.90 80.44
Loud	71.38 62.43	80.26 82.65	79.31 82.40	73.23 71.30	69.19 77.67	73.04 75.36
Lombard	67.24 62.20	77.43 79.94	77.50 81.61	71.54 68.48	69.23 77.01	71.51 74.26
Spectral Slope Estimate						
Normal	67.37 56.76	83.38 90.00	80.87 84.73	77.22 72.95	80.27 80.55	77.90 76.80
Loud	51.78 46.85	70.54 83.25	48.47 60.77	68.05 69.18	66.80 69.94	60.98 64.94
Lombard	49.63 45.62	59.58 75.63	41.59 54.84	65.24 65.97	60.20 61.96	55.36 59.31
Root Power Sums						
Normal	68.01 58.04	83.18 90.96	85.69 88.34	77.35 72.10	81.16 80.18	79.35 77.54
Loud	52.99 48.63	76.18 84.07	59.70 71.97	67.67 68.58	72.74 75.89	66.16 69.84
Lombard	51.04 47.62	65.00 77.57	52.02 65.44	67.67 68.37	67.29 70.67	61.28 65.76
Slope-Dependent Weighting*						
Normal	74.97 65.36	88.32 89.71	90.73 90.43	83.25 78.27	82.12 84.78	83.24 81.90
Loud	71.72 63.77	82.97 84.51	79.26 83.51	74.00 73.24	71.02 79.64	74.12 77.04
Lombard	67.34 62.95	78.40 82.09	78.15 83.07	73.64 71.22	70.90 78.55	72.70 75.87

* $s_T = 0.5$ is used for RNN, and $s_T = 1.0$ is used for SCD.

for this phoneme category.

Figures 44 and 45 show the performance of nasals with the baseline system and SDW-SCD, respectively. Nasals did quite well, improving individually in all three conditions as well as closing the abnormal-normal gaps. For the baseline, the gaps measured $F_{\Delta(loud-normal)} = -8.33$, and $F_{\Delta(Lombard-normal)} = -11.39$, while for SDW-SCD the gaps measured $F_{\Delta(loud-normal)} = -5.20$, and $F_{\Delta(Lombard-normal)} = -7.62$.

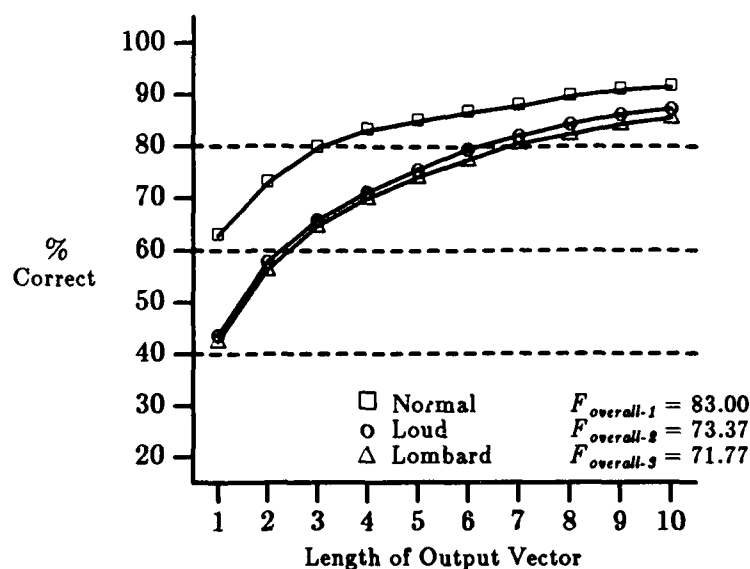


Figure 40. Recognition performance for baseline system, all phonemes, all speakers

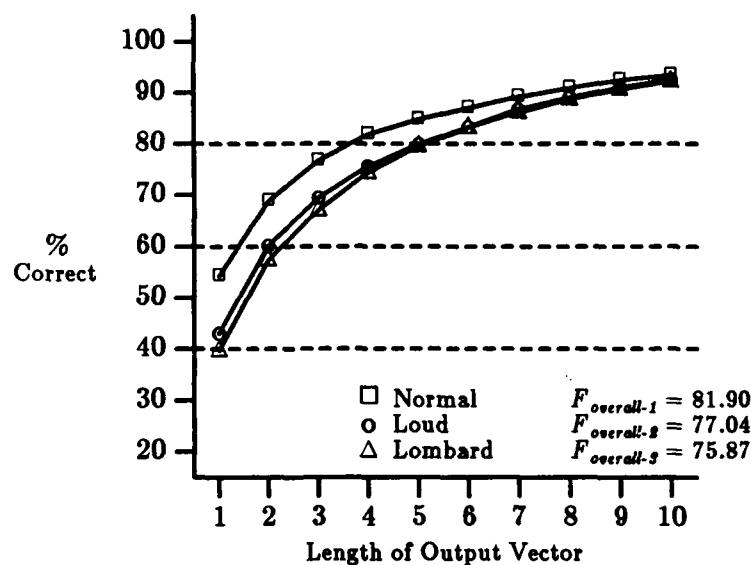


Figure 41. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, all phonemes, all speakers

The performance for fricatives, shown in Figures 46 and 47, was similar to that of the nasals, with all three conditions (normal, loud, and Lombard) improving for SDW-SCD. In fact, the normal curve in Figure 47 has a figure of merit of 90.43, the highest score with SDW-SCD for normal speech of all the

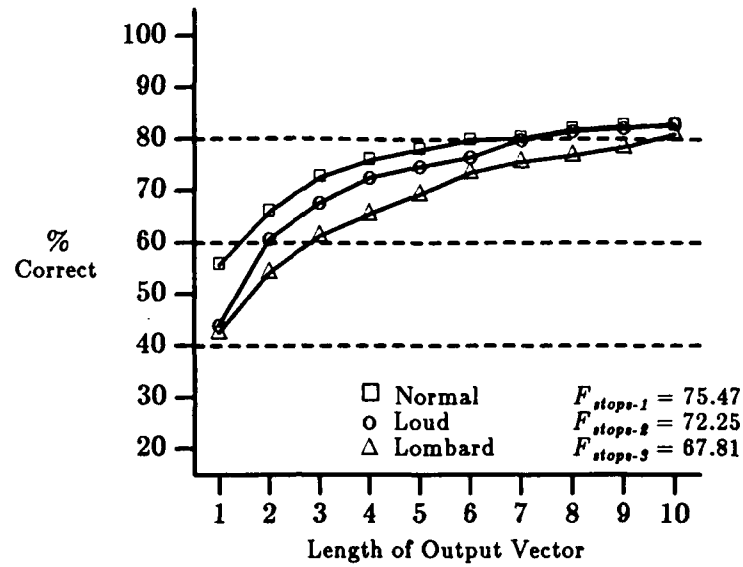


Figure 42. Recognition performance for baseline system, stops, all speakers

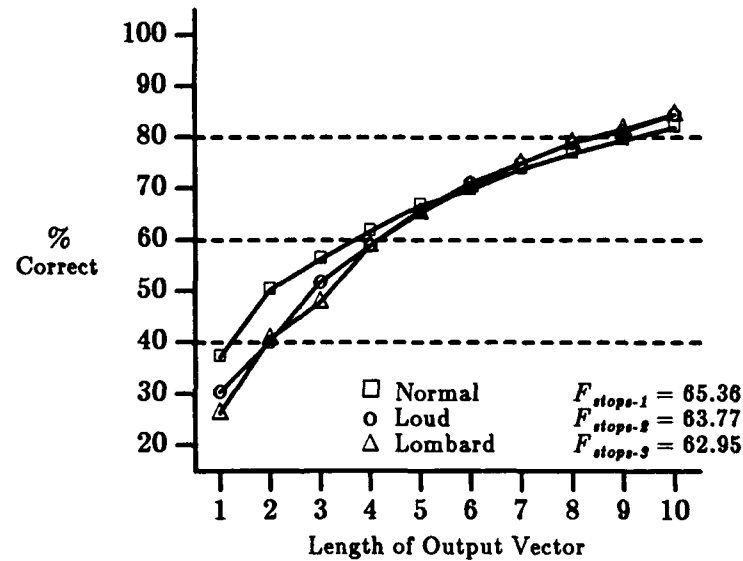


Figure 43. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, stops, all speakers

phoneme categories. Both nasals and fricatives exhibited sharp rises in the normal, loud, and Lombard curves with SDW-SCD for phoneme vector lengths, $M \leq 4$. Again, the abnormal-normal gaps improved from the baseline ($F_{\Delta(loud-normal)} = -9.60$, and $F_{\Delta(Lombard-normal)} = -10.37$) to SDW-SCD ($F_{\Delta(loud-normal)} = -6.92$, and $F_{\Delta(Lombard-normal)} = -7.36$).

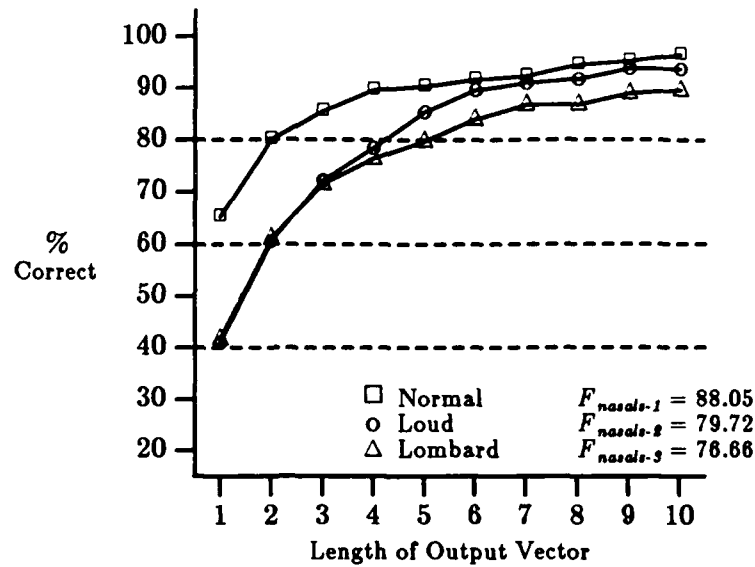


Figure 44. Recognition performance for baseline system, nasals, all speakers

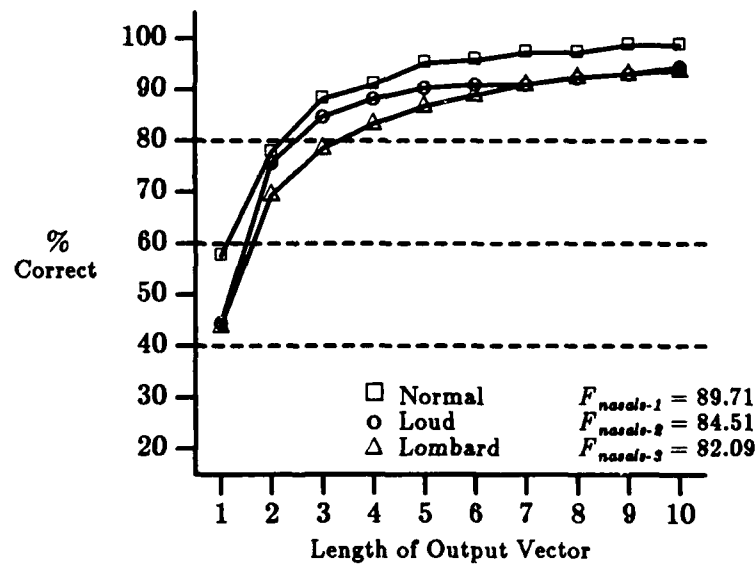


Figure 45. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, nasals, all speakers

Figures 48 and 49 display the performance of liquids with the baseline system and SDW-SCD, respectively. While the abnormal-normal gaps became smaller for SDW-SCD, it was at the expense of a degradation in normal recognition. The figure of merit for normal recognition dropped from $F_{normal} = 83.55$ in the baseline system to $F_{normal} = 78.27$ with SDW-SCD.

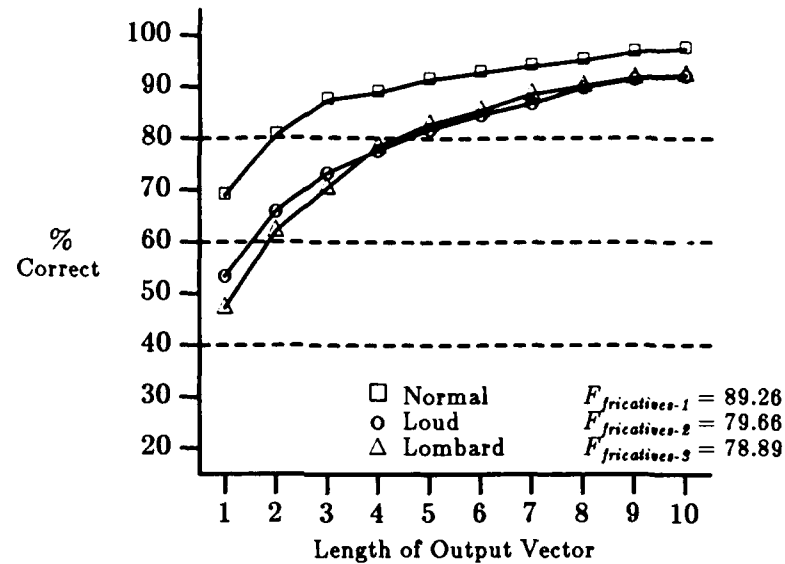


Figure 46. Recognition performance for baseline system, fricatives, all speakers

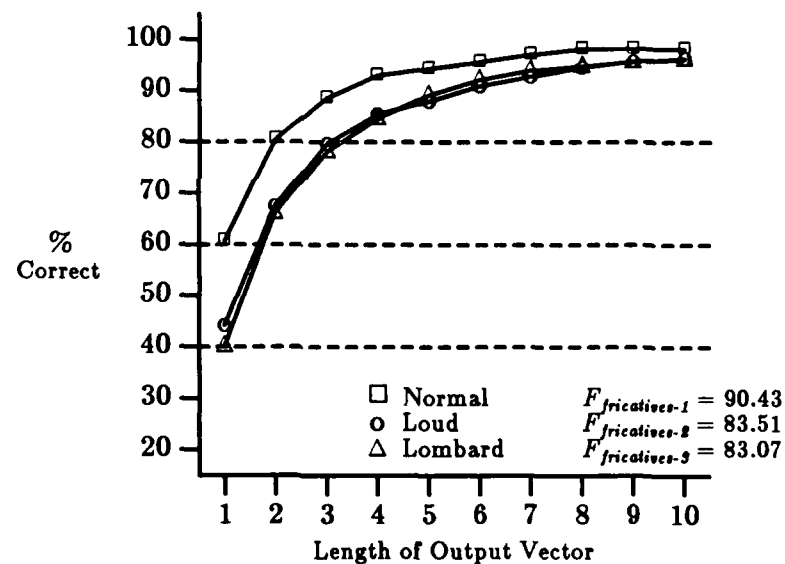


Figure 47. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, fricatives, all speakers

Conversely, the performance for liquids was minimally altered in loud and Lombard speech with the baseline scoring $F_{\text{loud}} = 73.33$, $F_{\text{Lombard}} = 71.73$, and SDW-SCD scoring $F_{\text{loud}} = 73.24$, $F_{\text{Lombard}} = 71.22$.

The performance of vowels is shown in Figures 50 and 51. Recall from Chapter 7 that it was in the vowels where the most reliable energy shifts were

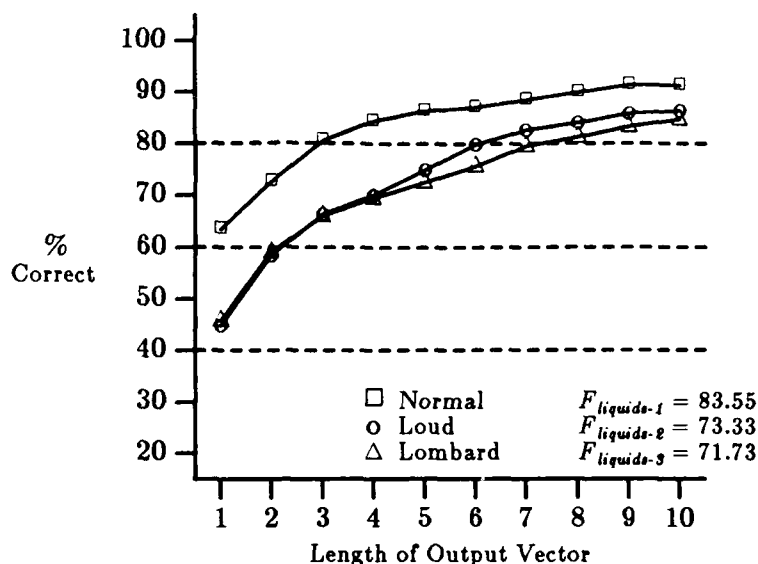


Figure 48. Recognition performance for baseline system, liquids, all speakers

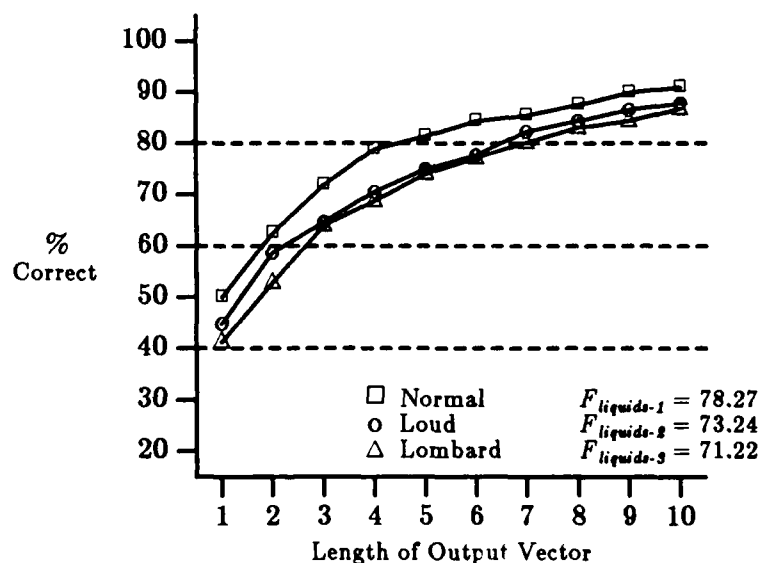


Figure 49. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, liquids, all speakers

observed, owing to the increased vocal effort in the abnormal speech. These energy migrations were the primary motivation in the development of slope-dependent weighting. Figure 51 shows clear improvement in performance with SDW-SCD. In fact, there was overall improvement for all three speech conditions. Figures of merit for the baseline were: $F_{\text{normal}} = 81.95$,

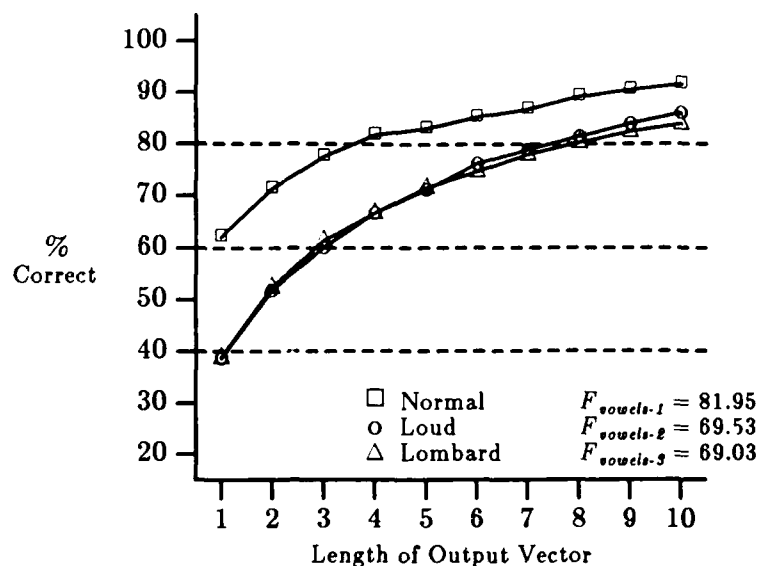


Figure 50. Recognition performance for baseline system, vowels, all speakers

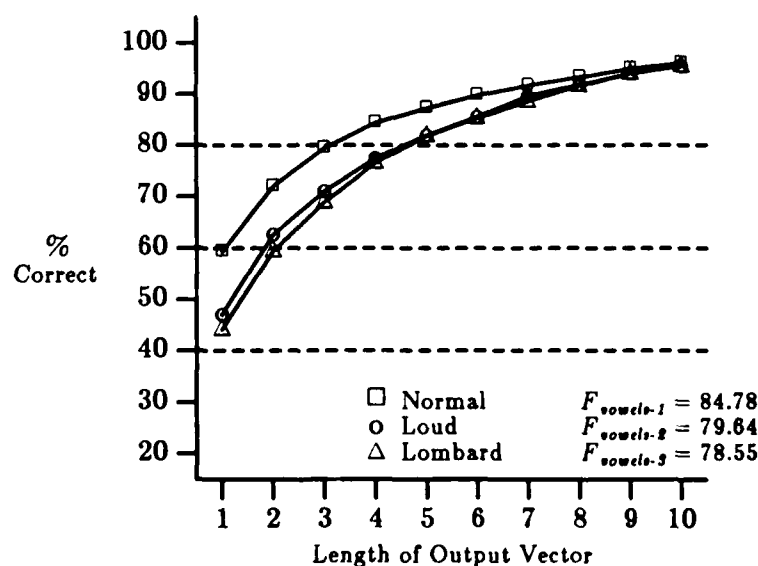


Figure 51. Recognition performance for slope-dependent weighting, $s_T = 1.0$, smallest cumulative distance, vowels, all speakers

$F_{\text{loud}} = 69.53$, and $F_{\text{Lombard}} = 69.03$; and the figures of merit for SDW-SCD were: $F_{\text{normal}} = 84.78$, $F_{\text{loud}} = 79.64$, and $F_{\text{Lombard}} = 78.55$. In addition, the abnormal-normal gaps were essentially reduced to half their original size. For the baseline, the gaps measured $F_{\Delta(\text{loud-normal})} = -12.42$, and $F_{\Delta(\text{Lombard-normal})} = -12.92$, while for SDW-SCD, the gaps measured $F_{\Delta(\text{loud-normal})} = -5.14$,

and

$$F_{\Delta(Lombard-normal)} = -6.23.$$

9.7 Performance in Terms of Error Rate

Up to this point, all results have been discussed in terms of the figure of merit, F , which is a gross summary of the recognition rate (% Correct) versus phoneme vector length ($1 \leq M \leq 10$) curve. Since the phoneme vector length, M , significantly affected recognition rate, it was important to use F in order to capture as much of the performance characteristics as possible. Now we consider the case where the phoneme vector length is fixed at $M = 5$ in order to obtain an example of performance in terms of error rate¹. An error is defined as the case when the correct phoneme is not included in the phoneme candidate vector. The results are broken down as in the previous section, that is, all speakers are grouped together, and distinctions are made by phoneme category.

Error rates for the case, $M = 5$, are listed in Table 25 for the various recognition methods and phoneme categories. As in Table 24, the baseline figures are in boldface for easy reference. Overall, the baseline error rates were 15.0%, 24.4% and 25.8% for normal, loud, and Lombard, respectively. These baseline performances are compared to the SDW-SCD error rates of 15.3%, 20.1% and 20.5% for normal, loud, and Lombard respectively. For SDW-SCD, the error rate for normal speech degraded slightly, but the error rates for abnormal speech decreased from about 25% to 20%, or by one-fifth. Noting that the difference in error rates between abnormal and normal speech for the baseline system was about 10%, this difference was cut in half with SDW-SCD. These results are in agreement with the assessments using figure of merit, F .

Results for the individual phoneme categories are also in line with previous sections. To simplify discussion, error rates in this paragraph are listed in triplets for the three speech conditions (normal, loud, and Lombard) unless otherwise noted. For the stops, the baseline error rates were 22.0%, 25.3%, and 30.7%. Errors increased for SDW-SCD, with very little distinction among

1. $M = 5$ was chosen as a tradeoff between minimizing the size of the phoneme vector and maximising the chance that the correct phoneme was included in the vector. Note that in most cases, the performance curves are relatively flat for $M > 5$, meaning that very little improvement in performance is gained for these higher values of M .

Table 25. Error rates for the case, $M = 5$, for all recognition methods, all speakers, broken down by phoneme category

	Error Rate, %											
	stops		nasals		fricatives		liquids		vowels		overall	
	RNN	SCD	RNN	SCD	RNN	SCD	RNN	SCD	RNN	SCD	RNN	SCD
Euclidean Measure												
Normal	22.0	35.1	9.7	3.5	8.6	6.8	13.6	20.5	16.8	13.6	15.0	16.2
Loud	25.3	36.9	14.6	9.7	18.2	11.2	25.0	26.4	28.8	20.5	24.4	21.6
Lombard	30.7	36.6	20.2	15.3	17.2	13.0	27.4	27.4	28.4	20.9	25.8	22.6
Cepstral Measure												
Normal	22.6	35.4	9.7	3.5	8.6	8.1	14.9	25.0	17.6	17.4	15.7	18.8
Loud	25.6	36.9	14.6	9.7	18.0	11.7	27.8	32.6	31.1	25.3	25.8	24.6
Lombard	31.3	37.2	21.6	14.6	18.5	12.7	27.1	32.6	30.1	25.3	26.9	25.1
Likelihood Ratio												
Normal	22.3	34.5	9.7	10.4	8.1	6.5	15.3	19.8	17.3	14.1	15.5	16.7
Loud	25.9	36.3	14.6	11.1	19.0	12.5	25.0	26.4	28.1	19.7	24.4	21.5
Lombard	31.6	35.1	18.1	13.9	18.0	12.8	26.0	28.5	28.9	21.1	25.9	22.5
Spectral Slope Estimate												
Normal	30.9	40.8	12.5	5.6	17.7	10.4	22.6	25.7	17.4	17.6	20.3	20.5
Loud	47.9	52.4	21.5	10.4	50.3	37.0	29.2	28.1	31.4	27.3	37.0	32.5
Lombard	49.1	55.1	39.6	18.0	57.0	42.7	32.6	31.9	37.8	36.3	43.0	38.8
Root Power Sums												
Normal	30.7	39.9	14.6	3.5	11.7	7.5	21.2	25.4	17.2	17.5	18.9	19.5
Loud	44.6	52.1	18.1	10.4	38.5	24.7	30.2	29.2	24.1	20.3	31.0	27.3
Lombard	46.4	50.6	35.4	18.1	46.3	32.5	30.5	28.8	31.0	25.9	37.0	31.4
Slope-Dependent Weighting*												
Normal	23.8	33.3	9.0	4.9	7.3	5.7	15.3	18.7	16.5	12.8	15.2	15.3
Loud	25.0	34.3	11.1	9.7	18.0	12.0	24.0	25.0	26.4	18.1	23.0	20.1
Lombard	32.4	34.8	16.7	13.2	18.5	10.9	23.6	26.0	26.4	18.2	24.7	20.5

* $s_T = 0.5$ is used for RNN, and $s_T = 1.0$ is used for SCD.

speech conditions (33.3%, 34.3%, and 34.8%). Error rates for nasals in the baseline system were 9.7%, 14.6%, and 20.2%. By applying SDW-SCD, these rates decreased to 4.9%, 9.7%, and 13.2%. Note that while the errors for each individual condition decreased, there were only minor improvements in discrepancies between normal and abnormal error rates. The fricatives in the baseline system had error rates of 8.6%, 18.2%, and 17.2%. These improved to 5.7%, 12.0%, and 10.9% using SDW-SCD. For liquids, the baseline error rate for normal speech of 13.6% degraded to 18.7% with SDW-SCD. Abnormal

speech was minimally affected however. The baseline error rates for liquids were 25.0% and 27.4% for loud and Lombard, respectively, and the SDW-SCD error rates were 25.0% and 26.0% for loud and Lombard, respectively. As noted before with the figure of merit, F , the vowels posted considerable gains with the method of SDW-SCD, and this is reiterated by using error rates as a method of comparison. The vowels posted baseline error rates of 16.8%, 28.8%, and 28.4%. By applying SDW-SCD, these error rates dropped substantially to 12.8%, 18.1%, and 18.2%. The error rates improved for all three speech conditions, and the gaps in abnormal-normal recognition were cut in half. Error rates for the other experiments in this research are included in Table 25 for reference.

9.8 Summary

The major goal of this research was to reduce the discrepancy in recognition performance between normal and abnormal speech, given that reference templates were derived only from normal speech. The baseline recognition system using direct computation of Euclidean distance was first tested against cepstral measurement of Euclidean distance and the likelihood ratio metric for LPC coefficients. Both cepstral measure and likelihood ratio performed worse than the baseline system in all three speech conditions, with cepstral measure exhibiting four times more degradation than the likelihood ratio. The slope of the log magnitude LPC spectrum was then chosen as an attribute to be exploited in reducing the gaps between normal and abnormal recognition. The first attempt used Euclidean distance of a spectral slope estimate derived from the log magnitude LPC spectrum. This method performed quite poorly, scoring -11.30 in overall figure of merit compared to the baseline system. The second attempt used the method of root power sums to calculate the difference in spectral slope between templates. Root power sums outperformed the spectral slope estimate method, but still registered much worse than the baseline system with an overall score of -7.11. A method was then devised that used the difference in spectral slope between LPC log magnitude spectra to weight the point-by-point energy differences between the spectra. The non-linear weighting function used the slope difference threshold, s_T , to determine whether or not energy weighting would be invoked. For several threshold values tested with the baseline system, it was found that $s_T = 0.5$ gave the best performance with an overall improvement in figure of merit ($F_G = 0.65$). Finally, the distances of all reference tokens of like phonemes were combined to form a *smallest cumulative distance* method of recognition in contrast to the *raw nearest neighbor* method used in the baseline system. SCD

performed worse for the cepstral measure of Euclidean distance, but otherwise registered improvements over RNN for the baseline Euclidean measure, likelihood ratio, spectral slope estimate, and root power sums. These net improvements resulted from significant gains in the performance of abnormal speech even though SCD caused moderate degradation in the performance of normal speech. When SCD was combined with the method of slope-dependent weighting (SDW), the most significant success was obtained. Among the threshold values tested, $s_T = 1.0$ was found to give the best performance with an overall improvement in figure of merit of $F_G = 2.24$. Numerous graphs were compared to illustrate the improvement in performance due to the method of SDW-SCD. Performance of all recognition experiments was then broken down by phoneme category. The nasals, fricatives, and vowels responded favorably to SDW-SCD, while performance was degraded somewhat for liquids, and significantly for stops. Finally, performance was expressed in terms of error rates for the case where the phoneme vector length, M , was set to five. Results were in agreement with those obtained using the figure of merit, F . SDW-SCD was found to reduce the difference in error rate between normal and abnormal speech by approximately 50%.

10. CONCLUSIONS

This research focused on the problem of recognizing loud and Lombard speech in the fighter cockpit environment, given that training was accomplished only on normal speech. The problem is of critical interest to the United States Air Force in view of the need for robust speech recognition systems to improve the interface between man and machine. Normal, loud, and Lombard speech data was obtained for this research from the Armstrong Aerospace Medical Research Laboratory at Wright Patterson Air Force Base, Ohio. Over 17500 phonemes were digitized and hand-labelled from eight different speakers.

The first of the two phases of this research consisted of a detailed analysis of the speech data to determine reliable differences between normal and abnormal speech. Eighteen different features were analyzed for each of the 40 phonemes in the lexicon. The most reliable differences were found to be in the spectral energies of the various frequency bands. Specifically, it was discovered that there was a consistent migration of energy in the sonorants out of the 0-500Hz and 4k-8kHz ranges, and into the 500-4kHz range. Considering all eight speakers combined, the average loss of energy in band 1 (0-250Hz) was 2.41 dB for loud speech and 1.23 dB for Lombard speech; for band 10 (7k-8kHz) the average loss was 1.45 dB for loud speech and 1.36 dB for Lombard speech. For both loud and Lombard speech, the largest increases were in band 5 (2k-3kHz) and ranged from 1.3 dB to 2.3 dB across the vowels.

The second phase of the research was devoted to the development of a method that would reduce the performance differences in the recognition of normal, loud and Lombard speech. All experiments were compared to a baseline recognition system using Euclidean distances between log magnitude LPC spectra. The performance of the baseline system compared favorably with the use of cepstral coefficients and the likelihood ratio. During the verification of the baseline system, it was found that the likelihood ratio was superior to cepstral coefficients, and that the baseline system was marginally better than the likelihood ratio.

Two methods of spectral slope distance (spectral slope estimate, and root power sums) were tested against the baseline system. The use of spectral slope as a distance measure was found to be inferior to the baseline system for the recognition of normal speech. In addition, spectral slope distance aggravated the degradation in recognition performance for loud and Lombard speech. The worsening of the discrepancy in performance between normal and abnormal speech is most likely attributable to the sensitivity of spectral slope distance to frequency shifts in the formants.

A distance measure was then developed that was designed to exploit the new knowledge of energy migrations in the sonorants discovered in the first phase of this research. With the hypothesis that similarity in spectral slope could be used as an indicator of energy shift due to abnormal speech, the Euclidean distances between spectral samples were weighted by the dissimilarity in spectral slope at the given frequency. If spectral slope was dissimilar, then the energy difference between spectra was fully weighted, and if the spectral slope was similar then the energy difference was reduced or eliminated under the premise that energy differences in regions of similar slope were due to the energy shifts in abnormal speech. The boundary between similar and dissimilar slope values was determined by a threshold value, s_T , in the weighting function. This method, termed slope-dependent weighting or SDW, performed better than the baseline system, and was optimized with a value of $s_T = 0.5$.

Recognition tests used two methods of managing the five reference templates for each phoneme in the lexicon. The first method, called raw nearest neighbor or RNN, treated each template independently. The distances from the test template to each reference template were preserved individually, and the ranking of the best-scoring template that matched the test template determined recognition performance. The other method, called smallest cumulative distance or SCD, used the performance of all five tokens of a given phoneme collectively to determine ranking. In general, SCD improved the recognition of loud and Lombard speech, but degraded to varying degrees the recognition of normal speech. For all except the cepstral measure, SCD provided gains in abnormal speech recognition that outweighed the losses in normal speech recognition.

The use of slope-dependent weighting with smallest cumulative distance best achieved the goals of this research. This combination of methods provided the most improvement in abnormal recognition with minimal degradation in normal recognition. The discrepancy between normal and abnormal recognition performance from the baseline system was cut in half with SDW-SCD. Broken down by phoneme categories, SDW-SCD performed quite well with vowels,

nasals, and fricatives, exhibited moderate degradation for liquids, and significant degradation for stops.

Table 26. Rank order of recognition methods from best to worst, according to the overall figure of merit

Method	F_G	Error Rate (%), $M=5$			% Change in Gap	
		Normal	Loud	Lombard	Loud	Lombard
1. Slope-Dependent Weighting, SCD ($s_T = 1.0$)	2.24	15.3	20.1	20.5	48.9	51.9
2. Baseline Euclidean, SCD	0.68	16.2	21.6	22.6	42.6	40.7
3. Likelihood Ratio, SCD	0.65	16.7	21.5	22.5	48.9	46.3
4. Slope-Dependent Weighting, RNN ($s_T = 0.5$)	0.65	15.2	23.0	24.7	17.0	12.0
5. Baseline Euclidean, RNN	0.0	15.0	24.4	25.8	0.0	0.0
6. Likelihood Ratio, RNN	-0.23	15.5	24.4	25.9	5.3	3.7
7. Cepstral Measure, RNN	-1.01	15.7	25.8	26.9	-7.4	-3.7
8. Cepstral Measure, SCD	-1.89	18.8	24.6	25.1	38.3	41.7
9. Root Power Sums, SCD	-4.99	19.5	27.3	31.4	17.0	-10.2
10. Root Power Sums, RNN	-7.11	18.9	31.0	37.0	-28.7	-67.6
11. Spectral Slope Estimate, SCD	-9.03	20.5	32.5	38.8	-27.7	-69.4
12. Spectral Slope Estimate, RNN	-11.30	20.3	37.0	43.0	-77.7	-110.2

In summary, there were several significant contributions in this research. First of all the differences between normal, loud, and Lombard speech have never been quantified to the level of detail reported in this work. The analyses in Chapter 7 clarify and expand previous studies that used limited data sets and feature sets. Second, the differences in recognition performance between normal, loud, and Lombard speech were quantified for a number of recognition methods, and the relative performances of these methods were established. Table 26 summarizes the rankings of the various recognition methods according to figure of merit. Error rates for $M = 5$ are also shown along with the ratios of change in error rate for abnormal speech. Third, a new method, SDW-SCD, was developed that improves the recognition of loud and Lombard speech using the training of *only* normal speech. SDW-SCD improves robustness by reducing the discrepancy in recognition performance between normal and abnormal speech by

approximately 50%. And finally, this research contributed to the overall understanding of the causes of recognition errors in abnormal speech. By quantifying the energy migrations in the sonorants and discovering a way to exploit the similarity in spectra of different speech conditions, a method was devised to reduce error rates in the recognition of abnormal speech without having to explicitly train on the abnormal speech.

There are several activities that would further the findings in this research:

1. Test the method of SDW-SCD on actual speech under stress from the cockpit environment.
2. Test additional threshold values in the range $0 < s_T < 2$ to more completely graph the effects of threshold value on RNN and SCD.
3. Incorporate additional speakers into the data base.
4. Incorporate additional speech conditions into the data base.
5. Incorporate a system for automatic segmentation and labelling.
6. Search for algorithms to streamline SDW-SCD.
7. Incorporate higher knowledge sources into the recognition system.
8. Explore parallel computing implementations of SDW-SCD.

LIST OF REFERENCES

LIST OF REFERENCES

- [AFA87] (no author cited) "What's Happening at ASD," *Air Force Magazine*, January 1987.
- [At71] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave" *Journal of the Acoustical Society of America*, Vol 50, 1971.
- [At74] B. S. Atal, "Linear prediction for speaker identification," *Journal of the Acoustical Society of America*, Vol 55, 1974.
- [B77] B. Beek, E. P. Neuberg, and D. C. Hodge, "An Assessment of the Technology of Automatic Speech Recognition for Military Applications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-25, No 4, August 1977.
- [Ba75] J. K. Baker, "The DRAGON System -- An Overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-23, No 1, February 1975.
- [Ba86] J. M. Baker and D. F. Pinto, "Optimal and Suboptimal Training Strategies for Automatic Speech Recognition in Noise, and the Effects of Adaptation on Performance," presented at the DARPA Workshop on Speech Recognition, Palo Alto, CA, February 1986.
- [Bl80] L. T. Blank, *Statistical Procedures for Engineering, Management, and Science*, McGraw-Hill Inc., 1980.
- [Bog63] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-Cepstrum, and Saphe Cracking," in *Proc. Symp. Time Series Analysis*, Edited by M. Rosenblatt, John Wiley & Sons, New York, 1963.
- [Bo86] Z. S. Bond, T. J. Moore, and T. R. Anderson, "The Effects of High Sustained Acceleration on the Acoustic Phonetic Structure of Speech: A preliminary Investigation," Aerospace Medical Research Laboratory Technical Report AAMRL-TR86-011, May 1986.

- [Bu74] R. R. Burton, S. D. Leverett, and E. D. Michaelson, "Man at High Sustained +Gz Acceleration: A Review," *Aerospace Medicine*, 45, 1974.
- [Ch88] Y. Chen, "Cepstral Domain Talker Stress Compensation for Robust Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-36, No 4, April 1988.
- [Co82] C. R. Coler, R. P. Plummer, and E. M. Heff, "Automatic Speech Recognition and Main Computer Interaction at DASA -- Ames Research Center," NASA -- American Research Center, Mountain View, CA, 1982.
- [Cy86] D. S. Cyphers, R. H. Kassel, D. H. Kaufman, H. C. Leung, M. A. Randolph, S. Seneff, J. E. Unverferth, III, T. Wilson, and V. W. Zue, "The Development of Speech Research Tools on MIT's Lisp Machine-Based Workstations," presented at the DARPA Workshop on Speech Recognition, Palo Alto, CA, February 1986.
- [D85] G. Doddington, "The DARPA Recognition Research and Evaluation Data Base," Message written to AAMRL, NBS, et al., 7 August 1985.
- [Da84] J. R. Davis, D. A. Ratino, R. E. Van Patten, D. W. Repperger, and J. W. Frazier, "Performance and Physiological Effects of Multiple, Sequential, +Gx Acceleration Exposures (Space Plane Boost Profiles)," Aerospace Medical Research Laboratory Technical Report AFAMRL-TR-84-012, 1984.
- [Do74] N. M. Downie and R. W. Heath, *Basic Statistical Methods, Fourth Edition*, Harper and Row, 1974.
- [Dr58] J. J. Dreher and J. J. O'Neill, "Effects of Ambient Noise on Speaker Intelligibility of Words and Phrases," *Laryngoscope*, Vol 68, 1958.
- [Er80] L. D. Erman, "The Hearsay-II Speech Understanding System: A Tutorial," in *Trends in Speech Recognition*, Edited by W. A. Lea, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [F60] G. Fant, *Acoustic Theory of Speech Production*, Mouton, 1960.
- [F73] G. Fant, *Speech Sounds and Features*, MIT Press, 1973.
- [Fl72] J. L. Flanagan, *Speech Analysis Synthesis and Perception, Second Edition*, Springer-Verlag, 1972.
- [Fu72] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1972.

- [G76] A. H. Gray Jr. and J. D. Markel, "Distance Measures for Speech Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-24, No 5 October 1976.
- [Ha49] T. D. Hanley and M. D. Steer, "Effect of Level of Distracting Noise Upon Speaking Rate, Duration, and Intensity," *Journal of Speech and Hearing Disorders*, Vol 14, 1949.
- [Han87] B. A. Hanson and H. Wakita, "Spectral Slope Distance Measures with Linear Prediction Analysis for Word Recognition in Noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-35, No 7 July 1987.
- [He83] W. Hess, *Pitch Determination of Speech Signals*, Springer-Verlag, 1983.
- [Ho83] R. Hockey, *Stress and Fatigue in Human Performance* Wiley, 1983.
- [Hv88] D. S. Harvey, "Talking with Airplanes," *Air Force Magazine* January 1988.
- [I75] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-23, No 1, February 1975.
- [J82] F. Jelinek, "Self-Organized Continuous Speech Recognition," IBM T. J. Watson Research Center, April 1982.
- [Ke82] Z. A. Kersteen, "An Evaluation of Automatic Speech Recognition Under Three Ambient Noise Levels," presented at the Workshop on Standardization for Speech I/O Technology, National Bureau of Standards, Gaithersburg, Maryland, March 1982.
- [Kl85] D. H. Klatt and C. Aoki, "Comparison of Several Spectral Distance Metrics," in Summary of Research in Speech Recognition, Massachusetts Institute of Technology, Cambridge, MA, November 1985.
- [L80] N. E. Lane, "Conversations with Weapons Systems: Crewstation Applications of Interactive Voice Technology," *Yearbook on Navy Manpower, Personnel and Training Research and Development*, 1980.
- [La70] H. L. Lane, B. Tranel, and C. Sisson, "Regulation of Voice Communication by Sensory Dynamics," *Journal of the Acoustical Society of America*, Vol 47, 1970.
- [La71] H. L. Lane and B. Tranel, "The Lombard Sign and the Role of Hearing in Speech," *Journal of Speech and Hearing Research*, Vol 14, 1971.

- [Le80] W. A. Lea, "Speech Recognition: Past, Present, and Future," in *Trends in Speech Recognition*, Edited by W. A. Lea, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [Le80-2] W. A. Lea, "Prosodic Aids to Speech Recognition", in *Trends in Speech Recognition*, Edited by W. A. Lea, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [Lo76] E. J. Lovesey, "The Development of Aircraft Instruments," in *Visual Presentation of Cockpit Information Including Special Devices Used for Particular Conditions of Flying*, NATO/AGARD Conference Proceedings CP201, November 1976.
- [Lom11] E. Lombard, "Le signe de l'elevation de la voix," *Ann. Maladiers Oreille, Larynx, Nez, Pharynx*, 37, 1911.
- [M70] J. E. McGrath, "A Conceptual Formulation for Research on Stress," in *Social and Psychological Factors in Stress*, Edited by J. E. McGrath, Holt, Rinehart & Winston, 1970.
- [MG76] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, 1976.
- [Mo87] T. J. Moore and Z. S. Bond, "Acoustic-Phonetic Changes in Speech due to Environmental Stressors: Implications for Speech Recognition in the Cockpit," *Proceedings of the Fourth International Symposium on Aviation Psychology*, Columbus, Ohio, April 1987.
- [N75] A. Newell et al., *Speech Understanding Systems*, Academic Press, 1975.
- [Ni80] Nils J. Nilsson, "*Principles of Artificial Intelligence*," Tioga, Palo Alto, CA, 1980.
- [O68] A. V. Oppenheim, R. W. Schafer, and T. C. Stockham, "Non-linear filtering of multiplied and convolved signals," *Proceedings of the IEEE*, Vol 56, pp. 1264-1291, Aug 1968.
- [OS75] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [Pal82] K. K. Paliwal, "On the performance of the quefrency-weighted cepstral coefficients in vowel recognition," *Speech Communication*, May, 1982.
- [Pa86] D. B. Paul, R. P. Lippmann, Y. Chen, C. J. Weinstein, "Robust HMM-Based Techniques for Recognition of Speech Produced Under Stress and in Noise," presented at the DARPA Workshop on Speech Recognition, Palo Alto, CA, February 1986.

- [Pi85] D. B. Pisoni, R. H. Bernacki, H. C. Nusbaum, and M. Yuchtman, "Some Acoustic-Phonetic Correlates of Speech Produced in Noise," *Proceedings, International Conference on Acoustics, Speech, and Signal Processing '85*, Tampa, FL, March 1985.
- [R76] D. R. Reddy, "Speech Recognition by Machine: A Review," *Proceedings of the IEEE*, Vol 64, No 4, April 1976.
- [Ra85] P. K. Rajasekaran and G. R. Doddington, "Speech Recognition in the F-16 Cockpit Using Principal Spectral Components," *Proceedings, International Conference on Acoustics, Speech, and Signal Processing '85*, Tampa, FL, March 1985.
- [Ra86] P. K. Rajasekaran and G. R. Doddington, "Recognition of Speech Under Stress and in Noise," *Proceedings, International Conference on Acoustics, Speech, and Signal Processing '86*, Tokyo, Japan, April 1986.
- [RD86] P. K. Rajasekaran and G. R. Doddington, "Initial Results and Progress," presented at the DARPA Workshop on Speech Recognition, Palo Alto, CA, February 1986.
- [RJ86] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, Vol 3, No 1, January 1986.
- [Ro83] A. Rollins and J. Wiesen, "Speech Recognition and Noise," *Proceedings, International Conference on Acoustics, Speech, and Signal Processing '83*, Boston, MA 1983.
- [RS78] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
- [S74] T. B. Sheridan and W. R. Ferrell, *Man-Machine Systems: Information, Control, and Decision Models of Human Performance*, MIT Press, 1974.
- [Sa78] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-26, No 1, February 1978.
- [Sc81] M. R. Schroeder, "Direct (Nonrecursive) Relations Between Cepstrum and Predictor Coefficients," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-29, No 2 April 1981.
- [Se84] S. Seneff, "Pitch and Spectral Estimation of Speech Based on Auditory Synchrony Model," *Proceedings, International Conference on Acoustics, Speech, and Signal Processing '84*, San Diego, CA, March 1984.

- [Se85] Stephanie Seneff, "A Computational Model for the Peripheral Auditory System: Application to Speech Recognition Research," in Summary of Research in Speech Recognition, Massachusetts Institute of Technology, Cambridge, MA, November 1985.
- [Sha49] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL, 1949.
- [Sh78] G. R. Sharp and J. Ernsting, "The Effects of Long Duration Acceleration," in *Aviation Medicine, Vol I, Physiology and Human Factors*, Edited by G. Dhenin and J. Ernsting, Tri-Med, London, 1978.
- [Sho80] June E. Shoup, "Phonological Aspects of Speech Recognition," in *Trends in Speech Recognition*, Edited by W. A. Lea, Prentice-Hall Inc, Englewood Cliffs, NJ, 1980
- [U86] E. Ulsamer, "Hard Calls on Tactical Technology," *Air Force Magazine*, April 1986.
- [W74] A. T. Welford, "Stress and Performance." in *Man Under Stress* Edited by A. T. Welford, Wiley, 1974.
- [We62] J. C. Webster and R. G. Klumpp, "Effects of Ambient Noise and Nearby Talkers on a Face-to-Face Communication Task," *Journal of the Acoustical Society of America*, Vol 34, 1962.
- [Z85] V. W. Zue, "The Use of Speech Knowledge in Automatic Speech Recognition," *Proceedings of the IEEE*, Vol 73, No 11, November 1985.

APPENDICES

Appendix A: Applications of Voice Interaction in the AFTI F-16

1. Selecting communication and navigation frequencies
2. Selecting navigation steerpoints
3. Entering steerpoint data
4. Entering IFF data
5. Control of cockpit displays
6. Control of radar mode, range, azimuth, and gain
7. Entering altitude, tracker, and fuel advisories
8. Obtaining weapon status, fuel status, and airspeed
9. Selecting weapon release interval
10. Selecting master attack mode
11. Implementing preset configurations

Appendix B: Lisp Code for Symbolics 3870

```
;;; -*. Syntax: Zetalisp; MODE: LISP; Package: ZL-SPI; Base: 10 -*.
```

```
;;;          CONVERT-WAVE-TO-UTT
;;;          6 Oct 87
```

```
;;;      Purdue customization routine for SPIRE
```

```
;;; This function is a derivative of read-utterance-wav-2
;;; originally provided by kaufman@mit. It is a utility
;;; function designed to convert a digitized wave form into
;;; an utterance file for Spire. The new utterance is returned
;;; by this function. There is also a user version that provides
;;; menu interaction. The binary file can reside either on the
;;; LMFS or in one of the UNIX directories on ECN. If the file
;;; is on ECN, the full pathname must be supplied in the following
;;; manner:
```

```
;;;      "ef:///f//stanton//digitized-speech-file"
```

```
;;; Note that the slashes must be doubled. By the way the
;;; code is currently written, the file must be on one of the
;;; "e*" machines (e.g. ec, ed, ee, ef, eg, eh, ei, el, en).
;;; More elegant treatment can be effected, given the
;;; time to hack around. Beware, that the Goulds (ei and en)
;;; must have the bytes swapped around!
```

```
;;; Also note that in file transfers from UNIX, the byte-size
;;; must be set to 16 and mode to binary (same as characters nil).
```

```
;;; Infile is the pathname for the file on the host machine.
;;; Utfile is the pathname for the utterance file to be created.
;;; Rate is the sampling frequency.
;;; Mag is a magnification factor for the wave form.
```

```
;;; In the One-Dimensional Signal Processing Laboratory at Purdue
;;; we digitize to 12-bit samples with sign-extension to 16-bit
;;; words; therefore some scaling needs to be done since Spire
;;; expects full 16-bit magnitudes.
;;;          bjs
```

```
(defun convert-wave-to-utt (infile &optional
                           (utfile "tmp.utt")
                           (rate 16000.)
                           (mag 16.))
```

```

(declare (special current-utt))

(cond
  ((and (equal (substring infile 0 1) "e")
        (equal (substring infile 2 3) ":"))
   (copyf infile "z:>tmp>tmp.wav" :characters nil :byte-size 16)
   (setq infile "z:>tmp>tmp.wav"))))

(with-open-file (stream infile :direction :input :characters nil :byte-size 16.)
  (let* ((length (send stream :length))
         (wave (make-fix-array length)))
    (declare (sys:array-register wave))
    (dotimes (i length)
      (setf (aref wave i) (* mag (sign-extend-16 (or (send stream :tyi) 0)))))
    (setf current-utt (spire:create-utterance-from-waveform wave rate))
    (spire:dump-utterance current-utt uttfile)
    current-utt)))

;;;

(defun user-convert-wave-to-utt ()

  (declare (special infile uttfile rate mag))

  (setf infile (fs:make-pathname
                :host      "ef"
                :raw-directory (list "sign" "stanton" "xx" "zbg")
                :raw-name    "nnn"
                :raw-type    "zbg"))

  (setf uttfile (fs:make-pathname
                 :host      "z"
                 :raw-directory (send (fs:user-homedir) :raw-directory)
                 :raw-name    "tmp"
                 :raw-type    "utt"))

  (setf rate 16000)
  (setf mag 16)

  (tv:choose-variable-values
   '("PATHNAME OPTIONS"
     (infile "Origin pathname containing raw waveform" :pathname)
     (uttfile "Destination pathname of utterance" :pathname)
     "PARAMETER OPTIONS"
     (rate "Sampling rate" :number)
     (mag "Multiplication factor" :number)
     ""))
   :label "Options for Converting Waveform to SPIRE Utterance")

  (convert-wave-to-utt infile uttfile rate mag)

```

```

nil)

;;; This is where the function is added to the main system menu.

(tv:add-to-system-menu-programs-column
 "Convert Wave to Utt"
 '(spire:user-convert-wave-to-utt)
 "Takes a raw speech waveform and makes it into a SPIRE utterance"
 nil)

;;; This is experimental code intended to incorporate the
;;; USER-CONVERT-WAVE-TO-UTT features into the Spire menu.
;;; I came across an unexplained bug in the initial implementation
;;; where it seemed to be trying to pass the automatically bound
;;; variables (e.g. spire::character, spire::mouse-x, etc) as
;;; arguments to the function. However, I have been unable to
;;; duplicate this error. Note that DEFINE-SPIRE-COMMAND is very
;;; similar to FS:DEFINE-CP-COMMAND as well as DEFUN. Instead of
;;; just naming a function in the body, the entire function definition
;;; could reside in the body.

(define-spire-command convert-waveform () ()

  "Takes a raw speech waveform and makes it into a SPIRE utterance"
  (user-convert-wave-to-utt))

;;;

(defun extract-features-for-group (&key
                                   (hostname "z")
                                   spkr-num-string
                                   (group-num-string "0"))

  "Updated version of EXTRACT-FEATURES-FOR-SPEAKER, which is now obsolete.
  Works with the smaller utterance groups in order to keep the computations
  more manageable. The original function caused the LISPM to run out of swap
  space, requiring a reboot. The complete utterance list then had to be
  restarted somewhere in mid-stream. This latest version is designed to be
  independent of the method of user interface. Handler functions will take
  care of prompting the user for input, consistent with the terminal being used."

  (let*
    ((utt-group-file
      (buildpath
        :hostname      "z"
        :group-num-string group-num-string
        :filetype      "grp")))

    (uttlist (make-utt-list-array utt-group-file)))

```



```

(do i 0 (1+ i) (equal i (array-length uttlist))

  (do condition 1 (1+ condition) (greaterp condition 3)

    (let*
      ((target-path
        (buildpath
          :hostname hostname
          :spkr-num-string spkr-num-string
          :cond-num condition
          :utt-num-string (aref uttlist i)
          :filetype "fmt"))))

      (cond
        ((probe (send target-path :new-row-type "f3"))
         (print (string-append "Skipping " target-path)))
        (t
         (calculate-and-write-formants
          spkr-num-string
          condition
          (aref uttlist i)
          target-path)))
      ))))

;;;
;;;

(defun console-extract-features-for-group ()

  "This is the handler function to obtain user input from the console
  using window menus. It calls EXTRACT-FEATURES-FOR-GROUP. Currently,
  the directory-list selection is meaningless since it is not passed on.
  The function BUILDPATH defaults the directory-list based on the hostname."

  ;;; Recall that TV:CHOOSE-VARIABLE-VALUES requires any variables used
  to be declared special.

  (declare (special utt-group-number
                    utt-group-file
                    uttlist
                    speaker
                    hostname
                    directory-list))

  (setf speaker "")
  (setf utt-group-number "")
  (setf hostname "")
  (setf directory-list ())

  (tv:choose-variable-values
   '("")
   (speaker "Speaker Number" :choose

```

```

      ("1" "2" "3" "4" "5" "6" "7" "8" "9" "0" "a"))
    (utt-group-number "Utterance Group Number" :choose
      ("1" "2" "3" "4" "5" "6" "7" "8" "9" "0" "a"))
    ...
    "ROOT PATHNAME OPTIONS"
    ...
    (hostname "Host" :choose
      ("ec" "ed" "ee" "ef" "ei" "en" "z"))
    (directory-list "Directory tree (no more than 4)" :choose-multiple
      (("bj0" "bj0")
       ("bj1" "bj1")
       ("sign" "sign")
       ("sun" "sun")
       ("users" "users")
       ("usr" "usr")
       ("harbor" "harbor")
       ("src" "src")
       ("bj" "bj")
       ("stanton" "stanton")
       ("lisp" "lisp")))
    ...)
  'label "Options for Computing and Writing Formant Files")
(setf directory-list
  (reverse (append (car directory-list)
                   (cadr directory-list)
                   (caddr directory-list)
                   (caddr directory-list))))

(extract-features-for-group
 :hostname hostname
 :spkr-num-string speaker
 :group-num-string utt-group-number))
;;;
;;;

(define-spire-command extract-formants-for-utterance-group () ()

  "Computes pitch and first three formants for a group of utterances."

  (console-extract-features-for-group))
;;;
;;;

(defun telnet-extract-features-for-group ()

  "This function uses simple user interfaces that are suitable for use
  through dumb terminals or network logins."

  (print "Extract Features for an Utterance Group")
  (let*
    ((speaker (prompt-and-read :string-trim "Enter Speaker number: "))
     (group-num-string (prompt-and-read :string-trim "Enter Utterance Group Number: ")))

```

```

(hostname (prompt-and-read :string-trim "Enter the Short Name of Target Host: "))

(extract-features-for-group
 :hostname hostname
 :spkr-num-string speaker
 :group-num-string group-num-string)))

;;
;;

(defun calculate-and-write-formants (speaker condition utt-num target-path)

  "Calculates the pitch and first three formant frequencies for the
  utterance indicated. Results are written to individual files having
  a common target path and unique file types: f0, f1, f2, and f3."

  (let*
    ((utt-path (buildpath
                  :hostname      "z"
                  :spkr-num-string speaker
                  :cond-num      condition
                  :utt-num-string utt-num
                  :filetype      "utt"))

      (write-array-to-file
       (att-val
        (send (utterance utt-path)
              :find-att "Pitch Frequency")
        nil)
       (send target-path :new-row-type "f0")))

    (write-array-to-file
     (att-val
      (send (utterance utt-path)
            :find-att "First Formant")
      nil)
     (send target-path :new-row-type "f1")))

    (write-array-to-file
     (att-val
      (send (utterance utt-path)
            :find-att "Second Formant")
      nil)
     (send target-path :new-row-type "f2")))

    (write-array-to-file
     (att-val
      (send (utterance utt-path)
            :find-att "Third Formant")
      nil)
     (send target-path :new-row-type "f3"))))

```

```

(send (utterance utt-path) :kill)))

;;;
;;;

#| This is the old version

(declare (special z-name utt-path current-utt local-att feature-array))

(setf z-name (string-append utt-num "-zbg"))

(setf utt-path (fs:make-pathname
                  :host "z"
                  :raw-directory (list "stanton" session)
                  :raw-name z-name
                  :raw-type "utt"))

(setf current-utt (utterance utt-path));obtain an utterance object
(setf local-att (send current-utt :find-att "Pitch Frequency"));find the att
(setf feature-array (att-val local-att nil));compute the att and put it in array
(write-array-to-file feature-array (send target-path :new-row-type "f0"))

(setf local-att (send current-utt :find-att "First Formant"))
(setf feature-array (att-val local-att nil))
(write-array-to-file feature-array (send target-path :new-row-type "f1"))

(setf local-att (send current-utt :find-att "Second Formant"))
(setf feature-array (att-val local-att nil))
(write-array-to-file feature-array (send target-path :new-row-type "f2"))

(setf local-att (send current-utt :find-att "Third Formant"))
(setf feature-array (att-val local-att nil))
(write-array-to-file feature-array (send target-path :new-row-type "f3"))

(send current-utt :kill)
|#

;;; This is the latest version of the function EXTRACT-LABELS
;;; which is designed to extract the hand-labeling
;;; information from the default utterance and store it both
;;; on EI and on the LISP file systems.

;;; It is designed to be used within SPIRE and work on the currently
;;; selected utterance. To use, simply invoke the function EXTRACT-LABELS
;;; in a command window with no arguments. It will automatically write
;;; the label data in a file having compatible pathnames and with type
;;; LBL.

;;;
;;;

```

```
(defun extract-and-write-labels (selected-utterance target-path)
```

"Obtains the phonetic transcription of the selected-utterance and writes it in ASCII form to the target-path."

```
(let*
  ((att (send selected-utterance :find-att "Phonetic Transcription"))
   (token-list (att-val att nil)))

  (cond ((not (null token-list))

         (write-token-slots token-list target-path))))
```

```
;;;
;;;
```

```
(defun user-extract-labels ()
```

```
(declare (special lbl-path))
```

```
(let*

  ((utt-path (utterance-pathname (default-utterance))))

  (setf lbl-path (make-lbl-path utt-path))

  (tv:choose-variable-values
   '(""
     (lbl-path "Pathname of label file for LISPM" :pathname)
     ""))
   :label "Phonetic Label File Options")
```

```
(extract-and-write-labels (default-utterance) lbl-path))
nil)
```

```
;;; This is where the function is added to the main system menu.
```

```
(tv:add-to-system-menu-programs-column
 "Extract Utt Phonetic Labels"
 '(spire:user-extract-labels)
 "Extracts the phonetic transcription for the default utterance and writes to separate files"
 nil)
```

```
;;; This is experimental code intended to incorporate the
;;; USER-EXTRACT-LABELS features into the Spire menu.
```

```
(define-spire-command extract-phonetic-labels () ())
```

```
"Writes phonetic transcription to separate files"
(user-extract-labels))
```

```
;;;
```

```
;;
```

```
(defun extract-labels-for-utterance-list (uttlist)
```

```
  "For a list of utterances, finds their pathnames, and then writes
  phonetic label files into the appropriate parallel LBL directory."
```

```
  (loop for utt in uttlist do
```

```
    (let*
```

```
      ((lbl-path (make-lbl-path (utterance-pathname utt))))
```

```
      (extract-and-write-labels utt lbl-path))))
```

```
;;
```

```
;;
```

```
(defun make-lbl-path (utt-path)
```

```
  "Returns the label path parallel to the supplied utterance path."
```

```
  ;; This trivial code puts the label path in the same directory as the
  ;; utterance and changes only the filetype. Not the preferred result,
  ;; but will serve as a band-aid until the other code can be worked on.
  ;; This is a last-ditch approach (send utt-path :new-row-type "lbl")
```

```
  (fs:make-pathname
```

```
    :host (send utt-path :host)
```

```
    :raw-directory (reverse (append '("lbl") (cdr (reverse
      (send utt-path
        :raw-directory))))))
```

```
    :raw-name (send utt-path :raw-name)
```

```
    :raw-type "lbl"))
```

```
  #|| This code is suspect to side-effects. I think it may be the cause
  of the utterance path being altered inadvertently. The mechanism is not
  clear. Recommend not using...
```

```
  (let*
```

```
    ((lbl-path (send utt-path :new-row-type "lbl"))
```

```
    (dirlist (send lbl-path :raw-directory))))
```

```
    (setf dirlist (cl:replace dirlist '("lbl")
```

```
      :start1 (1- (zl:length dirlist))))
```

```
    (setf lbl-path (send lbl-path :new-row-directory dirlist))
```

```
    lbl-path)
```

```
  ;; this closure was for the defun )
```

```
  ||#
```

```
(define-spire-command extract-loaded-phonetic-labels () ())
```

```
  "Processes utterances that are currently loaded"
```

```

(extract-labels-for-utterance-list *loaded-utterances*))

;;;
;;;

;;; Here is where I put it all together: The function
;;; GET-TOKEN-SLOTS iterates through the token list
;;; and writes to the stream. WRITE-TOKEN-SLOTS takes care of the
;;; actual file writing.

(defun write-token-slots (token-list filename)
  (with-open-file (stream filename :direction :output :characters t)
    (get-token-slots token-list stream)))

(defun get-token-slots (token-list stream)
  (declare (special next-token))
  (cond
    ((null token-list) nil)
    (t
     (setf next-token (car token-list))
     (format stream "~A ~10,2,14$ ~10,2,14$ ~%"
              (spire:token-name next-token)
              (spire:token-from-pos next-token)
              (spire:token-to-pos next-token))
     (get-token-slots (cdr token-list) stream))))

;;;
;;;

#||

;;; This was the original version of EXTRACT-LABELS. It is now obsolete.

(defun extract-labels ()
  (declare (special ei-path s-path next-token))
  (let*

    ;; First set up the pointer to the phonetic label data in the utterance.

    ((att (send (default-utterance) :find-att "Phonetic Transcription"))
     (token-list (att-val att nil)))

    ;; Now do the necessary string manipulations to build the proper pathnames
    ;; for both UNIX and the LISP.

    (s-name
     (send (utterance-pathname (default-utterance))
           :raw-name))

    (ei-name
     (string-append

```

```

(substring z-name 0 3)
"."
(substring z-name 4)))

(session
  (cadr
    (send (utterance-pathname (default-utterance))
      :raw-directory))))

(setf ei-path
  (fs:make-pathname
    :host "ei"
    :raw-directory (list "bj0" "bj" session "lbl")
    :raw-name ei-name
    :raw-type "lbl"))

(setf z-path
  (fs:make-pathname
    :host "z"
    :raw-directory (list "stanton" session)
    :raw-name z-name
    :raw-type "lbl"))

(setf ei-path
  (prompt-and-read
    '(:pathname :visible-default ,ei-path)
    "Enter name for EI label file "))

(write-token-slots token-list ei-path)

(setf z-path
  (prompt-and-read
    '(:pathname :visible-default ,z-path)
    "Enter name for Z label file "))

(write-token-slots token-list z-path))
nil)

;;;
;;; Here is an outdated function that pulled out label files.

(defun recursive-extract-labels (uttlist)

  "This function writes the label files into the same directory where
the utterance resides."

  (cond
    ((null uttlist) nil)
    (t
     (extract-and-write-labels (car uttlist)
                               (send (utterance-pathname (car uttlist))
                                :new-raw-type "lbl"))
     (recursive-extract-labels (cdr uttlist)))))

```



```

(recursive-extract-labels (cdr uttlist))))))

;;;
;;;

;;#

;;;

;;; This function is designed to load a set of utterances designated in
;;; the file UTT-GROUP-N where N is an integer. The idea
;;; is to also have the wide-band spectrograms pre-computed so that the
;;; user does not have to be when labelling the group of utterances.

(defun load-utt-group ()

  "Loads a group of utterances specified in file UTT-GROUP-N.TEXT where
  N is an integer. Spectrograms are computed as each utterance is loaded."

  (let*
    ((speaker
      (prompt-and-read :string-trim "Enter speaker number: "))

      (utt-group-number
        (prompt-and-read :string-trim "Enter utt-group number: "))

      (utt-group-file
        (buildpath
          :hostname      "z"
          :group-num-string utt-group-number
          :filetype      "grp"))))

    ;;; Here is some playing around with getting the utterance lists
    ;;; into the proper form. Simply requires some fundamental list
    ;;; manipulations.

    (with-open-file (in-stream utt-group-file :direction :input)
      (let*
        (
          (selection-keyword-alist
            (list '(1 "Normal" nil nil nil nil); The four nils are
                  '(2 "Loud" nil nil nil nil); implications that were
                  '(3 "Lombard" nil nil nil nil))); added with Genera 7.
          (utt-list ())
          (end-of-file nil))

          (do i 0 () end-of-file
            (let*
              ((line (send in-stream :line-in))
               (utt "")))

```

```

(cond
  ((equal 0 (string-length line))
   (setf end-of-file t))
  (t
   (setf utt (substring line 1 4))
   (setf utt-list (cons
                    (.ist utt
                     line
                     (choice-list speaker utt "lbl"))
                    utt-list))))))

(work-through-list (reverse (tv:multiple-choose
                             (string-append "Utterances to Load for Speaker "
                             speaker " ")
                             (reverse utt-list)
                             selection-keyword-alist)
                    speaker)
)))

;;;
;;;

(defun work-through-list (worklist speaker)

  "This function simply recurses through the list that was produced
  by TV:MULTIPLE-CHOOSE."

  (cond
    ((null worklist) nil)
    (t
     (get-single-wave (caar worklist) (cdar worklist) speaker)
     (work-through-list (cdr worklist) speaker))))

;;;
;;;

(defun get-single-wave (utt-num cond-list speaker)

  "This function operates on a list where the first element
  is the utterance number string and the subsequent elements would
  indicate the conditions that are to be loaded for that utterance"

  (cond
    ((null cond-list) nil)
    (t
     (load-unlabeled-utterance
      :spkr-num-string speaker
      :cond-num (car cond-list)
      :utt-num-string utt-num)
     (get-single-wave utt-num (cdr cond-list) speaker)
     )))

```

;;; Make this available as a Spire Command.

```
(define-spire-command load-utterance-group () ())
```

"Loads and calculates spectrograms for a set of utterances."
(load-utt-group))

```
;;;
;;;
```

```
(defun choice-list (speaker utt type)
```

"Determines whether or not files for the three conditions exists and returns a list of choices that are defaulted to NO if the file is already in the LISPM directory and YES if the file is not found."

```
(let*
  ((choices ()))
```

```
(do condition 3 (1- condition) (lessp condition 1)
```

```
  (setf
   choices
   (cons
    (list
     condition
     (null (probe-file
            (buildpath
             :hostname "z"
             :spkr-num-string speaker
             :cond-num condition
             :utt-num-string utt
             :filetype type))))
    choices)))
```

```
  choices))
```

```
;;;
;;;
```

```
(defun load-unlabeled-utterance (&key
                                spkr-num-string
                                cond-num
                                utt-num-string)
```

"First checks to see if the labels for requested utterance already exist. If so, it skips the loading. Otherwise it will load the utterance and compute the Wide-band Spectrogram. It will also check to see if the utterance even exists. If not, it will seek the raw waveform on EF and build the utterance from it."

```
(let*
  ((wave-path
```

```

(buildpath
  :hostname      "ef"
  :spkr-num-string spkr-num-string
  :cond-num      cond-num
  :utt-num-string utt-num-string
  :filetype      "zbg"))

(utt-path
  (buildpath
    :hostname      "z"
    :spkr-num-string spkr-num-string
    :cond-num      cond-num
    :utt-num-string utt-num-string
    :filetype      "utt"))

(lbl-path
  (buildpath
    :hostname      "z"
    :spkr-num-string spkr-num-string
    :cond-num      cond-num
    :utt-num-string utt-num-string
    :filetype      "lbl"))

(current-utt nil))

(cond
  ((null (probe! lbl-path))
    (cond
      ((null (probe! utt-path))
        (print (string-append "Converting " wave-path))
        (print (string-append "to      " utt-path))
        (setf current-utt (convert-wave-to-utt wave-path utt-path 16000 16)))
      (t
        (setf current-utt (utterance utt-path)))))

  (print (string-append "Computing Wide-Band Spectrogram for " utt-path " ..."))
  (att-val (send current-utt :find-att "Wide-Band Spectrogram") nil)
  )
  (t
    (print (string-append "Found " lbl-path " Not loading " utt-path))))))

;;;
;;;

(defun telnet-load-utt-group ()

  "Provides the capability of loading files for labeling purposes while
working from a remote terminal."

  (let*
    ((speaker

```

```

(prompt-and-read :string-trim "Enter speaker number: ")

(utt-group-number
  (prompt-and-read :string-trim "Enter utt-group number: "))

(utt-group-file
  (buildpath
    :hostname      "z"
    :group-num-string utt-group-number
    :filetype      "grp"))

(uttlist (make-utt-list-array utt-group-file)))

(loop for utt-num-string being each array-element in uttlist do

  (loop for condition from 1 to 3 do

    (load-unlabeled-utterance
      :spkr-num-string speaker
      :cond-num      condition
      :utt-num-string utt-num-string))))

#||
;;; This is an obsolete version of the function. It provided no flexibility.

(defun load-utt-group ()

  (declare (special speaker
    utt-group-number
    utt-group-file
    utt-number
    wave-path
    utt-path
    current-utt))

  (setf fs:*remember-passwords* t)

  (setf speaker
    (prompt-and-read :string-trim "Enter speaker number: "))

  (setf utt-group-number
    (prompt-and-read :string-trim "Enter utt-group number: "))

  (setf utt-group-file
    (fs:make-pathname
      :host      "z"
      :raw-directory '("stanton")
      :raw-name    (string-append "utt-group-" utt-group-number)
      :raw-type    "text")))

;; Now open up the utterance list and load the utterance number strings
;; into an array for easy processing.

```

```

(with-open-file (in-stream utt-group-file :direction :input)
  (let*
    ((length (/ (send in-stream :length) 4)))
    (dotimes (i length)
      (setf utt-number (send in-stream :line-in))

      (do condition 1 (1+ condition) (greaterp condition 3)

        (setf wave-path
          (fs:make-pathname
            :host      "ef"
            :raw-directory (list "sign" "stanton"
              (string-append speaker
                (string (+ 48 condition))))
            "zbg")
          :raw-name      utt-number
          :raw-type       "zbg"))

        (setf utt-path
          (fs:make-pathname
            :host      "z"
            :raw-directory (list "stanton"
              (string-append speaker
                (string (+ 48 condition))))
            :raw-name      (string-append utt-number "-zbg")
            :raw-type       "utt"))

        (print (string-append "Converting " wave-path))
        (print (string-append "to " utt-path))
        (setf current-utt (convert-wave-to-utt wave-path utt-path 16000 16))
        (print "Calculating Wide Band Spectrogram...")
        (att-val (send current-utt :find-att "Wide-Band Spectrogram") nil))))

  nil)
"#
;;;

```

;;; The following code contains a number of the utility functions that do
 ;;; simple things and support the main functions developed for my research.

```

(defun buildpath (&key
  hostname
  (dir-list () dir-supplied)
  spkr-num-string
  cond-num
  utt-num-string
  filetype
  group-num-string)

```

"Returns the proper pathname for the supplied arguments"

```

;;; First build the proper rootpath
(let*
  ((filename ""))

  (cond
    ((not dir-supplied)
     (cond
       ((equal hostname "z")
        (setf dir-list '("stanton"))))
       ((equal hostname "ef")
        (setf dir-list '("sign" "stanton"))))
       ((equal hostname "ei")
        (setf dir-list '("bj1" "bj"))))
       ((equal hostname "en")
        (setf dir-list '("usr" "src" "bj"))))
       ((equal hostname "col")
        (setf dir-list '("usr" "harbor" "stanton" "lispn"))))))

    (cond

      ((or (equal filetype "zbg") (equal filetype "utt"))
       (setf filename utt-num-string)
       (setf dir-list (append dir-list (list (session-string
                                                spkr-num-string
                                                cond-num)
                                              "zbg")))))

      ((equal filetype "lbl")
       (setf filename utt-num-string)
       (setf dir-list (append dir-list (list (session-string
                                                spkr-num-string
                                                cond-num)
                                              filetype))))

      ((equal filetype "fmt")
       (setf filename utt-num-string)
       (setf dir-list (append dir-list (list (session-string
                                                spkr-num-string
                                                cond-num)
                                              filetype))))

      ((equal filetype "grp")
       (setf filename (string-append "utt-group-" group-num-string))
       (setf filetype "text")))

    (fs:make-pathname
     :host      hostname
     :raw-directory dir-list
     :raw-name   filename
     :raw-type   filetype)))

;;;

```

```
(defun session-string (speaker condition)
```

"Creates the string of the session number from
the string SPEAKER and the integer CONDITION"

```
(string-append speaker
  (string (character (+ 48
    condition))))))
```

```
;;;
;;;
```

```
(defun make-utt-list-array (utt-list-file)
```

"Reads an utterance list from a file and loads it into
an array for easy processing."

```
(with-open-file (in-stream utt-list-file :direction :input)
  (let*
    ((end-of-file nil)
     (utt-list-array (make-fix-array
      (lines-in-file utt-list-file))))
    (declare (sys:array-register utt-list-array))
    (do i 0 (1+ i) end-of-file
      (let*
        ((line (send in-stream :line-in)))
        (cond
          ((equal 0 (string-length line))
           (setf end-of-file t))
          (t
           (setf (aref utt-list-array i) (substring line 1 4))))))
      utt-list-array)))
```

```
;;;
;;;
```

```
(defun write-array-to-file (feature-array target-path)
```

"Takes any arbitrary real array and writes it to a file in ASCII form."

```
(print (string-append "Writing to: " target-path))
(with-open-file (stream target-path :direction :output :characters t)
  (loop for i from 0 to (- (array-length feature-array) 1)
    do (format stream "~4D~%" (aref feature-array i)))))
```

```
;;;
;;;
```

```
(defun lines-in-file (pathname)
```

"Returns the number of lines in the given file"


```

(with-open-file (in-stream pathname :direction :input)

  (let*
    ((end-of-file nil)
     (number-of-lines 0))

    (do i 0 (1+ i) end-of-file

      (cond
        ((equal 0 (string-length
                    (send in-stream :line-in)))
         (setf end-of-file t)
         (setf number-of-lines i))))
      number-of-lines)))

;; -*- Mode: Lisp; Package: ZL-USER; Syntax: Zetalisp; Base: 10 -*-

(defun window-dump ()

  ;; (send terminal-io ':refresh)
  (if (equal 1 (tv:with-mouse-and-buttons-grabbed
                (setq tv:who-line-mouse-grabbed-documentation
                    "Left: Select window to be dumped to COLUMBIA. Middle or Right aborts.")
                (tv:wait-for-mouse-button-down "Waiting for Mouse Click"))))

    (let* (
      (bitmap-array (send (tv:window-under-mouse) :screen-array))
      (ydim (car (array-dimensions bitmap-array)))
      (xdim (cadr (array-dimensions bitmap-array)))
      (header-size 40)
      (file-version 6)
      (display-type 302)
      (display-planes 1)
      (pixmap-format 0)
      (window-bdr-width 2)
      (pixmap-width (* (1+ (/ xdim 16)) 16))
      (pixmap-height ydim)
      (window-width (- pixmap-width (* window-bdr-width 2)))
      (window-height (- pixmap-height (* window-bdr-width 2)))
      (window-x 0)
      (window-y 0)
      (window-ncolors 0))

      (with-open-file (stream "z:>stanton>lispm-screen.bin"
                             :direction :output
                             :characters nil
                             :byte-size 16)

        (tyo 0          stream)
        (tyo header-size stream)
        (tyo 0          stream)
        (tyo file-version stream)

```

```

(tyo 0      stream)
(tyo display-type  stream)
(tyo 0      stream)
(tyo display-planes  stream)
(tyo 0      stream)
(tyo pixmap-format  stream)
(tyo 0      stream)
(tyo pixmap-width  stream)
(tyo 0      stream)
(tyo pixmap-height  stream)
(tyo window-width  stream)
(tyo window-height  stream)
(tyo window-x      stream)
(tyo window-y      stream)
(tyo window-bdr-width stream)
(tyo window-ncolors stream)

```

```

(loop for y from 0 below ydim do

```

```

  (loop for xw from 0 to (// xdim 16) do

```

```

    (let* (
      (tot 0);
      (xmax
        (if (equal xw (// xdim 16))
            (sub1 ( xdim 16))
            15)))
      (loop for x from 0 to xmax do

        (setf tot
          (+ tot (* (expt 2 x)
                    (if (equal (aref bitmap-array
                                   y
                                   (+ (* xw 16) x))
                        1) 0 1))))))

```

```

      (if (equal xw (// xdim 16))
        (loop for x from ( xdim 16) to 15 do
          (setf tot (+ tot (expt 2 x))))))

```

```

      (tyo tot stream))))))

```

```

(tv:add-to-system-menu-programs-column
  "Window Dump"
  '(zl-user:window-dump)
  "Performs a raster dump of a window selected by the mouse"
  nil)

```

Appendix C: AAMRL Database Vocabulary

A	DIVE	INCREASE	OVERFLY	STEADY
ABOVE	DOGFIGHT	INDEX	OXYGEN	STOP
ACKNOWLEDGED	DOWN	INDICATED	P	STRAFE
AIM	DULL	INVENTORY	PAGE	SUP
ALTERNATE	E	J	PERFORMANCE	SURFACE
ALTITUDE	EASE	JETTISON	PIGEONS	T
AND	EAST	K	PLATE	TACAN
ARM	EASY	L	POINT	TARGET
AS	EIGHT	LARGER	POSSIBLE	TEN
AT	EIGHTEEN	LEFT	PREVIOUS	TERRAIN
AUDIO	EIGHTY	LEVEL	Q	TEST
AUTO	ELEVATION	LITTLE	R	THIRTEEN
AUTOPILOT	ELEVEN	LOCK	RACK	THIRTY
B	EMERGENCY	LOUDER	RADAR	THOUSAND
BACK	ENDURANCE	LOW	RESET	THREATS
BACKUP	ENGINE	M	RIGHT	THREE
BELOW	ENTER	MACH	S	THROTTLE
BORE	EXIT	MAP	SAFE	TIME
BRAKE	F	MARK	SCALES	TO
BREAK	FIFTEEN	MAX	SCAN	TOP
BRIGHTER	FIFTY	MID-RANGE	SEARCH	TRAIN
BURNER	FIVE	MIL	SET	TRUE
BY	FLARE	MISSILES	SEVEN	TWELVE
C	FORTY	MODE	SEVENTEEN	TWENTY
CAGE	FOUR	MORE	SEVENTY	TWO
CAUTION	FOURTEEN	MUTE	SHARP	U
CHAFF	FOX	N	SIGHT	UNCAGE
CHART	FUEL	NAV	SILENCE	UP
CHECKLIST	G	NEXT	SIM	V
CLEAR	GO	NINE	SIX	W
CLIMB	GROUND	NINETEEN	SIXTEEN	WARNING
CLOSE	GUNS	NINETY	SIXTY	WEAPON
COMM	H	NO	SLAVE	WEST
CONTROL	HARD	NORTH	SMALLER	WHY
D	HARDER	NOSE	SOFT	X
DECREASE	HEADING	NOW	SOFTER	Y
DESCENT	HIGH	O	SORT	YES
DESTINATION	HOLD	OFF	SOUTH	Z
DIMMER	HUNDRED	OH	SPEED	ZERO
DIRECT	I	ONE	SQUAWK	
DISPENSE	IDENT	OPEN	STAB-OUT	
DISPLAY	IDLE	OVER	STANDBY	

Appendix D: Complete AAMRL List of Enrollment Utterances

A001 AUTOPILOT HOLD DESCENT
A002 I R SCAN
A003 PAGE I F R SUP PREVIOUS
A004 SET NINETY TWO ELEVATION TEN THOUSAND EIGHT HUNDRED FOURTEEN ENTER
A005 DISPLAY BRIGHTER
A006 THROTTLE RIGHT BACK
A007 NINE ONE DIRECT
A008 I N S TIME FOURTEEN SIXTEEN ENTER
A009 DISPLAY CHAFF AND FLARE
A010 I R STAB-OUT THREE FOUR TWO BY TWO
A011 WHY
A012 CAUTION ACKNOWLEDGED
A013 DISPLAY TERRAIN
A014 JETTISON THIRTEEN AND SIX AND ONE SIX AND FIVE RACK NOW
A015 JETTISON FOUR AND ONE FOUR WEAPON NOW
A016 JETTISON ELEVEN AND SIX AND NINETEEN AND ONE FIVE WEAPON NOW
A017 CLEAR PERFORMANCE
A018 SET SIXTY Z Y THREE NINE ZERO FIVE ZERO FIVE TWO THREE ENTER
A019 AUTOPILOT LEVEL A LITTLE
A020 FOX TWO
A021 DOGFIGHT
A022 DISPLAY ENGINE
A023 AUTOPILOT BREAK RIGHT
A024 DISPLAY CHECKLIST AT D J
A025 STRAFE
A026 CLEAR HIGH ALTITUDE INDEX
A027 SET SIXTY ONE C B FIVE SIX SIX EIGHT TWO NINE EIGHT FOUR ENTER
A028 SET FORTY B K TWO NINE FOUR THREE ONE FOUR EIGHT SIX ENTER
A029 CLEAR LOW INDEX
A030 MID-RANGE
A031 NO
A032 I R AT TWO ZERO ZERO SEARCH ABOVE ONE NINE THREE
A033 AUTOPILOT DECREASE ONE FIVE
A034 I F F STANDBY
A035 SPEED BRAKE CLOSE
A036 AUTOPILOT DIVE
A037 SET TEN X V FIVE THREE THREE THREE THREE SEVEN THREE THREE ENTER
A038 DISPLAY LOW INDEX
A039 SET K TIME SEVENTEEN EIGHTEEN ENTER
A040 AUTOPILOT EASY LEFT
A041 AUTOPILOT LEFT BREAK
A042 SOFTER DOWN LOUDER ENTER
A043 SET FIFTY S K FIVE EIGHT ONE FOUR ONE THREE EIGHT SEVEN ENTER
A044 F M OVER
A045 AUTOPILOT RIGHT BREAK
A046 RADAR AT FIVE SEARCH BELOW ONE THOUSAND
A047 I R AT THREE FIVE SEARCH SURFACE
A048 U H F SET G C A THREE FIFTY POINT FIVE SEVEN FIVE ENTER
A049 CLEAR MAP
A050 E P U RESET NOW
A051 CLEAR ENGINE
A052 RADAR SORT
A053 AUTOPILOT NOSE UP
A054 AUTOPILOT HOLD DESCENT

A055 AUTOPILOT RIGHT HARD AS POSSIBLE
A056 CAGE
A057 STRAFE
A058 SET SIXTY J L TWO THREE SEVEN SIX FOUR FOUR TWO FOUR ENTER
A059 I L S AUTO
A060 RADAR ALTITUDE SHARP
A061 MIL
A062 CLEAR CHECKLIST P O
A063 AUTOPILOT HARD AS POSSIBLE LEFT
A064 DISPLAY BRIGHTER
A065 DOWN UP LARGER ENTER
A066 LARGER LARGER SMALLER ENTER
A067 RADAR AT TWO ZERO ZERO SEARCH SURFACE UP
A068 ARM NOW
A069 AUTOPILOT LEVEL OFF A LITTLE
A070 AT ONE EIGHT ZERO SEARCH SURFACE
A071 AUTOPILOT BREAK LEFT
A072 FOX ONE
A073 CLEAR MAP
A074 IDLE
A075 DOGFIGHT
A076 SMALLER SMALLER UP ENTER
A077 I R AT EIGHT ZERO SEARCH ABOVE TWO EIGHT POINT FIVE THOUSAND
A078 AUDIO INCREASE
A079 SET THIRTEEN EAST ZERO NINE EIGHT NINETEEN TWELVE ENTER
A080 DISPLAY CHECKLIST X T
A081 RADAR AT TWO ZERO ZERO SEARCH BELOW THREE FIVE POINT FIVE THOUSAND
A082 CLEAR MARK ONE OH SIX
A083 AUTOPILOT NOSE UP
A084 AUTOPILOT HARD AS POSSIBLE LEFT
A085 CAUTION TEST
A086 AUTOPILOT HARDER
A087 CAGE
A088 AUTOPILOT EASE OFF
A089 ONE FIVE DIRECT
A090 SET ALTERNATE WEST FIFTEEN FIFTEEN ENTER
A091 STRAFE
A092 AUTOPILOT INCREASE ONE ZERO
A093 E P U RESET NOW
A094 IDENT
A095 SET SEVENTY EAST ZERO NINE SEVEN FIFTEEN NINETEEN ENTER
A096 U H F SET GROUND TWO FORTY POINT THREE ZERO ZERO ENTER
A097 UNCAGE
A098 I N S OVERFLY UP
A099 AUTOPILOT EASE OFF
A100 WHY
A101 SIM
A102 CLEAR MAP
A103 WARNING TEST
A104 SPEED BRAKE STOP
A105 AUTOPILOT STEADY
A106 MIL
A107 DISPLAY MARK ONE OH SIX
A108 DISPLAY LOW ALTITUDE INDEX
A109 SPEED BRAKE CLOSE
A110 DISPLAY SCALES
A111 SET FORTY EAST ONE ZERO SEVEN FIFTY EIGHT ELEVEN ENTER

A112 CLEAR CONTROL
A113 GUNS
A114 CLEAR TERRAIN
A115 EXIT
A116 U H F SET G C A THREE NINETY POINT NINE ENTER
A117 AUTOPILOT CLIMB MORE
A118 MID-RANGE
A119 TACAN SET TWENTY Y ENTER
A120 AUTOPILOT HARD AS POSSIBLE RIGHT
A121 DISPENSE
A122 DISPLAY I F R SUP AT L Q
A123 GO SEVENTY SIX POINT TWO FIVE ZERO ENTER
A124 COMM SET EMERGENCY EIGHTY POINT SEVEN ENTER
A125 V H F MUTE
A126 DISPLAY SOFT
A127 UNCAGE
A128 SET ALTERNATE W W FOUR NINE ZERO THREE SIX SIX ONE EIGHT ENTER
A129 BORE SIGHT AIM NINE MISSILES
A130 JETTISON NINE AND TWELVE AND TWO WEAPON NOW
A131 AT ONE SIX ZERO SEARCH BELOW TWO POINT FIVE THOUSAND
A132 AUDIO
A133 AUTOPILOT BURNER CLIMB
A134 SET PREVIOUS T X TWO FIVE ENTER
A135 SET E NORTH ONE THIRTEEN SEVENTEEN ENTER
A136 AUTOPILOT DIVE
A137 SPEED BRAKE STOP
A138 PAGE PLATE PREVIOUS
A139 DISPLAY BRIGHTER
A140 AT FIVE SEARCH ABOVE SIX ZERO POINT FIVE THOUSAND
A141 CLEAR DESTINATION PREVIOUS
A142 SET PIGEONS SOUTH SIX THREE FOUR FIVE EIGHT ENTER
A143 WHY
A144 CAGE
A145 JETTISON ONE THREE AND ELEVEN AND TWELVE AND ONE WEAPON NOW
A146 PAGE PLATE PREVIOUS
A147 I N S OVERFLY UP
A148 AUTOPILOT HOLD DESCENT
A149 F M SET GROUND THIRTY NINE POINT ONE ZERO ZERO ENTER
A150 AUTOPILOT LEVEL A LITTLE
A151 TACAN SET THREE TWO ENTER
A152 I F F STANDBY
A153 I R SCAN
A154 NAV HEADING TWO EIGHT THREE ENTER
A155 SET TWENTY TWO EAST SIXTEEN THIRTY FOUR ENTER
A156 SET K WEST SIXTEEN EIGHTEEN ENTER
A157 R TWO TRAIN ONE
A158 FOX ONE
A159 AUTOPILOT DECREASE TWO FIVE
A160 I R AT TWO FIVE SEARCH SURFACE
A161 AUTOPILOT DIVE
A162 F M MUTE
A163 ARM NOW
A164 AUTOPILOT HOLD NINE THOUSAND FIVE HUNDRED TWELVE INDICATED
A165 GROUND OVER
A166 AUTOPILOT DOWN MORE
A167 RADAR SHARP
A168 BACKUP

A169 TACAN SET FIFTY FOUR Y ENTER
A170 MID-RANGE
A171 D DIRECT
A172 IDENT
A173 DISPLAY SOFT
A174 AUTOPILOT HOLD FOUR THOUSAND SIX HUNDRED NINETY TRUE
A175 BURNER
A176 I N S HEADING TWO ZERO THREE ENTER
A177 FUEL CLOSE
A178 RADAR ALTITUDE AUTO
A179 WARNING ACKNOWLEDGED
A180 E P U RESET NOW
A181 AUTOPILOT HOLD MAX ENDURANCE
A182 AUTOPILOT HARDER
A183 AUTOPILOT HOLD THREE SIX FIVE SEVEN TRUE
A184 SET TARGET SOUTH TWO FIVE SIXTEEN ENTER
A185 AUTOPILOT DIVE
A186 SET MODE THREE TWO TWO TWO SEVEN ENTER
A187 DISPLAY CONTROL
A188 TACAN SET NINE ZERO Y ENTER
A189 UNCAGE
A190 WARNING ACKNOWLEDGED
A191 DISPLAY INVENTORY
A192 AUTOPILOT DOWN MORE
A193 AUTOPILOT EASE OFF
A194 DISPLAY BRIGHTER
A195 DISPLAY INVENTORY
A196 I F F STANDBY
A197 EXIT
A198 SAFE
A199 I R SCAN
A200 H F MUTE
A201 V H F DECREASE
A202 CAGE
A203 CLEAR WARNING
A204 G C I OVER
A205 SQUAWK
A206 YES
A207 YES
A208 COMM SET EMERGENCY THREE TWO TWO ENTER
A209 PIGEONS DIRECT
A210 SPEED BRAKE STOP
A211 SPEED BRAKE CLOSE
A212 SILENCE
A213 SET TARGET WEST NINE FOUR THREE SEVENTEEN ENTER
A214 DISPLAY PERFORMANCE
A215 CLEAR DESTINATION ALTERNATE
A216 RADAR ALTITUDE LOCK
A217 DISPLAY ENGINE
A218 SMALLER UP LOUDER ENTER
A219 DISPENSE
A220 AUTOPILOT HOLD DESCENT
A221 CLEAR CHAFF AND FLARE
A222 BACKUP
A223 DISPLAY CONTROL
A224 CLEAR CONTROL
A225 DISPLAY DIMMER

A226 SIM
A227 SET NEXT T J SEVEN THREE TWO SIX ZERO FIVE TWO TWO ENTER
A228 JETTISON SIXTEEN AND EIGHT AND FIFTEEN AND NINE RACK NOW
A229 E P U OFF NOW
A230 SAFE
A231 DISPLAY MARK ONE OH SIX
A232 AUTOPILOT STEADY
A233 AUTOPILOT BURNER CLIMB
A234 TARGET DIRECT
A235 AUTOPILOT HOLD FOUR TWO THREE NINE INDICATED
A236 DESTINATION PIGEONS ENTER
A237 DISPLAY TERRAIN
A238 SET EIGHTY ELEVATION SEVENTEEN THOUSAND SEVENTY SEVEN ENTER
A239 AUTOPILOT HOLD SIX ZERO TWO SIX INDICATED
A240 ARM NOW
A241 SILENCE
A242 TOP
A243 SAFE
A244 WHY
A245 MIL
A246 NAV HEADING ZERO FOUR TWO ENTER
A247 AUTOPILOT HOLD MACH FOUR
A248 AUTOPILOT DIVE
A249 PAGE CHART RIGHT
A250 AUTOPILOT HOLD MACH NINE
A251 CLEAR LOW ALTITUDE INDEX
A252 DOGFIGHT
A253 EXIT
A254 CLEAR MARK ONE OH SIX
A255 RADAR ALTITUDE SHARP
A256 CHAFF
A257 CLEAR HIGH INDEX
A258 SET PIGEONS NORTH SIX THIRTY FIVE FORTY SIX ENTER
A259 SET SEVENTY TIME FOURTEEN TWELVE ENTER
A260 ARM NOW
A261 RADAR ALTITUDE DULL
A262 CLEAR ENGINE
A263 EXIT
A264 I R STAB-OUT NINE THREE BY FIVE ONE
A265 AUTOPILOT EASE OFF
A266 RADAR ALTITUDE SORT
A267 YES
A268 AUTOPILOT HARDER
A269 AUTOPILOT HOLD MACH TWO
A270 PAGE PLATE NEXT
A271 DISPLAY PLATE INDEX
A272 AUTOPILOT HARDER
A273 RADAR LOCK
A274 CLEAR ENGINE
A275 R W R AUTO
A276 I N S OVERFLY UP
A277 CLEAR CHAFF AND FLARE
A278 AUTOPILOT LEVEL A LITTLE
A279 COMM SET GROUND SEVENTY FOUR POINT FIVE ZERO ZERO ENTER
A280 NO
A281 CAGE
A282 SET EIGHTY WEST TEN FORTY TWO ENTER

A283 I L S MUTE
A284 RADAR STANDBY
A285 SET D NORTH THIRTEEN FIFTEEN ENTER
A286 AUTOPILOT NOSE UP
A287 BACKUP
A288 SET I P ELEVATION NINETEEN THOUSAND FOUR HUNDRED FORTY ENTER
A289 TOP
A290 SET J Z Z FIVE ONE FIVE FOUR ONE TWO ZERO EIGHT ENTER
A291 CLEAR SCALES
A292 SPEED BRAKE OPEN
A293 STRAFE
A294 SET MODE ONE TWO ONE ENTER
A295 BURNER
A296 DOGFIGHT
A297 IDENT
A298 I N S WEST ZERO NINE THREE TWENTY TWO FIFTY FIVE ENTER
A299 AUDIO DECREASE
A300 AUTOPILOT HOLD MAX ENDURANCE
A301 DISPENSE
A302 BORE AIM NINE
A303 AUTOPILOT HOLD SIX SIXTY FIVE TRUE
A304 AT TWO ZERO ZERO SEARCH ABOVE FOUR THREE ZERO
A305 YES
A306 FOX THREE
A307 SIM
A308 THROTTLE RIGHT BACK
A309 SET I P E Y THREE ZERO EIGHT SIX ZERO SEVEN TWO FIVE ENTER
A310 AUTOPILOT HOLD MACH ZERO
A311 DISPLAY THREATS
A312 FUEL OPEN
A313 CLEAR PERFORMANCE
A314 SET M J V SEVEN NINE ZERO SIX TWO ZERO SIX SIX ENTER
A315 AUTOPILOT STEADY
A316 RADAR DULL
A317 AIM SEVEN BORE SIGHT
A318 CAUTION ACKNOWLEDGED
A319 FUEL OPEN
A320 DISPLAY OXYGEN
A321 AUTOPILOT EASY RIGHT
A322 AUTOPILOT BREAK LEFT
A323 RADAR SORT
A324 WARNING TEST
A325 AUTOPILOT LEVEL MAX
A326 COMM SET GROUND ONE NINETEEN POINT SEVEN TWO FIVE ENTER
A327 RADAR ALTITUDE DULL
A328 MID-RANGE
A329 SET M ELEVATION NINETY SIX THOUSAND EIGHTY SIX ENTER
A330 SET NEXT Q D THREE ZERO EIGHT ONE EIGHT FOUR FIVE NINE ENTER
A331 SET MODE FOUR B ENTER
A332 AUTOPILOT HOLD MAX ENDURANCE
A333 SILENCE
A334 RADAR ALTITUDE DULL
A335 RADAR ALTITUDE STANDBY
A336 LOUDER DOWN SOFTER ENTER
A337 NAV HEADING ONE SEVEN THREE ENTER
A338 I R AT ONE THREE ZERO SEARCH BELOW THREE THREE ONE
A339 CLEAR FUEL TEST

A340 LARGER UP RIGHT ENTER
A341 SET D SOUTH SEVENTEEN EIGHTEEN ENTER
A342 DISPENSE
A343 SQUAWK LOW
A344 SAFE
A345 CLEAR PLATE
A346 CLEAR OXYGEN
A347 SET ALTERNATE SOUTH NINE ZERO ZERO ZERO ZERO ZERO ENTER
A348 AUTOPILOT LEFT HARD AS POSSIBLE
A349 GUNS
A350 SOFTER SMALLER LARGER ENTER
A351 CLEAR I F R SUP O G
A352 I R BORE SIGHT
A353 V O R AUTO
A354 IDLE
A355 RADAR ALTITUDE LOCK
A356 CLEAR THREATS
A357 I L S HEADING ZERO THREE EIGHT ENTER
A358 V O R MUTE
A359 JETTISON TEN AND TEN WEAPON NOW
A360 CLEAR TERRAIN
A361 DISPLAY I F R SUP Q Y
A362 ARM NOW
A363 AUTOPILOT NOSE UP
A364 DISPLAY MARK ONE OH SIX
A365 RADAR ALTITUDE SHARP
A366 SILENCE
A367 WHY
A368 DOGFIGHT
A369 DISPLAY SCALES
A370 GUNS
A371 PAGE I F R SUP NEXT
A372 AUTOPILOT LEVEL OFF A LITTLE
A373 GO EMERGENCY
A374 GO ONE EIGHT POINT ZERO TWO SIX ENTER
A375 BACKUP
A376 CLEAR CONTROL
A377 AUDIO INCREASE
A378 CLEAR CHART
A379 AUTOPILOT HOLD MACH SIX
A380 R TWO TRAIN NINE
A381 IDLE
A382 DISPENSE
A383 AUTOPILOT CLIMB MORE
A384 R TWO TRAIN FOUR
A385 CLEAR PERFORMANCE
A386 GO SIX FOUR POINT SIX ZERO ZERO ENTER
A387 CLEAR INVENTORY
A388 GUNS
A389 FUEL CLOSE
A390 MIL
A391 RADAR ALTITUDE SORT
A392 AUTOPILOT NOSE UP
A393 SET PIGEONS NORTH THREE TEN TWO TEN ENTER
A394 COMM SET G C A ELEVEN POINT SIX FOUR FOUR ENTER
A395 CLEAR THREATS
A396 SET TWENTY FOUR Z O ZERO FOUR SIX NINE SIX ZERO TWO TWO ENTER

A397 E P U RESET NOW
A398 DISPLAY MAP
A399 DESTINATION ALTERNATE ENTER
A400 CLEAR TERRAIN
A401 SLAVE AIM NINE TO R W R
A402 TACAN SET ONE HUNDRED SIXTEEN Y ENTER
A403 AUTOPILOT LEVEL OFF MAX
A404 I R SHARP
A405 SIM
A406 RADAR SORT
A407 R TWO TRAIN TWO
A408 DISPLAY MARK EIGHTY FOUR
A409 SET MODE THREE SIX ZERO ZERO SIX ENTER
A410 MIL
A411 I N S OVERFLY UP
A412 CLEAR SOFT
A413 G C A OVER
A414 CLEAR NAV
A415 SOFTER LEFT UP ENTER
A416 NO
A417 SPEED BRAKE OPEN
A418 I N S OVERFLY UP
A419 I R SCAN
A420 DISPLAY CHAFF AND FLARE
A421 AT ONE FIVE FIVE SEARCH BELOW NINE THOUSAND
A422 JETTISON ONE TWO AND FIFTEEN AND ELEVEN AND FOURTEEN RACK NOW
A423 IDLE
A424 SET NINETEEN Z X NINE SEVEN EIGHT SIX NINE SIX FIVE FIVE ENTER
A425 TOP
A426 DISPLAY DIMMER
A427 THROTTLE RIGHT BACK
A428 PAGE CHART DOWN
A429 E P U RESET NOW
A430 I N S SOUTH ZERO SEVEN ONE ZERO TWO THIRTEEN ENTER
A431 IDENT
A432 DISPLAY CHECKLIST AT T M
A433 JETTISON FLARE NOW
A434 SLAVE AIM NINE MISSILES TO SIGHT
A435 DISPLAY SOFT
A436 AUTOPILOT HOLD ONE THOUSAND ONE HUNDRED TWO TRUE
A437 YES
A438 CLEAR CHECKLIST B L
A439 I R LOCK
A440 DISPLAY BRIGHTER
A441 AUTOPILOT DOWN MORE
A442 SLAVE AIM NINE MISSILES TO SIGHT
A443 CLEAR SCALES
A444 RADAR STAB-OUT ONE TWO TWO BY THREE
A445 DISPLAY DIMMER
A446 CAUTION ACKNOWLEDGED
A447 SIM
A448 DISPLAY HIGH INDEX
A449 R TWO TRAIN SIX
A450 UNCAGE
A451 AUTOPILOT HOLD DESCENT
A452 SLAVE AIM NINE MISSILES TO R W R
A453 FOX TWO

A454 AUTOPILOT HOLD TWO THOUSAND TEN TRUE
A455 CLEAR OXYGEN
A456 SOFTER LOUDER LOUDER ENTER
A457 SET SIXTY V A TWO SIX ONE SEVEN ZERO TWO NINE TWO ENTER
A458 MID-RANGE
A459 PAGE CHART UP
A460 EMERGENCY OVER
A461 AUDIO DECREASE
A462 CLEAR DESTINATION TARGET
A463 JETTISON FOURTEEN AND THIRTEEN AND ONE THREE AND SIXTEEN RACK NOW
A464 SET Q T J SEVEN TWO FOUR FOUR FIVE TWO ZERO TWO ENTER
A465 V H F INCREASE
A466 GUNS
A467 AUTOPILOT HOLD MAX ENDURANCE
A468 I R AT ONE NINE FIVE SEARCH ABOVE NINE POINT FIVE THOUSAND
A469 STRAFE
A470 AUTOPILOT BURNER CLIMB
A471 TOP
A472 AIM SEVEN BORE
A473 I R DULL
A474 CLEAR THREATS
A475 AUTOPILOT RIGHT EASY
A476 SET MODE THREE FIVE THREE ONE TWO ENTER
A477 DISPLAY HIGH ALTITUDE INDEX
A478 SET ELEVEN ELEVATION NINETY FOUR THOUSAND SEVENTY FOUR ENTER
A479 AUTOPILOT HOLD THREE FOUR NINE SIX INDICATED
A480 CLEAR INVENTORY
A481 NO
A482 IDENT
A483 SET NEXT E Q SEVEN THREE ZERO NINE SIX FOUR SIX ONE ENTER
A484 SQUAWK EMERGENCY
A485 TOP
A486 SET EIGHTEEN TIME ZERO THREE TWELVE FIVE EIGHTEEN ENTER
A487 DISPLAY DIMMER
A488 AUTOPILOT LEVEL MAX
A489 SET TARGET NORTH EIGHT ZERO THREE ONE ONE FIVE ENTER
A490 DISPLAY MAP
A491 EXIT
A492 DISPLAY PERFORMANCE
A493 NO
A494 FUEL OPEN
A495 AUTOPILOT LEFT EASY
A496 GO V H F
A497 AUTOPILOT STEADY
A498 CLEAR MARK EIGHTY FOUR
A499 DISPLAY MARK THIRTY SIX
A500 AUTOPILOT EASY LEFT
A501 SLAVE AIM NINE MISSILES TO R W R
A502 SPEED BRAKE STOP
A503 DISPLAY THREATS
A504 DISPLAY CHART
A505 SILENCE
A506 SQUAWK
A507 AUTOPILOT HARDER
A508 SQUAWK
A509 DISPLAY OXYGEN
A510 AUTOPILOT HOLD MAX ENDURANCE

A511 SPEED BRAKE STOP
A512 CLEAR INVENTORY
A513 RADAR STAB-OUT THREE SIX ZERO BY TWO TWO FOUR
A514 I N S EAST ZERO NINE SIX THIRTY SIX THIRTY SIX ENTER
A515 NAV AUTO
A516 CLEAR SCALES
A517 AUTOPILOT INCREASE SIX ZERO
A518 CAUTION TEST
A519 CLEAR SOFT
A520 DISPLAY HIGH INDEX
A521 I N S TIME TWENTY HUNDRED ENTER
A522 RADAR STAB-OUT SIX ONE BY ONE TWO ZERO
A523 AUTOPILOT STEADY
A524 I R AT ONE SEVEN ZERO SEARCH SURFACE
A525 I R SCAN
A526 THROTTLE RIGHT BACK
A527 DISPLAY DIMMER
A528 OXYGEN
A529 SAFE
A530 UNCAGE
A531 THROTTLE RIGHT BACK
A532 JETTISON ONE FIVE AND ONE EIGHT AND TWELVE AND SEVENTEEN RACK NOW
A533 PAGE X K ENTER
A534 RADAR ALTITUDE LOCK
A535 IDLE
A536 AUTOPILOT HOLD ONE THOUSAND FIFTY INDICATED
A537 BACKUP
A538 SET FIVE B H FOUR NINE ONE EIGHT EIGHT EIGHT SEVEN SIX ENTER
A539 AUTOPILOT EASE OFF

Appendix E: List of Speech Sessions

Listed below are sessions of the ten speakers originally acquired from AAMRL. Speakers #9 and #0 were excluded from the actual research in this thesis when it was discovered that their data was contaminated by a loose oxygen mask and a faulty inhalation/exhalation valve.

Table 27. Identification data for speech sessions

Session	Name	Condition	Tape Code
11	Stanton, Bill	normal	G2L
12		loud	G2L
13		Lombard	G2R
21	Welde, William	normal	B2R
22		loud	C1L
23		Lombard	C2L
31	Goci, Michael	normal	E1L
32		loud	E2L
33		Lombard	E1R
41	Gilio, James	normal	A2R
42		loud	B1L
43		Lombard	B1R
51	Cordner, Tim	normal	D2L
52		loud	D1R
53		Lombard	D2R
61	Johnson, Steve	normal	F1R
62		loud	F2R
63		Lombard	G1L
71	Ericson, Mark	normal	H1L
72		loud	H1L
73		Lombard	G1R
81	Ota, Gary	normal	A1L
82		loud	A2L
83		Lombard	A1R
91	Cross, Lee	normal	C1R
92		loud	C2R
93		Lombard	D1L
01	Boring, Gene	normal	E2R
02		loud	F1L
03		Lombard	F2L

Appendix F: Vocabulary Transcriptions

A	/ EY /
ABOVE	/ AX / B / AH / V /
ACKNOWLEDGED	/ AE / KO / N / AA / L / EH / JH / D /
AIM	/ EY / M /
ALTERNATE	/ AO / L / T / AXR / N / AX / T /
ALTITUDE	/ AE / L / T / IH / T / UW / D /
AND	/ AE / N / D /
ARM	/ AA / ER / M /
AS	/ AE / Z /
AT	/ AE / T /
AUDIO	/ AO / D / IY / OW /
AUTO	/ AO / DX / OW /
AUTOPILOT	/ AO / DX / OW / P / AY / L / AX / T /
B	/ B / IY /
BACK	/ B / AE / K /
BACKUP	/ B / AE / K / AH / PO /
BELOW	/ B / AX / L / OW /
BORE	/ B / OW / ER /
BRAKE	/ B / R / EY / K /
BREAK	/ B / R / EY / K /
BRIGHTER	/ B / R / AY / DX / AXR /
BURNER	/ B / ER / N / AXR /
BY	/ B / AY /
C	/ S / IY /
CAGE	/ K / EY / JH /
CAUTION	/ K / AO / SH / AX / N /
CHAFF	/ CH / AE / F /
CHART	/ CH / AA / ER / T /
CHECKLIST	/ CH / EH / KO / L / IH / S / T /
CLEAR	/ K / L / IY / ER /
CLIMB	/ K / L / AY / M /
CLOSE	/ K / L / OW / Z /
COMM	/ K / AA / M /
CONTROL	/ K / AX / N / T / R / OW / L /
D	/ D / IY /
DECREASE	/ D / IY / K / R / IY / S /
DESCENT	/ D / AX / S / IH / N / TO /
DESTINATION	/ D / EH / S / T / AX / N / EY / SH / AX / N /
DIMMER	/ D / IH / M / AXR /
DIRECT	/ D / AXR / EH / KO / T /
DISPENSE	/ D / IH / S / P / IH / N / S /
DISPLAY	/ D / IH / S / P / L / EY /
DIVE	/ D / AY / V /
DOGFIGHT	/ D / AO / GO / F / AY / T /
DOWN	/ D / AW / N /
DULL	/ D / AH / L /
E	/ IY /
EASE	/ IY / Z /
EAST	/ IY / S / T /
EASY	/ IY / Z / IY /
EIGHT	/ EY / T /
EIGHTEEN	/ EY / T / IY / N /

EIGHTY	/ EY / DX / IY /
ELEVATION	/ EH / L / AX / V / EY / SH / AX / N /
ELEVEN	/ AX / L / EH / V / AX / N /
EMERGENCY	/ IH / M / ER / JH / AX / N / S / IY /
ENDURANCE	/ IH / N / D / UH / ER / AX / N / S /
ENGINE	/ IH / N / JH / IH / N /
ENTER	/ IH / N / T / AXR /
EXIT	/ EH / GO / Z / IH / TO /
F	/ EH / F /
FIFTEEN	/ F / IH / F / T / IY / N /
FIFTY	/ F / IH / F / T / IY /
FIVE	/ F / AY / V /
FLARE	/ F / L / EY / ER /
FORTY	/ F / OW / ER / T / IY /
FOUR	/ F / OW / ER /
FOURTEEN	/ F / OW / ER / T / IY / N /
FOX	/ F / AA / KO / S /
FUEL	/ F / Y / UW / L /
G	/ JH / IY /
GO	/ G / OW /
GROUND	/ G / R / AW / N / D /
GUNS	/ G / AH / N / Z /
H	/ EY / CH /
HARD	/ HH / AA / ER / D /
HARDER	/ HH / AA / ER / D / AXR /
HEADING	/ HH / EH / D / IH / NX /
HIGH	/ HH / AY /
HOLD	/ HH / OW / L / D /
HUNDRED	/ HH / AH / N / D / R / AX / D /
I	/ AY /
IDENT	/ AY / D / IH / N / T /
IDLE	/ AY / D / EL /
INCREASE	/ IH / N / K / R / IY / S /
INDEX	/ IH / N / D / EH / KO / S /
INDICATED	/ IH / N / D / AX / K / EY / DX / AX / D /
INVENTORY	/ IH / N / V / IH / N / T / OW / ER / R / IY /
J	/ JH / EY /
JETTISON	/ JH / EH / DX / AX / S / AX / N /
K	/ K / EY /
L	/ EH / L /
LARGER	/ L / AA / ER / JH / AXR /
LEFT	/ L / EH / F / T /
LEVEL	/ L / EH / V / EL /
LITTLE	/ L / IH / DX / EL /
LOCK	/ L / AA / K /
LOUDER	/ L / AW / D / AXR /
LOW	/ L / OW /
M	/ IH / M /
MACH	/ M / AA / K /
MAP	/ M / AE / P /
MARK	/ M / AA / ER / K /
MAX	/ M / AE / KO / S /
MID-RANGE	/ M / IH / D / R / EY / N / JH /
MIL	/ M / IH / L /
MISSILES	/ M / IH / S / EL / Z /
MODE	/ M / OW / D /
MORE	/ M / OW / ER /

MUTE	/ M / Y / UW / T /
N	/ IH / N /
NAV	/ N / AE / V /
NEXT	/ N / EH / KO / S / T /
NINE	/ N / AY / N /
NINETEEN	/ N / AY / N / T / IY / N /
NINETY	/ N / AY / N / T / IY /
NO	/ N / OW /
NORTH	/ N / OW / ER / TH /
NOSE	/ N / OW / Z /
NOW	/ N / AW /
O	/ OW /
OFF	/ AO / F /
OH	/ OW /
ONE	/ W / AH / N /
OPEN	/ OW / P / AX / N /
OVER	/ OW / V / AXR /
OVERFLY	/ OW / V / AXR / F / L / AY /
OXYGEN	/ AA / KO / S / AX / JH / AX / N /
P	/ P / IY /
PAGE	/ P / EY / JH /
PERFORMANCE	/ P / AXR / F / OW / ER / M / AX / EN / S /
PIGEONS	/ P / IH / JH / AX / N / Z /
PLATE	/ P / L / EY / T /
POINT	/ P / OY / N / T /
POSSIBLE	/ P / AA / S / AX / B / EL /
PREVIOUS	/ P / R / IY / V / IY / AX / S /
Q	/ K / Y / UW /
R	/ AA / ER /
RACK	/ R / AE / K /
RADARD	/ R / EY / D / AA / ER /
RESET	/ R / IY / S / EH / T /
RIGHT	/ R / AY / T /
S	/ EH / S /
SAFE	/ S / EY / F /
SCALES	/ S / K / EY / L / Z /
SCAN	/ S / K / AE / N /
SEARCH	/ S / ER / CH /
SET	/ S / EH / T /
SEVEN	/ S / EH / V / AX / N /
SEVENTEEN	/ S / EH / V / AX / N / T / IY / N /
SEVENTY	/ S / EH / V / AX / N / T / IY /
SHARP	/ SH / AA / ER / P /
SIGHT	/ S / AY / T /
SILENCE	/ S / AY / L / AX / N / S /
SIM	/ S / IH / M /
SIX	/ S / IH / KO / S /
SIXTEEN	/ S / IH / KO / S / T / IY / N /
SIXTY	/ S / IH / KO / S / T / IY /
SLAVE	/ S / L / EY / V /
SMALLER	/ S / M / AO / L / AXR /
SOFT	/ S / AO / F / T /
SOFTER	/ S / AO / F / T / AXR /
SORT	/ S / OW / ER / T /
SOUTH	/ S / AW / TH /
SPEED	/ S / P / IY / D /
SQUAWK	/ S / K / W / AO / K /

STAB-OUT	/ S / T / AE / B / AW / T /
STANDBY	/ S / T / AE / N / DO / B / AY /
STEADY	/ S / T / EH / D / IY /
STOP	/ S / T / AA / P /
STRAFE	/ S / T / R / EY / F /
SUPP	/ S / AH / P /
SURFACE	/ S / ER / F / AX / S /
T	/ T / IY /
TACAN	/ T / AE / K / AE / N /
TARGET	/ T / AA / ER / G / EH / T /
TEN	/ T / IH / N /
TERRAIN	/ T / AXR / R / EY / N /
TEST	/ T / EH / S / T /
THIRTEEN	/ TH / ER / T / IY / N /
THIRTY	/ TH / ER / T / IY /
THOUSAND	/ TH / AW / Z / AX / N /
THREATS	/ TH / R / EH / T / S /
THREE	/ TH / R / IY /
THROTTLE	/ TH / R / AA / DX / EL /
TIME	/ T / AY / M /
TO	/ T / UW /
TOP	/ T / AA / P /
TRAIN	/ T / R / EY / N /
TRUE	/ T / R / UW /
TWELVE	/ T / W / EH / L / V /
TWENTY	/ T / W / IH / N / T / IY /
TWO	/ T / UW /
U	/ Y / UW /
UNCAGE	/ AH / N / K / EY / JH /
UP	/ AH / P /
V	/ V / IY /
W	/ D / AH / B / EL / Y / UW /
WARNING	/ W / AA / ER / N / IH / NX /
WEAPON	/ W / EH / P / AX / N /
WEST	/ W / EH / S / T /
WHY	/ W / HV / AY /
X	/ EH / K / S /
Y	/ W / AY /
YES	/ Y / EH / S /
Z	/ Z / IY /
ZERO	/ Z / IY / R / OW /

Appendix G: List of Phonemes

Phoneme	ARPAbet	Example	Phoneme	ARPAbet	Example
p	P	<i>pet</i>	y	Y	<i>yes</i>
t	T	<i>tap</i>	h	HH	<i>hat</i>
k	K	<i>kit</i>	l	EL	<i>bottle</i>
b	B	<i>bat</i>	w	W	<i>win</i>
d	D	<i>dip</i>	ε	EH	<i>bet</i>
g	G	<i>go</i>	ɔ	AO	<i>bought</i>
r	DX	<i>butter</i>	a	AA	<i>pot</i>
m	M	<i>mat</i>	u	UW	<i>boot</i>
n	N	<i>net</i>	ʒ	ER	<i>bird</i>
ŋ	NX	<i>sing</i>	a ^y	AY	<i>bite</i>
s	S	<i>sap</i>	e ^y	EY	<i>bait</i>
z	Z	<i>zip</i>	a ^w	AW	<i>bout</i>
ʃ	CH	<i>chair</i>	ə	AX	<i>about</i>
θ	TH	<i>think</i>	ɪ	IH	<i>bit</i>
f	F	<i>fat</i>	æ	AE	<i>pan</i>
ʃ	SH	<i>shoe</i>	ʌ	AH	<i>up</i>
j	JH	<i>joke</i>	o ^y	OY	<i>void</i>
v	V	<i>vat</i>	i ^y	IY	<i>beet</i>
l	L	<i>link</i>	o ^w	OW	<i>boat</i>
r	R	<i>rap</i>	ɔ̃	AXR	<i>over</i>

Appendix H: Vocabulary Word - Phoneme Cross Reference

P

TOP
AUTOPILOT
MAP
SUP

OPEN
POSSIBLE
PLATE
PAGE

P
UP
PERFORMANCE
DISPENSE

SPEED
POINT
PIGEONS
WEAPON

STOP
DISPLAY
PREVIOUS
SHARP

T

TOP
RESET
IDENT
THIRTEEN
EIGHTEEN
FORTY
RIGHT
EAST
THREATS
INVENTORY
DESTINATION
AT
SIGHT
SORT

TWO
NEXT
SET
FOURTEEN
NINETEEN
FIFTY
NINETY
WEST
CHART
SOFT
ALTERNATE
CHECKLIST
MUTE
ENTER

SOFTER
TEST
TIME
FIFTEEN
TWENTY
STOP
SIXTY
POINT
PLATE
T
ALTERNATE
DOGFIGHT
TO

TRAIN
TEST
TEN
SIXTEEN
TWENTY
AUTOPILOT
SEVENTY
TRUE
CONTROL
ALTITUDE
TARGET
DIRECT
STAB-OUT

EIGHT
STANDBY
TWELVE
SEVENTEEN
THIRTY
LEFT
STEADY
TERRAIN
TACAN
ALTITUDE
TARGET
STRAFE
STAB-OUT

K

BACKUP
SQUAWK
MACH
CONTROL
X
LOCK

CLOSE
BREAK
INDICATED
SCALES
MARK
SCAN

BRAKE
CLIMB
BACK
TACAN
CAGE

CAUTION
INCREASE
CLEAR
K
UNCAGE

SQUAWK
DECREASE
COMM
Q
RACK

B

BACKUP
POSSIBLE
BORE

BRAKE
BURNER
STAB-OUT

STANDBY
BACK
ABOVE

B
W
BELOW

BREAK
BRIGHTER
BY

D

ACKNOWLEDGED
MODE
STEADY
ENDURANCE
MID-RANGE
ALTITUDE
DIMMER
AUDIO

LOUDER
HOLD
DOWN
INDICATED
DISPLAY
AND
DOGFIGHT

HUNDRED
SPEED
DESCENT
INDICATED
HEADING
DESTINATION
DIRECT

HUNDRED
HARD
DIVE
GROUND
W
D
DISPENSE

IDENT
HARDER
DECREASE
IDLE
RADAR
INDEX
DULL

G

GROUND

TARGET

GUNS

GO

DX

AUTOPILOT
BRIGHTER

EIGHTY
JETTISON

LITTLE
AUTO

INDICATED

THROTTLE

M

M
MORE
COMM
MISSILES
MUTE

EMERGENCY
MAX
MAP
AIM

MODE
MACH
PERFORMANCE
MARK

TIME
MID-RANGE
DIMMER
SIM

CLIMB
MIL
SMALLER
ARM

N

SILENCE
SEVEN
EMERGENCY
STANDBY
FOURTEEN
EIGHTEEN
NINETY
DESCENT
INCREASE
MID-RANGE
ENGINE
AND
INDEX
WEAPON

ACKNOWLEDGED
NINE
WARNING
IDENT
FIFTEEN
NINETEEN
NINETY
BURNER
ENDURANCE
TERRAIN
ENGINE
DESTINATION
GUNS
SCAN

NO
NINE
CAUTION
TEN
SIXTEEN
NINETEEN
SEVENTY
NOSE
ENDURANCE
CONTROL
INVENTORY
DESTINATION
UNCAGE
ELEVATION

TRAIN
OPEN
OXYGEN
ELEVEN
SEVENTEEN
NINETEEN
NORTH
POINT
INDICATED
NAV
INVENTORY
PIGEONS
DISPENSE
NOW

ONE
NEXT
HUNDRED
THIRTEEN
SEVENTEEN
TWENTY
DOWN
THOUSAND
GROUND
TACAN
N
ALTERNATE
JETTISON
ENTER

NX

WARNING

HEADING

S

SILENCE
SIX
EMERGENCY
SET
STOP
STEADY
MAX
THREATS
DESTINATION
CHECKLIST
SIM
FOX
SURFACE

SILENCE
SEVEN
TEST
SIXTEEN
POSSIBLE
SOUTH
INCREASE
PERFORMANCE
X
SMALLER
DISPENSE
SLAVE
SCAN

YES
RESET
OXYGEN
SIXTEEN
SIXTY
EAST
DECREASE
SCALES
PREVIOUS
MISSILES
DISPENSE
SEARCH
SORT

SOFTER
C
STANDBY
SEVENTEEN
SIXTY
WEST
ENDURANCE
SOFT
INDEX
STRAFE
JETTISON
STAB-OUT

SIX
NEXT
SQUAWK
SPEED
SEVENTY
DESCENT
DISPLAY
S
SUP
SAFE
SIGHT
SURFACE

Z

EXIT
NOSE
MISSILES

ZERO
THOUSAND
GUNS

EASY
SCALES

AS
Z

EASE
PIGEONS

CH

CHART

H

CHAFF

CHECKLIST

SEARCH

TH

THREE
THOUSAND

THIRTEEN
THROTTLE

THIRTY
THREATS

NORTH

SOUTH

F

SOFTER
F
FIFTY
CHAFF
FOX

FOUR
FOURTEEN
FIFTY
FLARE
SURFACE

FIVE
FIFTEEN
LEFT
DOGFIGHT
OVERFLY

FUEL
FIFTEEN
PERFORMANCE
STRAFE

OFF
FORTY
SOFT
SAFE

SH

CAUTION

DESTINATION

SHARP

ELEVATION

JH

ACKNOWLEDGED
ENGINE
CAGE

EMERGENCY
J
UNCAGE

OXYGEN
PIGEONS
JETTISON

G
LARGER

MID-RANGE
PAGE

V

FIVE
SEVENTEEN
INVENTORY
ELEVATION

SEVEN
SEVENTY
PREVIOUS
OVER

V
DIVE
SLAVE

ELEVEN
LEVEL
ABOVE

TWELVE
NAV
OVERFLY

L

SILENCE
LOW
LEFT
MIL
SCALES
LARGER
DULL

ACKNOWLEDGED
HOLD
CLIMB
DISPLAY
ALTITUDE
SMALLER
OVERFLY

LOUDER
ELEVEN
LEVEL
CLEAR
FLARE
SLAVE
ELEVATION

FUEL
TWELVE
LITTLE
PLATE
ALTERNATE
BELOW

CLOSE
AUTOPILOT
L
CONTROL
CHECKLIST
LOCK

R

R
BRAKE
FORTY
NORTH
TRUE
TERRAIN
INVENTORY
PREVIOUS
ARM

TRAIN
WARNING
RIGHT
MORE
GROUND
THREATS
RADAR
BRIGHTER
RACK

THREE
HUNDRED
BREAK
INCREASE
THROTTLE
CHART
RADAR
LARGER
BORE

FOUR
ZERO
HARD
DECREASE
MID-RANGE
CONTROL
FLARE
STRAFE
SHARP

RESET
FOURTEEN
HARDER
ENDURANCE
CLEAR
PERFORMANCE
TARGET
MARK
SORT

Y

YES
MUTE

FUEL

U

W

Q

HH

HUNDRED
HIGH

HOLD

HARD

HARDER

HEADING

EL

POSSIBLE
W

LEVEL
MISSILES

LITTLE

THROTTLE

IDLE

W

WHY
TWENTY

ONE
WEST

WARNING
Y

SQUAWK
WEAPON

TWELVE

EH

ACKNOWLEDGED
NEXT
TWELVE
WEST
S
CHECKLIST

YES
TEST
SEVENTEEN
LEVEL
DESTINATION
DIRECT

EXIT
F
LEFT
L
X
JETTISON

SEVEN
SET
SEVENTY
THREATS
TARGET
WEAPON

RESET
ELEVEN
STEADY
HEADING
INDEX
ELEVATION

AO

SOFTER
SOFT
AUDIO

OFF
ALTERNATE

CAUTION
SMALLER

SQUAWK
DOGFIGHT

AUTOPILOT
AUTO

AA

ACKNOWLEDGED
STOP
THROTTLE
LARGER
SHARP

TOP
HARD
COMM
MARK

R
POSSIBLE
CHART
ARM

WARNING
HARDER
RADAR
FOX

OXYGEN
MACH
TARGET
LOCK

UW

TWO
ALTITUDE

FUEL
Q

U
MUTE

TRUE
TO

W

ER

EMERGENCY
SEARCH

THIRTEEN
SURFACE

THIRTY

BURNER

PERFORMANCE

AY

SILENCE
STANDBY
RIGHT
Y
BY

WHY
IDENT
NINETY
HIGH
OVERFLY

FIVE
TIME
CLIMB
BRIGHTER

NINE
NINETEEN
DIVE
DOGFIGHT

I
AUTOPILOT
IDLE
SIGHT

EY

TRAIN
BREAK
TERRAIN
FLARE
STRAFE
SLAVE

EIGHT
EIGHTY
PLATE
DESTINATION
AIM
ELEVATION

BRAKE
INDICATED
SCALES
J
SAFE

A
MID-RANGE
H
K
CAGE

EIGHTEEN
DISPLAY
RADAR
PAGE
UNCAGE

AW

LOUDER
STAB-OUT

SOUTH
NOW

DOWN

THOUSAND

GROUND

AX

SILENCE
OXYGEN
SEVENTEEN
THOUSAND
PERFORMANCE
PREVIOUS
ABOVE

SEVEN
OXYGEN
AUTOPILOT
ENDURANCE
DESTINATION
JETTISON
BELOW

OPEN
HUNDRED
POSSIBLE
INDICATED
DESTINATION
JETTISON
ELEVATION

EMERGENCY
ELEVEN
SEVENTY
INDICATED
PIGEONS
WEAPON
ELEVATION

CAUTION
ELEVEN
DESCENT
CONTROL
ALTERNATE
SURFACE

IH

EXIT
IDENT
FIFTY
ENDURANCE
HEADING
N
DIMMER
ENTER

SIX
TEN
SIXTY
INDICATED
ENGINE
ALTITUDE
MISSILES

M
FIFTEEN
DESCENT
MID-RANGE
ENGINE
PIGEONS
SIM

EMERGENCY
SIXTEEN
LITTLE
MIL
INVENTORY
INDEX
DISPENSE

WARNING
TWENTY
INCREASE
DISPLAY
INVENTORY
CHECKLIST
DISPENSE

AE

ACKNOWLEDGED
BACK
ALTITUDE
STAB-OUT

BACKUP
MAP
CHAFF
SCAN

STANDBY
NAV
AND

AS
TACAN
AT

MAX
TACAN
RACK

AH

BACKUP
SUP

ONE
GUNS

HUNDRED
UNCAGE

UP
ABOVE

W
DULL

OY

POINT

IY

THREE
C
THIRTEEN
EIGHTEEN
FIFTY
NINETY
EAST
CLEAR
PREVIOUS

CLOSE
EMERGENCY
FOURTEEN
NINETEEN
SPEED
SIXTY
G
INVENTORY
PREVIOUS

E
V
FIFTEEN
TWENTY
EASY
SEVENTY
INCREASE
T
AUDIO

P
ZERO
SIXTEEN
THIRTY
EASY
EIGHTY
DECREASE
D

RESET
B
SEVENTEEN
FORTY
EASE
STEADY
DECREASE
Z

OW

NO
MODE
AUTOPILOT
PERFORMANCE
BELOW
GO

FOUR
HOLD
NORTH
INVENTORY
AUTO
OVER

OPEN
ZERO
MORE
O
SORT

CLOSE
FOURTEEN
NOSE
OH
OVERFLY

LOW
FORTY
CONTROL
BORE
AUDIO

AXR

LOUDER
ALTERNATE
DIRECT

SOFTER
BRIGHTER
OVERFLY

HARDER
DIMMER
ENTER

BURNER
LARGER
OVER

TERRAIN
SMALLER

Appendix I: Algorithm for Selection of Utterance Set

For all approaches, the isolated word phonetic transcriptions were used to obtain first-order approximations of the phonemes contained in a given utterance. The general idea was to look for a set of utterances that contained at least six repetitions of the phonemes listed in Appendix D without including an excessive amount of extra phonemes. The theoretical minimum phoneme count is simply

$$(\text{minimum phoneme coverage}) \times (\text{number of phonemes in set of interest})$$

So for this research the minimum phoneme count was 240. The first approach examined every utterance and selected the one that provided the highest total phoneme coverage¹. This utterance was then designated as belonging to the set of selected utterances, and the phoneme coverages were stored. Then the remaining utterances were scanned for the utterance providing the most new coverage. This process was continued until all phonemes were fully covered. It turned out that this method was not very efficient because it selected an utterance set containing roughly three times the minimal amount required. Upon examination of the order of selection, it was noted that the last 11 utterances selected were each contributing a single token of needed coverage, indicating that some phonemes were harder to cover than others. Therefore the following subset of *troublesome* phonemes was selected: /NX SH G CH Y EL DX JH AO ER/. The original approach was now broken into two stages where only the troublesome subset was covered in stage one, and then the remainder of the phonemes were covered in stage two. This yielded some improvement by reducing the total phoneme count by approximately 9%.

The next approach selected utterances by minimizing the number of excess phonemes (i.e. the phonemes that did not contribute to coverage) for a given utterance. Again the process was broken into two stages where the smaller phoneme set was first covered followed by the remaining phoneme set.

-
1. The term *coverage* in this context refers to the number of tokens obtained for any given phoneme. If the total number of tokens of a phoneme is less than the minimum number of tokens required, that phoneme is said to be *uncovered*. If the number of tokens for a phoneme is equal to or greater than the minimum number of required tokens, then that phoneme is *covered*. *Coverage* then is the number of tokens provided for an uncovered phoneme, not to exceed the number required to fully cover that phoneme.

This approach heavily favored the single-word utterances and produced negligible improvement in total phoneme count.

Finally, an algorithm was used that selected utterances by first maximizing the delta increase in phoneme coverage, and then searching for the shortest utterance that provided this delta increase. Again, the troublesome phoneme set was covered first, followed by the remaining phoneme set. Then a new iterative procedure was applied to the set of selected utterances whereby each utterance was temporarily removed from the set to determine its final contribution to phoneme coverage. If coverage was unaffected, then the utterance was discarded. If any phonemes were left uncovered, then the set of non-selected utterances was searched to determine if an utterance existed that would restore coverage and at the same time contribute less excess phonemes than the original utterance under scrutiny. This iteration continued until an utterance set was produced with each member optimally contributing to the required phoneme count. The reduction in total phoneme count was roughly 22% over the original selection method, with the total phoneme count being roughly 2.3 times the minimal phoneme count. The algorithm is illustrated in Figure 52.

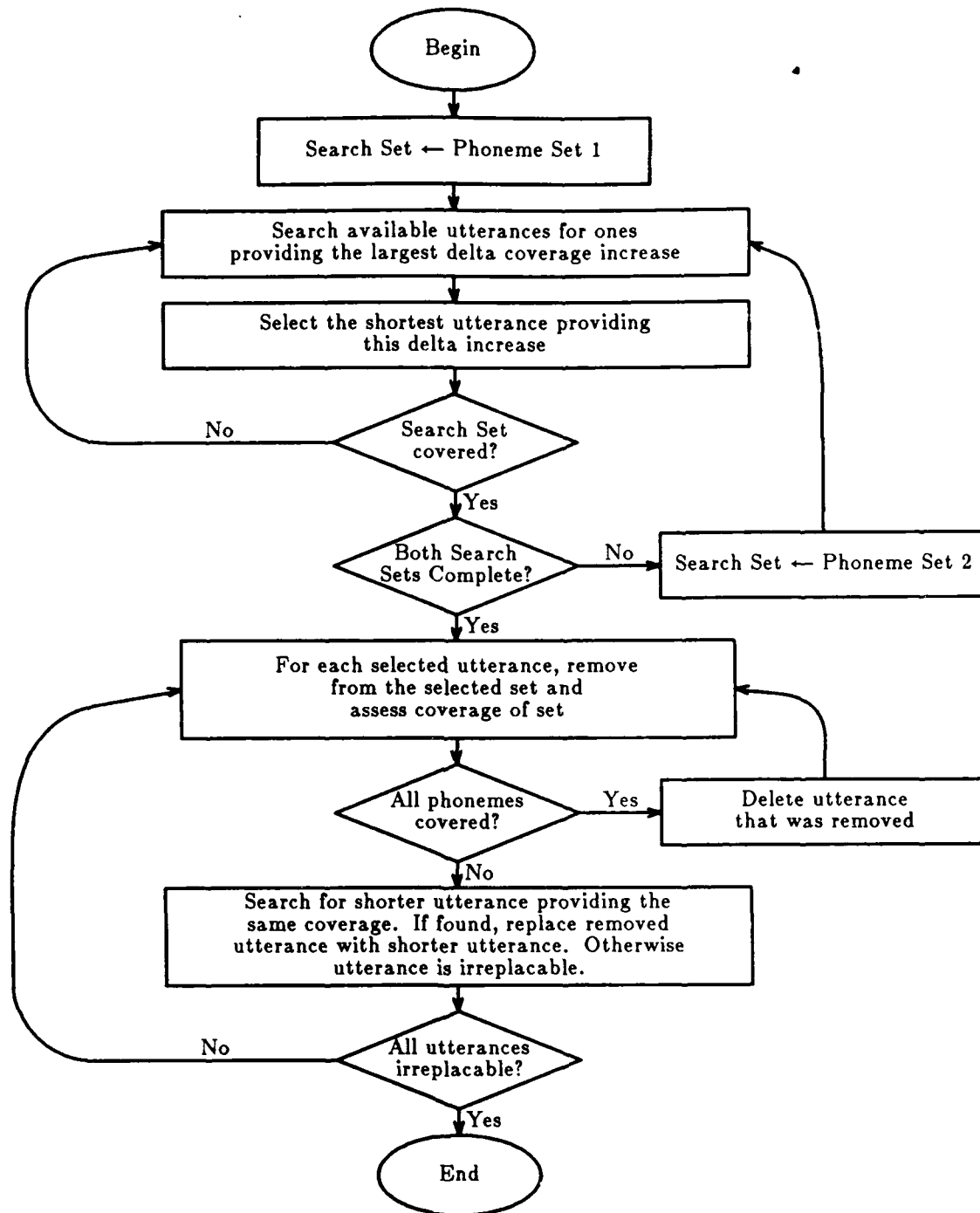


Figure 52. Utterance selection algorithm

Appendix J: Fortran Code

```

      subroutine accumdat(dat,itotspkr,adat1,adat2,adat3)
c This routine accumulates XBAR (averages) of all the features
c per phoneme for all speakers indicated by ITOTSPEAKER. The
c results are stored by condition (normal, loud, and Lombard)
c in ADAT1, ADAT2, and ADAT3.

```

```

      integer itotspkr,ispkr,iphone,xbar
      real dat(1200,20,3)
      real adat1(1200,20),adat2(1200,20),adat3(1200,20)

      do 100 iphone=1, 40

        xbar = 7*iphone - 4

        do 80 ifeat=1, 18

          adat1(xbar,ifeat) = adat1(xbar,ifeat) +
a          dat(xbar,ifeat,1)/itotspkr
          adat2(xbar,ifeat) = adat2(xbar,ifeat) +
a          dat(xbar,ifeat,2)/itotspkr
          adat3(xbar,ifeat) = adat3(xbar,ifeat) +
a          dat(xbar,ifeat,3)/itotspkr

        80      continue
      100      continue

      return
      end

```

c -----

```

      subroutine accumphon(i,icover)

c This routine adds the phone counts to the array ICOVER
c for utterance i. The phone lists for all the utterances
c are passed through the common block IP.

      integer i,icover(126),ipud(15000),ipun(15000),j

      common /ip/ipud,ipun

c J is the internal pointer for this routine.
c ICOVER is indexed by phoneme; i.e ICOVER(j) represents
c the number of occurrences of phoneme j.

```

```

      j = i

10      icover(ipud(j)) = icover(ipud(j)) + 1
      if (ipun(j).ne.0) then
        j = ipun(j)
        goto 10
      end if

      return
      end

```

c -----

```
subroutine analant2(x,itoken,ifatot,ispkr,icond)
```

```
c This subroutine is derived from the standalone program ANALANT.
c This version is intended to be merged with program ANALYZE in
c order to make the code more manageable and maintainable.
c It is designed to calculate and store average values of
c formants, pitch, and duration for the 40 phonemes of interest.
c It uses raw data that is stored in the label files and formant
c files generated by the LISPM on each utterance. The scheme is to
c work through all the utterances similar to the method used in
c TEMPLATES2. This stores formant data on the phonemes in an
c intermediate data structure. Then it iterates through the phonemes
c similar to the method in ANALYZE to transfer the statistics into
c the common data structure used for all analyses.
```

```
c F          is the buffer that contains the raw formant/pitch
c             samples that are read in from LISPM files.
c   F(K,0)    contains pitch samples (f0)
c   F(K,1)    contains first formant samples (f1)
c   F(K,2)    contains second formant samples (f2)
c   F(K,3)    contains third formant samples (f3)
c FORMANTFILE is the file containing the formant or pitch data
c FROMPOS()   contains the list of start times for the labels
c I1          is the starting index for formant/pitch samples for
c             a phoneme
c I2          is the ending index for formant/pitch samples for
c             a phoneme
c IALL        is the the total number of formant/pitch samples for
c IDIM        is the dimension of ILABEL(), FROMPOS(), AND TOPOS()
c ILABEL()    contains the list of phoneme labels for an utterance
c IPHONE      contains the short index of the phoneme being processed
c ISHORT(K)   contains the list of ascii codes (labels) of phonemes
c             of interest
c ISTR        is used to hold the character length of LISTNAME
c ITOKEN(K)   contains the number of occurrences of phoneme K
c ITOT        contains the total number of labels read for an utterance
c IUTT        is the number of each utterance being processed
c LABELFILE   is the file containing labels of the phonetic transcription
c LISTNAME    is the file containing the list of utterances
c TOPOS()     contains the list of end times for the labels
c X           is the 3-dimensional intermediate data structure
c             where data is held after it has been processed
c             from array F() and before it is transferred into
c             array DAT(). The first index points to individual
c             samples. The second index points to features:
c             0 = f0 (pitch)
c             1 = f1
c             2 = f2
c             3 = f3
c             4 = duration
c             The third index identifies the phoneme.
```

```
character*36 listname,labelfile,formantfile
integer i,j,i1,i2,iall,iphone,itoken(40)
integer istr,iutt,ilabel(2000),itot,idim
integer ishort(40),iptr(126),ifatot,ifeat,ispkr,icond
real frompos(2000),topos(2000),f(1:2000,0:3),b(0:3)
real x(1:100,0:10,1:40),dur
```

```
include "ml"
```

```
c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.
```

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

- c The IPTR array is the complement of the ISHORT array. Given the
 c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
 c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
 c means that phoneme K is not in the set of 40 phonemes.

```

data (iptr(i),i= 1, 70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i= 71, 80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i= 81, 90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i= 91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/

```

```

data idim      /2000/

```

```

ifatot = 5

```

```

do 5 i=1, 40
  itoken(i) = 0

```

```

5      continue

```

```

10     format(/"Program ANALANT..." /
a      "Enter filename for list of utterances: ",%)

```

```

      write (6,10)

```

```

12     format (a36)

```

```

      read (5,12)listname

```

```

      istr = 36

```

```

      call endstr(listname,istr)

```

```

      write (6,*)"Opening file ",listname

```

```

      open (unit=50,file=listname,status="old")

```

```

      rewind (50)

```

```

15     format (i3)

```

c ***** Beginning of the loop that processes utterances

```

20     read (50,15,end=100)iutt

```

```

      call buildpath(iutt,0,0,"lbl",ispkr,icond,labelfile)

```

```

      call buildpath(iutt,0,0,"fmt",ispkr,icond,formantfile)

```

```

      write (6,*)"Reading from",labelfile

```

```

      call readlabels(labelfile,ilabel,frompos,topos,itot,idim)

```

```

      write (6,*)"Reading from",formantfile

```

```

      do 30 ifeat=0, ifatot-2

```

```

        call readfmts(formantfile,ifeat,f(1,ifeat))

```

```

30     continue

```

c Now sift through the tokens of this utterance, processing data

c from qualifying phonemes.

```

      do 80 j=1, itot

```

```

        dur = topos(j) - frompos(j)

```

```

        if (iptr(ilabel(j)).ne.0.and.dur.ge.0.016) then

```

```

          iphone = iptr(ilabel(j))

```

```

          itoken(iphone) = itoken(iphone) + 1

```

c Note that duration is stored as feature number IFATOT-1 due to
 c zero indexing.

$x(\text{itoken}(\text{iphone}), \text{ifatot}-1, \text{iphone}) = \text{topos}(j) - \text{frompos}(j)$

c Calculate the starting and ending points in the formant buffers

$i1 = \text{int}(\text{frompos}(j)/0.005) + 1$
 $i2 = \text{int}(\text{topos}(j)/0.005)$
 $iall = i2 - i1 + 1$

c Cycle through the five features of interest. Note that the
 c IF-THEN allows the averaging and smoothing of the first
 c four features because there are multiples samples for each phoneme.
 c Feature five, duration, has only one value, so it bypasses the
 c averaging step.

call fsmooth(f,i1,i2,b)

do 50 i=0, ifatot-1
 if (i.le.3) then
 $x(\text{itoken}(\text{iphone}), i, \text{iphone}) = b(i)$
 end if

50 continue

end if

80 continue

goto 20

100 close (50)

c ***** End of loop that processes utterances

return
 end

c -----

program analyze2

c Designed to calculate features for each of 40 phonemes for a
c given session

c DAT is the formal data structure that contains the result
c of all analyses. The second index indicates features.
c At present, features 1-13 are supplied by program
c ANALYZE, and 14-18 are supplied by program ANALANT.
c Indices 1-10: Energy band data
c Index 11: Center of gravity data
c Index 12: Low band (0-3kHz) spectral tilt
c Index 13: High band (3-8kHz) spectral tilt
c Index 14: Pitch frequency data
c Indices 15-17: Formants 1-3 data
c Index 18: Duration data
c HERE logical used to test whether or not a template file
c exists
c IBG address containing the beginning address for the
c list of individual samples of a feature for a
c phoneme
c IFATOT Total number of features supplied by ANALANT2
c IFFTOT Total number of features supplied by FEATURES
c INOC address for the number of occurrences for a phoneme
c IPHONE contains the short index of the phoneme being processed
c IPT keeps track of the next available storage location for
c individual samples in the data structure DAT()
c ITOK is used to iterate through the tokens of a given phoneme
c PARFILE filename for a given template
c XBAR address for the sample mean of a feature for a
c phoneme
c VAR address for the sample variance of a feature for a
c phoneme
c SSUM address for the sum of samples of a feature for a
c phoneme
c S2SUM address for the sum of squares of samples of a
c feature for a phoneme
c X is the 3-dimensional intermediate data structure
c where data is held after it has been processed
c by routine ANALANT2 and before it is transferred into
c array DAT(). The first index points to individual
c samples. The second index points to features:
c 0 = f0 (pitch)
c 1 = f1
c 2 = f2
c 3 = f3
c 4 = duration
c The third index identifies the phoneme.

logical here
character*36 parfile,analysisfile
integer iphone,itok,ipt,ifeat,noc,ifatot,iffot
integer inoc,ibg,xbar,var,ssum,s2sum,itoken(40)
integer ispk,icond
real ts(128,50),eb(20),dat(1200,20)
real x(1:100,0:10,1:40)

ispkr = -1
icond = -1

10 format (/ "Program ANALYZE2...")

```

write (6,10)

c Starting point in the data structure where individual samples
c for each phoneme are stored.

ipt = 301

call analant2(x,itoken,ifatot,ispkr,icond)

do 100 iphone=1, 40

  inoc = 7*iphone - 6
  ibg = 7*iphone - 5
  xbar = 7*iphone - 4
  var = 7*iphone - 3
  ssum = 7*iphone - 2
  s2sum = 7*iphone - 1

  noc = 0

  do 20 ifeat=1, 20
    dat(ibg,ifeat) = ipt
    dat(ssum,ifeat) = 0.0
    dat(s2sum,ifeat) = 0.0
20    continue

  do 80 itok=1, 100

    call buildpath(0,itok,iphone,"pft",ispkr,icond,parfile)
    write (6,*) parfile
    inquire (file=parfile,exist=here)
    if (.not.here) then
      noc = itok - 1
      itok = 100
      goto 75
    end if

    open (unit=1,file=parfile,status="old",form="unformatted")
    rewind (1)
    read (1) ts
    close (1)
    call features(ts,eb,iffatot)

    do 60 ifeat=1, iffatot+ifatot
      if (ifeat.le.iffatot) then
        dat(ipt,ifeat) = eb(ifeat)
      else
        dat(ipt,ifeat) = x(itok,ifeat-iffatot-1,iphone)
      end if
      dat(ssum,ifeat) = dat(ssum,ifeat) + dat(ipt,ifeat)
      dat(s2sum,ifeat) = dat(s2sum,ifeat) + dat(ipt,ifeat)**2
60    continue

    ipt = ipt + 1

75    continue
80    continue

  do 90 ifeat=1, iffatot+ifatot

    if (noc.ne.itoken(iphone)) then
85      format ("Disagreement in number of tokens for phoneme: ",i3/
a        "Program execution is terminated prematurely"/

```



```

a      "NOC VS ITOKEN(IPHONE): ", 2i8)
      write (6,85)iphone,noc,itoken(iphone)
      stop
      end if
      dat(xbar,ifeat) = dat(ssum,ifeat)/noc
      dat(var,ifeat) = (dat(s2sum,ifeat)-noc*(dat(xbar,ifeat)**2))/
a                                     (noc-1)
      dat(inoc,ifeat) = noc
90     continue

```

c Now save the results so far and print out a summary:

```

      call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
      open (unit = 2,
a         file = analysisfile,
a         status = "unknown",
a         form = "unformatted")
      rewind (2)
      write (2) dat
      close (2)

92     format(/"Statistics on Phoneme: ",i3)
93     format("Feature",20i11)
94     format(a5," = ",%)
95     format(20g11.4e1)

      write (6,92)iphone
      write (6,93) (ifeat,ifeat=1,ifftot+ifatot)
      write (6,94)"inoc"
      write (6,95) (dat(inoc,ifeat),ifeat=1,ifftot+ifatot)
      write (6,94)"ibg"
      write (6,95) (dat(ibg,ifeat),ifeat=1,ifftot+ifatot)
      write (6,94)"xbar"
      write (6,95) (dat(xbar,ifeat),ifeat=1,ifftot+ifatot)
      write (6,94)"var"
      write (6,95) (dat(var,ifeat),ifeat=1,ifftot+ifatot)
c     write (6,94)"ssum"
c     write (6,95) (dat(ssum,ifeat),ifeat=1,ifftot+ifatot)
c     write (6,94)"s2sum"
c     write (6,95) (dat(s2sum,ifeat),ifeat=1,ifftot+ifatot)

100    continue

      stop
      end
c -----

```

```
subroutine andpfa(pfa,pfaall)
```

```
c This routine does a logical AND of the elements in the
c two PFA arrays, storing the results in PFAALL. The ANDing
c operation is defined as:
c both elements equal --> element preserved
c both elements unequal --> element set to 0
```

```
integer pfa(40,20),pfaall(40,20)
integer i,j
```

```
do 100 i=1, 40
  do 80 j=1, 20
```

```
    if (pfaall(i,j).ne.pfa(i,j)) pfaall(i,j) = 0
```

```
80      continue
100     continue
```

```
return
end
```

```
c -----
```

```
subroutine anova2(d,nd,kd,n,k,f,dfb,dfw,xm)
```

```
c This routine is designed to perform a simple Analysis of Variance
c on a set of data. It is designed with reference to the discussion
c found on pp 207-213 of BASIC STATISTICAL METHODS, FOURTH EDITION,
c by N. M. Downie and R. W. Heath.
```

```
c INPUTS:
```

```
c D   is the array containing all the raw data
c ND  is the row dimension of the D array
c KD  is the column dimension of the D array
c N   is the number of entries considered in each set
c K   is the number of sets of data
```

```
c OUTPUT:
```

```
c F   is the computed F-ratio
c DFB Degrees of freedom between groups
c DFW Degrees of freedom within a group
c XM(J) Sample mean for group J
```

```
c LOCAL VARIABLES
```

```
c I   row index
c J   column index
c NN  Total number of data points: N*K
c DFT Total degrees of freedom
c XT(J) Sum of data points for group J
c SST Sum of squares total
c SSB Sum of squares between groups
c SSW Sum of squares within groups
c MSB Mean square between groups
c MSW Mean square within groups
c XSQT Grand sum of all squared data points
c XSMSQ Square of the grand sum of all data points
```

```
integer n,kd,i,j,nn,dft,dfb,dfw,nd,k
real d(nd,kd),f,xm(20)
double precision xt(20),xtt,ssbn,xsmsg
double precision sst,ssb,ssw,msb,msw,xsqt
```

```
nn = n*k
xsqt = 0.0
xsmsg = 0.0
xtt = 0.0
ssbn = 0.0
ssb = 0.0
```

```
do 50 j=1, k
  xm(j) = 0.0
  xt(j) = 0.0
  do 40 i=1, n
    xt(j) = xt(j) + d(i,j)
    xsqt = xsqt + d(i,j)**2
40  continue
  xm(j) = xt(j)/n
  xsmsg = xsmsg + xt(j)
  xtt = xtt + xt(j)**2
50  continue
```

```
xsmsg = (xsmsg**2)/nn
sst = xsqt - xsmsg
ssbn = xtt/n
ssb = ssbn - xsmsg
ssw = sst - ssb
```

```

dft = nn - 1
dfb = k - 1
dfw = k*(n-1)
msb = ssb/dfb
msw = ssw/dfw
f = msb/msw

```

c This section produces a diagnostic dump to stdout in the event
c that a squirrely f-ratio is produced.

```

      if (f.lt.0.0) then

60      format(/"Negative F-ratio in subroutine ANOVA2"/
a      "Raw data follows:")
62      format (1x,g12.6,$)
64      format (" ")
66      format (a5," = ",g12.6)
68      format (a5," = ",i12)

      write (6,60)

      do 80 i=1, n
        do 75 j=1, k
          write (6,62) d(i,j)
75      continue
        write (6,64)
80      continue

      write (6,64)
      write (6,66) "xtt" , xtt
      write (6,66) "xsmsq", xsmsq
      write (6,66) "sst" , sst
      write (6,66) "ssbn" , ssbn
      write (6,66) "ssb" , ssb
      write (6,66) "ssw" , ssw
      write (6,66) "dft" , dft
      write (6,66) "dfb" , dfb
      write (6,66) "dfw" , dfw
      write (6,66) "msb" , msb
      write (6,66) "msw" , msw
      write (6,66) "f" , f
      end if

      return
      end

```

c -----

```

      subroutine auto(x,n,m,a,errn,rmsl)
c.....compute lpc coefficients b(1),...,b(m) for m <= 40
c      to approximate signal x(1)...x(n), where n<=1024.
c      levinson's formulation of autocorrelation method is used.
c      convention used: signs of the b(k)'s are such that the
c      denominator of the transfer function is of the form
c      1 + (sum from k=1 to p of b(k) * z ** (-k))
c      (normal convention for inverse filtering formulation)
c
c.....errn=normalized minimum error
c.....rmsl=root mean square energy of x(i)'s
c.....n= number of data points in frame. n <= 1024
c.....m= number of coefficients=degree of inverse filter polynomial
c.....m <= 40
c
      integer n,m
      real x(1024),a(40)
      real errn,rmsl
      real f(40),tf(40),r(41)
      real ss,alpha,beta,gamma,c,sum,q
c
c      calculation of m+1 length autocorrelation sequence
      mp1= m+1
      do 11 jj= 1,mp1
        j= jj-1
        nmj=n-j
        ss= 0.
        do 10 i= 1,nmj
          ipj= i+j
10      ss=ss+x(i)*x(ipj)
11      r(jj)= ss
c      levinson's method
      f(1)= 1.
      alpha=r(1)
      beta=r(2)
      a(1)= -r(2)/r(1)
      gamma=a(1)*r(2)
      do 1 k=2,m
        km1= k-1
        c= -beta/alpha
        if(k-2)2,2,3
3      do4 j= 2,km1
        kk=k-j+1
4      tf(j)= f(j)+c*f(kk)
        do 5 j= 2,km1
5      f(j)= tf(j)
2      f(k)= c*f(1)
        alpha=alpha+c*beta
        beta=0.
        do 6 j= 1,k
          kk=k-j+2
6      beta= beta+f(j)*r(kk)
          q=-(r(k+1)+gamma)/alpha
          do 7 j= 1,km1
            kk=k-j+1
7      a(j)= a(j)+q*f(kk)
            a(k)= q*f(1)
            gamma=0.
            do 8 j= 1,k
              kk=k-j+2
8      gamma=gamma+a(j)*r(kk)
1      continue
c      calculate normalized error errn

```

```
sum=0.  
do 77 j= 1,m  
77 sum= sum+a(j)*r(j+1)  
errn= 1.+sum/r(1)  
rmsl=sqrt(r(1)/float(n))  
return  
end
```

c -----

program averageall

c This program takes results from EXP6 and averages them across all 8
 c speakers, putting the results into spkr-a, where spkr-a has the same
 c directory tree structure as the actual speakers. If any data for a given
 c speaker is not available, the program will abort for the incomplete data set
 c and move on to the next one. The user will supply the experiment ident, and
 c then the program will cycle through the six phoneme classes for each of the
 c three conditions, averaging on a point-by-point basis (phoneme vector
 c length) across the 8 speakers.

```

character*80 pathin,pathout,datafile(18)*36,expident*36
real avg(10),p1(10)
integer ispkr,icond,j,k,istr
data datafile /"rnn05","rnn05st","rnn05na","rnn05fr","rnn05li",
a          "rnn05vo","scr05","scr05st","scr05na","scr05fr",
a          "scr05li","scr05vo","scd05","scd05st","scd05na",
a          "scd05fr","scd05li","scd05vo"/

10  format(/"Enter experiment ident: ",%)
12  format(a36)
   write (6,10)
   read (5,12) expident

   do 150 icond=1, 3
     do 100 j=1, 18
       call zeroit(avg,10)
       call recogpath(ichar("a")-48,icond,expident,datafile(j),
a          pathout)
       do 50 ispkr=1, 8
         call recogpath(ispkr,icond,expident,datafile(j),pathin)
         if (tmerit(pathin).eq.0.0) then
           istr = 80
           call endstr(pathin,istr)
           format("Data missing for the path:")
           write (6,20)
           write (6,*) pathin(1:istr)
           goto 100
         end if
         open (unit=1,file=pathin,status="old")
         rewind (1)
         read (1,*) p1
         close (1)

         do 30 k=1, 10
           avg(k) = avg(k) + p1(k)/8.0
           continue
           30
         continue
           50

         open (unit=2,file=pathout,status="unknown")
         istr = 80
         call endstr(pathout,istr)
         write (6,*)pathout(1:istr)
         rewind (2)
           60
         format(10f6.1)
         write (2,60) avg
         close (2)
           100
         continue
           150
         continue
       stop
     end
  
```

c -----

program avganadif

```

c This program cycles through the 8 speakers and calculates the
c differences between loud-normal (21) Lombard-normal (31), and
c Lombard-loud (32) for each phoneme of each speaker. These values
c are printed. It also compiles the averages of each phoneme across
c all 8 speakers of these differences and then prints them at the end.
c The average values in the energy bands are converted to dB, and
c the spectral tilt values are converted to dB/octave. These conversions
c are performed in the routine FINDDIF.
c As an added feature, it compiles the grand mean of each feature mean
c (xbar) across all eight speakers, and stores these values into
c look-alike ANALYSIS.DAT arrays for SPKR-A. This allows other existing
c routines to easily access this averaged data.

c ADIF(IPHONE,I)    the array where the average differences are compiled

      integer i,iphone,xbar,itc,ibc
      integer ispk, itotspkr
      real dat(1200,20,3),dif(54,40),adif(54,40)
      real adat1(1200,20),adat2(1200,20),adat3(1200,20)

      itotspkr = 8
      do 100 ispk=1, itotspkr
        call getdat(ispk,dat)
        call accumdat(dat,itotspkr,adat1,adat2,adat3)
        do 80 iphone=1, 40
          call finddif(dat,iphone,2,1,dif(1,iphone))
          call finddif(dat,iphone,3,1,dif(19,iphone))
          call finddif(dat,iphone,3,2,dif(37,iphone))

c Accumulate values for the overall averages

          do 60 i=1, 54
            adif(i,iphone) = adif(i,iphone) + dif(i,iphone)/8.0
60          continue
80          continue

        call pranadif(ispk,dif)
cccc      call prfeat(ispk,14,dif)

100      continue

      write (6,*) "Average across all speakers"
      call pranadif(ichar("a")-48,adif)
cccc      call prfeat(ispk,14,adif)

      write (6,*) "Storing average energy differences in AVGENG.DAT ..."
      open (unit=1,file="avgeng.dat",status="unknown")
      rewind (1)
      write (1,*)adif
      close (1)

      call putdat(49,1,adat1)
      call putdat(49,2,adat2)
      call putdat(49,3,adat3)

      stop
      end
c -----

```



```
subroutine buildpath(iutt,itok,iphs,path,ispkr,icond,pathname)
```

```
c This routine builds pathnames for accessing the various
c types of data used in this research
```

```
c INPUT
```

```
c IUTT is the number of the utterance being processed.
c ITOK is the occurrence number of a particular phoneme.
c IPHS is the short index identifying a particular phoneme
c PATH is a character string indicating the type of pathname
c ISPKR is the optional speaker number. It must be set to -1 for
c routine to prompt for the speaker.
c ICOND is the optional condition number. It must be set to -1 for
c routine to prompt for the condition.
c requested:
c "wav" - original speechfile
c "lbl" - label file for the original speechfile
c "fmt" - data file containing pitch or formant frequencies
c "pft" - parameter file used as the one being tested
c "pfr" - parameter file used as the reference
c "lpc" - parameter file containing lpc coefficients
c "ana" - parameter file containing analysis data
```

```
c OUTPUT
```

```
c PATHNAME is the complete pathname requested
```

```
c INITED indicates whether or not pathnames have been initialized
c with the root path, speaker number, and condition number.
```

```
logical inited,ispkrsupplied,icondsupplied
integer iutt,itok,iphs,iroot,ipf,isf,ispkr,icond
character*36 pathname,speechfile,labelfile,lpcfile
character*36 partestfile,parreffile,formantfile,analysisfile
character*5 root1
character*8 root2
character*9 root3
character*3 path
character*1 spkr,cond
```

```
data root1 / "data/" /
data root2 / "/bj1/bj/" /
data root3 / "/hogs/bj/" /
```

```
c Preserve the values of all local variables in this routine
c from one call to the next.
```

```
save
```

```
if (.not.inited) then
  inited = .true.
```

```
if (ispkr .eq. -1) then
  ispkrsupplied = .false.
else
  ispkrsupplied = .true.
end if
```

```
if (icond .eq. -1 ) then
  icondsupplied = .false.
else
  icondsupplied = .true.
end if
```

```

call getroot(root1,root2,root3,iroot)
if (.not. ispkrsupplied) call getspkr(ispkr)
if (.not. icondsupplied) call getcond(icond)

```

```

end if

```

```

spkr = char(48+ispkr)
cond = char(48+icond)

```

c Here is where root paths are merged with the speaker and condition numbers..

```

if (iroot.eq.1) then
  partestfile = root1//spkr//cond//"/par/skkmmu.dat"
  parreffile = root1//spkr//"1"//"/par/skkmmu.dat"
  lpfile = root1//spkr//cond//"/par/skkmmu.lpc"
  speechfile = root1//spkr//cond//"/zbg/nnn.zbg"
  labelfile = root1//spkr//cond//"/lbl/nnn.lbl"
  formantfile = root1//spkr//cond//"/fmt/nnn.f0"
  analysisfile = root1//spkr//cond//"/ana/analysis.dat"

```

c Note these values are +9 and +8 of the root string

```

  ipf = 14
  isf = 13
else if (iroot.eq.2) then
  partestfile = root2//spkr//cond//"/par/skkmmu.dat"
  parreffile = root2//spkr//"1"//"/par/skkmmu.dat"
  lpfile = root2//spkr//cond//"/par/skkmmu.lpc"
  speechfile = root2//spkr//cond//"/zbg/nnn.zbg"
  labelfile = root2//spkr//cond//"/lbl/nnn.lbl"
  formantfile = root2//spkr//cond//"/fmt/nnn.f0"
  analysisfile = root2//spkr//cond//"/ana/analysis.dat"
  ipf = 17
  isf = 16

```

```

else if (iroot.eq.3) then
  partestfile = root3//spkr//cond//"/par/skkmmu.dat"
  parreffile = root3//spkr//"1"//"/par/skkmmu.dat"
  lpfile = root3//spkr//cond//"/par/skkmmu.lpc"
  speechfile = root3//spkr//cond//"/zbg/nnn.zbg"
  labelfile = root3//spkr//cond//"/lbl/nnn.lbl"
  formantfile = root3//spkr//cond//"/fmt/nnn.f0"
  analysisfile = root3//spkr//cond//"/ana/analysis.dat"
  ipf = 18
  isf = 17

```

```

else

```

```

45   format("/Invalid selection for root pathname!"/
a     "Program Halted."/)

```

```

  write (6,45)
  stop

```

c This is not standard Fortran 77, although some compilers allow
c it (i.e. transferring back into an if-then-else block).

```

c     goto 20
end if

```

```

if (path.eq."wav") then

```

```

a   speechfile(isf:isf+2) = char(48+int(iutt/100))//
a                               char(48+int(mod(iutt,100)/10))//
a                               char(48+mod(iutt,10))
  pathname = speechfile

```

```

else if (path.eq."lbl") then

```

```

    labelfile(isf:isf+2) = char(48+int(iutt/100))//
a                          char(48+int(mod(iutt,100)/10))//
a                          char(48+mod(iutt,10))
    pathname = labelfile

else if (path.eq."fmt") then
    formantfile(isf:isf+2) = char(48+int(iutt/100))//
a                          char(48+int(mod(iutt,100)/10))//
a                          char(48+mod(iutt,10))
    pathname = formantfile

else if (path.eq."pft") then
    partestfile(ipf:ipf+3) = char(48+int(itok/10))//
a                          char(48+mod(itok,10))//
a                          char(48+int(iphs/10))//
a                          char(48+mod(iphs,10))
    pathname = partestfile

else if (path.eq."pfr") then
    parreffile(ipf:ipf+3) = char(48+int(itok/10))//
a                          char(48+mod(itok,10))//
a                          char(48+int(iphs/10))//
a                          char(48+mod(iphs,10))
    pathname = parreffile

else if (path.eq."lpc") then
    lpcfile(ipf:ipf+3) = char(48+int(itok/10))//
a                          char(48+mod(itok,10))//
a                          char(48+int(iphs/10))//
a                          char(48+mod(iphs,10))
    pathname = lpcfile

else if (path.eq."ana") then
    pathname = analysisfile
end if

return
end
c -----

```

program callanova3

c This is a revision of CALLANOVA2. It incorporates calculation of
 c the threshold of the F distribution using IMSL. The output is
 c also formatted more concisely, using the subroutine ANOVA3 which
 c does no printing but instead passes more data back to this program.

c This program looks for significant differences from the analyses
 c produced by extracting various features from normal, loud, and
 c Lombard speech.

c IFTOT total number of features being examined

```
character*36 file1,file2,file3
character*8  file4
character*24 date
integer iphone,ifeat,n,ibg1,ibg2,ibg3,i,dfb,dfw,ier
integer inoc,ibg,xbar,var,ssum,s2sum,iftot,ispkr,pfa(40,20)
real d1(1200,20),d2(1200,20),d3(1200,20)
real dd(100,3),f,mean(20),conf,p
```

```
iftot = 18
conf = 0.99
```

```
10  call fdate(date)
    format(/"Program CALLANOVA3... ",a24//
    a  "Enter desired confidence: ",%)
    write (6,10)date
    read (5,*)conf
```

c Load in the files that contain the feature values for every
 c phoneme, every condition, for a given speaker.
 c ISPKR is set to -1 so BUILDPATHNP will prompt for one.

```
ispkr = -1
call buildpath(0,0,0,"ana",ispkr,1,file1)
write (6,*)file1
call buildpath(0,0,0,"ana",ispkr,2,file2)
write (6,*)file2
call buildpath(0,0,0,"ana",ispkr,3,file3)
write (6,*)file3
open (unit=1,file=file1,status="old",form="unformatted")
rewind (1)
read (1) d1
close (1)
open (unit=2,file=file2,status="old",form="unformatted")
rewind (2)
read (2) d2
close (2)
open (unit=3,file=file3,status="old",form="unformatted")
rewind (3)
read (3) d3
close (3)
```

```
do 100 iphone=1, 40
```

```
  do 90 ifeat=1, iftot
```

```
    inoc = 7*iphone - 6
    ibg  = 7*iphone - 5
    xbar = 7*iphone - 4
    var  = 7*iphone - 3
    ssum = 7*iphone - 2
```

```

s2sum = 7*iphone - 1
ibg1 = d1(ibg,ifeat)
ibg2 = d2(ibg,ifeat)
ibg3 = d3(ibg,ifeat)
n = min1(d1(inoc,ifeat),d2(inoc,ifeat),d3(inoc,ifeat))

c Now load the interfacing data array

do 50 i=1, n

    dd(i,1) = d1(ibg1+i-1,ifeat)
    dd(i,2) = d2(ibg2+i-1,ifeat)
    dd(i,3) = d3(ibg3+i-1,ifeat)
50    continue

c This is for diagnostic printing
51    format (i2,1x,g12.6,2i3,3(1x,g12.6))
    call anova2(dd,100,3,n,3,f,dfb,dfw,mean)
c    write (6,51) n,f,dfb,dfw,mean(1),mean(2),mean(3)
    call mdfd(f,dfb,dfw,p,ier)

    if (p.gt.conf) then
        if (mean(2).gt.mean(1).and.mean(3).gt.mean(1)) then
c            sym = "."
            pfa(iphone,ifeat) = 1
        else if (mean(2).lt.mean(1).and.mean(3).lt.mean(1)) then
c            sym = "v"
            pfa(iphone,ifeat) = -1
        else if (mean(2).lt.mean(1).and.mean(3).gt.mean(1)) then
c            sym = "3"
            pfa(iphone,ifeat) = 3
        else if (mean(2).gt.mean(1).and.mean(3).lt.mean(1)) then
c            sym = "2"
            pfa(iphone,ifeat) = 2
        else
c            sym = "?"
            pfa(iphone,ifeat) = 9
        end if
    else
c        sym = " "
        pfa(iphone,ifeat) = 0
    end if

90    continue

100    continue

c Now save the PFA array
    file4 = "pfa"//char(48 + ispkr)//".dat"
    open (unit=1,file=file4,status="unknown",form="unformatted")
    rewind (1)
    write (1)pfa
    close (1)

    call printpfa(pfa,iftot)

    stop
    end
c -----

```

```
subroutine cepcoef(a,nlpc,ncep,c)
```

```
c This routine computes cepstral coefficients directly from LPC
c coefficients, using the recursive method discussed in Gray &
c Markel, "Distance Measures for Speech Processing", Vol ASSP-24,
c No. 5, Oct 76, p390. Also mentioned in Schroeder, Vol ASSP-29,
c No. 2, Apr 81, p298.
```

```
c INPUTS
```

```
c A() Array of LPC coefficients
c NLPC    Number of LPC coefficients
c NCEP    Number of cepstral coefficients desired
```

```
c OUTPUT
```

```
c C() Array of cepstral coefficients
```

```
integer nlpc,ncep,n,k
real a(40),c(40)
```

```
do 100 n=1, ncep
```

```
  if (n.le.nlpc) then
```

```
    c(n) = -a(n)
```

```
    do 50 k=1, n-1
```

```
      c(n) = c(n) - (k*c(k)*a(n-k))/n
```

```
50    continue
```

```
  else if (n.gt.nlpc) then
```

```
    c(n) = 0.0
```

```
    do 70 k=1, nlpc
```

```
      c(n) = c(n) - ((n-k)*c(n-k)*a(k))/n
```

```
70    continue
```

```
  end if
```

```
100  continue
```

```
  return
```

```
end
```

```
c -----
```

```
subroutine ceptemplate(ispeech,from,to,cep,lpc)
```

```
c This routine builds a template of 50 frames of the phoneme passed
c to it. Each frame is a 40-point vector representing the
c cepstral coefficients derived from the LPC coefficients.
c The window size is set to 16 milliseconds (256 points) and
c the stepsize is determined by the duration of the phoneme.
```

```
c INPUTS
```

```
c ISPEECH is the array containing digitized speech
c FROM is the starting position of the phoneme in seconds
c TO is the ending position of the phoneme in seconds
```

```
c OUTPUT
```

```
c TS is the array where all the 50-frame template is stored
c LPC is the array where the lpc coefficients are stored
```

```
integer*2 ispeech(160000)
real from,to,cep(40,50),lpc(40,50)
integer nlpc,i,j,k,ifrom,iframes,ifs,ncep
real sframe(1024),hs(1024),b(40),ss,errn,rmsl
```

```
c Set the frame size to 256 pts and LPC coefficients to 24
ifs = 256
nlpc = 24
ncep = 40
iframes = 50
```

```
c To find the starting and ending points in the phoneme:
```

```
c ifrom = int(from*16000.)
c ito = int(to*16000.)
c itpts = ito - ifrom
```

```
c Calculate the stepsize based on the duration of the phoneme.
```

```
ss = (to-from-0.016)/(iframes-1)
```

```
do 100 i=1,iframes
ifrom = (from + ss*(i-1)) * 16000
```

```
do 75 j=1,ifs
sframe(j) = ispeech(ifrom+j)
75 continue
```

```
call hamm(sframe,hs,ifs)
call auto(hs,ifs,nlpc,b,errn,rmsl)
call cepcoef(b,nlpc,ncep,cep(1,i))
```

```
c Note that we exploit the column-major format of 2-D arrays in
c fortran by sending CEP COEF column I which is really frame I.
c CEP COEF expects to be passed a 1-D array of dimension 40.
```

```
do 80 k=1, nlpc
lpc(k,i) = b(k)
80 continue
```

```
100 continue
```

```
return
end
```

```
c -----
```

program checkcover

- c Built from parts of SELUTT and COVERUP, this program takes
- c a list of utterances and shows the theoretical phoneme coverage
- c based on the transcriptions of the individual vocabulary words

```
character*36 listname
character*70 g(-2:40)
integer i,j,iutt,ii,jj,ipt,iptr(126),icover(40),ishort(40)
integer ipud(15000),ipun(15000),ips(539)
```

```
common /ip/ipud,ipun
```

```
include "ml"
```

- c The ISHORT array preserves the order in IORDER, but only contains
- c the 40 phonemes selected in Appendix D.

```
data (ishort(i),i=1,i0)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/
```

- c The IPTR array is the complement of the ISHORT array. Given the
- c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
- c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
- c means that phoneme K is not in the set of 40 phonemes.

```
data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i=81,90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i=91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/
```

```
3      format(i3)
10     format(/"Program CHECKCOVER..." /
      a      "Enter filename containing utterance list: ",%)
      write (6,10)
      read (5,*)listname
```

- c First build the linked lists for each of the 539 utterances

```
call uttphon(ipud,ipun,ips)
```

- c Next, initialize the grid array with blanks

```
do 50 i=-2, 40
  do 40 j=1, 70
    g(i)(j:j) = " "
40    continue
50    continue

open (unit=1,file=listname,status="old")
rewind (1)
```

- c *** Start of the main loop

```
jj = 0

60    read(1,3,end=100)iutt
      ii = iutt
```



```

      jj = jj + 1
      g(-2)(jj:jj) = char(48+int(iutt/100))
      g(-1)(jj:jj) = char(48+int(mod(iutt,100)/10))
      g(0)(jj:jj) = char(48+mod(iutt,10))

70    if (iptr(ipud(ii)).ne.0) then
        icover(iptr(ipud(ii))) = icover(iptr(ipud(ii))) + 1
        g(iptr(ipud(ii)))(jj:jj) = "x"
      end if
      if (ipun(ii).ne.0) then
        ii = ipun(ii)
        goto 70
      end if

      goto 60

100   continue

c *** End of the main loop

120   format(/"Utts -->",t10,a70/t10,a70/"Phonemes",t10,a70)
      write (6,120)(g(i),i=-2,0)

140   format(a3,2x,i3,t10,a70)

      do 160 i=1, 40
        write (6,140)ml(ishort(i)),icover(i),g(i)
        ipt = ipt + icover(i)
160   continue

180   format(/"Total phonemes:",t20,i3/
a      "Total utterances:",t20,i3)

      write (6,180)ipt,jj

      stop
      end
c -----

```

program comparemer

c This program will calculate a figure of merit on a particular
 c recognition experiment as well as provide a means of comparing one
 c experiment to another.

c In the development stage, it will be a bare-bones caller to test
 c the supporting subroutines.

```

      integer icond,ispkr,i
      real tmerit,tot(6,4)
      character*36 expident1,datafile1,expident2,datafile2
      character*80 path1,path2

10    format ("Enter ",a8," experiment ident (e.g. lpc24r1): ",$)
12    format ("Enter ",a8," data filename (e.g. rnn05): ",$)
15    format (a36)
      write (6,10) "baseline"
      read (5,15)expident1
      write (6,12) "baseline"
      read (5,15)datafile1
      write (6,10) "tested"
      read (5,15)expident2
      write (6,12) "tested"
      read (5,15)datafile2
20    format("/"Session",t10,"Baseline",t22,"Test",t30,"Difference")
      write (6,20)

      do 100 ispkr=1, 8
        do 80 icond=1,3
          call recogpath(ispkr,icond,expident1,datafile1,path1)
          tot(icond,1) = tmerit(path1)
          call recogpath(ispkr,icond,expident2,datafile2,path2)
          tot(icond,2) = tmerit(path2)
80      continue

          call printtot(ispkr,tot)
100     continue
      stop
      end

```

c -----

program comparemer2

- c This program will calculate a figure of merit on all complete
 c recognition experiments, broken down by phoneme category.

```

integer icond,ispkr,i
real tmerit,tpc(3,12)
character*36 expident,datafile(12),cond(3)*7
character*80 path

data (datafile(i),i=1,4)/"rnn05st","scd05st","rnn05na","scd05na"/
data (datafile(i),i=5,8)/"rnn05fr","scd05fr","rnn05li","scd05li"/
data (datafile(i),i=9,12)/"rnn05vo","scd05vo","rnn05","scd05"/
data (cond(i),i=1,3)/"Normal ","Loud  ","Lombard"/

10  format ("Enter experiment ident (e.g. lpc24r1): ",%)
15  format (a36)
   write (6,10)
   read (5,15)expident

   ispk=49
   do 80 icond=1,3
     do 60 i=1, 12
       call recogpath(ispk,icond,expident,datafile(i),path)
       tpc(icond,i) = tmerit(path)
60    continue
80    continue

85  format (a7,2x,12f6.1)
   do 90 icond=1, 3
     write (6,85) cond(icond), (tpc(icond,i),i=1,12)
90    continue

   stop
   end
c -----

```

program comparepfa

c Designed to consolidate a number of Phoneme-Feature arrays
c to determine similarities across speakers

c PFA Phoneme-Feature array where the following values
c have specific meaning:
c -1 = both loud and Lombard were less than normal
c 0 = no significant difference between normal,
c loud, and Lombard
c 1 = both loud and Lombard were more than normal
c 2 = loud > normal and Lombard < normal
c 3 = loud < normal and Lombard > normal
c 9 = undetermined state or error
c PFAALL same as PFA except used to store consolidated results
c IFTOT total number of features being examined

integer pfa(40,20),pfaall(40,20),iftot,iphone,ifeat
integer pfab(40,20),pfac(40,20),pfaball(40,20),pfacall(40,20)
character*24 date
character*1 spkr,ans

```

iftot = 18
call fdate(date)
10  format("/Program COMPAREPFA... ",a24)
    write (6,10)date
20  format(a1)
30  format("/Enter Speaker number: ",%)
    write (6,30)
    read (5,20)spkr

    call loadpfa(pfaall,spkr)
    call pfafilter2(pfaall,pfaball)
    call pfafilter3(pfaall,pfacall)
35  write (6,30)
    read (5,20)spkr
    call loadpfa(pfa,spkr)
    call pfafilter2(pfa,pfab)
    call pfafilter3(pfa,pfac)
    call andpfa(pfa,pfaall)
    call andpfa(pfab,pfaball)
    call andpfa(pfac,pfacall)
    write (6,*)"Comparing Normal and Loud only"
    call printpfa(pfaball,iftot)
    write (6,*)"Comparing Normal and Lombard only"
    call printpfa(pfacall,iftot)
    write (6,*)"Comparing Normal and Abnormal (Loud and Lombard)"
    call printpfa(pfaall,iftot)

40  format("/Include another speaker? (y/n): ",%)
    write (6,40)
    read (5,20)ans
    if (ans.eq."y".or.ans.eq."Y") goto 35
    open (unit=1,file="pfaz.dat",status="unknown",form="unformatted")
    rewind (1)
    write (1) pfaall
    close (1)

    stop
    end

```

c -----

program counttok

c Designed to count the number of occurrences of all the
 c phonemes in a list of files, and then calculate statistics
 c on the durations. The statistics are listed to the standard
 c output for the entire SPIRE phoneme set, and an option is
 c given to save statistics on the 40-phoneme set in ascii files
 c suitable for plotting with QPLOT. The old program LISTTOK has
 c been incorporated as a subroutine, as well as an additional feature
 c of showing the utterances in which a given phoneme occurs (phoneme
 c membership).

c UMEM() is the array containing utterance numbers that forms
 c the data portion of the linked-list structure for
 c printing the utterance memberships of a given phoneme.
 c UNEXT() is the next-array used with UMEM() in the linked-list
 c ULIST(K) is the pointer array that points to the tail of each
 c utterance list in the linked-list structure, where
 c K is the short-index of the phoneme of interest.
 c NEXT points to the next unused storage location while
 c the linked lists are being built, and iterates through
 c a phoneme linked list as the results are being printed.

```
character*36 filename,listname
character*1 ans
integer ilabel(2000),itoken(126),i,k,itot,ulist(70)
integer iorder(70),ishort(40),is2(40),umem(2000),unext(2000)
integer idim,j,ises,iutt,istring,iptr(126),next,ispkr,icond
real xplot(40)
real frompos(2000),topos(2000)
real dur,spread,sprmax,omin,omax
real dmean(126),dsq(126),dmin(126),dmax(126),dsd(126)
include "ml"
```

c This array orders all the phoneme symbols to coincide with the
 c order in SPIRE. The only significance is there is a loose
 c grouping according to phonetic events.

```
data (iorder(i),i=1,10)/112,116,107,13,14,15,98,100,103,10/  

data (iorder(i),i=11,20)/11,12,70,63,109,110,71,77,78,7/  

data (iorder(i),i=21,30)/6,115,122,67,84,102,83,90,74,68/  

data (iorder(i),i=31,40)/118,108,114,121,8,104,76,119,16,72/  

data (iorder(i),i=41,50)/69,99,97,117,82,89,101,87,120,124/  

data (iorder(i),i=51,60)/73,64,94,85,79,105,111,88,9,58/  

data (iorder(i),i=61,68)/35,42,36,43,45,39,34,126/
```

c The ISHORT array preserves the order in IORDER, but only contains
 c the 40 phonemes selected in Appendix D.

```
data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/  

data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/  

data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/  

data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/
```

c The IS2 array contains the same 40 phonemes as above, but shuffles
 c phonemes within acoustic classes according to duration averages.

```
data (is2(i),i=1,10)/70,98,100,103,112,116,107,109,110,71/  

data (is2(i),i=11,20)/115,83,74,67,122,84,118,102,104,108/  

data (is2(i),i=21,30)/114,121,76,119,120,73,88,82,94,111/  

data (is2(i),i=31,40)/105,99,69,97,117,84,101,79,89,87/
```

c The IPTR array is the complement of the ISHORT array. Given the

- c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
 c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
 c means that phoneme K is not in the set of 40 phonemes.

```

data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i=81,90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i=91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/

data idim/2000/
data dmin/126*1000000.0/
data umem,unext /2000*0,2000*0/

```

- c This initialization is for the linked list data structure.

```

do 2 i=1, 70
  ulist (i) = i
2   continue

```

- c Setting these variables to -1 causes routine BUILDPATH to
 c prompt for the values.

```

ispkr = -1
icond = -1

3   format(/"Program COUNTTOK..."/)
    write (6,3)

4   format(/"Enter filename containing list of utterances: ",%)
    write (6,4)
36  format (a36)
    read (5,36)listname
    istring = 36
    call endstr(listname,istring)
    open (unit=2,file=listname,status="old")
    rewind (2)

5   format(i3)
    write (6,476)
    next = 70

```

- c **** Beginning of the loop that reads the label files for each
 c utterance in the listfile.

```

6   read (2,5,end=10)iutt

    call buildpath(iutt,0,0,"lbl",ispkr,icond,filename)
7   call readlabels(filename,ilabel,frompos,topos,itot,idim)
    call listtok (iutt,itot,ilabel)

```

- c Now count all the tokens of each phoneme, and compile durational
 c statistics.

```

do 100 j=1,itot
  itoken(ilabel(j)) = itoken(ilabel(j)) + 1
  dur = topos(j) - frompos(j)
c   write (6,*) j,topos(j),frompos(j),dur
  if (dur.lt.dmin(ilabel(j))) dmin(ilabel(j)) = dur
  if (dur.gt.dmax(ilabel(j))) dmax(ilabel(j)) = dur
  dmean(ilabel(j)) = dmean(ilabel(j)) + dur

```

```

      dsq(ilabel(j)) = dsq(ilabel(j)) + dur**2
c      Here is where the linked lists of utterances for a
c      given phoneme are built. Format 99 is for diagnostic
c      purposes only.
99     format(i3,i4,i4,a4,i5,i5,i5,i5)
      if (iptr(ilabel(j)).ne.0) then
          umem(ulist(iptr(ilabel(j)))) = iutt
          unext(ulist(iptr(ilabel(j)))) = next
          ulist(iptr(ilabel(j))) = next
          next = next + 1
c      write (6,99)iutt,j,ilabel(j),ml(ilabel(j)),iptr(ilabel(j)),
c      a      ulist(iptr(ilabel(j))),umem(ulist(iptr(ilabel(j))))),
c      a      unext(ulist(iptr(ilabel(j))))),next
          end if
100    continue

      goto 6

10    continue

c **** End of loop for reading label files

c Temporary patch to allow the reading of a single arbitrary
c file
c      if (imax.eq.0) then
c6     format(/"Enter 7-character filename for single file: ",%)
c      write (6,6)
c      read (5,*)flen
c      goto 7
c      end if

c Section that calculates duration mean and std dev and
c prints out the results.
c For future reference, it would be nice to have a breakdown of
c phonemes in each utterance.

15     format(/t17,"Duration Statistics.....",
a      /"Token",t7,"N",t17,"Min",t29,"Max",t41,"Spread",
a      t53,"Mean",t65,"Std Dev")
16     format(a3,t5,i3,t10,5f12.3)

      write (6,15)
      itot = 0
      sprmax = 0.0
      do 400 k=1,68
          if (itoken(iorder(k)).gt.1) then
              dsd(iorder(k)) =
a              itoken(iorder(k))*dsq(iorder(k))-dmean(iorder(k))**2
              dsd(iorder(k)) =
a              dsd(iorder(k))/(itoken(iorder(k))*(itoken(iorder(k))-1))
              dsd(iorder(k)) = sqrt(dsd(iorder(k)))
              dmean(iorder(k)) = dmean(iorder(k))/itoken(iorder(k))
              end if
              if (dmin(iorder(k)).eq.1000000.0) dmin(iorder(k)) = 0.0
              spread = dmax(iorder(k)) - dmin(iorder(k))
              sprmax = max(sprmax,spread)
              write(6,16)ml(iorder(k)),itoken(iorder(k)),dmin(iorder(k)),
a              dmax(iorder(k)),spread,dmean(iorder(k)),dsd(iorder(k))
              itot = itot + itoken(iorder(k))
400    continue

17     format(/"Total tokens: ",i4," Max Duration Spread: ",f6.3)
      write (6,17)itot,sprmax

```

```

18  format (// "For the 40-phoneme set:")
    write (6,18)
    write (6,15)
    itot = 0
    omax = 0
    omin = 10.
    sprmax = 0.0

    do 450 k=1,40
        spread = dmax(ishort(k)) - dmin(ishort(k))
        sprmax = max(sprmax,spread)
        omax = max(omax,dmax(ishort(k)))
        omin = min(omin,dmin(ishort(k)))
        write(6,16)ml(ishort(k)),itoken(ishort(k)),dmin(ishort(k)),
a      dmax(ishort(k)),spread,dmean(ishort(k)),dsd(ishort(k))
        itot = itot + itoken(ishort(k))
450  continue
    write (6,17)itot,sprmax
460  format (// "Overall min: ",f6.3," Overall max: ",f6.3)
    write (6,460)omin,omax

470  format (// "Phoneme Membership:")
472  format (a3," ",%)
474  format (i4,%)
476  format (" ")
c    write (6,470)

    do 500 k=1, 40
        write (6,472)ml(ishort(k))
        i = 1
        next = k
480  if (umem(next).ne.0) then
            write (6,474) umem(next)
            i = i + 1
            if (i.ge.18) then
                i = 0
                write (6,476)
            end if
            next = unext(next)
            goto 480
        end if
        write (6,476)
500  continue

520  format("Save min, max, and mean for plotting? (y/n): ",%)
    write (6,520)
522  format (a1)
    read (5,522)ans
    if (ans.eq."y".or.ans.eq."Y") then
c    if (ians.ne.0) then
        do 525 i=1,40
            xplot(i) = dmin(is2(i))
525  continue
        ises = 99
        filename = "min"//char(48+int(ises/10))//char(48+mod(ises,10))
        call savenoprmt(xplot,40,40,filename)

        do 530 i=1,40
            xplot(i) = dmax(is2(i))
530  continue
        filename = "max"//char(48+int(ises/10))//char(48+mod(ises,10))
        call savenoprmt(xplot,40,40,filename)

```



```
do 535 i=1,40
  xplot(i) = dmean(is2(i))
  continue
535 filename = "mean"//char(48+int(ises/10))//char(48+mod(ises,10))
  call savenoprmt(xplot,40,40,filename)
  end if

stop
end
c -----
```

program coverup

c This program is derived from SELUTT. It will select additional
 c utterances to provide coverage for a set of still-uncovered
 c phonemes. It uses the same algorithm as SELUTT, but is designed
 c to be much more flexible by allowing data and search sets to be
 c entered interactively.

```

      integer i,j,k,mincov,itot
      integer icov(126),iflg(539),ipset(40)
      integer ipud(15000),ipun(15000),ips(539)
      integer iptmp(40),iptmax,ipmax,iutt
      character*24 listfile
      logical uncovered,found,altered
      common /ip/ipud,ipun
      include "ml"

10    format (/"Program COVERUP..."/
      a    "Enter the minimum coverage desired: ",%)
      write (6,10)
      read (5,*)mincov

15    format (/"Enter the total number of phones to cover up: ",%)
      write (6,15)
      read (5,*)ipmax

20    format (/"Enter the ASCII code for each phone as prompted...")
      write (6,20)

      do 30 i=1,ipmax
25      format("phone ",i2," : ",%)
      write (6,25)i
      read (5,*)ipset(i)
30      continue

35    format(/"Now enter filename containing list of used utts: ",%)
      write (6,35)
      read (5,*)listfile
      call endstr(listfile,20)
      open (unit=2,file=listfile,status="old")

40    format(i3)

c **** Beginning of the loop that reads the label files for each
c    utterance in the listfile.
c Note that each utterance from the listfile will be marked with
c a "2". This is to distinguish them from new utterances that will
c be selected by this program and marked with a "1". In so doing, the
c replacement section that begins with label 495 will only consider
c the newly selected utterances. This scheme allows subroutine
c COVMAXDEL to work correctly because it will test for "0" in IFLG()
c to determine whether or not an utterance is available.

42    read (2,40,end=50)iutt
      iflg(iutt) = 2
      goto 42

50    continue

c **** End of loop for reading label files

c First build the phoneme linked lists for each of the 539
c utterances.
```

```

55    format(/"Now standby as linked lists are built...")
      write (6,55)
      call uttphon(ipud,ipun,ips)

c Search iteratively for utterances having the maximum
c delta coverage and the minimum total phones.
c Use the scarce phone set first then the complete set

90    format(/"Selected Utt  Delta  Utt Size")
      write (6,90)

      call covmaxdel(icov,mincov,ipset,ipmax,iflg,ips)

c This is the final section that iterates through the
c chosen list in order to try and find replacement utts.

495    altered = .false.
      do 500 i=1,539
        if (iflg(i).eq.1) then
          call uncovtest(i,icov,mincov,ipset,ipmax,uncovered,
a          iptmp,iptmax)
          write (6,450)i
          if (uncovered) then
            write(6,460)(ml(iptmp(k)),k=1,iptmax)
            call deaccum(i,icov)
            call recover(i,icov,mincov,iptmp,iptmax,ips,
a            iflg,found,j)
            if (found) then
              altered = .true.
              write(6,470)i,j,ips(i)-ips(j)
              iflg(i) = 0
              iflg(j) = 1
              call accumphon(j,icov)
            else
              call accumphon(i,icov)
              end if
            else
              altered = .true.
              call deaccum(i,icov)
              iflg(i) = 0
              write(6,455)
              end if
            end if
          500    continue
            if (altered) goto 495

c Print out the final results

260    format(/"List of selected utterances"/)
270    format(i4,$)
275    format(" ")
305    format(/"Total utterances selected: ",i3)
310    format(/"Phoneme Coverage")
320    format(a3,t11,i3)
405    format(/"Total number of phones: ",i4/)
450    format("Eliminating ",i3," uncovers ",$)
455    format("nothing!!!")
460    format(10(a4))
470    format("Utt ",i3," replacable by Utt ",i3," saving ",i3)

      itot = 0
      write (6,310)
      do 400 j=1,ipmax

```

```

        itot = itot + icov(ipset(j))
        write (6,320)ml(ipset(j)),icov(ipset(j))
400      continue
        write (6,405)itot

        itot = 0
        write (6,260)
        do 300, k=1,539
            if (iflg(k).eq.1) then
                itot = itot + 1
                write (6,270)k
                if (mod(itot,20).eq.0) write (6,275)
            end if
300      continue
        write (6,305)itot

        stop
        end
c -----

```

```
subroutine covmaxdel(icov,mincov,ipset,ipmax,iflg,ips)
```

```
c This routine searches for utterances that will cover phones
c that are listed in IPSET. It selects utterances by the criterion
c of maximizing the delta increase and then selecting the shortest
c utterance giving this delta increase.
```

```
c INPUTS
```

```
c ICOV is the array of accumulated coverages for each phoneme.
c MINCOV is the lower bound on the number of occurrences of
c each phoneme.
c IPSET is the array that lists the phonemes of interest.
c IPMAX is the number of phonemes in IPSET.
c IFLG is the array showing which utterances are unselected (0)
c and selected (1)
c IPS is the array containing the total number of phonemes in
c each utterance.
```

```
c OUTPUTS
```

```
c ICOV is altered as utterances are selected.
c IFLG is updated as utterances are selected.
```

```
integer i,j,mincov,maxdel,maxdelp,ipmax,ysize
integer icov(126),iflg(539),ipset(40)
integer ips(539),idelta(539)
```

```
7 format(t6,i3,t15,i3,t24,i3)
```

```
c Now search iteratively for utterances having the maximum
c contribution. MAXDEL stores the maximum delta increase found
c so far, and MAXDELP is the pointer to the utterance achieving
c the maximum delta increase.
```

```
c ----- Beginning of the loop
```

```
10 maxdel = 0
```

```
do 100 i=1,539
  if (iflg(i).eq.0) then
    call delcov(i,icov,mincov,ipset,ipmax,idelta(i))
    if (maxdel.lt.idelta(i)) then
      maxdel = idelta(i)
      maxdelp = i
      ysize = ips(i)
    end if
  end if
```

```
100 continue
```

```
c Now with the maximum delta coverage, search for the smallest
c utterance that will give this coverage.
```

```
do 120 i=1,539
  if (iflg(i).eq.0.and.idelta(i).eq.maxdel.and.ips(i).lt.ysize)
    a then
      ysize = ips(i)
      maxdelp = i
    end if
```

```
120 continue
```

```
c Flag the utterance that provided the greatest contribution.
c Then update the total coverage so far.
```

```

ifg(maxdelp) = 1
call accumphon(maxdelp,icov)
write (6,7)maxdelp,delta(maxdelp),ips(maxdelp)

```

c Next check to see if any phones are uncovered. If so, loop back
c to select another utterance.

```

      do 200 j=1,ipmax
        if(icov(ipset(j)).lt.mincov) then
          j = 40
          goto 10
        end if
200    continue

```

c ----- End of the loop

```

      return
    end

```

c -----

```
subroutine covminex(icov,mincov,ipset,ipmax,iflg,ips)
```

```
c This routine searches for utterances that will cover phones
c that are listed in IPSET. It selects utterances by the criterion
c of minimizing the excess number of phones. This is just the
c difference between the total number of phones, IPS(i), and
c the delta coverage, IDELTA(i) for utterance i.
```

```
c INPUTS
```

```
c ICOV is the array of accumulated coverages for each phoneme.
c MINCOV is the lower bound on the number of occurrences of
c each phoneme.
c IPSET is the array that lists the phonemes of interest.
c IPMAX is the number of phonemes in IPSET.
c IFLG is the array showing which utterances are unselected (0)
c and selected (1)
c IPS is the array containing the total number of phonemes in
c each utterance.
```

```
c OUTPUTS
```

```
c ICOV is altered as utterances are selected.
c IMAX is incremented as utterances are selected.
c IFLG is updated as utterances are selected.
```

```
integer i,j,mincov,minex,minexp,ipmax
integer icov(126),iflg(539),ipset(40)
integer ips(539),idelta(539)
```

```
7 format(t6,i3,t15,i3,t24,i3)
```

```
c MINEX is the minimum excess phones
c found, and MINEXP is the pointer to the utterance having
c the minimum excess phones to contribute.
```

```
c ----- Beginning of the loop
```

```
10 minex = 1e6
```

```
do 100 i=1,539
  if (iflg(i).eq.0) then
    call delcov(i,icov,mincov,ipset,ipmax,idelta(i))
    if ((ips(i)-idelta(i)).lt.minex.and.idelta(i).gt.0)
      a then
        minex = ips(i) - idelta(i)
        minexp = i
      end if
    end if
```

```
100 continue
```

```
c Flag the utterance that provided the greatest contribution.
c Then update the total coverage so far. IMAX is the number
c of utterances selected.
```

```
iflg(minexp) = 1
call accumphon(minexp,icov)
write (6,7)minexp,idelta(minexp),ips(minexp)
```

```
c Next check to see if any phones are uncovered.
c If so, loop back to select another utterance.
```

```
do 200 j=1,ipmax
```

```
        if(icov(ipset(j)).lt.mincov) then
            j = 40
            goto 10
        end if
200    continue

    return
end
c -----
```



```
subroutine deaccum(i,icover)
```

```
c This routine subtracts the phone counts from the array ICOVER
c for utterance i. The phone lists for all the utterances
c are passed through the common block IP.
```

```
integer i,icover(126),ipud(15000),ipun(15000),j
```

```
common /ip/ipud,ipun
```

```
c J is the internal pointer for this routine.
c ICOVER is indexed by phoneme; i.e ICOVER(j) represents
c the number of occurrences of phoneme j.
```

```
j = i
```

```
10  icover(ipud(j)) = icover(ipud(j)) - 1
    if (ipun(j).ne.0) then
      j = ipun(j)
      goto 10
    end if
```

```
return
end
```

```
c -----
```

```
subroutine delcov(i,icov,mincov,ipset,ipmax,idelta)
```

```
c This routine calculates the delta coverage to be gained
c by adding utterance i to the list of selected utterances.
```

```
c INPUTS
```

```
c I is the utterance index
```

```
c ICOV is the array of accumulated coverages of each phoneme.
```

```
c MINCOV is the lower bound on the number of occurrences of
c each phoneme.
```

```
c IPSET is the array that lists the phonemes of interest.
```

```
c IPMAX is the number of phonemes in IPSET.
```

```
c OUTPUT
```

```
c IDELTA is the increase in coverage to be gained if utterance
c i were to be added to the list of selected utterances.
```

```
integer i,icov(126),mincov,ipset(40),idelta
integer j,itcov(126),ipmax,k
```

```
c First work through the phoneme list for this utterance and
c accumulate the coverages in the temporary buffer ITCOV(j).
```

```
do 50 j=1,126
  itcov(j) = 0
50 continue
```

```
call accumphon(i,itcov)
```

```
c Now calculate the total increase. This algorithm gives no
c credit for coverage of any phone over the minimum cover
c requirement.
```

```
idelta = 0
do 100 j=1,ipmax
  k = min(max((mincov-icov(ipset(j))),0),itcov(ipset(j)))
  idelta = idelta + k
100 continue
```

```
return
end
```

```
c -----
```

program testdist

c Test for routines DIST and FIND from Gray and Markel

```

dimension r(5),rp(5),dm(9)
data r/1.,8.,388.,07784003,-.0754063/
data rp/2.,1.6,.776,.15568006,-.0098079/
m=4
l=16
call dist(m,l,r,rp,dm)
write (6,50) dm
50 format(2f10.6,/,4f10.6,/,3f10.6)
call exit
end

```

c Here is the output that it produced:

```

c Script started on Tue Mar 1 17:20:59 1988
c ei51% a.out
c -0.800000 0.700000 -0.600002 0.900009
c -0.800000 0.700001 -0.600008 0.300027
c 2.894516 1.395550
c 6.066723 8.542278 11.334484 5.744059
c 4.635481 8.232235 10.853126
c ei52%

```

subroutine dist(m,l,r,rp,dm)

c This comes directly from Gray and Markel, Vol ASSP-24, No 5,
c Oct 76, p389.

c Calculates the various distance measures discussed in the above
c paper.

```

dimension r(1),rp(1),dm(1)
dimension c(60),cp(60),ra(21),rap(21)
dimension a(21),ap(21),rc(21),rcp(21)
data dbfac/4.342944819/
fn(z) = dbfac*log(1.0+z+sqrt(z*(2.0+z)))
call find(m,l,r,c,ra,alp,a,rc)
call find(m,l,rp,cp,rap,alpp,ap,rcp)
mp = m+1
del = r(1)*rap(1)
delp = rp(1)*ra(1)
do 90 j=2, mp
del = del + 2.0*r(j)*rap(j)
90 delp = delp + 2.0*rp(j)*ra(j)
dm(1) = del/alp
dm(2) = delp/alpp
q = (dm(1) + dm(2))/2.0 - 1.0
dm(3) = fn(q)
q1 = alpp*r(1)/(alp*rp(1))
q = (dm(1)/q1 + dm(2)*q1)*0.5 - 1.0
dm(4) = fn(q)
q2 = alpp/alp
q = (dm(1)/q2 + dm(2)*q2)*0.5 - 1.0
dm(5) = fn(q)
q = sqrt(dm(1)*dm(2)) - 1.0
dm(6) = fn(q)
sum = 0.0
do 110 k=1, l
q = c(k) - cp(k)

```

```
110  sum = sum + q*q
      sum = sum + sum
      dm(7) = dbfac*sqrt(sum)
      q = alog(q1)
      dm(8) = dbfac*sqrt(sum+q*q)
      q = alog(q2)
      dm(9) = dbfac*sqrt(sum+q*q)
      return
      end
```

c -----

```
subroutine find(m,nf,r,cep,ra,alpha,a,rc)
```

```
c This comes directly from Gray and Markel, Vol ASSP-24, No 5,  
c Oct 76, p390.
```

```
c Calculates the polynomials A(Z), the cepstral terms other than  
c C(0), and the polynomial autocorrelation.
```

```
dimension r(1),cep(1),ra(1),a(1),rc(1)
```

```
mp = m + 1  
a(1) = 1.  
a(2) = -r(2)/r(1)  
rc(1) = a(2)  
alpha = r(1) * (1.0 - a(2)*a(2))  
do 450 j=2, m  
mh = j/2  
jm = j-1  
q = r(j+1)  
do 420 l=1, jm  
lb = j+1-l  
420 q = q + a(l+1)*r(lb)  
q = -q/alpha  
rc(j) = q  
do 430 k=1, mh  
kb = j-k+1  
at = a(k+1) + q*a(kb)  
a(kb) = a(kb) + q*a(k+1)  
430 a(k+1) = at  
a(j+1) = q  
alpha = alpha*(1.0 - q*q)
```

```
c Kill job if unstable filter  
if (alpha.le.0.0) call exit  
450 continue
```

```
c .....
```

```
c Evaluation of cepstrum
```

```
cep(1) = a(2)  
do 455 j=2, m  
cep(j) = float(j)*a(j+1)  
jm = j-1  
do 455 k=1, jm  
kb = j-k+1  
455 cep(j) = cep(j) - cep(k)*a(j-k+1)  
if (nf.le.m) goto 480  
do 460 j=mp, nf  
cep(j) = 0.0  
do 460 k=1, m  
460 cep(j) = cep(j) - cep(j-k)*a(k+1)  
do 470 j=1, nf  
470 cep(j) = -cep(j)/float(j)
```

```
c .....
```

```
c Evaluation of polynomial autocorrelation
```

```
c480 do 500 l=1, mp  
c k=mp+l-1  
c ra(l) = 0.0  
c do 500 j=1, k  
c jl = l+j-1  
c500 ra(l) = ra(l) + a(j)*a(jl)
```

```
c Above was the code provided in the paper. Below is my  
c own routine that I wrote to do autocorrelations.
```

```
480 call myauto(a,mp,mp,ra)
```

```
555      write (6,555) (rc(j),j=1,m)
      format (4f10.6)
      return
      end
```

c -----

subroutine distance(tx,ty,itx,r,ifwdev,d)

c This routine calculates the matrix of distance measures, D(I,J),
c between the two templates TX and TY. It was patterned after
c subroutine DME ^S. Changes include the assumption of constant-
c length templates of 50 frames, and initializing the distance
c array to very large values outside the search path.

c INPUTS

c TX() template array for the x-axis
c TY() template array for the y-axis
c ITX First dimension of TX and TY (feature vector length)
c R Width of the DTW search space from the main diagonal
c IFWDEV frequency deviation index for frequency warping. A
c value of 0 means no frequency warping, and each unit
c gives 62.5 Hz either side of center; e.g. a value of
c 2 would provide a frequency window of 250 Hz.

c OUTPUT

c D(,) matrix of distances, where D(I,J) is the distance between
c vector I of template TX() and vector J of template TY().

c R is the width control for the search path. Outside the path,
c distances will be set to a very large number.

integer i,j,k,l,r,itx,ifwdev
real tx(itx,50),ty(itx,50),d(-2:50,-2:50),x,x1

c First initialize:

```

do 20 j=-2, 50
  do 10 i=-2, 50
    d(i,j) = 1.0e12
  10 continue
20 continue

```

c Now calculate the distances within the region of interest.

```

do 40 j=1,50
  do 30 i= max0(1,j-r), min0(50,j+r)
    x = 0.0
    do 25 k=1, itx
      x1 = 1.e12
      do 23 l= max0(1,k-ifwdev), min0(itx,k+ifwdev)
        x1 = amin1(x1,(tx(l,i) - ty(k,j))**2)
      23 continue
    c diagnostic print statement to check frequency warping
    c22 format(4i4,3(2x,g12.6))
    c write(6,22)j,i,k,l,tx(l,i),ty(k,j),x1
    23 continue
    x = x + x1
    c x = x + (tx(k,i) - ty(k,j))**2
    25 continue
    d(i,j) = sqrt(x)
    30 continue
  40 continue

```

return
end

c -----

```
subroutine distancecep(tx,ty,itx,r,ifwdev,d)
```

```
c This routine calculates the matrix of distance measures, D(I,J),
c between the two templates TX and TY. It was patterned after
c subroutine DMEAS. Changes include the assumption of constant-
c length templates of 50 frames, and initializing the distance
c array to very large values outside the search path.
```

```
c INPUTS
```

```
c TX() template array for the x-axis
c TY() template array for the y-axis
c ITX First dimension of TX and TY (feature vector length)
c R Width of the DTW search space from the main diagonal
c IFWDEV frequency deviation index for frequency warping. A
c value of 0 means no frequency warping, and each unit
c gives 62.5 Hz either side of center; e.g. a value of
c 2 would provide a frequency window of 250 Hz.
```

```
c OUTPUT
```

```
c D(,) matrix of distances, where D(I,J) is the distance between
c vector I of template TX() and vector J of template TY().
```

```
c R is the width control for the search path. Outside the path,
c distances will be set to a very large number.
```

```
c NCEP is the number of cepstral coefficients that are actually
c being used to compute the root-power-sums
```

```
integer i,j,k,l,r,itx,ifwdev,ncep
real tx(itx,50),ty(itx,50),d(-2:50,-2:50),x,x1
```

```
c First initialize:
```

```
ncep = 24
```

```
do 20 j=-2, 50
do 10 i=-2, 50
d(i,j) = 1.0e35
```

```
10 continue
20 continue
```

```
c Now calculate the distances within the region of interest.
```

```
do 40 j=1,50
do 30 i= max0(1,j-r), min0(50,j+r)
x = 0.0
do 25 k=1, ncep
x = x + ( k * (tx(k,i) - ty(k,j) ) )**2
25 continue
d(i,j) = sqrt(x)
```

```
c Diagnostic print statement
c27 format(2i4,2(2x,g12.6))
c write (6,27)i,j,d(i,j),x
30 continue
40 continue
```

```
return
end
```

```
c -----
```



```
subroutine distancecep2(tx,ty,itx,r,ifwdev,d)
```

```
c This routine calculates the matrix of distance measures, D(I,J),
c between the two templates TX and TY. It was patterned after
c subroutine DMEAS. Changes include the assumption of constant-
c length templates of 50 frames, and initializing the distance
c array to very large values outside the search path.
c It is patterned after DISTANCECEP except that it uses unweighted
c cepstral measures, giving an approximation to the L2 distance.
```

```
c INPUTS
```

```
c TX() template array for the x-axis
c TY() template array for the y-axis
c ITX First dimension of TX and TY (feature vector length)
c R Width of the DTW search space from the main diagonal
c IFWDEV frequency deviation index for frequency warping. A
c value of 0 means no frequency warping, and each unit
c gives 62.5 Hz either side of center; e.g. a value of
c 2 would provide a frequency window of 250 Hz.
```

```
c OUTPUT
```

```
c D(,) matrix of distances, where D(I,J) is the distance between
c vector I of template TX() and vector J of template TY().
```

```
c R is the width control for the search path. Outside the path,
c distances will be set to a very large number.
```

```
c NCEP is the number of cepstral coefficients that are actually
c being used to compute the cepstral distance
```

```
integer i,j,k,l,r,itx,ifwdev,ncep
real tx(itx,50),ty(itx,50),d(-2:50,-2:50),x,x1
```

```
c First initialize:
```

```
ncep = 24
```

```
do 20 j=-2, 50
```

```
do 10 i=-2, 50
```

```
d(i,j) = 1.0e35
```

```
10 continue
```

```
20 continue
```

```
c Now calculate the distances within the region of interest.
```

```
do 40 j=1,50
```

```
do 30 i= max0(1,j-r), min0(50,j+r)
```

```
x = 0.0
```

```
do 25 k=1, ncep
```

```
x = x + ( tx(k,i) - ty(k,j) )**2
```

```
25 continue
```

```
d(i,j) = sqrt(x)
```

```
c Diagnostic print statement
```

```
c27 format(2i4,2(2x,g12.6))
```

```
c write (6,27)i,j,d(i,j),x
```

```
30 continue
```

```
40 continue
```

```
return
```

```
end
```

```
c -----
```

```
subroutine distancelik(tx,ty,itx,r,d)
```

```
c This routine calculates the matrix of distance measures, D(I,J),
c between the two templates TX and TY. It was patterned after
c subroutine DMEAS. Changes include the assumption of constant-
c length templates of 50 frames, and initializing the distance
c array to very large values outside the search path. This version
c uses symmetrical likelihood ratios as the actual distance measure.
```

```
c INPUTS
```

```
c TX()  template array for the x-axis
c TY()  template array for the y-axis
c ITX   First dimension of TX and TY (feature vector length)
c R     Width of the DTW search space from the main diagonal
```

```
c OUTPUT
```

```
c D(i)  matrix of distances, where D(I,J) is the distance between
c       vector I of template TX() and vector J of template TY().
```

```
c R     is the width control for the search path. Outside the path,
c       distances will be set to a very large number.
```

```
integer i,j,k,l,r,itx
real tx(itx,50),ty(itx,50),d(-2:50,-2:50),x,x1
```

```
c First initialize:
```

```
do 20 j=-2, 50
  do 10 i=-2, 50
    d(i,j) = 1.0e12
  10 continue
20 continue
```

```
c Now calculate the distances within the region of interest.
```

```
do 40 j=1,50
  do 30 i= max0(1,j-r), min0(50,j+r)
    call likratio(tx(1,i),tx(65,i),tx(60,i),
a               ty(1,j),ty(65,j),ty(60,j),24,d(i,j))
  30 continue
40 continue
```

```
return
end
```

```
c -----
```

```
subroutine distancesw(tx,ty,itx,r,ifwdev,d)
```

c This routine calculates the matrix of distance measures, $D(I,J)$,
 c between the two templates TX and TY. It was patterned after
 c subroutine DISTANCE. The difference is that it incorporates a
 c Slope-dependent Weighting (SW) feature, as implemented with the
 c functions SLPWT and SLOPE.

c INPUTS

c TX() template array for the x-axis
 c TY() template array for the y-axis
 c ITX First dimension of TX and TY (feature vector length)
 c R Width of the DTW search space from the main diagonal
 c IFWDEV frequency deviation index for frequency warping. A
 c value of 0 means no frequency warping, and each unit
 c gives 62.5 Hz either side of center; e.g. a value of
 c 2 would provide a frequency window of 250 Hz.

c OUTPUT

c D(,) matrix of distances, where $D(I,J)$ is the distance between
 c vector I of template TX() and vector J of template TY().

c R is the width control for the search path. Outside the path,
 c distances will be set to a very large number.

```
integer i,j,k,l,r,itx,ifwdev
real tx(itx,50),ty(itx,50),d(-2:50,-2:50),x,x1,x2,slpwt
```

c First initialize:

```
do 20 j=-2, 50
  do 10 i=-2, 50
    d(i,j) = 1.0e12
  10 continue
  20 continue
```

c Now calculate the distances within the region of interest.

```
do 40 j=1,50
  do 30 i= max0(1,j-r), min0(50,j+r)
    x = 0.0
    do 25 k=1, itx
      x1 = 1.e12
      do 23 l=max0(1,k-ifwdev), min0(itx,k+ifwdev)
        x2 = slpwt(tx,l,i,ty,k,j,itx) *
a          ((tx(l,i) - ty(k,j))**2)
        x1 = amin1(x1,x2)
c      diagnostic print statement to check frequency warping
c22 format(4i4,3(2x,g12.6))
c      write(6,22)j,i,k,l,tx(l,i),ty(k,j),x1
      23 continue
      x = x + x1
c      x = x + (tx(k,i) - ty(k,j))**2
      25 continue
      d(i,j) = sqrt(x)
      30 continue
    40 continue

    return
  end
```

c -----

program e7r1

c This is experiment 7. It is a derivation of EXP1A and EXP5 and
 c combines all their features into one flexible program. The
 c purpose of this program is to load the TD() array, which is
 c of dimension ITDxITD. It clashes normal, loud, and Lombard
 c conditions against the normal templates as reference. Within
 c the TD() array, there are blocks of zero entries that are of
 c dimension IPxIP, and situated along the main diagonal. If
 c normal speech is being tested, then the TD() array will be
 c symmetrical, and this economy is exploited in the code. For
 c abnormal speech conditions, the TD() will not be symmetrical,
 c and will be $(IP-1)*100/IP$ % filled. The PLUS suffix indicates
 c that the features of EXP7FILL have been incorporated. If the
 c MASSFILL option is selected (available only when clashing
 c loud-normal or Lombard-normal), the program will exhaustively
 c test for and fill any zero entries in the TD() array. This
 c option can be used to either fill the zero blocks on the
 c main diagonal or to completely load the TD() array from scratch.

c TXS() is the single template array for the x-axis
 c TYS() is the single template array for the y-axis
 c ITX is the feature vector length (1st dimension of
 c TXS and TYS)
 c D() Distance array used by the warping algorithm to find
 c the minimal path
 c TD() Total distance array that is the result from warping
 c all applicable combinations of templates This is
 c where the results of this program are stored.
 c MASSFILL indicates whether MASSFILL mode has been selected.
 c SAVEIT indicates whether new values of the TD() need saving.
 c NEEDTESTTEMP indicates whether the test template has already
 c been loaded.
 c FILE1 is the name of the data file for the template being
 c tested
 c FILE2 is the name of the data file for the template being
 c referenced
 c FILE3 is the name of the data file where the TD() array is
 c stored
 c IROOT is the index for the root path name selected
 c II is the pointer into the strings FILE1 and FILE2 to
 c indicate where the identification numbers start
 c IOCC is the occurrence number to start with
 c for the phoneme tested
 c IPHONE is the short index number to start with for the
 c phoneme tested
 c I is the occurrence index of the phoneme being tested
 c J is the short index of the phoneme being tested
 c K is the occurrence index of the phoneme being referenced
 c L is the short index of the phoneme being referenced
 c JJ is the row index for the TD() array. The row refers to
 c the phoneme being tested.
 c LL is the column index for the TD() array. The column refers
 c to the phoneme being referenced.
 c ISTR passes the number of characters in a string to ENDSTR.
 c When returned, it contains the actual length of the string.
 c ISPKR is the speaker number being processed.
 c ICOND is the condition being processed
 c IFWDEV frequency deviation index for frequency warping. A
 c value of 0 means no frequency warping, and each unit
 c gives 62.5 Hz either side of center; e.g. a value of
 c 2 would provide a frequency window of 250 Hz.

```

c IT      is the number of templates per phoneme
c IP      is the size of the phoneme set
c ITD     is the dimension of the TD array (IT*IP)
c R       Width of the DTW search space from the main diagonal

```

```

logical massfill,saveit,needtesttemp
character*36 file1,file2,tdfilename,pname
integer i,j,k,l,jj,ll,it,ip,itd,iss(40),r,itx,ifwdev
integer iocc,iphone,iroot,ii,istr,ispkr,icond,itt,icount
integer rowstartindex
real d(-2:50,-2:50)

```

```

c Caution: the dimension of the TD array must be >= IT*IP
real td(280,280)
real txs(128,50),tys(128,50)
c Caution: the ISS array must have IP entries. This is a pointer
c array that allows subsets of the 40 phonemes to be easily accessed.
c The way it is now initialized, it is transparent since all 40
c phonemes are being tested. It is used in the program in the
c calls to BUILDPATH where phoneme templates are sought.
data (iss(i),i=1,15)/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/
data (iss(i),i=16,27)/16,17,18,19,20,21,22,23,24,25,26,27/
data (iss(i),i=28,40)/28,29,30,31,32,33,34,35,36,37,38,39,40/
c This was the initialization when considering only 29 phonemes.
c data (iss(i),i=1,15)/1,2,3,4,7,9,11,12,14,15,18,19,20,23,24/
c data (iss(i),i=16,29)/25,26,27,28,29,30,31,33,34,35,36,38,39,40/

```

```

35      format("TD(",i3,",",i3,") =",f9.4)

```

```

c Note that IT has been set to 6 because there are only 6 tokens
c of some phonemes.

```

```

icount = 0
r = 1
it = 6
ip = 40
itd = it*ip
itx = 128
iocc = 1
iphone = 1
saveit = .false.
pname = "E7R1"

```

```

call usrinit(pname,it,ip,ifwdev,ispkr,icond,massfill,tdfilename)
call gettd(tdfilename,td)
if (.not.massfill) call tdstatus(td,it,ip,itd,iocc,iphone)

```

```

c Now comes the nested iterations that will clash the templates together.
c Either normal, loud, or Lombard will be used as test while the IT normal
c occurrence sets will be used as reference. If normal is tested, then the
c TD() array will be symmetrical. For all cases, the IPxIP blocks on the main
c diagonal will be empty, unless the MASSFILL option has been selected.
c Recall that the normal-normal template is symmetrically filled, meaning that
c the last block of rows will have already been calculated by the time the
c iterations progress that far.

```

```

if (icond.eq.1) then
  itt = it - 1
else
  itt = it
end if

```

```

do 300 i=iocc, itt

```

```

  do 280 j=iphone, ip

```

```

needtesttemp = .true.
call buildpath(0,i,iss(j),"pft",ispkr,icond,file1)

do 260 k=rowstartindex(icond,i), it
  if (k.eq.i.and..not.massfill) goto 260

  do 240 l=1, ip
    jj = j + ip*(i-1)
    ll = l + ip*(k-1)

    if (td(jj,ll).eq.0.0) then
      if (needtesttemp) then
        call readtp(file1,txs,itx,.true.)
        needtesttemp = .false.
      end if
      call buildpath(0,k,iss(l),"pfr",ispkr,icond,file2)
      call readtp(file2,tys,itx,.false.)

      call distance(txs,tys,itx,r,ifwdev,d)
c      call distancecep2(txs,tys,itx,r,ifwdev,d)
c      call distancecep(txs,tys,itx,r,ifwdev,d)
c      call distancelik(txs,tys,itx,r,d)
      call warper(d,50,50,r,td(jj,ll))
      icount = icount + 1
      if (icond.eq.1) td(ll,jj) = td(jj,ll)
c      write (6,35)jj,ll,td(jj,ll)
      saveit = .true.
    end if

240      continue
260      continue

    if (saveit) then
      call puttd(tdfilename,td)
      saveit = .false.
    end if

280      continue

c      In the event the program is restarted in the middle of an unfinished
c      block, IPHONE will have been set in TDSTATUS. After that block
c      has been finished, IPHONE must then be reset.

      iphone = 1
300      continue

      call report(pname,ispkr,icond,ifwdev,icount)

      stop
      end
c -----

```

program e7r1sw025

c Derived from the E7R1 series of experiments. This manages the
c slope-dependent weighting algorithm. It Incorporates the
c slope difference threshold in a common block with the necessary
c subroutine.

```
logical massfill,saveit,needtesttemp
character*36 file1,file2,tdfilename,pname
integer i,j,k,l,jj,ll,it,ip,itd,iss(40),r,itx,ifwdev
integer iocc,iphone,iroot,ii,istr,ispkr,icond,itt,icount
integer rowstartindex
real d(-2:50,-2:50)
real td(280,280)
real txs(128,50),tys(128,50),threshold

common /knee/threshold

data (iss(i),i=1,15)/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/
data (iss(i),i=16,27)/16,17,18,19,20,21,22,23,24,25,26,27/
data (iss(i),i=28,40)/28,29,30,31,32,33,34,35,36,37,38,39,40/
```

35 format("TD(",i3,",",i3,") =",f9.4)

```
icount = 0
r = 1
it = 6
ip = 40
itd = it*ip
itx = 128
iocc = 1
iphone = 1
saveit = .false.
pname = "E7R1SW025"
threshold = 0.25

call usrinit(pname,it,ip,ifwdev,ispkr,icond,massfill,tdfilename)
call gettd(tdfilename,td)
if (.not.massfill) call tdstatus(td,it,ip,itd,iocc,iphone)

if (icond.eq.1) then
  itt = it - 1
else
  itt = it
end if

do 300 i=iocc, itt

  do 280 j=iphone, ip
    needtesttemp = .true.
    call buildpath(0,i,iss(j),"pft",ispkr,icond,file1)

    do 260 k=rowstartindex(icond,i), it
      if (k.eq.i.and..not.massfill) goto 260

      do 240 l=1, ip
        jj = j + ip*(i-1)
        ll = l + ip*(k-1)

        if (td(jj,ll).eq.0.0) then
          if (needtesttemp) then
            call readtp(file1,txs,itx,.true.)
            needtesttemp = .false.
          end if
        end if
      end do
    end do
  end do
end do
```

```

        end if
        call buildpath(0,k,iss(l),"pfr",ispkr,icond,file2)
        call readtp(file2,tys,itx,.false.)

        call distancesw(txs,tys,itx,r,ifwdev,d)
c      call distance(txs,tys,itx,r,ifwdev,d)
        call warper(d,50,50,r,td(jj,ll))
        icount = icount + 1
        if (icond.eq.1) td(ll,jj) = td(jj,ll)
c      write (6,35)jj,ll,td(jj,ll)
        saveit = .true.
        end if

240      continue
260      continue

        if (saveit) then
            call puttd(tdfilename,td)
            saveit = .false.
        end if

280      continue
        iphone = 1
300      continue

        call report(pname,ispkr,icond,ifwdev,icount)

        stop
        end
c -----

```



```
subroutine endstr(a,ic)
```

```
c This routine finds the end of a string and marks
c it with a null character. It can be used to truncate
c a string that has trailing blanks, and is needed
c specifically when passing filenames to library routine
c UOPEN. A feature added on 27 May 87 is the return of
c the actual length of the string through the variable IC.
c Therefore, arguments to this routine must be variables
c rather than constants.
```

```
integer i,j,ic
character*1 a(ic)

j = ic
do 10 i=1,ic
    if(a(i).lt.' ') then
        a(i) = ' '
        j = i-1
        i = ic
    endif
```

```
10 continue
   ic = j
   return
end
```

```
c -----
```

```
function euclid(x,y,n)
```

```
c Designed to calculate the Euclidean distance between the two
c vectors X and Y.
```

```
c INPUTS:
c X holds the x-vector
c Y holds the y-vector
c N is the dimension of X and Y
```

```
c OUTPUT:
c EUCLID holds the Euclidean distance between X and Y
```

```
real euclid,x(n),y(n),xx
integer n,i
```

```
xx = 0.0
```

```
do 10 i=1, n
    xx = xx + (x(i) - y(i))**2
10 continue
```

```
euclid = sqrt(xx)
```

```
return
end
```

```
c -----
```

```
real function errate(pathname,m)
```

```
c Provides the error rate of the recognition experiment stored
c in PATHNAME, for phoneme vector length M
```

```
character*(*) pathname
integer m
real p1(10)
```

```
errate = 0.0
open (unit=1,file=pathname,status="old",err=60)
rewind (1)
read (1,*)p1
close (1)
errate = 100.0 - p1(m)
```

```
60    return
      end
```

```
c -----
```

program exp6

c This is Experiment 6. It is derived from Experiment 2 and is
 c very similar except that it provides the flexibility of producing
 c recognition results from using N of the available reference tokens
 c ($1 \leq N \leq IT-1$). It is designed to take the results from
 c Experiment 5 (Program EXP5) and produce recognition hypotheses.
 c It operates on the array TD(), which it expects to find in the
 c file named by the user. A row of array TD() represents the
 c calculated distances from the test template to the $(IT-1)*IP$ reference
 c templates. This program will take from IP to $(IT-1)*IP$ distances
 c (in multiples of IP) for each row and find out what reference
 c phonemes scored best with the smallest distances. Efforts are
 c also made to combine the scores of the reference templates of
 c a given phoneme to obtain improved results. Relative performance
 c of different ranking methods are then calculated and displayed.

c NOTFOUND is a flag used to catch the first occurrence of the
 c correct phoneme in the raw nearest neighbor (RNN) ranking
 c PRINTIT is a flag for printing optional results on individual
 c phonemes
 c STOREM is a flag for storing percentage performances in files
 c for qplot
 c FULLARRAY is a flag telling whether or not we are working with an
 c array that has been completely filled.
 c TD() Total distance array that is the result from warping
 c all applicable combinations of templates.
 c A() Scratch array used to hold one row of TD() at a time.
 c CR() Array used to store the cumulative rank of each of the
 c IP phonemes.
 c CD() Array used to store the cumulative distance of each of
 c the IP phonemes.
 c COL() is used to preserve the original column location in TD()
 c of a total distance value
 c P1() is the permutation array that is used in the sorting of
 c distances between templates
 c P2() is the permutation array that is used in the sorting of
 c cumulative ranks of each reference phoneme.
 c P3() is the permutation array that is used in the sorting of
 c cumulative distances of each reference phoneme.
 c IROW is the index for rows in the TD() array.
 c IROWMAX contains the number of rows in TD() that have been
 c already loaded with total distance values.
 c IZERO is the number of zeroes in a given row of the TD() array.
 c If $IZERO > IP$, then it means that this row has not been
 c fully calculated. The allowance of IP zeroes accounts for
 c the block of the TD() array that represents clashing
 c a given occurrence with itself. Obviously, this is not
 c done because it would give meaningless data when clashing
 c normal vs normal.
 c II is used in index conversion for the test phoneme
 c JJ is used in index conversion for the reference phoneme
 c ML is the label array that gives ARPABET strings for each
 c of the ascii codes for the phonemes.
 c ICMP stores the value used as a threshold to skip IP values
 c when a row of TD() is loaded into array A().
 c ICOL is the column index for loading a row of TD() into
 c array A().
 c RNN(I) keeps track of how many times the BEST rank of the
 c correct phoneme occurred in position I for the
 c Raw Nearest Neighbor method. (Recall that
 c there are multiple occurrences equal to the number of
 c templates for a given phoneme.)

```

c SCR(I) keeps track of how many times the rank of the correct
c phoneme occurred in position I for the Smallest
c Cumulative Rank method.
c SCD(I) keeps track of how many times the rank of the correct
c phoneme occurred in position I for the Smallest
c Cumulative Distance method.
c CRNN(I) keeps track of percentage of times the correct phoneme
c occurred in positions 1 through I for the RNN method.
c CSCR(I) keeps track of percentage of times the correct phoneme
c occurred in positions 1 through I for the SCR method.
c CSCD(I) keeps track of percentage of times the correct phoneme
c occurred in positions 1 through I for the SCD method.
c IRNN is the number of times the RNN scored best (no ties)
c ISCR is the number of times the SCR scored best (no ties)
c ISCD is the number of times the SCD scored best (no ties)
c JRNN is the best rank of the RNN method for a given test
c JSCR is the best rank of the SCR method for a given test
c JSCD is the best rank of the SCD method for a given test
c ISTR passes the number of characters in a string to ENDSTR.
c ITMAX is the number of reference tokens used for each phoneme
c ITSTART is the number of reference tokens to start with in the
c major iteration loop for different numbers of reference
c tokens

```

```

logical notfound, printit, storem, fullarray
character*1 ans
character*7 file1, file2, file3
character*10 class(10)
character*36 tdfilename
real a(300), cr(300), cd(300)
real rnn(40,10), scr(40,10), scd(40,10)
real crnn(40,10), cscr(40,10), cscd(40,10)

```

```

c Caution: the dimension of TD() and COL() must be at least IP*IT
real td(280,280)
integer col(280)

```

```

integer i,j,ishort(40),ishort2(40),irow,irowmax,izero
integer ii,jj,irnn,iscr,iscd,jrnn,jscr,jscd,p1(300)
integer iclass,ipclass,itstart,ispkr,icond
integer icmp,icol,itmax,it,ip,itd,itt,p2(300),p3(300)
include "ml"

```

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

```

c ISHORT2 is built along the same idea as ISHORT, except that it only
c has IP phonemes that had adequate population tokens, as determined
c and set in by hand. Caution: ISHORT2 must have IP entries.

```

```

data (ishort2(i),i=1,10)/112,116,107,98,70,110,115,122,84,102/
data (ishort2(i),i=11,20)/118,108,114,76,119,69,99,97,117,82/
data (ishort2(i),i=21,29)/89,101,120,73,64,94,105,111,88/

```

```

a data (class(i),i=1,6)/" all","stops","nasals","fricatives",
"liquids","vowels"/

```

```

c JRNN, JSCR, and JSCD must be initialized to the max rank value because

```

c they are used to determine which of the three ranking methods had
 c the correct phoneme in the lowest position.

```

    ip = 40
    it = 6
    itd = it*ip
    jrnn = ip
    jscr = ip
    jscd = ip
c Let's start with an empty deck...
    do 2 j=1, 10
        do 1 i=1, 40
            rnn(i,j) = 0.0
            scr(i,j) = 0.0
            scd(i,j) = 0.0
            crnn(i,j) = 0.0
            cscr(i,j) = 0.0
            cscd(i,j) = 0.0
1            continue
2        continue

    write (6,*)

4    format("/Program EXP6....")
    write (6,4)
    call getspkr(ispkr)
    call getcond(icond)
    call tddpath(ispkr,icond,tddfilename)
    call gettd(tddfilename,td)

    fullarray = .false.
    if (td(1,1).ne.0.0) then
10    format("/Block diagonals of TD appear to be full..."/
        a    "Do you wish to treat TD as a full array? ",%)
        write (6,10)
        read (5,*)ans
        if (ans.eq."y".or.ans.eq."Y") fullarray = .true.
    else
        write (6,*)"Block diagonals of TD appear to be zero..."
    end if

c Check to see how far the calculations have progressed
c The code below is a prime candidate to be replaced by subroutine
c TDSTATUS. However, it will be done at another time. Hint:
c IROWMAX would be set to (IOCC-1)*40 + IPHONE.

```

```

    irowmax = 0

    if (fullarray) then
        do 15 i=1, itd
            do 14 j=1,itd
                if (td(i,j).eq.0.0) then
                    irowmax = i
                    i = itd
                    j = itd
                end if
14            continue
15        continue
    else
        do 17 i=1, itd
            izero = 0
            do 16 j=1, itd
                if (td(i,j).eq.0.0) izero = izero + 1

```

```

16      continue
      if (izero.gt.ip) then
        irowmax = i
        i = itd
      end if
17      continue
      end if

      if (irowmax.ne.0) then
        irowmax = irowmax - 1
        write (6,*)"Calculations complete only through row ",irowmax
      else
        write (6,*)"Calculations are complete. TD fully loaded."
        irowmax = itd
      end if

18      format(/"Do you want individual phoneme results printed? ",%)
      write (6,18)
      read (5,*)ans
      if (ans.eq."Y".or.ans.eq."y") then
        printit = .true.
      else
        printit = .false.
      end if

19      format(/"Do you want percentages stored for qplotting? ",%)
      write (6,19)
      read (5,*)ans
      if (ans.eq."Y".or.ans.eq."y") then
        storem = .true.
      else
        storem = .false.
      end if

```

c Beginning of major iteration loop for testing different numbers
 c of templates for each phoneme.

```

      if (fullarray) then
        itt = it
      else
        itt = it-1
      end if

20      format (/ "Enter number of reference tokens to start with: ",%)
      write (6,20)
      read (5,*)itstart

      do 400 itmax=itstart, itt

      do 200 irow=1, irowmax

```

c For each row, load the values into the working array A(). Note
 c that IROW indicates what phoneme is being tested, modulo(40)
 c Note that we immediately count the number of tests in
 c each class, storing the value in row 40 of the RANKARRAYs so the
 c total will be available when calculating percentages.

```

      rnn(40,1) = rnn(40,1) + 1
      scr(40,1) = scr(40,1) + 1
      scd(40,1) = scd(40,1) + 1
      iclass = ipclass(irow)
      rnn(40,iclass) = rnn(40,iclass) + 1
      scr(40,iclass) = scr(40,iclass) + 1

```

```

scd(40,iclass) = scd(40,iclass) + 1

c Diagnostic print statements for debugging
c201 format(3i4,6f5.0)
c write(6,201)irow,iclass,irowmax,rnn(40,1),scr(40,1),scd(40,1),
c a rnn(40,iclass),scr(40,iclass),scd(40,iclass)

if (fullarray) then
do 23 i=1, ip*itmax
a(i) = td(irow,i)
col(i) = i
23 continue
else
c This loop leaves out the IP values that correspond to the
c occurrence of the phoneme being tested.

icmp = int((irow-1)/ip)*ip

do 26 i=1, ip*itmax
if (i.le.icmp) then
icol = i
else
icol = i+ip
end if
a(i) = td(irow,icol)
col(i) = icol
26 continue
end if

c Sort the total distances in the A() array, lowest to highest.
c From this sort, the reference templates can be assigned a ranking
c of nearness to the template being tested.

c The permutation array cannot have any invalid subscript entries
c for the array A(). Otherwise, the QSORT algorithm will not work.
c Therefore, make sure that INITPI is always called before QSORT

i = 1
j = ip*itmax
call initpi(p1,300)
call qsort(a,p1,i,j)

30 format(/"Test phoneme: ",a3,/"RNN: ",%)
32 format(" ")
34 format(a3," ",%)
36 format("SCR: ",%)
38 format("SCD: ",%)
ii = mod(irow,ip)
if (ii.eq.0) ii = ip
if (printit) write (6,30)ml(ishort(ii))

c Use the flag NOTFOUND to catch the FIRST occurrence of the correct
c phoneme in the ranking. Remember that in the RNN, there are multiple
c occurrences (4) of each phoneme. We only want the lowest ranking one.

notfound = .true.

do 50 i=1,10
jj = mod(col(p1(i)),ip)
if (jj.eq.0) jj = ip
if (printit) write (6,34)ml(ishort(jj))
if (ii.eq.jj.and.notfound) then
call update(rnn,i,irow)

```

```

        jrnn = i
        notfound = .false.
        end if
50      continue
        if (printit) write (6,32)

```

c Now initialize the cumulative rank and distance arrays. They
c will have old data from a previous row.

```

        do 70 i=1, ip
            cr(i) = 0.0
            cd(i) = 0.0
70      continue

```

c Next calculate the cumulative rank and distance simultaneously.

```

        do 80 i=1, ip*itmax
            jj = mod(col(p1(i)),ip)
            if (jj.eq.0) jj = ip
            cr(jj) = cr(jj) + i
            cd(jj) = cd(jj) + a(p1(i))
80      continue

```

c Sort the cumulative ranks

```

        i = 1
        j = ip
        call initpi(p2,300)
        call qsort(cr,p2,i,j)
        if (printit) write (6,36)
        do 90 i=1, 10
            if (printit) write (6,34)ml(ishort(p2(i)))
            if (ii.eq.p2(i)) then
                call update(scr,i,irow)
                jscr = i
            end if
90      continue
        if (printit) write (6,32)

```

c Sort the cumulative distances

```

        i = 1
        j = ip
        call initpi(p3,300)
        call qsort(cd,p3,i,j)
        if (printit) write (6,38)
        do 100 i=1, 10
            if (printit) write (6,34)ml(ishort(p3(i)))
            if (ii.eq.p3(i)) then
                call update(scd,i,irow)
                jscd = i
            end if
100     continue
        if (printit) write (6,32)

```

c Next determine which method scored best and then reset score
c variables.

```

        if (jrnn.lt.jscr.and.jrnn.lt.jscd) then
            irnn = irnn + 1
        else if (jscr.lt.jrnn.and.jscr.lt.jscd) then
            iscr = iscr + 1
        else if (jscd.lt.jrnn.and.jscd.lt.jscr) then

```



```

        iscd = iscd + 1
        end if

        jrnn = ip
        jscr = ip
        jscd = ip

200      continue

c At this point, all that is left is to massage the accumulated data
c a bit, and then print out the results in an easily readable format.
c Here is the temporary entry point to test all this formatting
225      continue

c Newly added (3 Feb 88): Cycle through the various phoneme classes

        do 380 iclass=1, 6

250      format(// "Reference tokens = ",i3," Phoneme Class: ",a10//
a          "Correct match occurred",/"in position:",t20,
a          10(i2,4x),/" (Percent)")
        write (6,250)itmax,class(iclass),(i,i=1,10)

252      format("RNN individual   ",$)
254      format("RNN cumulative   ",$)
256      format("SCR individual   ",$)
258      format("SCR cumulative   ",$)
260      format("SCD individual   ",$)
262      format("SCD cumulative   ",$)
270      format(10f6.1)
272      format(// "RNN scored best ",i3," times"/
a          "SCR scored best ",i3," times"/
a          "SCD scored best ",i3," times")

c Change the data to percentages and calculate cumulatives

        do 300 i=1, 10
            rnn(i,iclass) = rnn(i,iclass)*100/rnn(40,iclass)
            scr(i,iclass) = scr(i,iclass)*100/scr(40,iclass)
            scd(i,iclass) = scd(i,iclass)*100/scd(40,iclass)
            do 290 j=1, i
                crnn(i,iclass) = crnn(i,iclass) + rnn(j,iclass)
                cscr(i,iclass) = cscr(i,iclass) + scr(j,iclass)
                cscd(i,iclass) = cscd(i,iclass) + scd(j,iclass)
290          continue
300        continue

c      write (6,252)
c      write (6,270)(rnn(i,iclass), i=1, 10)
c      write (6,254)
c      write (6,270)(crnn(i,iclass), i=1, 10)
c      write (6,32)
c      write (6,256)
c      write (6,270)(scr(i,iclass), i=1, 10)
c      write (6,258)
c      write (6,270)(cscr(i,iclass), i=1, 10)
c      write (6,32)
c      write (6,260)
c      write (6,270)(scd(i,iclass), i=1, 10)
c      write (6,262)
c      write (6,270)(cscd(i,iclass), i=1, 10)
c      write (6,32)
c      write (6,272)jrnn,iscr,iscd

```

```

if (storem) then
  file1 = "rnn0"//char(48+itmax)//class(iclass)
  file2 = "scr0"//char(48+itmax)//class(iclass)
  file3 = "scd0"//char(48+itmax)//class(iclass)
  open (unit=1,file=file1,status="unknown")
  rewind (1)
  write (1,270)(crnn(i,iclass), i=1, 10)
  close (1)
  open (unit=2,file=file2,status="unknown")
  rewind (2)
  write (2,270)(cscr(i,iclass), i=1, 10)
  close (2)
  open (unit=3,file=file3,status="unknown")
  rewind (3)
  write (3,270)(cscd(i,iclass), i=1, 10)
  close (3)
end if

```

c Finally, zero out all arrays and variables that accumulate

```

      irnn = 0
      iscr = 0
      iscd = 0

      do 350 i=1, 40
        rnn(i,iclass) = 0.0
        scr(i,iclass) = 0.0
        scd(i,iclass) = 0.0
        crnn(i,iclass) = 0.0
        cscr(i,iclass) = 0.0
        cscd(i,iclass) = 0.0
350    continue

380    continue

400    continue

      stop
      end
c -----

```

subroutine features(ts,eb,ifftot)

c This routine calculates the LPC center of gravity, energy in
 c different frequency bands for a phoneme, and spectral tilt
 c for low and high bands from the template that has been
 c passed to it.

c INPUT

c TS Array that contains 50 frames of 128-pt LPC log-mag spectra

c OUTPUT

c EB is a vector having the 50-frame average of each feature
 c calculated

c IFTOT is the total number of features calculated in this routine

c RES is the frequency resolution of the spectra

c NFB is the number of frequency bands calculated

c SMIN is used to find the minimum value in the spectrum

c CN is the cumulative numerator term for Center of Gravity (COG)

c CD is the cumulative denominator term for COG

c COR is the correction factor used to give non-negative weight to
 c all samples of the spectrum for COG calculations.

c EBB(L,J) is used as a buffer for features of individual frames.
 c L is the feature number
 c J is the frame number

c XYL(I,J) is the input array for IMSL routine RLLAV. I is the
 c index for individual energy band samples. J=1 contains
 c the independent variable (log frequency in this case).
 c J=1 contains the dependent variable (energy in a particular
 c band). The other columns are used as scratch area.
 c This array contains the low band data from 0 to 3kHz.

c XYH(I,J) same as XYL() except for the high band data, 3kHz to 8kHz.

c BETA contains the spectral tilt (slope) value calculated by RLLAV.

c SUMRE is the sum of residuals from RLLAV.

c WK, IWK are scratch buffers required by RLLAV.

c ITER is the number of iterations required by RLLAV.

c IRANK is the rank of the matrix of independent variables from RLLAV.

c IER is the error parameter from RLLAV.

c ISWITCH is the energy band after which begins the high band for
 c purposes of computing spectral tilt

c I general purpose iteration index

c IFRAME index for frame numbers in a template

c IBAND index for the frequency bands

```
real ts(128,50),res,cn,cd,smin,cor,xyl(5,6),xyh(5,6),beta(2)
real fl(20),fh(20),ebb(20,50),eb(20),sumre,wk(6),a1,a0,r2
integer nfb,iband,iter,irank,iwk(5),ier,iswitch
integer i,iframe,ifftot
```

```
data (fl(i),i=1,10)/ 0, 250, 500, 1000, 2000, 3000, 4000,
a      5000, 6000, 7000/
data (fh(i),i=1,10)/250, 500, 1000, 2000, 3000, 4000, 5000,
a      6000, 7000, 8000/
```

```
res = 8000.0/128
nfb = 10
ifftot = nfb + 3
iswitch = 5
eb(nfb+1) = 0.0
```

```
do 60 iframe=1, 50
```

```

c ---- Section to compute the center of gravity of each
c      LPC spectrum

      smin = 1.0e6
      cn = 0.0
      cd = 0.0

c      First, find the minimum value in the spectrum. Use
c      this to offset the entire spectrum to the positive side.

      do 40 i=1, 128
        smin = amin1(smin,ts(i,iframe))
40      continue

      if (smin.lt.0.0) then
        cor = abs(smin)
      else
        cor = 0.0
      end if

      do 50 i=1, 128
        cn = cn + i * (ts(i,iframe)+cor)
        cd = cd +      ts(i,iframe)+cor
50      continue

      ebb(nfb+1,iframe) = res*cn/cd
      eb(nfb+1) = eb(nfb+1) + ebb(nfb+1,iframe)

c ---- Section to compute energy in different frequency bands
c      for the frame under consideration

      do 56 iband=1, nfb
        ebb(iband,iframe) = 0.0

        do 54 i=int(fl(iband)/res)+1, int(fh(iband)/res)
          ebb(iband,iframe) = ebb(iband,iframe) + ts(i,iframe) + 10
54          continue

          ebb(iband,iframe) = ebb(iband,iframe)/(int(fh(iband)/res) -
a                                int(fl(iband)/res))
          eb(iband) = eb(iband) + ebb(iband,iframe)
56          continue

60      continue

c      Now compute the sample means

      do 70 iband=1, nfb+1

70      eb(iband) = eb(iband)/50.0
          continue

c      Here is where the spectral tilt for low and high bands are
c      computed. Note that the average energies of the individual
c      bands are conveniently available in EB(). First compute
c      the low-band tilt.

      do 80 iband=1, iswitch
        xyl(iband,1) = alog((fh(iband)+fl(iband))/2)
        xyl(iband,2) = eb(iband)
80      continue
cccc  call rllav(xyl,5,5,1,0,beta,sumre,iter,irank,iwk,wk,ier)
      call lreg(xyl(1,1),xyl(1,2),5,a1,a0,r2)

```

```

      eb(nfb+2) = a1

      do 85 iband=iswitch+1, nfb
        xyh(iband-iswitch,1) = alog((fh(iband)+f1(iband))/2)
        xyh(iband-iswitch,2) = eb(iband)
85      continue
cccc  call rllav(xyh,5,5,1,0,beta,sumre,iter,irank,iwk,wk,ier)
      call lreg(xyh(1,1),xyh(1,2),5,a1,a0,r2)
      eb(nfb+3) = a1

c      End of the computing section of this routine.

c Temporary section to show results and make available for plotting

c      open (unit=1,file="fcc",status="unknown")
c      rewind (1)
c      write (1,*)fcc
c      close (1)
c      write (6,*)"Avg COG (Hz): ",fc

c      do 180 i=1, nfb+1
c      open (unit=1,file="ebb"//char(48+i),status="unknown")
c      rewind (1)
c      write (1,*)(ebb(i,iframe),iframe=1,50)
c      close (1)
c175  format("Avg energy from ",f7.1," to ",f7.1," = ",f14.3)
c      write (6,175)f1(i),fh(i),eb(i)
180    continue

      return
      end
c -----

```

```
subroutine finddif(dat,iphone,itc,ibc,dif)
```

- c Finds the difference between two conditions for all 18 features.
- c The average energies in features 1 - 10 are converted to dB.
- c The spectral tilts in features 12 and 13 are converted to dB/octave.
- c All other features are in correct units:
- c Center of Gravity, Pitch, Formants -- Hz
- c Duration -- Seconds

```
real dat(1200,20,3),dif(1)
integer iphone,itc,ibcb,xbar
real dbfac,dbocfac
```

```
dbfac = 10.0/log(10.0)
dbocfac = dbfac * log(2.0)
```

```
xbar = 7*iphone - 4
```

```
do 10 i=1, 18
  if (i.le.10) then
    dif(i) = dbfac * (dat(xbar,i,itc) - dat(xbar,i,ibc))
  else if (i.eq.12.or.i.eq.13) then
    dif(i) = dbocfac * (dat(xbar,i,itc) - dat(xbar,i,ibc))
  else
    dif(i) = dat(xbar,i,itc) - dat(xbar,i,ibc)
  end if
```

```
10  if (abs(dif(i)).gt.1.0e18) dif(i) = 0.0
    continue
```

```
return
end
```

```
c -----
```

```
subroutine fsmooth(f,i1,i2,b)
```

c This routine is derived from DIAGNOSTICF. It uses the algorithm
c described on page 117 of my personal notes to eliminate erroneous
c extreme points in formant and pitch data from the LISPM.

```
real f(1:2000,0:3),b(0:3),tol(0:3),fmax,fmin  
double precision b2(0:3),b3(0:3)  
integer i1,i2,i,j,itot(0:3),icount
```

```
data tol /100., 100., 200., 300./
```

```
do 20 i=0, 3  
  icount = 1
```

```
8   b3(i) = 0.0  
    b2(i) = 0.0  
    itot(i) = 0
```

```
c   write (6,*) "i1= ",i1," i2= ",i2
```

```
do 10 j=i1, i2  
c   write (6,*) "J, I, F(J,I) ===",j,i,f(j,i)  
    if (f(j,i).ne.0.0) then  
        b3(i) = b3(i) + f(j,i)  
        b2(i) = b2(i) + f(j,i)**2  
        itot(i) = itot(i) + 1  
    end if  
10  continue
```

c There is a possibility of having either none or only one non-zero
c value of F(J,I) (especially in pitch). If this is the case, B3(I) and
c B2(I) must be zeroed out rather than dividing by zero. In fact, I will
c trap out any time there are less than three valid samples.

```
    if (itot(i).lt.3) then  
        b3(i) = 0.0  
        b2(i) = 0.0  
    else  
        b3(i) = b3(i)/itot(i)  
        b2(i) = (b2(i)-itot(i)*(b3(i)**2))/(itot(i)-1)  
    end if
```

```
c   Diagnostic trap in the event we have a negative value  
    if (b2(i).lt.0.0) then  
12  format (/ "Value less than zero in subroutine FSMOOTH: "//  
a    "b2(i) = ",g12.6)
```

```
    write (6,12) b2(i)  
    write (6,*) "i=",i  
    write (6,*) "b2=",b2  
    write (6,*) "itot=",itot  
    write (6,*) "b3=",b3  
    write (6,*) "No harm done to overall results..."  
    b2(i) = 0.0  
    end if  
    b2(i) = sqrt(b2(i))
```

```
    if (b2(i).gt.tol(i).and.icount.lt.3) then  
        icount = icount + 1  
        fmax = b3(i) + b2(i)  
        fmin = b3(i) - b2(i)
```

```

do 15 j=i1, i2
  if (f(j,i).gt.fmax.or.f(j,i).lt.fmin) f(j,i) = 0.0
15  continue

  goto 8
end if

20  continue

do 30 i=0, 3
  b(i) = b3(i)
25  format("B(",i1,") has value: ",g12.6)
  if (b(i).gt.1.0e12) write(6,25) i,b(i)
30  continue

  return
end
c -----

```



```

subroutine getcond(icond)

integer icond
character*1 cond

35      format (a1)
40      format(/"Enter the condition being tested:"/
a         t10,"1 = normal"/
a         t10,"2 = loud"/
a         t10,"3 = Lombard"/
a         t10,": ",$)
        write (6,40)
        read (5,35)cond
        icond = ichar(cond) - 48

        return
        end

```

c -----

```
subroutine getdat(ispkr,dat)
```

```
c Gets all the analysis data for a given speaker and loads it into
c a the array DAT(1,IFEATURE,ICOND)
```

```
integer ispkr,icond
real dat1(1200,20),dat2(1200,20),dat3(1200,20),dat(1200,20,3)
character*36 analysisfile
```

```
icond = 1
call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
write (6,*) analysisfile
open (unit = 2,
a      file = analysisfile,
a      status = "old",
a      form = "unformatted")
rewind (2)
read (2) dat1
close (2)
```

```
icond = 2
call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
write (6,*) analysisfile
open (unit = 2,
a      file = analysisfile,
a      status = "old",
a      form = "unformatted")
rewind (2)
read (2) dat2
close (2)
```

```
icond = 3
call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
write (6,*) analysisfile
open (unit = 2,
a      file = analysisfile,
a      status = "old",
a      form = "unformatted")
rewind (2)
read (2) dat3
close (2)
```

```
c Now transfer all data into a three-dimensional array:
```

```
write (6,*)"Building 3-d DAT array..."
do 20 i=1, 1200
  do 15 j=1, 20
    dat(i,j,1) = dat1(i,j)
    dat(i,j,2) = dat2(i,j)
    dat(i,j,3) = dat3(i,j)
15  continue
20  continue

return
end
```

```
c -----
```

```

subroutine getroot(root1,root2,root3,iroot)

character*5 root1
character*8 root2
character*9 root3
integer iroot

10      format(/"Enter root pathname to be using: "/t10,
a         "1 = ",a20/t10,
a         "2 = ",a20/t10,
a         "3 = ",a20/t30," : ",$)
20      write (6,10) root1, root2, root3
25      format (i1)
        read (5,25)iroot

        return
        end
c -----

subroutine getspkr(ispkr)

integer ispkr
cccc character*1 spkr
30      format(/"Enter number of speaker to be processed: ",$)
        write (6,30)
cccc35  format (a1)
cccc    read (5,35)spkr
        read (5,*)ispkr
cccc    ispkr = ichar(spkr) - 48
        return
        end
c -----

```

```
subroutine gettd(tdfilename,td)
```

```
c Checks to see if there is a TD file already in existence, and if so,  
c loads it
```

```
character*36 tdfilename  
real td(280,280)  
logical itshere
```

```
c      itshere = .false.  
      inquire(file=tdfilename,exist=itshere)  
      if (itshere) then  
70      format("TD data found in file ",a36)  
         write (6,70)tdfilename  
         open (unit = 3,  
a         file = tdfilename,  
a         status = "old",  
a         form = "unformatted")  
         rewind (unit=3)  
         read (3)td  
         close (3)  
      else  
75      format("No file found with name: ",a36)  
         write (6,75)tdfilename  
         end if
```

```
      return  
      end
```

```
subroutine hamm(s,hs,n)
```

```
c.....applies hamming window to s an returns windowed signal in hs  
c.....n is frame length, n<= 1024
```

```
c      integer n  
      real s(1024),hs(1024)  
  
      real omega,w  
  
c      hamming window  $w(k) = 0.54 - 0.46 \cos((2 \cdot \pi \cdot k)/(n-1))$   $k=0, \dots, n-1$   
c      omega=2.*3.141593/float(n-1)  
      do 10 k=1,n  
      kminus=k-1  
      w = 0.54 - 0.46 * cos(float(kminus) * omega)  
10 hs(k)=s(k)*w  
      return  
      end
```

```
c -----
```

program grapify8

c This routine reads the recognition performances as computed by EXP6 and
 c then writes out an ASCII files suitable for use in GRAP. The first
 c column represents the x-axis values (length of the phoneme vector),
 c the second column is the performance for normal speech, the third column
 c is the performance for loud speech, and the fourth column is the
 c performance for Lombard speech.

```

      integer icond,ispkr,i
      real p(10,3),p1(10)
      character*36 expident,datafile
      character*80 path1

10      format ("Enter experiment ident (e.g. lpc24r1): ",%)
12      format ("Enter data filename (e.g. rnn05): ",%)
15      format (a36)
      write (6,10)
      read (5,15)expident
      write (6,12)
      read (5,15)datafile

      call getspkr(ispkr)
      do 80 icond=1,3
        call recogpath(ispkr,icond,expident,datafile,path1)
        open (unit=1,file=path1,status="old",err=90)
        rewind (1)
        read (1,*)p1
        close (1)
        do 70 i=1, 10
          p(i,icond) = p1(i)
70      continue
80      continue

      open (unit=2,file="g6.data",status="unknown")
      rewind (2)
      write (6,*) "Writing to file g6.data . . ."
      do 85 i=1, 10
82      format(i2,3f7.1)
        write (2,82) i,p(i,1),p(i,2),p(i,3)
85      continue
      close (2)
      goto 100
90      write (6,95)ispkr,icond,expident,datafile
95      format("NOT FOUND: Session ",2i1," ",2a36)

100     continue

      stop
      end

```

c -----

```
subroutine initpi(pi,ic)
```

```
c This routine initializes a permutation array so routine QSORT
c will work properly.
```

```
integer i,ic
integer pi(ic)

do 10 i=1, ic
  pi(i) = i
10  continue
  return
end
```

```
c -----
```

```
integer function ipclass(irow)
```

```
c Returns the integer corresponding to the class of the phoneme
c contained in IROW. If an error occurs, then -1 will be
c returned.
```

```
integer irow, i, mymod

i = mymod(irow,40)
ipclass = -1
if (1.le.i.and.i.le.7) ipclass = 2
if (8.le.i.and.i.le.10) ipclass = 3
if (11.le.i.and.i.le.18) ipclass = 4
if (19.le.i.and.i.le.24) ipclass = 5
if (25.le.i.and.i.le.40) ipclass = 6

return
end
```

```
c -----
```

```
integer function isp()
```

```
character*1 spkr
30  format(/"Enter number of speaker to be processed: ",%)
  write (6,30)
35  format (a1)
  read (5,35)spkr
  isp = ichar(spkr) - 48
return
end
```

```
c -----
```

```

subroutine ldcondstring(string,icond)

integer icond
character*(*) string

if (icond.eq.1) then
  string = "normal"
else if (icond.eq.2) then
  string = "loud"
else if (icond.eq.3) then
  string = "Lombard"
end if
return
end
c -----

subroutine likratio(r,ra,alp,rp,rap,alpp,m,dis)

c This routine calculates the symmetrical likelihood ratio with
c unity weighting as discussed in Gray and Markel (what they call
c DM(3)) Vol ASSP-24, No 5, Oct 76, p389.

c INPUTS
c R() autocorrelation sequence of the speech data for reference
c RA() autocorrelation sequence of the lpc coefficients for reference
c ALP value of alpha for reference
c RP() autocorrelation sequence of the speech data for test
c RAP() autocorrelation sequence of the lpc coefficients for test
c ALFP value of alpha for test
c M number of LPC coefficients

c OUTPUT
c DIS symmetrical likelihood ratio as a measure of distance
c between reference and test

dimension r(1),ra(1),rp(1),rap(1)
real alp,alpp,dis,dbfac,del,delp,da,dap,q
integer m,mp,j

data dbfac/4.342944819/

fn(z) = dbfac*log(1.0+z+sqrt(z*(2.0+z)))

mp = m+1
del = r(1)*rap(1)
delp = rp(1)*ra(1)
do 90 j=2, mp
  del = del + 2.0*r(j)*rap(j)
90 delp = delp + 2.0*rp(j)*ra(j)
da = del/alp
dap = delp/alpp
q = (da + dap)/2.0 - 1.0
dis = fn(q)

return
end
c -----

```

subroutine liktemplate(ispeech,from,to,ts,lpc)

c This routine builds a template of 50 frames of the phoneme passed
 c to it. Each frame contains the information necessary to compute
 c the Likelihood Ratio in comparing two different sets of LPC
 c coefficients. The autocorrelation sequence of the speech data
 c begins at TS(1,I) of each frame I, the value of ALPHA is stored
 c in TS(60,I), and the autocorrelation sequence for the LPC
 c coefficients begins at TS(65,I).

c INPUTS

c ISPEECH is the array containing digitized speech
 c FROM is the starting position of the phoneme in seconds
 c TO is the ending position of the phoneme in seconds

c OUTPUT

c TS is the array where all the 50-frame template is stored
 c LPC is normally the array where the lpc coefficients are stored,
 c but in this case, contains nothing

```
integer*2 ispeech(160000)
real from,to,ts(128,50),lpc(40,50)
integer nlpc,nlpcpl,i,j,k,ifrom,iframes,ifs
real sframe(1024),hs(1024),b(40),bspec(513),ss,errn,rmsl
```

c Set the frame size to 256 pts and LPC coefficients to 24

```
ifs = 256
nlpc = 24
nlpcpl = nlpc + 1
```

```
iframes = 50
```

c To find the starting and ending points in the phoneme:

```
c ifrom = int(from*16000.)
c ito = int(to*16000.)
c itpts = ito - ifrom
```

c Calculate the stepsize based on the duration of the phoneme.

```
ss = (to-from-0.016)/(iframes-1)
```

```
do 100 i=1,iframes
  ifrom = (from + ss*(i-1)) * 16000
```

```
75   do 75 j=1,ifs
      sframe(j) = ispeech(ifrom+j)
      continue
```

```
      call hamm(sframe,hs,ifs)
      call myauto(hs,ifs,nlpcpl,ts(1,i))
      call lpcauto(ts(1,i),nlpc,ts(60,i),ts(65,i))
```

```
100   continue
```

```
      return
      end
```

c -----


```
subroutine listtok (iutt,itot,ilabel)
```

```
c Designed to list the token strings for a given utterance. Replaces
c the old program LISTTOK.
```

```
c INPUTS
```

```
c IUTT      utterance number
```

```
c ITOT      total number of labels in the given utterance
```

```
c ILABEL()  contains the list of phoneme labels for the utterance
```

```
c ITOK      keeps track of the number of phoneme labels printed on
c           a given line so that a new line can be inserted when
c           necessary.
```

```
integer iutt,itot,ilabel(2000),itok,j
include "ml"
```

```
8      format(i3," ",%)
```

```
9      format(a3,%)
```

```
11     format(" ")
```

```
write(6,8)iutt
```

```
itok = 1
```

```
do 100 j=1,itot
```

```
write(6,9)ml(ilabel(j))
```

```
itok = itok + 1
```

```
if (itok.ge.25) then
```

```
itok = 0
```

```
write (6,11)
```

```
end if
```

```
100    continue
```

```
write(6,11)
```

```
return
```

```
end
```

```
c -----
```

```
subroutine loadpfa(pfa,spkr)
```

```
c Reads the appropriate data file and loads it into an array
```

```
integer pfa(40,20)
character*1 spkr
character*8 pfafile

pfafile = "pfa"//spkr//".dat"
open (unit=1,file=pfafile,status="old",form="unformatted")
rewind (1)
read (1) pfa
close (1)

return
end
```

```
c -----
```

```
subroutine loadspeech(speechfile,isppeech,n)
```

```
c This routine loads the array ISPEECH with speech data that is
c contained in the file SPEECHFILE.
```

```
c INPUT
```

```
c SPEECHFILE string containing the pathname for the file where
c speech data is stored.
```

```
c OUTPUTS
```

```
c ISPEECH is the array containing digitized speech
```

```
c N is the total number of samples of digitized speech
```

```
character*36 speechfile
integer*2 ispeech(160000)
integer n,uopen,uclose,uread,ibytes,ifd,istring
```

```
istring = 36
call endstr(speechfile,istring)
ifd = uopen(speechfile,0)
ibytes = uread(ifd,isppeech,160000*2)
ifd = uclose(ifd)
```

```
c UREAD reads, in this case, 160000*2 bytes from the unit IFD
c into the buffer called ISPEECH.
c IBYTES = 0 means EOF.
c IBYTES < 0 means ERROR.
c IBYTES=160000*2 means probably did not get the whole file.
c 0 < IBYTES < 160000*2 means IBYTES=number of bytes read in.
```

```
if (ibytes.eq.0) then
  write (8,*)"Error in routine LOADSPEECH: EOF encountered."
else if (ibytes.lt.0) then
  write (8,*)"Error in routine LOADSPEECH: IBYTES < 0."
else if (ibytes.eq.160000*2) then
  write (8,*)"From routine LOADSPEECH: Speechfile too long."
  n = float(ibytes)/2
else
  n = float(ibytes)/2
end if
```

```
return
end
```

```
c -----
```

program lookatlabels

c This program is designed to view the raw labels for a set of utterances.
 c It is derived from TEMPLATES4 and therefore has the same general
 c structure and flow. This program was necessitated by the fact that
 c label files from Spire 18.3 under Genera 7.1 have some differences from
 c those written with Spire 17.5 under Release 6.0. This program will be
 c used to diagnose those differences and rewrite subroutine READLABELS as
 c necessary.

c SPEECHFILE is a file containing digitized speech
 c LISTNAME is the file containing the list of utterances
 c DATAFILE is the file where the template data will be stored
 c ANS is used to receive interactive responses from the user
 c ILABEL() contains the list of phoneme labels for an utterance
 c FROMPOS() contains the list of start times for the labels
 c TOPOS() contains the list of end times for the labels
 c IDIM is the dimension of ILABEL(), FROMPOS(), AND TOPOS()
 c IToken(K) contains the number of occurrences of phoneme K
 c ITEMP(K) contains the number of templates actually built for
 c phoneme k
 c ISHORT(K) contains the list of ascii codes (labels) of phonemes
 c of interest
 c ISES is the session number being processed
 c IUTT is the number of each utterance being processed
 c ITOT contains the total number of labels read for an utterance
 c IOCC is the max number of templates of each phoneme that
 c is sought

logical loaded
 c logical itsphere
 character*36 listname,labelfile
 c character*36 speechfile,recordfile,datafile
 c character*24 date
 integer ilabel(2000),i,itot
 integer ishort(40),iptr(126)
 integer idim,iutt,istr2
 c integer itoken(126),itemp(126),iocc,ii,iphetot,jj,j,n
 c integer*2 ispeech(160000)

real frompos(2000), topos(2000)
 c real dur, ts(128,50), lpc(40,50)
 include "ml"

c The ISHORT array preserves the order in IORDER, but only contains
 c the 40 phonemes selected in Appendix D.

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
 data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
 data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
 data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

c The IPTR array is the complement of the ISHORT array. Given the
 c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
 c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
 c means that phoneme K is not in the set of 40 phonemes.

data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
 data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
 data (iptr(i),i=81,90)/0,29,18,14,2*0,32,40,30,0/
 data (iptr(i),i=91,100)/3*0,36,2*0,27,4,28,5/
 data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/

```

data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/

data idim/2000/

c Build in the maximum flexibility for pathnames since the data
c could be scattered almost anywhere!

1   format(/"Program LOOKATLABELS..." /
a   "Enter filename for list of utterances: ",%)
    write (6,1)
    read (5,*)listname
    istr2 = 36
    call endstr(listname,istr2)
    write (6,*)"Opening file ",listname
    open (unit=2,file=listname,status="old")
    rewind (unit=2)

5   format(i3)

c **** Beginning of the loop that reads the label files for each
c   utterance in the listfile.

6   read (2,5,end=10)iutt

    loaded = .false.
    call buildpath(iutt,0,0,"lbl",labelfile)
    write (6,*)"Labelfile: ",labelfile
7   call readlabels(labelfile,ilabel,frompos,topos,itot,idim)

    do 9 i=1,itot

8       format(i3,2(2x,g14.6))
        if (ilabel(i).ge.126)
            write (6,8)ilabel(i),frompos(i),topos(i)
        end if
9       continue
        goto 6
10      continue

c **** End of loop for reading label files

    close (unit=2)

    stop
    end
c -----

```

```
subroutine lpcauto(xa,nlpc,alpha,aa)
```

```
c This routine (borrowing from Gray and Markel) is my own version
c that takes as input, the autocorrelation of the speech signal,
c XA(), and computes the autocorrelation of the LPC coefficients,
c storing them in array AA(). ALPHA is also computed, and is used
c for calculating the likelihood ratio (in a different routine).
c Reference to Gray and Markel: Vol ASSP-24, No 5
c Note that as a matter of convenience, the reflection coefficients
c are also stored in RC(), but at present, they are not made available
c outside the routine
```

```
dimension xa(1),aa(1)
real alpha,rc(60),a(60)
integer nlpc
```

```
nlpcp = nlpc + 1
```

```
a(1) = 1.
a(2) = -xa(2)/xa(1)
rc(1) = a(2)
alpha = xa(1) * (1.0 - a(2)*a(2))
```

```
do 450 j=2, nlpc
  nlpch = j/2
  jnlpc = j-1
  q = xa(j+1)
```

```
do 420 l=1, jnlpc
  lb = j+1-l
420   q = q + a(l+1)*xa(lb)
```

```
q = -q/alpha
rc(j) = q
```

```
do 430 k=1, nlpch
  kb = j-k+1
  at = a(k+1) + q*a(kb)
  a(kb) = a(kb) + q*a(k+1)
430   a(k+1) = at
```

```
a(j+1) = q
alpha = alpha*(1.0 - q*q)
```

```
450   continue
```

```
c Now calculate the autocorrelation of the LPC coefficients
```

```
call myauto(a,nlpcp,nlpcp,aa)
```

```
return
end
```

```
c -----
```

```

program lpcphon

c EXPERIMENTAL VERSION

c      Designed to do an LPC analysis of a particular phoneme
c      and produce output suitable for display with plot3d.

integer*2 speech(160000)
integer uopen,uclose,uread,bytes,ifd,ifs,ifile,n
real s(160000),frompos(100),topos(100),u
integer itpts,iframes
integer ilabel(100),i,j,k,iphone,ifrom,ito,istep,iptr
real sframe(1024),hs(1024),b(40),bspec(513)
real errn,rmsl,sec1,sec2
integer nspec,nlpc,jfile,itot,idim
character*1 label,answer
character*9 plotfile
c character*10 filename
c character*7 filen
character*9 filename
character*6 filen

character*3 char3
include "ml"

data idim/100/

2      format("/" Enter number of samples per frame: ",$)
      write (6,2)
      read (5,*)ifs

c Main re-entry point
4      continue

c6     format("/" Enter filename to open: ",$)
c      write (6,6)
c      read (5,*)ifile

6      format("/" Enter 6-char filename to open (.zb assumed): ",$)
7      format(a6)
      write (6,6)
      read (5,7)filen

c *****
      char3 = char(48+int(ifile/100))//
a      char(48+int(mod(ifile,100)/10))//char(48+mod(ifile,10))
c *****

c      filename = "12-a"//char3//".zb"
      filename = filen//".zb"
c      filen = "12-a"//char3
c      call endstr(filename,40)
      print *,filename
      ifd = uopen(filename,0)
      bytes = uread(ifd,speech,160000*2)
      ifd = uclose(ifd)

c UREAD reads, in this case, 160000*2 bytes from the standard input
c into the buffer called speech. BYTES=0 means EOF. BYTES<0 means
c ERROR. BYTES=160000*2 means probably did not get the whole file.
c 0<BYTES<160000*2 means BYTES=number of bytes read in.

      n = float(bytes)/2

```

```

        u = 0.0

c signal has no DC offset. At the same time we convert
c from integer to real format.

        do 75 i=1,n
        s(i) = speech(i)
75      continue

c Now open the label file and locate a phoneme of the type
c to be analyzed.

        call readlabels(filn,ilabel,frompos,topos,itot,idim)

130      format(/"Enter the number of the phone to analyze: ",%)
131      write (6,130)
        read (5,*)iphone

132      format(/t5,"Label String  Ascii Code")
        write (6,132)
133      format(i3,t9,a3,t23,i3)

        j = 0
        do 150 i=1,itot
            write (6,133)i,ml(ilabel(i)),ilabel(i)
            if (ilabel(i).eq.iphone) then
134          format("Is this the phone you want? ",%)
                write (6,134)
                read (5,*)answer
                if (answer.eq."y") then
                    j = i
                    i = 101
                end if
            end if
150      continue

        if (j.eq.0) then
            write (6,*) "Phone not found. "
            goto 131
        end if

152      format("Frompos = ",f7.3,"  Topos = ",f7.3)
        write (6,152)frompos(j),topos(j)

c Find the starting and ending points in the phoneme.
        ifrom = int(frompos(j)*16000.)
        ito = int(topos(j)*16000.)

c Option to override the length of the phone

153      format(/"Do you want to manually change the time interval? ",%)
        write (6,153)
        read (5,*)answer
        if (answer.eq."y") then
154          format(/" Enter starting time in seconds: ",%)
                write (6,154)
                read (5,*)sec1
                ifrom = int(sec1*16000)
155          format(/" Enter duration in seconds for analysis: ",%)
                write (6,155)
                read (5,*)sec2
                ito = int((sec1+sec2)*16000)

```

```

        end if
        itpts = ito - ifrom

156   format("/"Enter stepsize in msec: ",$)
        write (6,156)
        read (5,*)istep

        istep = istep*16
        iframes = (itpts - ifs)/istep
        write (6,*)"SELECTED PHONE: ",ml(ilabel(j))
        write (6,*)"IFRAMES = ",iframes
        write (6,*)"FRAMESIZE = ",ifs
        write (6,*)"STEP SIZE = ",istep

c Now compute the lpc spectrum of each frame, storing the
c data into PLOTDATAn where n is an integer.

        jfile = jfile + 1
        plotfile = "plotdata"//char(48+jfile)
        write (6,*)"DATA FILE = ",plotfile
        open(unit=1,status="unknown",file=plotfile)
c For a 128-pt log-magnitude plot, I am hardwiring NSPEC
        nspec = 256
c Also hardwiring the number of LPC coefficients
        nlpc = 14
c Provide for x-axis scaling if necessary
175   format ("X-axis scaling required? ",$)
        write (6,175)
        read (5,*)answer
        if (answer.eq."y".or.answer.eq."Y") call xscale(128)

        do 200 k=0,iframes-1
            iptr = ifrom + k * istep
            do 180 i=1,ifs
                sframe(i) = s(i+iptr)
180         continue

            write (6,*)"loop #",k
            call hamm(sframe,hs,ifs)
            call auto(hs,ifs,nlpc,b,errn,rmsl)
            call spect(b,nlpc,nspec,bspec)

182   format(g12.6)

c Now write the info to the output file

        do 184 i=1,nspec/2 + 1
184         write (1,182)alog(bspec(i))

200   continue

        close (unit=1)
        format("/"Analyze another phone? ",$)
210   write (6,210)
        read (5,*)answer
        if (answer.eq."y") goto 4

500   stop
        end
c -----

```



```
subroutine lpctemplate(ispeech,from,to,ts,lpc)
```

```
c This routine builds a template of 50 frames of the phoneme passed
c to it. Each frame is a 128-point vector representing the log
c magnitude of the LPC spectrum. The window size is set to 16
c milliseconds (256 points) and the stepsize is determined by the
c duration of the phoneme.
```

```
c INPUTS
```

```
c ISPEECH is the array containing digitized speech
c FROM is the starting position of the phoneme in seconds
c TO is the ending position of the phoneme in seconds
```

```
c OUTPUT
```

```
c TS is the array where all the 50-frame template is stored
c LPC is the array where the lpc coefficients are stored
```

```
integer*2 ispeech(160000)
real from,to,ts(128,50),lpc(40,50)
integer nspec,nlpc,i,j,k,ifrom,iframes,ifs
real sframe(1024),hs(1024),b(40),bspec(513),ss,errn,rmsl
```

```
c Set the frame size to 256 pts and LPC coefficients to 24
ifs = 256
nlpc = 24
```

```
c Set NSPEC to give a 128-point LPC spectrum
nspec = 256
iframes = 50
```

```
c To find the starting and ending points in the phoneme:
```

```
c ifrom = int(from*16000.)
c ito = int(to*16000.)
c itpts = ito - ifrom
```

```
c Calculate the stepsize based on the duration of the phoneme.
ss = (to-from-0.016)/(iframes-1)
```

```
do 100 i=1,iframes
ifrom = (from + ss*(i-1)) * 16000
```

```
do 75 j=1,ifs
sframe(j) = ispeech(ifrom+j)
75 continue
```

```
call hamm(sframe,hs,ifs)
call auto(hs,ifs,nlpc,b,errn,rmsl)
```

```
do 80 k=1, nlpc
lpc(k,i) = b(k)
80 continue
```

```
call spect(b,nlpc,nspec,bspec)
```

```
do 85 k=1,128
ts(k,i) = alog(bspec(k))
85 continue
100 continue
```

```
return
end
```

```
c -----
```

```
subroutine lreg(x,y,n,a1,a0,r2)
```

c Performs linear regression by the method of Least Squares on
 c a set of N points having coordinates X(I),Y(I).
 c Outputs are the attributes of the line $y = A1x + A0$, along
 c with the coefficient of determination, R2.

```
real x(1),y(1),a0,a1,r2,xs,ys,x2s,y2s,xys,c,dx,dy  
integer i,n
```

```
xs = 0.0  
ys = 0.0  
xys = 0.0  
x2s = 0.0  
y2s = 0.0
```

```
do 10 i=1, n  
  xs = xs + x(i)  
  ys = ys + y(i)  
  x2s = x2s + x(i)**2  
  y2s = y2s + y(i)**2  
  xys = xys + x(i)*y(i)  
10 continue
```

```
c = xys - xs*ys/n  
dx = x2s - xs**2/n  
dy = y2s - ys**2/n  
a1 = c/dx  
a0 = ys/n - a1*xs/n  
r2 = c**2/(dx*dy)
```

```
return  
end
```

c -----

```
program makeplot
```

```
c This routine is designed to read a template file and write an
c ASCII version of it so it can be easily accessed by QPLOT and
c PLOT3D in UNIX.
```

```

      character*36 infile
      character*12 outfile
      character*1 s1
      character*2 s2

      real ts(128,50)
      integer j,k,istring,itok,iphs

      outfile = "plotdata.out"

1      format(a1)
2      format(a2)
8      format(/"Program MAKEPLOT..."/
a      "Enter occurrence number of phoneme: ",%)
      write (6,8)
      read (5,*)itok
12     format(/"Enter the short index of desired phoneme: ",%)
      write (6,12)
      read (5,*)iphs
      call buildpath(0,itok,iphs,"pft",infile)

14     open(unit=1,file=infile,status="old",form="unformatted")
      format(/"Reading the template file: ",a36)
      write (6,14)infile
      rewind(1)
      read(1)ts
      close(unit=1)

      open(unit=1,file=outfile,status="unknown")
      do 18 k=1, 50
        do 16 j=1, 128
          write (1,*)ts(j,k)
16         continue
18         continue
        close (unit=1)

      stop
      end
```

```
c -----
```

```

      subroutine maxmin(tx,itx,txmax,txmin)
c Finds the max and min values in the template TX, where ITX is
c the vector-length of the 50-frame template.

```

```

      integer itx, i, j
      real tx(itx,50), txmin, txmax

      txmin = 1.0e20
      txmax = -1.0e20

      do 100 j=1, 50
        do 80 i=1, itx
          txmin = amin1(txmin,tx(i,j))
          txmax = amax1(txmax,tx(i,j))
60          continue
100        continue

110      format(2(2x,g12.6))
c      write (8,110)txmin,txmax

      return
      end

```

```

c -----

```

```

      subroutine myauto(x,ix,ia,a)
c This is my routine for autocorrelations only. It is meant
c to be generic in order to provide flexibility. It returns the
c IA-point autocorrelation of the first IX points of the data in
c X(). The autocorrelation is stored in array A().

```

```

      dimension x(1),a(1)
      integer ix,ia,im1

      do 100 i=1, ia
        im1 = i-1
        a(i) = 0.0

        do 50 j=1, ix-im1
          a(i) = a(i) + x(j)*x(j+im1)
50          continue

100        continue

      return
      end
      integer function mymod(i1,i2)

```

```

c This gives a remainder equal to the divisor, i2, rather than
c zero when i1 is an integer multiple of i2.

```

```

      integer i1,i2

      mymod = mod(i1,i2)
      if (mymod.eq.0) mymod = i2
      return
      end

```

```

c -----

```

```

      subroutine pfafilter2(pfain,pfaout)
c filters out the differences between normal and loud speech

      integer pfain(40,20),pfaout(40,20)
      integer i,j

      do 100 i=1, 40
        do 80 j=1, 20

          if (pfain(i,j).eq.-1.or.pfain(i,j).eq.3) then
            pfaout(i,j) = -1
          else if (pfain(i,j).eq.1.or.pfain(i,j).eq.2) then
            pfaout(i,j) = 1
          else
            pfaout(i,j) = 0
          end if

60          continue
100         continue

      return
      end
c -----

      subroutine pfafilter3(pfain,pfaout)
c filters out the differences between normal and Lombard speech

      integer pfain(40,20),pfaout(40,20)
      integer i,j

      do 100 i=1, 40
        do 80 j=1, 20

          if (pfain(i,j).eq.-1.or.pfain(i,j).eq.2) then
            pfaout(i,j) = -1
          else if (pfain(i,j).eq.1.or.pfain(i,j).eq.3) then
            pfaout(i,j) = 1
          else
            pfaout(i,j) = 0
          end if

80          continue
100         continue

      return
      end
c -----

```

program plotanaf

```

c This program reads data from an existing analysis file of a
c desired session and writes it out into an ascii-formatted file
c for examination or troff processing.

c DAT      is the formal data structure that contains the result
c           of all analyses. The second index indicates features.
c           Indices 1-10: Energy band data
c           Index 11: Center of gravity data
c           Index 12: Low band (0-3kHz) spectral tilt
c           Index 13: High band (3-8kHz) spectral tilt
c           Index 14: Pitch frequency data
c           Indices 15-17: Formants 1-3 data
c           Index 18: Duration data
c IBG      address containing the beginning address for the
c           list of individual samples of a feature for a
c           phoneme
c INOC      address for the number of occurrences for a phoneme
c IPHONE    contains the short index of the phoneme being processed
c ITCK      is used to iterate through the tokens of a given phoneme
c XBAR      address for the sample mean of a feature for a
c           phoneme
c VAR      address for the sample variance of a feature for a
c           phoneme
c SSUM      address for the sum of samples of a feature for a
c           phoneme
c S2SUM     address for the sum of squares of samples of a
c           feature for a phoneme
c ISHORT(K) contains the list of ascii codes (labels) of phonemes
c           of interest
c IPL(K)    List of phoneme indices whose data are to be retrieved
c IPLT      Total number of phonemes to be retrieved
c DBFAC     is 10/LN(10) or 4.342944819. It is the conversion factor
c           that takes natural log magnitudes and converts them into
c           units of decibels, where  $dB(A) = 10 * \text{Log}_{\text{base}10}(A)$ .
c IFEAT     is the index selecting the various analysis features
c XDB       average energy in dB
c ITC       test condition; determines the string for TESTCOND
c IBC       base condition; determines the string for BASECOND

```

```

integer i,iphone,inoc,ibg,xbar,var,ssum,s2sum,itok
integer ishort(40),ispkr,icond,ifeat,ipstart,ipstop
integer ifstart,ifstop,iftot
integer istatus,itc,ibc
real dat(1200,20,3)
real dbfac
character*1 ans,output,command*80
character*7 testcond,basecond
include "ml"

```

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

```

dbfac = 10.0/log(10.0)

```

```

1 format (a1)
c This initialization flags BUILDPATH to prompt for speaker (within
c GETDAT in this program).

```

```

ispkr = -1
5   format (i3)
10  format (/ "Program PLOTANAF..."/
a   "Enter type of output <[s]oft, [h]ard, or [f]ile>: ", $)
    write (6,10)
    read (5,1) output

    call getdat(ispkr,dat)

30  format (/ "Enter ",a4," condition index: ", $)
35  write (6,30) "test"
    read (5,*) itc
    write (6,30) "base"
    read (5,*) ibc
    call ldcondstring(testcond,itc)
    call ldcondstring(basecond,ibc)

60  format (/ "Enter beginning and ending phoneme index: ", $)
    write (6,60)
    read (5,*) ipstart,ipstop
    iptot = ipstop-ipstart+1
65  format (/ "Enter beginning and ending feature index: ", $)
    write (6,65)
    read (5,*) ifstart,ifstop
    iftot = ifstop-ifstart+1

    open (unit=1,file="x",status="unknown")
    rewind (1)
    open (unit=2,file="y",status="unknown")
    rewind (2)
    open (unit=3,file="z",status="unknown")
    rewind (3)

    do 100 iphone=ipstart, ipstop
        write (2,5) iphone

        inoc = 7*iphone - 6
        ibg = 7*iphone - 5
        xbar = 7*iphone - 4
        var = 7*iphone - 3
        ssum = 7*iphone - 2
        s2sum = 7*iphone - 1

        do 80 ifeat=ifstart, ifstop
            write (3,*) dbfac*(dat(xbar,ifeat,itc)-dat(xbar,ifeat,ibc))
80          continue
100         continue

        do 120 ifeat=ifstart, ifstop
            write (1,5) ifeat
120          continue

    close (1)
    close (2)
    close (3)

    command = "plotana " //
a           output // " " //
a           char(48 + ispkr) // " " //
a           testcond // " " //
a           basecond // " " //
a           char(48 + int(iftot/10)) //
a           char(48 + mod(iftot,10)) // " " //

```

```
a          char(48 + int(iptot/10))//  
a          char(48 + mod(iptot,10))  
          write (6,*) command  
          istatus = system(command)  
  
150        format("Another plot? (y/n): ",%)  
          write (6,150)  
          read (5,1)ans  
          if (ans.eq."y") goto 35  
  
          stop  
          end  
c -----
```



```

subroutine pranadif(ispkr,dif)

integer ishort(40),ispkr,iphone
real dif(54,40)
character newpage
include "ml"

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

newpage = char(12)

40  format(a1/t26,a17/" Phone",18i6)
55  format(a1,1x,i2,1x,a3,10f6.1,f6.0,2f6.1,4f6.0,f7.3)

      write (6,40) newpage,"Loud vs Normal", (i,i=1,18)
      do 82 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
          a      (dif(i,iphone),i=1,18)
82      continue

      write (6,40) newpage,"Lombard vs Normal", (i,i=1,18)
      do 84 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
          a      (dif(i,iphone),i=19,36)
84      continue

      write (6,40) newpage,"Lombard vs Loud", (i,i=1,18)
      do 86 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
          a      (dif(i,iphone),i=37,54)
86      continue

      return
      end

```

c -----

```

subroutine prfeat(ispkr,ifeat,dif)

integer ishort(40),ispkr,iphone,ifeat
real dif(54,40)
include "ml"

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

40  format(t26,a17/" Phone",i18)
55  format(a1,1x,i2,1x,a3,g18.6)

      write (6,40) "Loud vs Normal", ifeat
      do 82 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
a          dif(ifeat,iphone)
82    continue

      write (6,40) "Lombard vs Normal", ifeat
      do 84 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
a          dif(ifeat,iphone)
84    continue

      write (6,40) "Lombard vs Loud", ifeat
      do 86 iphone=1, 40
        write (6,55) char(48+ispkr),iphone,ml(ishort(iphone)),
a          dif(ifeat,iphone)
86    continue

      return
      end
c -----

```

```
subroutine printpfa(pfa,iftot)
```

```
c This routine takes the phoneme-feature array for a given
c speaker and prints it in a readable format. It also has
c provision for making a troff-able file.
```

```
c INPUT
```

```
c PFA Phoneme-Feature array where the following values
c have specific meaning:
c -1 = both loud and Lombard were less than normal
c 0 = no significant difference between normal,
c loud, and Lombard
c 1 = both loud and Lombard were more than normal
c 2 = loud > normal and Lombard < normal
c 3 = loud < normal and Lombard > normal
c 9 = undetermined state or error
```

```
c IFTOT total number of features being examined
```

```
c OUTPUT to STDOUT and to file
```

```
c ISHORT(K) contains the list of ascii codes (labels) of phonemes
c of interest
```

```
character*1 sym,ans
```

```
integer pfa(40,20),iftot
```

```
integer iphone,ifeat
```

```
integer ishort(40)
```

```
include "ml"
```

```
c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.
```

```
data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/
```

```
10 format(/"Subroutine PRINTPFA..."/
a "Normal of Troffable form? (n/t): ",%)
```

```
write (6,10)
```

```
11 format(a1)
```

```
read (5,11) ans
```

```
15 format(/"Feature: ",20i3)
write (6,15) (i, i=1, iftot)
```

```
20 format (a3," ",%)
```

```
21 format (a3,%)
```

```
25 format (a3,%)
```

```
26 format ("@" ,a1,%)
```

```
30 format (" ")
```

```
32 format ("_")
```

```
c Open up a supplementary file that has the results in troff-able form
```

```
c open (unit=60,file="callanova3-troff.out",status="unknown")
```

```
do 100 iphone=1, 40
```

```
if (ans.eq."n".or.ans.eq."N") then
```

```
write (6,20)ml(ishort(iphone))
```

```
else if (ans.eq."t".or.ans.eq."T") then
```

```
write (6,21)ml(ishort(iphone))
```

```

end if

do 80 ifeat=1, iftot
  if (pfa(iphone,ifeat) .eq. -1) then
    sym = "v"
  else if (pfa(iphone,ifeat) .eq. 0) then
    sym = ""
  else if (pfa(iphone,ifeat) .eq. 1) then
    sym = "."
  else if (pfa(iphone,ifeat) .eq. 2) then
    sym = "2"
  else if (pfa(iphone,ifeat) .eq. 3) then
    sym = "3"
  else if (pfa(iphone,ifeat) .eq. 9) then
    sym = "?"
  else
    sym = "*"
  end if

  if (ans.eq."n".or.ans.eq."N") then
    write (6,25)sym
  else if (ans.eq."t".or.ans.eq."T") then
    write (6,26)sym
  end if

80    continue

    write (6,30)

c Place horizontal boundaries between phoneme categories

    if (iphone.eq.6.or.iphone.eq.7.or.iphone.eq.10
      a .or.iphone.eq.18.or.iphone.eq.24) then
      if (ans.eq."t".or.ans.eq."T") write (6,32)
    end if

100    continue

c    close (60)

    return
end
c -----

```

```
subroutine printtot(ispkr,tot)
```

```
c Prints out all the merit comparisons as well as calculating the
c summation of the various differences, giving overall figures of
c merit across all 8 speakers.
```

```
c TOT(1,4) = Sum of absolute differences for normal speech
c TOT(2,4) = Sum of absolute differences for loud speech
c TOT(3,4) = Sum of absolute differences for Lombard speech
c TOT(4,4) = Sum of relative differences for loud speech
c TOT(5,4) = Sum of relative differences for Lombard speech
c TOT(6,4) = Sum of absolute differences for all speech, i.e.
c          TOT(1,4) + TOT(2,4) + TOT(3,4)
```

```
integer ispkr,icond
real tot(6,4),gtot(6,4),basedif,testdif,totdif
character string*8,spkr*1
```

```
1      format(a1)
50     format(i2,t10,f7.1,t20,f7.1,t30,f7.1,t40,f7.1)

      do 100 icond=1, 3
        gtot(icond,1) = gtot(icond,1) + tot(icond,1)
        gtot(icond,2) = gtot(icond,2) + tot(icond,2)
        gtot(6,1) = gtot(6,1) + tot(icond,1)
        gtot(6,2) = gtot(6,2) + tot(icond,2)
        tot(6,1) = tot(6,1) + tot(icond,1)
        tot(6,2) = tot(6,2) + tot(icond,2)
        tot(icond,3) = tot(icond,2)-tot(icond,1)
        if (tot(icond,2).ne.0.0.and.tot(icond,1).ne.0.0) then
          tot(icond,4) = tot(icond,4) + tot(icond,3)
          tot(6,4) = tot(6,4) + tot(icond,3)
        end if

        write (6,50) ispkr*10+icond,tot(icond,1),tot(icond,2),
a          tot(icond,3)

c        write (6,50) ispkr*10+icond,tot(icond,1),tot(icond,2),
c a          tot(icond,3),tot(icond,4)
c        end if

100    continue

150    format(a8,t10,f7.1,t20,f7.1,t30,f7.1,t40,f7.1)
155    format("Overall",t40,f7.1)
      icond = 2
      tot(4,1) = tot(icond,1) - tot(1,1)
      tot(4,2) = tot(icond,2) - tot(1,2)
      tot(4,3) = tot(4,2) - tot(4,1)
      tot(4,4) = tot(4,4) + tot(4,3)
      icond = 3
      tot(5,1) = tot(icond,1) - tot(1,1)
      tot(5,2) = tot(icond,2) - tot(1,2)
      tot(5,3) = tot(5,2) - tot(5,1)
      tot(5,4) = tot(5,4) + tot(5,3)

      spkr = char(48+ispkr)
      string = spkr//"2-"/spkr//"1"
      write (6,150)string,tot(4,1),tot(4,2),tot(4,3)
      string = spkr//"3-"/spkr//"1"
      write (6,150)string,tot(5,1),tot(5,2),tot(5,3)
      write (6,1) "-"
c    else
```

```

c      write (6,150)"Loud  ",tot(4,1),tot(4,2),tot(4,3),tot(4,4)
c      write (6,150)"Lombard",tot(5,1),tot(5,2),tot(5,3),tot(5,4)
c      write (6,155) tot(6,4)
c      end if

      if (ispkr.eq.8) then
        write (6,150)"Tot Norm",gtot(1,1),gtot(1,2),tot(1,4)
        write (6,150)"Tot Loud",gtot(2,1),gtot(2,2),tot(2,4)
        write (6,150)"Tot Lomb",gtot(3,1),gtot(3,2),tot(3,4)
        write (6,150)"Overall",gtot(6,1),gtot(6,2),tot(6,4)
        end if

      return
      end
c -----

```

```
program prntwdbls
```

```
c Experimental Version
```

```
c Designed to read the phonetic label file written by  
c EXTRACT on the LISPM and print out the phonetic transcriptions  
c of the words as defined by the word boundaries.
```

```

character*7 filen
character*14 word(210)
integer ilabel(2000), itot, idim, i, j
real frompos(2000), topos(2000)
include "ml"

data idim/2000/

3  format(/"Enter 7-character file code: ",%)
4  format(a7)
   write (6,3)
   read (5,4)filen

   call readlabels(filen, ilabel, frompos, topos, itot, idim)

   open(unit=1, file="/f/stanton/research/amrl/amrlvoc", status="old")
5  format(a14)
   do 7 i=1, 207
       read(1,5)word(i)
7  continue
   close (unit=1)

8  format(/" Output of program PRNTWDLBLS"/)
   write (6,8)
9  format(" / ", a3, %)
10 format(/a14, " ", %)

   j = 1
   do 100 i=1, itot
c    call endstr(ml(ilabel(i)), 3)
       if (ml(ilabel(i))(1:1).eq. "#") then
           write (6,10)word(j)
           j = j + 1
       else
           write (6,9) ml(ilabel(i))
       end if
100  continue

   stop
   end
c -----
```

```
subroutine putdat(ispkr,icond,dat)
```

c Writes analysis data out to the proper location

```
integer ispkr,icond
real dat(1200,20)
character*36 analysisfile

call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
write (6,*) "Writing to ",analysisfile
open (unit = 2,
a      file = analysisfile,
a      status = "unknown",
a      form = "unformatted")
rewind (2)
write (2) dat
close (2)

return
end
```

c -----

```
subroutine puttd(tdfilename,td)
```

c Saves the TD array in the appropriate file

```
character*36 tdfilename
real td(280,280)

open (unit = 3,
a      file = tdfilename,
a      status = "unknown",
a      form = "unformatted")
rewind (unit=3)
write (3)td
close (3)

return
end
```

c -----


```
subroutine qsort(a,pi,i,j)
```

```
c This routine implements the algorithm QUICKSORT, as discussed in EE608,
c Spring 1986. It is described on pp 92-96 of Aho, Hopcroft, and Ullman, "The
c Design and Analysis of Computer Algorithms", and is further summarized in my
c notes, page R3-5. In the worst case, its time complexity is  $n^2$ , but on
c average it is  $n \log_2(n)$ . In this implementation, I use a permutation array
c to order the list of real numbers in the data array A. Note that this is a
c recursive algorithm, and as such, may not be compatible with some compilers.
```

```
c Beware! This algorithm has the potential of running past the initial upper
c boundary, J. While the value in A(J+1) can be arbitrary, since it will not
c directly enter into the sort, an invalid entry in the permutation array
c (i.e.  $PI(J+1) = 0$ ) will blow it out of the water for a subscript to A()
c being out of bounds.
```

```
c INPUTS
```

```
c A() contains real numbers on which to sort
c PI() contains the permutation array. It must be initialized
c linearly for this routine to work.
c I pointer to the start of the list to be sorted
c J pointer to the end of the list to be sorted
```

```
c OUTPUT
```

```
c PI() Altered permutation array, showing the proper order
c for the elements of A().
```

```
real a(300),aa
integer i,j,lo,hi,pi(300),ii
```

```
if (i.lt.j) then
  lc = i
  hi = j+1
  aa = a(pi(i))
```

```
8   if (lo.le.hi) then
```

```
10  lo = lo + 1
    if (a(pi(lo)).lt.aa.and.lo.le.hi) goto 10
```

```
20  hi = hi - 1
    if (a(pi(hi)).gt.aa.and.lo.le.hi) goto 20
```

```
    if (lo.lt.hi) then
      ii = pi(lo)
      pi(lo) = pi(hi)
      pi(hi) = ii
    end if
  end if
```

```
    if (lo.le.hi) goto 8
    ii = pi(i)
    pi(i) = pi(hi)
    pi(hi) = ii
    call qsort(a,pi,i,hi-1)
    call qsort(a,pi,hi+1,j)
  end if
```

```
  return
end
```

```
c -----
```

program readana

- c This simple program is designed to look at individual entries into
c the common analysis data structure stored in ANALYSIS.DAT

```

character*1 ans
real dat(1200,20)
integer iphone, islot, i, ii

open (unit=1, file="analysis.dat", status="old", form="unformatted")
rewind (1)
read (1) dat
close (1)

10  format(/"Program READANA..."//
a   "Enter short index of desired phoneme: ", $)
15  write (6, 10)
    read (5, *) iphone

20  format(/"Enter value of slot to be examined: "/
a   " 1 = s2sum"/
a   " 2 = ssum"/
a   " 3 = var"/
a   " 4 = xbar"/
a   " 5 = ibg"/
a   " 6 = inoc"/
a   "      : ", $)
    write (6, 20)
    read (5, *) islot

    ii = 7*iphone - islot

30  format("Feature Value for IPHONE =", i3, " ISLOT =", i2)
    write (6, 30) iphone, islot
40  format(i4, t9, g12.6)
    write (6, 40) (i, dat(ii, i), i=1, 20)

50  format ("Another slot? ", $)
    write (6, 50)
    read (5, *) ans
    if (ans.eq."y".or.ans.eq."Y") goto 15

    stop
    end
c -----

```

program readanadat

```

c This program reads data from an existing analysis file of a
c desired session and writes it out into an ascii-formatted file
c for examination or troff processing

c DAT      is the formal data structure that contains the result
c           of all analyses. The second index indicates features.
c           Indices 1-10: Energy band data
c           Index 11: Center of gravity data
c           Index 12: Low band (0-3kHz) spectral tilt
c           Index 13: High band (3-8kHz) spectral tilt
c           Index 14: Pitch frequency data
c           Indices 15-17: Formants 1-3 data
c           Index 18: Duration data
c IBG      address containing the beginning address for the
c           list of individual samples of a feature for a
c           phoneme
c INOC      address for the number of occurrences for a phoneme
c IPHONE    contains the short index of the phoneme being processed
c ITOK      is used to iterate through the tokens of a given phoneme
c XBAR      address for the sample mean of a feature for a
c           phoneme
c VAR      address for the sample variance of a feature for a
c           phoneme
c SSUM      address for the sum of samples of a feature for a
c           phoneme
c S2SUM     address for the sum of squares of samples of a
c           feature for a phoneme
c ISHORT(K) contains the list of ascii codes (labels) of phonemes
c           of interest
c IPL(K)    List of phoneme indices whose data are to be retrieved
c IPLT      Total number of phonemes to be retrieved

```

```

integer i,iphone,inoc,ibg,xbar,var,ssum,s2sum,itok
integer ishort(40),ispkr,icond,ipl(40)
real dat(1200,20),f(40,3,3)
character*36 analysisfile
character*12 datafile
character*1 ans
include "ml"

```

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

```

data (ipl(i),i=1,10)/25,27,28,29,33,34,35,36,38,39/
iplt = 10

```

```

c This initialization flags BUILDPATH to prompt for speaker and
c condition.

```

```
ispkr = -1
```

```

1   format (a1)
10  format (/ "Program READANADAT..." )
   write (6,10)
20  format (/ "Means or Individual samples? (m/i): ", $)
   write (6,20)
   read (5,1)ans

```

```

do 200 icond=1, 3

call buildpath(0,0,0,"ana",ispkr,icond,analysisfile)
open (unit = 2,
a      file = analysisfile,
a      status = "old",
a      form = "unformatted")
rewind (2)
read (2) dat
close (2)

c50  format (/"Enter short index of phoneme to be observed: ",$)
c55  write (6,50)
c    read (5,*)iphone

do 100 i=1, iplt
    iphone = ipl(i)

    inoc = 7*iphone - 6
    ibg = 7*iphone - 5
    xbar = 7*iphone - 4
    var = 7*iphone - 3
    ssum = 7*iphone - 2
    s2sum = 7*iphone - 1

c Load the formant buffer with sample means to provide a convenient
c way to access the data for printing or troffing.

    f(i,1,icond) = dat(xbar,15)
    f(i,2,icond) = dat(xbar,16)
    f(i,3,icond) = dat(xbar,17)

c This is where data is printed on the fly. Will probably make
c this mode selectable later.

70    format(a1,a3,3f8.1)

    if (ans.eq."i".or.ans.eq."I") then
do 80 itok=dat(ibg,15), dat(ibg,15)+dat(inoc,15)-1
        write (6,70)char(48+icond),ml(ishort(iphone)),
a            dat(itok,15),
a            dat(itok,16),
a            dat(itok,17)
80    continue
    else if (ans.eq."m".or.ans.eq."M") then
        write (6,70)char(48+icond),ml(ishort(iphone)),
a            dat(xbar,15),
a            dat(xbar,16),
a            dat(xbar,17)
    end if

100    continue
200    continue

    write (6,1) " "

c Now let's print the data of F1, F2, and F3 in a different GRAP
c format.

    datafile = "f1f2f3-spkr"//char(48+ispkr)
    open (unit=1, file=datafile, status="unknown")
    rewind (1)
250    format (/"Writing to file ",a12)

```

```
write (6,250)datafile
do 320 i=1, iplt
do 300 icond=1, 3
    write (1,70)char(48+icond),ml(ishort(ipl(i))),
a      f(i,1,icond),f(i,2,icond),f(i,3,icond)
300    continue
320    continue

close (1)

stop
end
c -----
```

subroutine readfmts(formantfile,indx,buf)

c This routine loads data from ASCII files into a one-dimensional
c array.

c INPUTS

c FORMANTFILE filename for the ASCII data. Every part of the
c filename is correct except for the last character.
c INDX Single digit (0,1,2, or 3) indicating what the
c last character of FORMANTFILE should be.

c OUTPUT

c BUF One-dimensional array holding the data.

character*36 formantfile
integer indx, isize, istr
real buf(2000)

isize = 1
istr = 36

call endstr(formantfile,istr)
formantfile(istr:istr) = char(48+indx)
open (unit=99,file=formantfile,status="old")
rewind (99)

10 read (99,*,end=20)buf(isize)
isize = isize + 1
goto 10

20 close (99)
return
end

c -----

```
subroutine readlabels(labelfile,ilabel,frompos,topos,itot,idim)
```

```
c This routine retrieves labeling information from the
c specified phonetic label file written by function EXTRACT
c on the Symbolics 3670.
```

```
c INPUT
```

```
c FILEN is the basic filename where the info is stored
c IDIM is the array dimension
```

```
c OUTPUT
```

```
c ILABEL(i) contains the ascii code for label i
c FROMPOS(i) contains the starting time for label i
c TOPOS(i) contains the ending time for label i
c ITOT contains the total number of labels read
```

```
character*36 labelfile
character*1 label
integer ilabel(idim),i,itot,idim
real frompos(idim), topos(idim)
```

```
open (unit=1,file=labelfile,status="old")
rewind (unit=1)
```

```
5 format(a1,2(1x,f14.10))
```

```
i = 1
```

```
10 read (1,5,end=100)label,frompos(i),topos(i)
ilabel(i) = ichar(label)
```

```
c Patch for differences found in Spire 18.3. Appears to be
c nothing more than the 8th bit being set for some characters
c (i.e. a difference of 128) but will not incorporate the
c following commented-out code until I observe more glitches
c that follow this consistency. The active lines represent
c values > 128 that have been actually observed.
```

```
c if (ilabel(i).gt.128) ilabel(i) = ilabel(i) - 128
c if (ilabel(i).eq.136) ilabel(i) = 8
c if (ilabel(i).eq.140) ilabel(i) = 12
c if (ilabel(i).eq.141) ilabel(i) = 13
c if (ilabel(i).eq.32.and.frompos(i).eq.0.and.topos(i).eq.0.) then
c read (1,5,end=100)label,frompos(i),topos(i)
c if (ichar(label).eq.32) ilabel(i) = 10
c end if
```

```
i = i + 1
```

```
goto 10
```

```
100 continue
itot = i - 1
close (unit=1)
```

```
return
end
```

```
c -----
```

```
program readtd
```

c This routine finds a selected TD file, reads it in, then writes
 c the data to the temporary file Z in free format so that it can
 c be plotted by UNIX's PLOT3D.

```

      character*1 ans
      character*36 pathname
      integer i,j
      real td(280,280)

10      format(/"Program READTD...")
12      format(a1)
      write (6,10)

22      format (/ "Enter pathname for template: ", $)
24      format (a36)
      write (6,22)
      read (5,24)pathname
      istr = 36
      call endstr(pathname,istr)

      open (unit=1,file=pathname,status="old",form="unformatted")
      rewind (1)
      read (1)td
      close (1)

30      format("View all or just zeros? (a/z): ", $)
      write (6,30)
      read (5,12)ans

      do 40 j=1, 240
        do 35 i=1, 240
33          format(2i4,2x,g12.6)
            if (ans.eq."a".or.td(i,j).eq.0.0) write (6,33)i,j,td(i,j)
35          continue
40          continue

      open (unit=2,file="z",status="unknown")
      rewind (2)
      write (2,*)td
      close (2)

      stop
      end

```

c -----


```
subroutine readtp(tppath,tp,itx,verbose)
```

c Routine to load in the template in the file TPPATH.

```

character*36 tppath
integer itx
real tp(itx,50)
logical verbose

open (unit = 1,
a   file = tppath,
a   status = "old",
a   form = "unformatted")
rewind (1)
10  format ("Reading template: ",a36)
    if (verbose) write (6,10) tppath
    read (1)tp
    close (1)

return
end
```

c -----

```
subroutine recogpath(ispkr,icond,expident,datafile,pathname)
```

c This routine builds the pathname for the file containing recognition
c results of a particular experiment.

c INPUTS

c ISPKR speaker number
c ICOND condition number
c EXPIDENT string identifying the experiment e.g. LPC24R1, LPC24R1SW
c DATAFILE string identifying the particular data e.g. RNN05, SCD05VO

c OUTPUT

c PATHNAME complete pathname of the file

```

integer ispkr,icond,istr1,istr2
character*36 expident,datafile
character*80 pathname

istr1 = 36
istr2 = 36
call endstr(expident,istr1)
call endstr(datafile,istr2)
pathname = "/hogs/bj/plots/exp6-results/spkr-" //
a         char(48+ispkr) // "/" // expident(1:istr1) //
a         "/" // char(48+ispkr) // char(48+icond) // "1/" //
a         datafile(1:istr2)

return
end
```

c -----

```

      subroutine recover(i,icov,mincov,ipset,ipmax,ips,
a         iflg,found,j)

```

```

c This routine will attempt to find a replacement for utterance i
c that provides the same coverage of critical phones listed in
c ipset yet is shorter in length.

```

```

c INPUTS

```

```

c I is the utterance index
c ICOV is the array of accumulated coverages of each phoneme.
c MINCOV is the lower bound on the number of occurrences of
c   each phoneme.
c IPSET is the array that lists the phonemes of interest.
c IPMAX is the number of phonemes in IPSET.
c IPS is the array of utterance lengths.

```

```

c OUTPUTS

```

```

c FOUND is the logical variable indicating whether or not
c   a replacement utterance was found.
c J is the index of the replacement utterance, if found.
c   Otherwise it will have value zero.

```

```

      integer i,icov(126),mincov,ipset(40),ipmax
      integer ips(539),iflg(539),j
      integer ii,k,ysize,itcov(126)
      logical found,covers

```

```

      j = 0
      found = .false.
      ize = ips(i)

      do 100 k=1,539
        if (iflg(k).eq.0) then
          do 20 ii=1,126
            itcov(ii) = 0
100          continue
            call accumphon(k,itcov)
            covers = .true.
            do 30 ii=1,ipmax
              write(6,*)"icov",icov(ipset(ii)),"itcov",itcov(ipset(ii))
c              if ((icov(ipset(ii))+itcov(ipset(ii))).lt.mincov)
a              covers = .false.
30            continue
            if(covers.and.ips(k).lt.ize) then
              ize = ips(k)
              found = .true.
              j = k
              end if
            end if
100          continue

      return
      end
c -----

```

```
subroutine report(pname,ispkr,icond,ifwdev,icount)
```

- c Records the details of the completion of a program in the
 c EXP7PLUS series. All arguments are inputs. ISPKR and ICOND
 c are altered if they are -1 in order to print a lower case
 c "s" in the recordfile path.

```
character*36 pname,recordfile
character*24 date
integer ispkr,icond,ifwdev,icount,istr
real totaltime,tarray(2),avgt
```

```
call fdate(date)
if (ispkr.eq.-1) ispkr = 74
if (icond.eq.-1) icond = 74
istr = 36
call endstr(pname,istr)
recordfile = pname(1:istr)//
a      "-"/date(12:16)//"/"-"/
a      char(48+ispkr)//char(48+icond)
istr = 36
call endstr(recordfile,istr)
write (6,*)"Writing to ",recordfile
open (unit=1,file=recordfile,status="unknown")
rewind (1)
totaltime = etime(tarray)
avgt = totaltime/float(icount)
```

```
410 format(a36,"finished: ",a24/
a      "Speaker and Condition (session): ",2i1/
a      "Frequency deviation index: ",i3/
a      "Total number of distances calculated: ",i5/
a      "User time: ",g12.4/
a      "System time: ",g12.4/
a      "Total time: ",g12.4/
a      "Average time per calculation: ",f12.4)
```

```
write (1,410)pname,date,ispkr,icond,ifwdev,icount,
a      tarray(1),tarray(2),totaltime,avgt
close (1)
```

- c Here is another way to get the date. Also, in general, a
 c way to execute shell commands directly within Fortran.
 c istatus = system("date")

```
return
end
```

c -----

```
integer function rowstartindex(icond,i)
```

```
c This function is used within the EXP7PLUS series of programs in
c the iteration of filling the TD array. It determines where
c calculations will begin on each row (test template). It accounts
c for the fact that the normal-normal TD array is symmetrical,
c meaning that calculations start to the right of the zero block
c for this case. Otherwise, calculations must start at the beginning
c of each row. ICOND is the condition number, and I is the row being
c calculated.
```

```
integer icond,i
```

```
if (icond.eq.1) then
  rowstartindex = i+1
else
  rowstartindex = 1
end if
```

```
return
end
```

```
c -----
```

```
subroutine savenoprmt(x,ix,nx,filename)
```

```
c This subroutine will store a vector of values in an
c ASCII formatted file. The file is named by the calling
c program rather than prompting the user.
```

```
c X      = Array of values
c IX     = Number of values to be stored
c NX     = Dimension of array X
c FILENAME = String containing filename
```

```
integer ix,j,nx
real x(nx)
character*20 filename
```

```
open(unit=1,file=filename,status="new")
20  format(f12.4)
    do 50,j=1,ix
      write (1,20)x(j)
50  continue
close(unit=1)
```

```
return
end
```

```
c -----
```

program selutt

c Designed to select a minimal subset of utterances that will
c give the required coverage for a desired set of phonemes.

```
integer i,j,k,mincov,itot
integer icov(126),iflg(539),ipset(40),ip1(40)
integer ipud(15000),ipun(15000),ips(539)
integer iptmp(40),iptmax
logical uncovered,found,altered
common /ip/ipud,ipun
include "ml"

data (ip1(i),i=1,10)/71,83,103,67,121,76,70,74,99,82/
data (ipset(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ipset(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ipset(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ipset(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/
```

c First build the phoneme linked lists for each of the 539
c utterances.

```
2      format(/"Program SELUTT..."/
a      "Enter the desired minimum coverage per phoneme: ",$)
      write (6,2)
      read (5,*)mincov
```

```
5      format(/"Selected Utt  Delta  Utt Size")
      write (6,5)
```

c Temporarily skip over the number crunching since
c I am not changing this part of the code.

```
      call uttphon(ipud,ipun,ips)
c      open (unit=1,file="ip.dat",status="unknown")
c      rewind (1)
c      write (1,*) ipud,ipun,ips
c      read (1,*) ipud,ipun,ips
c      close (unit=1)
c      goto 100
```

c Search iteratively for utterances having the minimum
c excess phones first using the scarce phoneme set then
c the complete phoneme set.

```
c      call covminex(icov,mincov,ip1,10,iflg,ips)
c      call covminex(icov,mincov,ipset,40,iflg,ips)
```

c -----
c Search iteratively for utterances having the maximum
c delta coverage and the minimum total phones.
c Use the scarce phone set first then the complete set

```
      call covmaxdel(icov,mincov,ip1,10,iflg,ips)
      call covmaxdel(icov,mincov,ipset,40,iflg,ips)
```

c Now to store some results to shorten run time.

```
c100  open(unit=1,file="icovetc.dat",status="unknown")
c      rewind (1)
c      write (1,*)icov,iflg
c      read (1,*)icov,iflg
c      close(unit=1)
```

c This is the final section that iterates through the
c chosen list in order to try and find replacement utts.

```
495  altered = .false.
      do 500 i=1,539
```

```

      if (iflg(i).eq.1) then
        call uncovtest(i,icov,mincov,ipset,40,uncovered,
a       iptmp,iptmax)
        write (6,450)i
        if (uncovered) then
          write(6,460)(ml(iptmp(k)),k=1,iptmax)
          call deaccum(i,icov)
          call recover(i,icov,mincov,iptmp,iptmax,ips,
a       iflg,found,j)
          if (found) then
            altered = .true.
            write(6,470)i,j,ips(i)-ips(j)
            iflg(i) = 0
            iflg(j) = 1
            call accumphon(j,icov)
          else
            call accumphon(i,icov)
            end if
          else
            altered = .true.
            call deaccum(i,icov)
            iflg(i) = 0
            write(6,455)
            end if
          end if
500    continue
      if (altered) goto 495

c Print out the final results
260    format(/"List of selected utterances"/)
270    format(i4,$)
275    format("")
305    format(/"Total utterances selected: ",i3)
310    format(/"Phoneme Coverage")
320    format(a3,t11,i3)
405    format(/"Total number of phones: ",i4/)
450    format("Eliminating ",i3," uncovers ",$)
455    format("nothing!!!")
460    format(10(a4))
470    format("Utt ",i3," replacable by Utt ",i3," saving ",i3)
      itot = 0
      write (6,310)
      do 400 j=1,40
        itot = itot + icov(ipset(j))
        write (6,320)ml(ipset(j)),icov(ipset(j))
400    continue
      write (6,405)itot

      itot = 0
      write (6,260)
      do 300, k=1,539
        if (iflg(k).eq.1) then
          itot = itot + 1
          write (6,270)k
          if (mod(itot,20).eq.0) write (6,275)
        end if
300    continue
      write (6,305)itot

      stop
      end
c -----

```

real function slope(tx,ix,jx,itx)

- c Calculates the slope of the curve at point LX,LY in the template
- c TX. ITX is the vector length of the 50-frame template.

integer ix,jx,itx
real tx(itx,50)

- c Take care of the endpoints of the curve and the peaks by assigning
- c a slope of zero for these cases.

```

      if (ix.eq.1.or.ix.eq.itx) then
        slope = 0.0
      else if (tx(ix,jx).gt.tx(ix-1,jx).and.
a         tx(ix,jx).gt.tx(ix+1,jx)) then
        slope = 0.0
      else
        slope = tx(ix+1,jx) - tx(ix-1,jx)
      end if

```

return
end

c -----

program slopedifcom

c combines data from program SLOPEDIFHIS, normalizes it, and
 c puts it into a form that GRAP can understand. In the output,
 c the first column is the x axis scale, the second column is the
 c normalized histogram (distribution) for slopes of the same
 c phoneme, the third column is the normalized histogram for the
 c slopes of different phonemes, and the fourth column is the
 c normalized histogram of both same and different phonemes combined.

```

      real smhs(0:400),smhd(0:400),smht(0:400)
      real ts,td,tt
      integer i

      open (unit=1,file="y-difhis-s",status="old")
      rewind (1)
      read (1,*) smhs
      close (1)
      open (unit=1,file="y-difhis-d",status="old")
      rewind (1)
      read (1,*) smhd
      close (1)

      do 10 i=0, 400
        smht(i) = smhs(i) + smhd(i)
        ts = ts + smhs(i)
        td = td + smhd(i)
10      continue

      tt = ts + td

      do 20 i=0, 400
        smhs(i) = smhs(i)/ts
        smhd(i) = smhd(i)/td
        smht(i) = smht(i)/tt
20      continue

      open (unit=1,file="y-difhis.g",status="unknown")
      rewind (1)
25      format(4f8.4)
      do 30 i=0, 400
        write (1,25) i*0.02, smhs(i), smhd(i), smht(i)
30      continue

      close (1)

      stop
      end
c -----

```


program slopedifhis

c Similar to SLOPEHIS except calculates histogram of slope differences
c that occur during a recognition experiment.

c SMHS collects data from like phonemes
c SMHD collects data from different phonemes

```
real smhs(0:400),smhd(0:400)
integer iutt,itok
```

```
logical massfill,saveit,needtesttemp
character*36 file1,file2,tdfilename,pname
integer i,j,k,l,jj,ll,it,ip,itd,iss(40),r,itx,ifwdev
integer iocc,iphone,iroot,ii,istr,ispkr,icond,itt,icount
integer rowstartindex
real d(-2:50,-2:50)
real txs(128,50),tys(128,50)
```

c Caution: the ISS array must have IP entries. This is a pointer
c array that allows subsets of the 40 phonemes to be easily accessed.
c The way it is now initialized, it is transparent since all 40
c phonemes are being tested. It is used in the program in the
c calls to BUILDPATH where phoneme templates are sought.

```
data (iss(i),i=1,15)/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/
data (iss(i),i=16,27)/16,17,18,19,20,21,22,23,24,25,26,27/
data (iss(i),i=28,40)/28,29,30,31,32,33,34,35,36,37,38,39,40/
```

```
icount = 0
it = 6
ip = 40
itd = it*ip
itx = 128
iocc = 1
iphone = 1
saveit = .false.
pname = "SLOPEDIFHIS"
threshold = 0.25
```

```
call usrinit(pname,it,ip,ifwdev,ispkr,icond,massfill,tdfilename)
```

c Now comes the nested iterations that will clash the templates together.
c Either normal, loud, or Lombard will be used as test while the IT normal
c occurrence sets will be used as reference.

```
if (icond.eq.1) then
  itt = it - 1
else
  itt = it
end if
```

```
do 300 i=iocc, itt
```

```
  do 280 j=iphone, ip
    needtesttemp = .true.
    call buildpath(0,i,iss(j),"pft",ispkr,icond,file1)
```

```
  do 260 k=rowstartindex(icond,i), it
    if (k.eq.i.and..not.massfill) goto 260
```

```
  do 240 l=1, ip
    jj = j + ip*(i-1)
    ll = l + ip*(k-1)
```

```

        if (needtesttemp) then
            call readtp(file1,txs,itx,.true.)
            needtesttemp = .false.
        end if
        call buildpath(0,k,iss(l),"pfr",ispkr,icond,file2)
        call readtp(file2,tys,itx,.true.)

c   at this point, txs and tys are now available.

        do 70 iframe=1, 50
            do 60 js=1, 128
                dif = abs(slope(txs,js,iframe,128) -
a                slope(tys,js,iframe,128))
                kh = int(dif/0.02 + 0.5)
                if (iss(j).eq.1) then
                    smhs(kh) = smhs(kh) + 1.0
                else
                    smhd(kh) = smhd(kh) + 1.0
                end if
60            continue
70        continue

            icount = icount + 1

240        continue
260        continue

        open (unit=2,file="y-difhis-s",status="unknown")
        rewind (2)
        write (2,*) smhs
        close (2)
        open (unit=2,file="y-difhis-d",status="unknown")
        rewind (2)
        write (2,*) smhd
        close (2)

280        continue

300        continue

        call report(pname,ispkr,icond,ifwdev,icount)

        stop
        end
c -----

```

program slopehis

c Compiles a histogram of slope magnitude, |S1|, for a given
c speaker

```
real smh(0:200),tx(128,50)
integer iuttt,itok,iphone,ispkr,icond
character*36 pathname
```

c Have BUILDPATH prompt for a speaker
ispkr = -1

```
do 100 icond=1, 3
  do 90 itok=1, 6
    do 80 iphone=1, 40
```

```
      call buildpath(0,itok,iphone,"pft",ispkr,icond,pathname)
```

```
      write (6,*)"Reading ",pathname
```

```
      open (unit=1,
```

```
a          file=pathname,
```

```
a          status="old",
```

```
a          form="unformatted")
```

```
      rewind (1)
```

```
      read (1) tx
```

```
      close (1)
```

```
      do 70 iframe=1, 50
```

```
        do 60 j=1, 128
```

```
          k = int(abs(slope(tx,j,iframe,128))/0.02 + 0.5)
```

```
          smh(k) = smh(k) + 1.0
```

```
60          continue
```

```
70          continue
```

```
80          continue
```

```
90          continue
```

```
100         continue
```

```
      open (unit=2,file="y-his",status="unknown")
```

```
      rewind (2)
```

```
      write (2,*) smh
```

```
      close (2)
```

```
      open (unit=2,file="y.g",status="unknown")
```

```
      rewind (2)
```

```
      do 200 i=0, 200
```

```
195      format (f8.2,i10)
```

```
      write (2,195) i*0.02, int(smh(i))
```

```
200      continue
```

```
      close (2)
```

```
      stop
```

```
      end
```

c -----

subroutine slopeify(ts)

c This routine takes any template array and computes the slope of the
c data on a frame-by-frame basis. The results are then passed back
c in the same array, thus destroying the original data.

c Note, that as written, there is no need for the full size temporary
c array, TSS; a single dimension TSS(128) is all that is necessary.
c For the time being, it will be left this way in order to facilitate
c the writing of a non-destructive version of this routine if it
c becomes necessary.

real ts(128,50),tss(128,50)
integer i,j

c First, zero out the slope endpoints of each frame

```

do 50 j=1, 50
  tss(1,j) = 0.0
  tss(128,j) = 0.0
50  continue

```

c Now compute the slope for points 2 to 127 of each frame

```

do 100 j=1, 50
  do 80 i=3, 128
    if (ts(i-1,j).gt.ts(i-2,j).and.
a      ts(i-1,j).gt.ts(i,j)) then
      tss(i-1,j) = 0.0
    else
      tss(i-1,j) = (ts(i,j)-ts(i-2,j))/125.0
    end if
80  continue

```

c Replace the original frame with slope data

```

do 90 i=1, 128
  ts(i,j) = tss(i,j)
90  continue

100 continue

```

return
end

c -----

```

      real function slpwt(tx,ix,jx,ty,iy,jy,itx)

c  Calculates the weighting function to be assigned for finding the
c  distance between the points of two curves, designated at TX(DX,JX)
c  and TY(IY,JY). ITX is the length of the vectors in the 50-frame
c  templates

      integer ix,jx,iy,jy,itx
      real tx(itx,50),ty(itx,50),slope,threshold
      common /knee/threshold
cccc  threshold = 2.0

      slpwt = abs(slope(tx,ix,jx,itx)-slope(ty,iy,jy,itx))

      if (slpwt.le.threshold) then
        slpwt = slpwt/threshold
      else
        slpwt = 1.0
      end if

      return
    end
c -----

```

```

subroutine spect(b,m,n,bspec)

c.....compute (n/2)+1 samples (from 0 to pi) in the spectrum
c      |1/d|, where d is the all-pole filter
c      1+b(1)z**-1 + b(2)z**-2 +...+b(m)z**-m
c      bspec(1..n/2+1) contains the resulting spectrum samples
c
integer m,n
real b(40),bspec(513)
complex zinv,p,dinv
real f,fscale,pi

c
pi = 3.14159265358979
fscale = 2.0*pi/float(n)
lim=n/2+1
do 10 k=1,lim
f= float(k-1)*fscale
zinv= cexp(cmplx(0.,-f))
dinv= cmplx(1.,0.)
p= dinv
do 20 i= 1,m
p= p*zinv
20 dinv= dinv+b(i)*p
bspec(k)= 1./cabs(dinv)
10 continue
return
end
c -----

```

```
subroutine tddpath(ispkr,icond,path)
```

```
c Builds the pathname (local directory) for the file used to store
c a TD array. Must have the speaker number ISPKR and the condition
c number ICOND. Output is the string PATH.
```

```
integer ispkr,icond,istr
character*36 chid,path
```

```
10 format(/"Enter unique ident string for TD array file: ",%)
   write (6,10)
12 format(a36)
   read (5,12)chid
   istr = 36
   call endstr(chid,istr)
   path = "td"//char(48+ispkr)//char(48+icond)//
a      "1"//chid(1:istr)//".dat"

   return
   end
```

```
c -----
```

```
c Code used to check the subroutine
```

```
c program testtdpath
c character*36 testpath
c integer ispkr,icond
c ispkr = 4
c icond = 3
c call tddpath(ispkr,icond,testpath)
c10 format(/"Testpath is: ",a36)
c write (6,10)testpath
c stop
c end
c subroutine tdstatus(td,it,ip,itd,iocc,iphone)
```

```
c Checks to see whether or not the TD is full. If not, it returns
c the altered values of IOCC and IPHONE so processing can begin at
c that point.
```

```
real td(280,280)
integer it,ip,itd,iocc,iphone,jj,kkk
logical arrayfull
```

```
arrayfull = .true.
```

```
c Note that starting KKK out at the far right of the array
c and then switching it over to the left side for the last
c block of rows effectively avoids any of the zero blocks
c in the array
```

```
kkk = itd
```

```
do 100 jj=1, itd
  if (jj.gt.(it-1)*ip) kkk = 1
  if (td(jj,kkk).eq.0.0) then
    arrayfull = .false.
    iocc = int((jj-1)/ip) + 1
    iphone = mymod(jj,ip)
```

```
90 format(/"Row ",i3," is incomplete. ",
a      "Row indices set to the following values:"/
a      "IOCC = ",i3/
a      "IPHONE = ",i3/)
   write (6,90)jj,iocc,iphone
```

```
        jj = itd
        end if
100    continue
    if (arrayfull) then
        write (6,*)"TD Array is full. Program terminating."
        stop
    end if

    return
end
c -----
```


program templates2

```

c This program is designed to build a set of templates for the
c 40-phoneme set of interest. It will work through the
c utterance list that has been designated, and build up to K
c templates of each phoneme, or the maximum number of occurrences
c of a given phoneme, whichever is less. The templates
c will be stored in a two-dimensional arrays and written to the files
c root-path/xx/skknnu.dat where xx is the session number, kk is
c the occurrence number, and nn is the short index number of the phoneme.
c ('s' denotes a single template, and 'u' denotes unformatted.)

c TEMPLATES2 means that LPC24 templates are built that contain a 128-pt
c log-magnitude spectrum vector for each of the 50 frames.

c SLOPEFLAG determines whether or not the subroutine SLOPEIFY
c is run on the templates built
c SPEECHFILE is a file containing digitized speech
c LISTNAME is the file containing the list of utterances
c DATAFILE is the file where the template data will be stored
c ANS is used to receive interactive responses from the user
c ILABEL() contains the list of phoneme labels for an utterance
c FROMPOS() contains the list of start times for the labels
c TOPOS() contains the list of end times for the labels
c IDIM is the dimension of ILABEL(), FROMPOS(), AND TOPOS()
c ITOKEN(K) contains the number of occurrences of phoneme K
c ITEMP(K) contains the number of templates actually built for
c phoneme k
c ISHORT(K) contains the list of ascii codes (labels) of phonemes
c of interest
c ISES is the session number being processed
c IUTT is the number of each utterance being processed
c ITOT contains the total number of labels read for an utterance
c IOCC is the max number of templates of each phoneme that
c is sought

```

```

logical itshere,loaded,hadtocompute
character*1 slopeflag
character*36 speechfile,listname,labelfile,datafile,recordfile
character*24 date
integer ilabel(2000),itoken(126),itemp(126),i,itot,n
integer ishort(40),iptr(126),ispkr,icond
integer idim,j,iutt,istr,iocc,ii,iphetot,jj
integer*2 ispeech(160000)

```

```

real frompos(2000),topos(2000),ts(128,50)
real lpc(40,50)
real dur
real totaltime,tarray(2)
include "ml"

```

```

c The ISHORT array preserves the order in IORDER, but only contains
c the 40 phonemes selected in Appendix D.

```

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

```

c The IPTR array is the complement of the ISHORT array. Given the
c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)

```

c means that phoneme K is not in the set of 40 phonemes.

```
data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i=81,90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i=91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,8,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/
```

```
data idim/2000/
```

c This initialization flags BUILDPATH to prompt for speaker and condition.

```
ispkr = -1
icond = -1
slopeflag = "n"
hadtocompute = .false.
```

c Build in the maximum flexibility for pathnames since the data could be scattered almost anywhere!

```
1      format(/"Program TEMPLATES2 using LPC24..." /
a      "Enter filename for list of utterances: ", $)
      write (6,1)
2      format (a36)
      read (5,2)listname
      istr = 36
      call endstr(listname,istr)
      write (6,*)"Opening file ",listname
      open (unit=2,file=listname,status="old")
      rewind (unit=2)

3      format (a1)
4      format(/"Enter the max number of templates desired per phone: ", $)
      write (6,4)
      read (5,*)ioccc

41     format (/ "Do you want templates SLOPEIFIED? (y/n): ", $)
      write (6,41)
      read (5,3)slopeflag
5      format(i3)
```

c **** Beginning of the loop that reads the label files for each utterance in the listfile.

```
6      read (2,5,end=10)iutt

      loaded = .false.
      call buildpath(iutt,0,0,"wav",ispkr,icond,speechfile)
      call buildpath(iutt,0,0,"lbl",ispkr,icond,labelfile)
      write (6,*)"Speechfile: ",speechfile
      write (6,*)"Labelfile: ",labelfile
54     format("Tok IPTR Sum")
      write (6,54)
7      call readlabels(labelfile,ilabel,frompos,topos,itot,idim)
```

c Now sift through the tokens of this utterance, making templates of the ones that qualify.

```
do 100 j=1,itot
  dur = topos(j) - frompos(j)
```

c If the token is at least 16 milliseconds long, then count it as valid

```

        if (dur.ge.0.016) then
            itoken(ilabel(j)) = itoken(ilabel(j)) + 1
c Now if this is one of the 40 phonemes and if it is the Kth occurrence
c that we seek, then process it and add it to the total number of
c templates built.

55      format(i3,2i5)
        write (6,5)ilabel(j),iptr(ilabel(j)),iphonetot+1

        if (iptr(ilabel(j)).ne.0.and.itoken(ilabel(j)).le.iocc) then
            itshere = .false.
            itemp(ilabel(j)) = itemp(ilabel(j)) + 1
            iphonetot = iphonetot + 1
            ii = iptr(ilabel(j))
            jj = itoken(ilabel(j))
            call buildpath(iutt,jj,ii,"pft",ispkr,icond,datafile)
            inquire(file=datafile,exist=itshere)

            if (.not.itshere) then
                hadtocompute = .true.
                write (6,*)"Building template for ",ml(ilabel(j))
                if (.not.loaded) then
                    call loadspeech(speechfile,ispkch,n)
                    loaded = .true.
                end if

                call lpctemplate(ispeech,frompos(j),topos(j),ts,lpc)
                if (slopeflag.eq."y".or.slopeflag.eq."Y")
                    call slopeify(ts)
a
                write (6,*)"Writing datafile: ",datafile
                open  (unit = 3,
a                  file = datafile,
a                  status = "new",
a                  form  = "unformatted")
                write (3)ts
                close (unit=3)

c Writing LPC coefficients into parameter files has been disabled because
c I currently have no need to do so. It can be brought back later if
c necessary...
c      call buildpath(iutt,jj,ii,"lpc",ispkr,icond,datafile)
c      open  (unit = 3,
c      a      file = datafile,
c      a      status = "unknown",
c      a      form  = "unformatted")
c      rewind (3)
c      write (3)lpc
c      close (unit=3)

            end if

            end if

            end if

100      continue

            if (iphonetot.lt.(40*iocc)) goto 6

10      continue

```

```

c **** End of loop for reading label files

      close (unit=2)

      if (hadtocompute) then
        call fdate(date)
c      recordfile = "templates2-record"//date(12:19)
        recordfile = "tplts2-"//date(12:18)//"-"//
a      char(48+ispkr)//char(48+icond)
        open (unit=1,file=recordfile,status="unknown")
        rewind (1)

110      format("TEMPLATES2 using LPC24 finished at: ",a24/
a      "Total number of templates built: ",i4/
a      "Speaker and Condition: ",2i1/
a      "Slopeify status: ",a1//
a      "Phoneme Number of Templates"/)
        write (1,110)date,iphonetot,ispkr,icond,slopeflag
        totaltime = etime(tarray)
115      format("/User time: ",g12.3/
a      "System time: ",g12.3/
a      "Total time: ",g12.3)
        write (1,115)tarray(1),tarray(2),totaltime

120      format(a3,t17,i3)

        do 140 i=1, 40
          write (1,120)ml(ishort(i)),itemp(ishort(i))
140          continue

        close (1)
        end if

        stop
        end
c -----

```

program templatescep

c (Derived from TEMPLATES2)
 c This program is designed to build a set of templates for the
 c 40-phoneme set of interest. It will work through the
 c utterance list that has been designated, and build up to K
 c templates of each phoneme, or the maximum number of occurrences
 c of a given phoneme, whichever is less. The templates
 c will be stored in a two-dimensional arrays and written to the files
 c root-path/xx/skknnu.dat where xx is the session number, kk is
 c the occurrence number, and nn is the short index number of the phoneme.
 c ('s' denotes a single template, and 'u' denotes unformatted.)

c TEMPLATESCEP means that LPC24 vectors are derived from the speech
 c data and then 24-point cepstral coefficient vectors are calculated
 c for each of the 50 frames.

c SPEECHFILE is a file containing digitized speech
 c LISTNAME is the file containing the list of utterances
 c DATAFILE is the file where the template data will be stored
 c ANS is used to receive interactive responses from the user
 c ILABEL() contains the list of phoneme labels for an utterance
 c FROMPOS() contains the list of start times for the labels
 c TOPOS() contains the list of end times for the labels
 c IDIM is the dimension of ILABEL(), FROMPOS(), AND TOPOS()
 c ITOKEN(K) contains the number of occurrences of phoneme K
 c ITEMP(K) contains the number of templates actually built for
 c phoneme k
 c ISHORT(K) contains the list of ascii codes (labels) of phonemes
 c of interest
 c ISES is the session number being processed
 c IUTT is the number of each utterance being processed
 c ITOT contains the total number of labels read for an utterance
 c IOCC is the max number of templates of each phoneme that
 c is sought

```

logical itshere,loaded,hadtocompute
character*36 speechfile,listname,labelfile,datafile,recordfile
character*24 date
integer ilabel(2000),itoken(126),itemp(126),i,itot,n
integer ishort(40),iptr(126),ispkr,icond
integer idim,j,iutt,istr,iocc,ii,iphetot,jj
integer*2 ispeech(160000)

```

```

real frompos(2000), topos(2000), ts(40,50)
real lpc(40,50)
real dur
real totaltime,tarray(2)
include "ml"

```

c The ISHORT array preserves the order in IORDER, but only contains
 c the 40 phonemes selected in Apper.dix D.

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,67,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,64,94,79,105,111,88/

```

c The IPTR array is the complement of the ISHORT array. Given the
 c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
 c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
 c means that phoneme K is not in the set of 40 phonemes.

```

data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i=81,90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i=91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/

```

```

data idim/2000/

```

c This initialization flags BUILDPATH to prompt for speaker and condition.

```

ispkr = -1
icond = -1
hadtocompute = .false.

```

c Build in the maximum flexibility for pathnames since the data could be scattered almost anywhere!

```

1      format(/"Program TEMPLATESCEP using LPC24 for 24 Cepstral Coefs..."/
a      "Enter filename for list of utterances: ",%)
      write (6,1)
2      format (a36)
      read (5,2)listname
      istr = 36
      call endstr(listname,istr)
      write (6,*)"Opening file ",listname
      open (unit=2,file=listname,status="old")
      rewind (unit=2)

3      format (a1)
4      format(/"Enter the max number of templates desired per phone: ",%)
      write (6,4)
      read (5,*)iocc

5      format(i3)

```

c **** Beginning of the loop that reads the label files for each utterance in the listfile.

```

6      read (2,5,end=10)iutt

      loaded = .false.
      call buildpath(iutt,0,0,"wav",ispkr,icond,speechfile)
      call buildpath(iutt,0,0,"lbl",ispkr,icond,labelfile)
      write (6,*)"Speechfile: ",speechfile
      write (6,*)"Labelfile: ",labelfile
54     format("Tok IPTR Sum")
      write (6,54)
7      call readlabels(labelfile,ilabel,frompos,topos,itot,idim)

```

c Now sift through the tokens of this utterance, making templates of the ones that qualify.

```

      do 100 j=1,itot
        dur = topos(j) - frompos(j)

```

c If the token is at least 16 milliseconds long, then count it as valid

```

      if (dur.ge.0.016) then
        itoken(ilabel(j)) = itoken(ilabel(j)) + 1

```

c Now if this is one of the 40 phonemes and if it is the Kth occurrence of that we seek, then process it and add it to the total number of

c templates built.

```

55      format(i3,2i5)
      write (6,55)ilabel(j),iptr(ilabel(j)),iphonetot+1

      if (iptr(ilabel(j)).ne.0.and.itoken(ilabel(j)).le.iocc) then
        itshere = .false.
        itemp(ilabel(j)) = itemp(ilabel(j)) + 1
        iphonetot = iphonetot + 1
        ii = iptr(ilabel(j))
        jj = itoken(ilabel(j))
        call buildpath(iutt,jj,ii,"pft",ispkr,icond,datafile)
        inquire(file=datafile,exist=itshere)

        if (.not.itshere) then
          hadtocompute = .true.
          write (6,*)"Building template for ",ml(ilabel(j))
          if (.not.loaded) then
            call loadspeech(speechfile,ispkch,n)
            loaded = .true.
          end if

          call ceptemplate(ispkch,frompos(j),topos(j),ts,lpc)

          write (6,*)"Writing datafile: ",datafile
          open      (unit = 3,
a                file = datafile,
a                status = "new",
a                form  = "unformatted")
          write (3)ts
          close (unit=3)

c      Writing LPC coefficients into parameter files has been disabled because
c      I currently have no need to do so. It can be brought back later if
c      necessary...
c      call buildpath(iutt,jj,ii,"lpc",ispkr,icond,datafile)
c      open      (unit = 3,
c      a        file = datafile,
c      a        status = "unknown",
c      a        form  = "unformatted")
c      rewind (3)
c      write (3)lpc
c      close (unit=3)

          end if

        end if

      end if

100    continue

      if (iphonetot.lt.(40*iocc)) goto 6

10    continue

c **** End of loop for reading label files

      close (unit=2)

      if (hadtocompute) then
        call fdate(date)
        recordfile = "tpltscep-//date(12:16)//"-"/"//

```

```

a      char(48+ispkr)//char(48+icond)
      open (unit=1,file=recordfile,status="unknown")
      rewind (1)

110    format("TEMPLATESCET LPC24, 24 Ceps finished at: ",a24/
a      "Total number of templates built: ",i4/
a      "Speaker and Condition: ",2i1//
a      "Phoneme Number of Templates"/)
      write (1,110)date,iphonetot,ispkr,icond
      totaltime = etime(tarray)
115    format("/User time:",g12.3/
a      "System time:",g12.3/
a      "Total time:",g12.3)
      write (1,115)tarray(1),tarray(2),totaltime

120    format(a3,t17,i3)

      do 140 i=1, 40
      write (1,120)ml(ishort(i)),itemp(ishort(i))
140    continue

      close (1)
      end if

      stop
      end
c -----

```


program templateslik

c This program is designed to build a set of templates for the
 c 40-phoneme set of interest. It will work through the
 c utterance list that has been designated, and build up to K
 c templates of each phoneme, or the maximum number of occurrences
 c of a given phoneme, whichever is less. The templates
 c will be stored in a two-dimensional arrays and written to the files
 c root-path/xx/skknnu.dat where xx is the session number, kk is
 c the occurrence number, and nn is the short index number of the phoneme.
 c ('s' denotes a single template, and 'u' denotes unformatted.)

c TEMPLATES2 means that LPC24 templates are built that contain a 128-pt
 c log-magnitude spectrum vector for each of the 50 frames.

c SLOPEFLAG determines whether or not the subroutine SLOPEIFY
 c is run on the templates built

c SPEECHFILE is a file containing digitized speech

c LISTNAME is the file containing the list of utterances

c DATAFILE is the file where the template data will be stored

c ANS is used to receive interactive responses from the user

c ILABEL() contains the list of phoneme labels for an utterance

c FROMPOS() contains the list of start times for the labels

c TOPOS() contains the list of end times for the labels

c IDIM is the dimension of ILABEL(), FROMPOS(), AND TOPOS()

c ITOKEN(K) contains the number of occurrences of phoneme K

c ITEMP(K) contains the number of templates actually built for
 c phoneme k

c ISHORT(K) contains the list of ascii codes (labels) of phonemes
 c of interest

c ISES is the session number being processed

c IUTT is the number of each utterance being processed

c ITOT contains the total number of labels read for an utterance

c IOCC is the max number of templates of each phoneme that
 c is sought

```

logical itshere,loaded,hadtocompute
character*36 speechfile,listname,labelfile,datafile,recordfile
character*24 date
integer ilabel(2000),itoken(126),itemp(126),i,itot,n
integer ishort(40),iptr(126),ispkr,icond
integer idim,j,iutt,istr,iocc,ii,iphonetot,jj
integer*2 ispeech(160000)

```

```

real frompos(2000), topos(2000), ts(128,50)
real lpc(40,50)
real dur
real totaltime,tarray(2)
include "ml"

```

c The ISHORT array preserves the order in IORDER, but only contains
 c the 40 phonemes selected in Appendix D.

```

data (ishort(i),i=1,10)/112,116,107,98,100,103,70,109,110,71/
data (ishort(i),i=11,20)/115,122,87,84,102,83,74,118,108,114/
data (ishort(i),i=21,30)/121,104,76,119,69,99,97,117,82,89/
data (ishort(i),i=31,40)/101,87,120,73,84,94,79,105,111,88/

```

c The IPTR array is the complement of the ISHORT array. Given the
 c SPIRE ascii label of a phoneme, K, IPTR(K) will give the working
 c index of that phoneme, between 1 and 40. A value of 0 for IPTR(K)
 c means that phoneme K is not in the set of 40 phonemes.

```

data (iptr(i),i=1,70)/63*0,35,2*0,13,0,25,7/
data (iptr(i),i=71,80)/10,0,34,17,0,23,2*0,37,0/
data (iptr(i),i=81,90)/0,29,16,14,2*0,32,40,30,0/
data (iptr(i),i=91,100)/3*0,36,2*0,27,4,26,5/
data (iptr(i),i=101,110)/31,15,6,22,38,0,3,19,8,9/
data (iptr(i),i=111,120)/39,1,0,20,11,2,28,18,24,33/
data (iptr(i),i=121,126)/21,12,4*0/

```

```

data idim/2000/

```

c This initialization flags BUILDPATH to prompt for speaker and condition.

```

ispkr = -1
icond = -1
hadtocompute = .false.

```

c Build in the maximum flexibility for pathnames since the data could be scattered almost anywhere!

```

1  format(/"Program TEMPLATESLIK using LPC24..." /
   a  "Enter filename for list of utterances: ", $)
   write (6,1)
2  format (a36)
   read (5,2)listname
   istr = 36
   call endstr(listname,istr)
   write (6,*)"Opening file ",listname
   open (unit=2,file=listname,status="old")
   rewind (unit=2)

3  format (a1)
4  format(/"Enter the max number of templates desired per phone: ", $)
   write (6,4)
   read (5,*)iocc

5  format(i3)

```

c **** Beginning of the loop that reads the label files for each utterance in the listfile.

```

6  read (2,5,end=10)iutt

   loaded = .false.
   call buildpath(iutt,0,0,"wav",ispkr,icond,speechfile)
   call buildpath(iutt,0,0,"lbl",ispkr,icond,labelfile)
   write (6,*)"Speechfile: ",speechfile
   write (6,*)"Labelfile: ",labelfile
54 format("Tok IPTR Sum")
   write (6,54)
7  call readlabels(labelfile,ilabel,frompos,topos,itot,idim)

```

c Now sift through the tokens of this utterance, making templates of the ones that qualify.

```

do 100 j=1,itot
  dur = topos(j) - frompos(j)

```

c If the token is at least 16 milliseconds long, then count it as valid

```

if (dur.ge.0.016) then
  itoken(ilabel(j)) = itoken(ilabel(j)) + 1

```

c Now if this is one of the 40 phonemes and if it is the Kth occurrence of that we seek, then process it and add it to the total number of

c templates built.

```

55      format(i3,2i5)
      write (6,55)ilabel(j),iptr(ilabel(j)),iphonetot+1

      if (iptr(ilabel(j)).ne.0.and.itoken(ilabel(j)).le.iocc) then
        itshere = .false.
        itemp(ilabel(j)) = itemp(ilabel(j)) + 1
        iphonetot = iphonetot + 1
        ii = iptr(ilabel(j))
        jj = itoken(ilabel(j))
        call buildpath(iutt,jj,ii,"pft",ispkr,icond,datafile)
        inquire(file=datafile,exist=itshere)

        if (.not.itshere) then
          hadtocompute = .true.
          write (6,*)"Building template for ",ml(ilabel(j))
          if (.not.loaded) then
            call loadspeech(speechfile,isperch,n)
            loaded = .true.
          end if

          call liktemplate(ispeech,frompos(j),topos(j),ts,lpc)

          write (6,*)"Writing datafile: ",datafile
          open  (unit = 3,
a           file = datafile,
a           status = "new",
a           form  = "unformatted")
          write (3)ts
          close (unit=3)

c      Writing LPC coefficients into parameter files has been disabled because
c      I currently have no need to do so. It can be brought back later if
c      necessary...
c           call buildpath(iutt,jj,ii,"lpc",ispkr,icond,datafile)
c           open  (unit = 3,
c           a           file = datafile,
c           a           status = "unknown",
c           a           form  = "unformatted")
c           rewind (3)
c           write (3)lpc
c           close (unit=3)

          end if

        end if

      end if

100      continue

      if (iphonetot.lt.(40*iocc)) goto 6

10      continue

c **** End of loop for reading label files

      close (unit=2)

      if (hadtocompute) then
        call fdate(date)
        recordfile = "tpltslik-//date(12:16)//"-//"

```

```

a      char(48+ispkr)//char(48+icond)
      open (unit=1,file=recordfile,status="unknown")
      rewind (1)

110    format("TEMPLATESLIK using LPC24 finished at: ",a24/
a      "Total number of templates built: ",i4/
a      "Speaker and Condition: ",2i1/
a      "Phoneme Number of Templates"/)
      write (1,110)date,iphetot,ispkr,icond
      totaltime = etime(tarray)
115    format("/User time: ",g12.3/
a      "System time: ",g12.3/
a      "Total time: ",g12.3)
      write (1,115)tarray(1),tarray(2),totaltime

120    format(a3,t17,i3)

      do 140 i=1, 40
        write (1,120)ml(ishort(i)),itemp(ishort(i))
140    continue

      close (1)
      end if

      stop
      end

      real function tmerit(pathname)

c Provides the total merit of the recognition experiment stored
c in PATHNAME

      character*(*) pathname
      real p1(10),ssum

      tmerit = 0.0
      open (unit=1,file=pathname,status="old",err=60)
      rewind (1)
      read (1,*)p1
      close (1)
      tmerit = ssum(p1,10)

60    return
      end
c -----

```

```

      subroutine uncovtest(i,icov,mincov,ipset,ipmax,
a         ans,iptmp,iptmpmax)

c This routine determines whether or not removing an utterance
c will produce an uncovered phone(s). If so, ANS will be TRUE
c and the set of uncovered phones will be listed.

c INPUTS
c I is the utterance index
c ICOV is the array of accumulated coverages of each phoneme.
c MINCOV is the lower bound on the number of occurrences of
c   each phoneme.
c IPSET is the array that lists the phonemes of interest.
c IPMAX is the number of phonemes in IPSET.

c OUTPUTS
c ANS is a logical variable indicating whether or not removal
c   of utterance i uncovered any phones.
c IPTMP is the array containing the list of uncovered phones.
c IPTMPMAX is the number of phones in IPTMPMAX.

      integer i,icov(126),mincov,ipset(40),ipmax
      integer iptmp(40),iptmpmax
      integer j,itcov(126)
      logical ans

c Initially, accumulate the coverages for this utterance and
c store them in the temporary buffer ITCOV(j).
c ITCOV must be zeroed out first.

      do 50 j=1,126
        itcov(j) = 0
50      continue

      call accumphon(i,itcov)
      ans = .false.
      iptmpmax = 0

c Now check through the phone set to see if any would be
c uncovered.

      do 100 j=1,ipmax
        if ((icov(ipset(j))-itcov(ipset(j))).lt.mincov) then
          ans = .true.
          iptmpmax = iptmpmax + 1
          iptmp(iptmpmax) = ipset(j)
        end if
100     continue

      return
      end
c -----

```

```
subroutine update(rankarray,irank,irow)
```

```
c This routine is designed to update the two-dimensional arrays
c RNN, SCR, and SCD, where IRANK is the position where the hit
c occurred in recognition, and IROW designates which phoneme was
c being tested.
```

```
integer irank, irow, iclass, ipclass
real rankarray(40,10)
```

```
c First of all, we update the overall accumulation regardless of
c phoneme class.
```

```
rankarray(irank,1) = rankarray(irank,1) + 1
```

```
c Now determine the phoneme class, and update accordingly.
```

```
iclass = ipclass(irow)
rankarray(irank,iclass) = rankarray(irank,iclass) + 1
```

```
return
end
```

```
c -----
```

```

      subroutine usrinit(pname,it,ip,ifwdev,ispkr,icond,
a      massfill,tdfilename)

c Subroutine to handle the user initialization for the EXP7PLUS-series
c of programs. Its purpose is to cut down the clutter in all the
c different versions of EXP7PLUS.

c INPUTS
c PNAME      program name for the herald
c IT         number of templates per phoneme (shown in the herald)
c IP         size of the phoneme set (shown in the herald)

c OUTPUTS
c ISPKR      is the speaker number being processed.
c ICOND      is the condition being processed
c IFWDEV     frequency deviation index for frequency warping. A
c            value of 0 means no frequency warping, and each unit
c            gives 62.5 Hz either side of center; e.g. a value of
c            2 would provide a frequency window of 250 Hz.
c MASSFILL   indicates whether MASSFILL mode has been selected.
c TDFILENAME self-explanatory

      character*36 pname,tdfilename
      character*1 ans
      integer it,ip,ifwdev,ispkr,icond,istr
      logical massfill

      ispkr = -1
      icond = -1
      massfill = .false.

10      format(/"Program ",a36/
a      "Calculates TD data for all types of clashing"/
a      "using ",i3," tokens of each of ",i3," phonemes."/)

20      write (6,10)pname,it,ip
30      format(/"Enter value for frequency deviation index: ",%)
      write (6,30)
      read (5,*)ifwdev

c BUILDPATH is called here simply to establish the values of ISPKR
c and ICOND.

      call buildpath(0,0,0,"pft",ispkr,icond,tdfilename)

      if (icond.eq.2.or.icond.eq.3) then
60      format(/"Do you want to use MASSFILL mode? ",%)
      write (6,60)
85      format(a1)
      read (5,65)ans
      if (ans.eq."y".or.ans.eq."Y") massfill = .true.
      end if

      call tddpath(ispkr,icond,tdfilename)
      istr = 36
      call endstr(tdfilename,istr)

      return
      end
c -----

```

```
subroutine uttphon(ipud,ipun,ips)
```

c This routine produces a linked list of phonemes for each of
c the 539 utterances in the AMRL training set. The linked lists
c are implemented by arrays IPUD and IPUN. The head of each list
c is indexed by the utterance number and the tail is indicated
c by a null marker in the next pointer. The size of each list
c is also calculated and stored in IPS.

c Input data comes from the vocabulary transcription,
c amrlvoc.LBL and from the word-sentence index, ws-index.

```
integer ipud(15000),ipun(15000),ips(539)
integer iwd(2400),iwn(2400),iwp(539),iutt(100)
integer ipd(1000),ipn(1000),ipp(207)
integer i,j,k,kk,n,il(1200),idim,itot
real fp(1200),tp(1200)
character*24 filename
```

```
data idim/1200/
```

c First create the linked list of words for given utterances.
c The data will come from the word-sentence index. Working
c through each word, we will add it to the word list of each
c of the utterances in which it is found.

c The linked lists are implemented in IWD(i) and IWN(i) with
c IWP(i) pointing to the tail of list i. IWN(1) - IWN(539)
c contain the heads of each list indexed by utterance.

c N is the pointer to the next available storage location.
c J is the word index number.
c IUTT(i) is the list of utterances in which word J is found.

```
n = 540
```

```
open (unit=1,file="ws-index",status="old")
rewind (1)
```

```
10 format(1x,i3,2x,90(1x,i3))
```

```
do 100 i=1,207
  read(1,10),iutt
  if (i.ne.j) write (6,*)"Error in reading ws-index."
```

```
do 50 k=1,90
  if (iutt(k).eq.0) then
    k = 91
  else if (iwp(iutt(k)).eq.0) then
    iwp(iutt(k)) = iutt(k)
    iwd(iutt(k)) = i
    iwn(iutt(k)) = 0
  else
    iwn(iwp(iutt(k))) = n
    iwp(iutt(k)) = n
    iwd(n) = i
    iwn(n) = 0
    n = n + 1
  end if
```

```
50 continue
100 continue
close (unit=1)
```


c Next, create linked lists of phones for all the words
 c in the vocabulary. The data will come from the
 c vocabulary transcription contained in amrlvoc.LBL.
 c The only info we will use is the label list that is
 c read into IL(i) with IPP(i) pointing to the tail of
 c list i. IPD(1) - IPD(207) contains the heads of each
 c list indexed by word.

c N is the pointer to the next available storage location.
 c J is the word index

```
n = 208
j = 0
```

```
filename = "amrlvoc"
call readlabels(filename,il,fp,tp,itot,idim)
```

```
do 200 i=1,itot
  if (il(i).eq.8) then
    goto 200
  else if (il(i).eq.35) then
    j = j + 1
  else if (ipp(j).eq.0) then
    ipp(j) = j
    ipd(j) = il(i)
    ipn(j) = 0
  else
    ipn(ipp(j)) = n
    ipp(j) = n
    ipd(n) = il(i)
    ipn(n) = 0
    n = n + 1
  end if
200 continue
```

c Now to build the phone list for each utterance. To do
 c this, we will work through the word list of the utterance
 c and insert the phone list of the word into the list we
 c are making.

c N is the pointer to the next available storage location.
 c I is the utterance index.
 c J is the word list pointer.
 c K is the phone list pointer.
 c KK is the working list pointer.

```
n = 540

do 300 i=1,539
  ips(i) = 0
  j = i
  kk = i
218 k = iwd(j)
220 ipud(kk) = ipd(k)
  ipun(kk) = 0
  ips(i) = ips(i) + 1
  if (ipn(k).ne.0) then
    k = ipn(k)
    ipun(kk) = n
    kk = n
    n = n + 1
    goto 220
  else if (iwn(j).ne.0) then
```

```
      j = iwn(j)
      ipun(kk) = n
      kk = n
      n = n + 1
      goto 218
    end if
300    continue

    return
  end
c -----
```

```
subroutine warper(d,it,jt,r,tnd)
```

```
c This routine was derived from subroutine WARP. For the present,
c the major operating characteristics of WARP have been preserved,
c but the argument list has been changed to accommodate new routines.
c This routine will accomplish the dynamic programming algorithm
c to time warp two sets of feature vectors together and produce
c the total normalized distance. It will also build the parent
c array so the warping function can be traced from its termination
c at (IT,JT) back to the origin (1,1). It assumes the array of
c distance measures, D(I,J) have already been calculated.
```

```
c INPUT:
```

```
c      D(i,j)  Array of distance measures between two sets of vectors
c      IT      Total number of I vectors
c      JT      Total number of J vectors
c      R       Width of the DTW search space from the main diagonal
```

```
c OUTPUT:
```

```
c      TND     Total normalized distance between the vector sets
```

```
c OTHER VARIABLES:
```

```
c      R       Integer controlling the width of the search path
c      G(i,j)  Dynamic programming search array
c      P(i,j,k) Parent array for backtracking.
c              P(i,j,1) = i coordinate of parent of G(i,j)
c              P(i,j,2) = j coordinate of parent of G(i,j)
```

```
real d(-2:50,-2:50),g(-2:50,-2:50)
real tnd,x1,x2,x3
integer p(50,50,2)
integer i,j,n,it,jt,r,rp1
```

```
cccccc tnd = 1.0e5
        rp1 = r+1
        n = 50
```

```
c Now do the DP-matching. This section implements the flowchart
c in Figure 4, page 47, of Sakoe and Chiba paper. Start with
c the initial conditions:
```

```
        i = 1
        j = 1
        g(i,j) = 2*d(i,j)
```

```
20      i = i + 1
```

```
30      if (i.gt.(j+r)) then
           goto 50
        else if (i.le.0.or.i.gt.it) then
           goto 20
        else
           x1 = g(i-1,j-2) + 2*d(i,j-1) + d(i,j)
           x2 = g(i-1,j-1) + 2*d(i,j)
           x3 = g(i-2,j-1) + 2*d(i-1,j) + d(i,j)
           g(i,j) = amin1(x1,x2,x3)
```

```
c Code for determining who the parent was and storing it
```

```
c      if (g(i,j).eq.x1) then
c          p(i,j,1) = i
```

```

c      p(i,j,2) = j-1
c      else if (g(i,j).eq.x2) then
c      p(i,j,1) = i-1
c      p(i,j,2) = j-1
c      else
c      p(i,j,1) = i-1
c      p(i,j,2) = j
c      endif
c      goto 20
endif

50      j = j+1
      if (j.le.jt) then
        i = j-r
        goto 30
      else
        tnd = g(it,jt)/float(it+jt)
      endif

c      This may very well be a needless check now. Let's comment it
c      out and see if we get any ripples...
c      i = it
c55      if (tnd.gt.1.e6) then
c      i = i - 1
c      tnd = g(i,jt)/float(it+jt)
c      goto 55
c      endif

      return
end
c -----

```

program wordlist

c EXPERIMENTAL VERSION

c Designed to produce a lists of words that contain a given
c phoneme.

```

real frompos(1200),topos(1200)
integer ilabel(1200),i,j,k,n,itot,idim,iorder(70)
integer ipt(126),iwr(1000),next(1000)
character*15 voc(207)
include "ml"

data (iorder(i),i=1,10)/112,116,107,13,14,15,98,100,103,10/
data (iorder(i),i=11,20)/11,12,70,63,109,110,71,77,78,7/
data (iorder(i),i=21,30)/6,115,122,67,84,102,83,90,74,68/
data (iorder(i),i=31,40)/118,108,114,121,8,104,76,119,16,72/
data (iorder(i),i=41,50)/69,99,97,117,82,89,101,87,120,124/
data (iorder(i),i=51,60)/73,64,94,85,79,105,111,88,9,58/
data (iorder(i),i=61,68)/35,42,36,43,45,39,34,126/
data idim/1200/

```

c First load the vocabulary into its buffer array.

```

open(unit=1,file="amrlvoc.dat",status="old")

10  format(a15)
    do 100 i=1,207
        read (1,10)voc(i)
        call endstr(voc(i),15)
100  continue
    close (unit=1)

```

c Next, load the vocabulary transcription

```
call readlabels("amrlvoc",ilabel,frompos,topos,itot,idim)
```

c Now work through the label list and build word lists accordingly.

c Depends on the format of the labels having the word separator

c symbol (#) preceding the phonetic transcription of each word.

c This is in fact the delimiter between word transcriptions.

c N is the pointer to the next available storage location when
c building linked lists.

c J is the word index.

c IPT(i) is a pointer array that keeps track of the last element
c in the word list for each phoneme i. If its value is 0 then the
c list has not yet been activated for that phoneme.

c IWORD(i) and NEXT(i) implement the linked lists. IWORD contains
c the integer indices representing words. IWORD(1) - IWORD(126)
c contain the initial words in the linked lists for the respective
c phonemes.

```

n = 127
j = 0

do 200 i=1,itot
    if (ilabel(i).eq.35) then
        j = j + 1
    else if (ipt(ilabel(i)).eq.0) then

```

```

        ipt(ilabel(i)) = ilabel(i)
        iwrđ(ilabel(i)) = j
        next(ilabel(i)) = 0
    else
        next(ipt(ilabel(i))) = n
        ipt(ilabel(i)) = n
        iwrđ(n) = j
        next(n) = 0
        n = n + 1
    end if
200    continue

c Now to print out the results. I want the phonemes in the proper
c order, so I make use of the IORDER lookup array.

250    format ("/" fB\s+6",a3,"fR\s-6")
260    format (a15," ",$)
265    format ("")

    do 300 i=1, 68
        j = 0
        k = iorder(i)
        write (6,250)ml(k)

        if (iwrđ(k).ne.0) then
270            write (6,260) voc(iwrđ(k))
                j = j + 1
                if (j.ge.5) then
                    j = 0
                    write (6,265)
                    end if
                if (next(k).ne.0) then
                    k = next(k)
                    goto 270
                else
                    if (j.ne.0) write (6,265)
                    goto 300
                    end if
                end if
        end if
300    continue

    stop
    end
c -----

```

```
subroutine xscale(n)
```

```
c This routine will take the sampling frequency, FS, and the number
c of points, N, and calculate an array of frequencies to be used
c to scale the x-axis for qplotting. Note that only the first (n/2)+1
c values correspond to the positive frequencies up to the Nyquist rate.
```

```
character*20 string
real x(1024),fs
integer n
```

```
5 format(/" Enter sampling frequency in Hz: ",%)
write (6,5)
read (5,*)fs
```

```
do 10 i=1,n
10 x(i) = (fs*(i-1))/n
```

```
string = "x-axis scaling"
call saveprmt(x,n,1024,string)
```

```
return
end
```

```
c -----
```

```
subroutine zeroit(x,i)
```

```
c Zeros out the given array
```

```
real x(1)
integer i,j
```

```
do 10 j=1, i
10 x(j) = 0.0
continue
return
end
```

Appendix K: Utterance Subset for Research

A010 I R STAB-OUT THREE FOUR TWO BY TWO
A019 AUTOPILOT LEVEL A LITTLE
A021 DOGFIGHT
A048 U H F SET G C A THREE FIFTY POINT FIVE SEVEN FIVE ENTER
A056 CAGE
A061 MIL
A074 IDLE
A075 DOGFIGHT
A085 CAUTION TEST
A086 AUTOPILOT HARDER
A087 CAGE
A096 U H F SET GROUND TWO FORTY POINT THREE ZERO ZERO ENTER
A103 WARNING TEST
A116 U H F SET G C A THREE NINETY POINT NINE ENTER
A119 TACAN SET TWENTY Y ENTER
A129 BORE SIGHT AIM NINE MISSILES
A140 AT FIVE SEARCH ABOVE SIX ZERO POINT FIVE THOUSAND
A144 CAGE
A149 F M SET GROUND THIRTY NINE POINT ONE ZERO ZERO ENTER
A151 TACAN SET THREE TWO ENTER
A160 I R AT TWO FIVE SEARCH SURFACE
A167 RADAR SHARP
A168 BACKUP
A171 D DIRECT
A174 AUTOPILOT HOLD FOUR THOUSAND SIX HUNDRED NINETY TRUE
A175 BURNUP
A176 I N S HEADING TWO ZERO THREE ENTER
A179 WARNING ACKNOWLEDGED
A182 AUTOPILOT HARDER
A190 WARNING ACKNOWLEDGED
A202 CAGE
A203 CLEAR WARNING
A206 YES
A222 BACKUP
A246 NAV HEADING ZERO FOUR TWO ENTER
A252 DOGFIGHT
A256 CHAFF
A257 CLEAR HIGH INDEX
A267 YES
A269 AUTOPILOT HOLD MACH TWO
A272 AUTOPILOT HARDER
A275 R W R AUTO
A279 COMM SET GROUND SEVENTY FOUR POINT FIVE ZERO ZERO ENTER
A281 CAGE
A305 YES
A318 CAUTION ACKNOWLEDGED
A324 WARNING TEST
A349 GUNS
A357 I L S HEADING ZERO THREE EIGHT ENTER
A373 GO EMERGENCY
A386 GO SIX FOUR POINT SIX ZERO ZERO ENTER
A404 I R SHARP
A436 AUTOPILOT HOLD ONE THOUSAND ONE HUNDRED TWO TRUE
A446 CAUTION ACKNOWLEDGED
A462 CLEAR DESTINATION TARGET
A518 CAUTION TEST

Appendix L: Analyses of Normal, Loud, and Lombard Speech

This appendix contains the data obtained from the extensive analyses of normal, loud, and Lombard speech, as described in Chapter 7. The first set of tables lists the quantitative differences found in the various features. The next series of displays contain graphs of formant trajectories. Finally, the last series depicts significant differences in features as determined by three-way analysis of variance for each phoneme of each speaker.

Table 28. Average differences in phoneme features between Loud and normal speech, all speakers

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.2	-0.8	-1.0	-0.3	0.7	0.5	1.1	0.3	-0.3	-0.9	140	0.4	-1.9	--	-13	-49	-22	-0.012
T	-1.5	-0.8	-0.7	-0.7	-0.4	-0.6	0.9	0.7	0.6	0.3	176	0.2	0.3	--	-13	-17	76	-0.005
K	0.0	0.7	0.3	0.6	0.8	-0.3	0.2	-0.4	-0.8	-0.4	-41	0.2	-0.4	--	-23	6	-2	-0.009
B	0.5	1.1	-0.2	-0.4	0.4	0.1	0.1	-0.1	-0.3	-0.1	-16	-0.2	-0.1	38	-152	-210	-164	-0.002
D	0.8	0.5	-1.0	-0.3	-0.1	-0.3	0.4	0.1	0.0	0.3	14	0.0	0.4	33	-55	15	-149	0.000
G	-0.4	0.2	-0.1	-0.4	0.9	-0.4	0.3	0.0	-0.5	0.1	-23	0.1	0.2	20	-10	62	55	-0.004
DX	-0.4	0.9	-0.3	1.0	1.5	-0.4	-0.6	-0.4	-0.5	-0.6	-47	0.6	0.1	54	8	-2	-42	-0.003
M	-1.6	0.6	-0.3	0.1	0.2	0.0	0.2	0.2	-0.3	-0.2	57	0.2	-0.4	51	10	-67	-210	-0.005
N	-1.0	1.1	-0.1	0.2	0.7	-0.4	0.0	0.0	-0.3	-0.2	2	0.5	0.1	52	-28	26	-39	-0.003
NX	-1.1	1.5	0.2	-0.2	0.9	-0.1	-0.1	-0.2	-0.3	-0.3	-14	0.5	0.1	41	-127	-18	-222	-0.015
S	-0.7	-0.3	-0.3	-0.2	-0.2	-1.6	0.0	0.1	1.0	1.2	165	0.0	2.9	--	48	-6	18	-0.008
Z	0.1	0.6	0.4	0.6	-0.3	-2.3	-0.3	0.0	0.7	1.1	49	-0.2	2.9	37	-69	-276	-176	-0.006
CH	-1.2	-0.6	-0.4	-0.8	-0.8	-0.7	1.0	0.6	0.6	0.6	171	0.2	0.9	--	25	71	100	-0.012
TH	-1.5	-0.4	-0.4	0.0	0.6	-0.1	0.1	0.1	0.1	-0.1	42	0.4	-0.3	--	15	-41	-78	-0.014
F	-0.8	0.0	-0.2	-0.5	0.0	-0.3	0.3	0.4	0.4	0.0	44	0.1	0.6	--	-58	-23	-93	-0.014
SH	-2.1	-1.8	-1.8	-1.9	-0.9	0.2	1.7	1.4	0.7	0.6	282	0.0	-0.4	--	206	203	229	-0.008
JH	-0.8	-0.3	-0.3	-0.6	-0.5	-0.4	1.0	0.5	0.1	0.2	101	0.1	-0.2	8	-74	26	0	-0.010
V	-0.4	1.1	-0.4	0.5	0.9	0.2	-0.2	-0.2	-0.4	-0.6	13	0.6	-0.4	48	-40	-100	-69	-0.002
L	-1.5	0.4	0.9	0.6	1.3	0.8	-0.2	-0.7	-0.8	-1.2	32	0.6	-1.5	51	23	-185	-129	0.002
R	-1.9	-0.5	0.8	1.2	1.2	0.4	-0.1	-0.6	-0.8	-1.0	30	0.7	-1.1	61	28	-10	-127	0.006
Y	-1.2	-0.2	0.1	0.3	1.2	0.7	-0.5	-0.3	-0.4	-0.7	29	0.4	-1.2	45	2	-35	-165	0.013
HH	0.4	0.5	-0.2	-0.2	0.3	-0.3	0.1	0.0	-0.1	0.0	31	-0.1	0.3	32	-2	90	45	-0.009
EL	-1.4	0.1	0.7	0.8	1.5	1.0	-0.3	-0.9	-0.8	-1.3	49	0.6	-1.8	38	28	-144	-58	0.008
W	-1.0	1.0	1.3	0.6	0.5	0.0	-0.3	-0.3	-0.5	-0.7	-31	0.1	-0.6	45	19	67	109	0.017
EH	-2.5	-1.0	0.5	0.8	2.2	0.7	-0.1	-0.5	-0.9	-1.5	113	1.2	-1.4	58	50	39	26	0.021
AO	-2.4	-1.0	-0.2	0.6	1.8	0.6	0.2	-0.2	-0.7	-1.4	121	1.1	-1.6	55	67	-11	62	0.024
AA	-2.8	-1.4	-0.3	0.6	1.9	0.9	0.1	-0.2	-0.7	-1.5	160	1.1	-2.1	64	59	37	12	0.023
UW	-1.8	-0.1	1.2	1.1	1.5	0.1	-0.1	-0.5	-1.1	-1.3	31	0.5	-1.6	54	31	6	-21	0.022
ER	-2.3	-0.6	0.6	1.2	1.4	0.4	-0.2	-0.5	-0.7	-1.3	93	0.7	-1.1	63	47	74	-93	0.007
AY	-2.7	-1.0	-0.2	0.7	2.0	1.2	0.0	-0.4	-0.8	-1.6	143	1.2	-1.9	55	65	21	57	0.020
EY	-2.4	-0.3	1.3	0.5	1.7	0.4	0.1	-0.5	-0.9	-1.4	70	1.2	-2.2	57	37	12	-40	0.026
AW	-2.8	-1.0	-0.4	1.0	1.9	1.3	-0.2	-0.4	-0.9	-1.6	147	1.1	-1.3	55	62	45	49	0.022
AX	-2.2	-0.3	0.2	0.9	2.2	1.0	-0.4	-0.8	-0.9	-1.4	76	1.3	-1.9	49	25	45	36	0.001
IH	-3.0	-0.9	0.4	0.5	2.3	0.8	0.2	-0.5	-1.1	-1.5	114	1.5	-2.2	63	28	56	23	0.011
AE	-2.7	-1.0	-0.1	0.7	2.0	1.1	0.1	-0.4	-0.9	-1.5	130	1.1	-1.8	56	56	40	29	0.024
AH	-3.0	-1.1	0.4	1.4	2.2	0.7	-0.2	-0.5	-1.2	-1.7	119	1.2	-1.8	62	67	50	0	0.020
OY	-2.4	-0.1	0.6	0.8	1.8	1.1	0.0	-0.8	-1.2	-1.6	95	1.0	-2.1	53	42	97	-4	0.019
IY	-1.8	0.0	0.7	0.3	1.4	0.3	0.3	-0.4	-0.7	-1.1	47	0.9	-1.8	56	18	24	-76	0.008
OW	-2.3	-1.0	0.4	0.9	1.6	0.7	0.0	-0.6	-0.7	-1.4	119	0.9	-1.7	58	54	55	84	0.014
AXR	-1.4	0.4	1.2	1.7	1.7	0.6	-0.7	-1.1	-1.2	-1.4	9	0.6	-0.9	32	16	-29	-206	0.034

Table 29. Average differences in phoneme features between Lombard and normal speech, all speakers

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)											COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	Lo		Hi	1		2	3		
P	-1.2	-1.1	-0.8	0.5	1.3	0.9	1.3	0.2	-1.0	-2.0	154	0.7	-3.5	--	53	-57	-86	-0.013	
T	-1.2	-0.8	-0.3	-0.9	-0.8	-0.6	1.1	0.8	0.8	0.3	190	0.0	0.6	--	11	-14	70	-0.002	
K	0.4	0.7	0.2	0.1	0.4	-0.4	0.5	-0.2	-0.3	-0.4	3	-0.1	-0.3	--	-35	-5	11	-0.009	
B	1.5	1.6	0.3	-0.2	0.3	-0.1	-0.1	-0.3	-0.2	-0.2	-80	-0.3	0.1	23	-104	-81	-178	0.002	
D	0.8	0.0	-1.0	-0.6	-0.1	-0.1	0.5	0.2	0.2	0.3	30	-0.2	0.2	19	-70	-61	-89	0.001	
G	0.4	-0.3	-0.2	-0.3	0.4	-0.3	0.3	0.2	-0.4	0.1	-10	0.1	0.0	12	53	219	119	0.001	
DX	0.7	1.0	-0.3	0.8	0.9	-0.4	-0.3	-0.3	-0.5	-0.3	-83	0.2	0.0	30	-7	10	49	-0.001	
M	0.0	1.4	-0.1	0.8	0.4	-0.2	-0.4	-0.2	-0.5	-0.3	-47	0.0	-0.1	31	9	-117	-214	-0.003	
N	0.7	2.2	0.3	0.5	0.6	-0.6	-0.3	-0.3	-0.4	-0.3	-96	0.1	0.2	29	-32	-58	-153	0.000	
NX	0.7	2.6	1.0	0.4	0.4	-0.3	-0.3	-0.4	-0.6	-0.5	-118	0.1	0.1	29	-133	-91	-240	0.001	
S	-0.7	-0.3	-0.1	-0.4	-0.6	-1.8	0.6	0.2	1.0	1.3	204	-0.1	2.4	--	13	-18	3	0.003	
Z	0.5	0.0	0.3	-0.1	-0.7	-2.3	0.4	0.2	0.8	1.3	122	-0.3	2.6	25	-89	-215	-170	-0.006	
CH	-0.9	-0.4	-0.4	-1.0	-0.7	-0.6	1.2	0.6	0.6	0.5	180	0.1	0.8	--	58	133	120	-0.006	
TH	-0.5	-0.3	-0.4	-0.1	0.0	-0.5	0.4	0.3	0.3	-0.1	64	0.1	-0.2	--	33	26	-23	-0.008	
F	-0.4	0.0	0.2	-0.2	0.1	-0.6	0.3	0.3	0.2	-0.2	29	0.0	0.3	--	-16	-30	-79	-0.007	
SH	-1.9	-1.3	-0.9	-1.8	-0.8	0.1	1.6	1.1	0.7	0.4	224	0.2	0.2	--	72	143	110	-0.003	
JH	-0.9	0.0	0.4	-0.5	-0.5	-0.5	1.2	0.5	0.1	-0.4	105	0.3	0.0	28	-87	-27	-77	0.013	
V	0.8	1.7	0.2	0.4	0.7	-0.2	-0.1	-0.5	-0.5	-0.6	-31	0.1	-0.3	31	-63	-140	-32	0.000	
L	-0.7	0.5	0.8	0.3	1.3	1.1	-0.1	-0.9	-0.9	-1.1	3	0.6	-1.0	29	15	-172	-64	0.002	
R	-1.1	-0.4	0.7	1.3	1.4	0.6	-0.2	-0.9	-1.1	-1.2	35	0.8	-1.7	31	22	-31	-295	0.005	
Y	-0.2	0.7	0.8	0.5	1.3	0.3	-0.3	-0.8	-0.8	-0.8	-56	0.3	-0.6	29	15	-38	-232	0.005	
HH	1.6	1.2	0.6	0.1	0.1	-0.6	0.0	-0.1	0.0	-0.4	2	-0.3	0.2	34	9	-70	-29	-0.007	
EL	-0.2	0.8	0.9	0.5	1.3	1.0	-0.2	-1.0	-1.0	-1.1	-13	0.5	-1.0	21	14	-111	-19	0.013	
W	-0.2	0.7	1.7	1.1	1.0	0.0	-0.4	-0.8	-0.9	-1.0	-63	0.1	-0.8	25	32	-83	-99	0.004	
EH	-1.3	-0.4	0.4	0.4	1.6	0.8	0.2	-0.5	-0.9	-1.4	86	0.9	-1.8	33	34	-41	-10	0.018	
AO	-1.1	-0.2	0.0	0.7	1.4	1.0	-0.1	-0.7	-1.1	-1.0	41	0.6	-1.7	28	53	23	47	0.015	
AA	-1.3	-0.7	-0.3	0.7	1.7	1.3	0.0	-0.7	-1.1	-1.4	92	0.8	-2.1	34	45	24	-23	0.021	
UW	-1.1	0.3	1.1	1.0	1.7	0.8	-0.1	-1.0	-1.3	-1.4	3	0.6	-1.7	31	26	-18	-19	0.006	
ER	-0.9	0.0	0.6	1.2	1.3	0.6	-0.2	-0.8	-1.0	-1.2	31	0.5	-1.3	35	29	29	-71	0.005	
AY	-1.2	-0.1	0.1	0.5	1.8	1.0	-0.1	-0.7	-1.1	-1.2	56	0.8	-1.9	31	48	9	40	0.016	
EY	-1.3	0.1	1.1	0.5	1.3	0.4	0.1	-0.5	-0.9	-1.2	34	0.8	-1.6	31	33	-32	-121	0.025	
AW	-1.4	-0.2	-0.1	0.8	1.6	0.9	-0.1	-0.6	-1.0	-1.2	72	0.7	-1.5	32	43	41	11	0.014	
AX	-1.3	0.0	0.6	0.9	2.0	1.3	-0.2	-1.0	-1.4	-1.6	41	1.1	-2.3	28	21	-34	28	0.002	
IH	-1.7	-0.4	0.4	0.6	1.8	1.1	0.1	-0.6	-1.2	-1.5	84	1.0	-2.2	30	21	-23	-33	0.009	
AE	-1.0	0.1	0.2	0.3	1.4	1.0	0.0	-0.6	-0.8	-1.1	54	0.7	-1.8	33	39	-9	30	0.022	
AH	-1.5	-0.4	0.2	0.9	1.8	1.4	0.2	-0.8	-1.4	-1.6	93	0.7	-2.5	31	51	37	-45	0.011	
OY	-1.4	-0.2	0.6	1.3	1.9	1.2	-0.2	-1.0	-1.4	-1.8	74	0.7	-2.5	32	37	59	-109	0.019	
IY	-1.2	0.3	0.8	0.6	1.4	0.5	0.1	-0.7	-1.0	-1.2	16	0.6	-1.7	36	23	-56	-241	0.011	
OW	-1.2	-0.7	0.2	1.2	1.5	0.9	0.1	-0.8	-1.1	-1.5	87	0.6	-1.9	33	46	70	27	0.013	
AXR	-0.7	0.6	1.4	1.9	2.0	0.5	-0.9	-1.2	-1.4	-1.4	-27	0.8	-0.8	18	15	-55	-227	0.024	

Table 30. Average differences in phoneme features between Lombard and loud speech, all speakers

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)											COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	Lo		Hi	1		2	3		
P	0.0	-0.3	0.2	0.8	0.6	0.4	0.2	-0.2	-0.7	-1.1	14	0.3	-1.6	--	66	-8	-64	-0.001	
T	0.3	0.0	0.3	-0.2	-0.4	-0.1	0.2	0.1	0.2	0.0	14	-0.2	0.2	--	24	3	-6	0.003	
K	0.4	0.1	-0.1	-0.5	-0.4	-0.1	0.3	0.3	0.5	0.0	44	-0.2	0.1	--	-12	-11	12	0.000	
B	1.0	0.4	0.4	0.2	-0.1	-0.2	-0.2	-0.2	0.1	-0.1	-64	-0.2	0.2	-14	47	128	-14	0.004	
D	0.0	-0.5	0.0	-0.3	0.0	0.1	0.0	0.1	0.1	0.0	16	-0.2	-0.3	-14	-16	-76	60	0.000	
G	0.8	-0.5	-0.1	0.1	-0.5	0.1	0.0	0.2	0.2	0.0	13	0.0	-0.2	-8	63	157	65	0.004	
DX	1.1	0.1	0.0	-0.2	-0.6	0.0	0.2	0.1	0.0	0.2	-36	-0.4	-0.1	-24	-16	13	91	0.002	
M	1.6	0.8	0.3	0.6	0.1	-0.2	-0.6	-0.4	-0.2	-0.2	-104	-0.2	0.2	-19	-1	-50	-4	0.002	
N	1.7	1.0	0.5	0.3	-0.1	-0.2	-0.3	-0.3	-0.2	-0.1	-98	-0.3	0.1	-23	-4	-84	-114	0.003	
NX	1.8	1.0	0.8	0.6	-0.5	-0.2	-0.2	-0.2	-0.3	-0.2	-104	-0.4	0.0	-12	-6	-73	-18	0.016	
S	-0.1	0.0	0.2	-0.3	-0.3	-0.2	0.5	0.1	0.0	0.1	39	-0.1	-0.4	--	-36	-12	-15	0.011	
Z	0.4	-0.5	-0.1	-0.7	-0.3	0.0	0.7	0.2	0.1	0.2	73	-0.1	-0.2	-12	-20	61	6	0.000	
CH	0.3	0.2	0.0	-0.3	0.0	0.1	0.2	0.0	0.0	-0.1	9	0.0	-0.1	--	33	63	20	0.007	
TH	1.0	0.1	0.0	-0.1	-0.6	-0.4	0.2	0.3	0.2	0.0	22	-0.3	0.0	--	19	67	55	0.006	
F	0.5	0.0	0.4	0.3	0.1	-0.3	0.0	-0.1	-0.2	-0.2	-15	-0.1	-0.3	--	42	-7	15	0.007	
SH	0.2	0.5	0.9	0.1	0.0	-0.1	-0.1	-0.3	0.0	-0.3	-58	0.1	0.6	--	-134	-60	-119	0.005	
JH	-0.1	0.3	0.7	0.1	0.0	-0.1	0.3	0.0	0.0	-0.6	4	0.2	0.2	20	-13	-53	-77	0.022	
V	1.2	0.6	0.6	-0.1	-0.1	-0.3	0.2	-0.2	-0.1	0.0	-44	-0.6	0.1	-16	-23	-40	36	0.002	
L	0.8	0.1	0.0	-0.3	0.0	0.3	0.0	-0.2	-0.1	0.1	-28	0.0	0.5	-22	-7	13	64	0.000	
R	0.9	0.1	-0.1	0.2	0.3	0.3	-0.1	-0.3	-0.3	-0.2	5	0.1	-0.6	-27	-6	-21	-168	-0.001	
Y	1.0	0.9	0.7	0.2	0.2	-0.4	0.2	-0.5	-0.3	-0.1	-86	-0.2	0.6	-16	14	-3	-67	-0.009	
HH	1.2	0.7	0.7	0.3	-0.2	-0.3	-0.1	-0.2	0.1	-0.4	-29	-0.3	-0.1	3	11	-160	-74	0.002	
EL	1.2	0.6	0.2	-0.3	-0.2	0.0	0.1	-0.1	-0.2	0.2	-62	-0.1	0.8	-17	-14	33	39	0.005	
W	0.9	-0.3	0.4	0.5	0.5	0.0	-0.1	-0.5	-0.4	-0.3	-32	0.0	-0.1	-17	13	-150	-207	-0.013	
EH	1.1	0.6	0.0	-0.3	-0.6	0.1	0.2	0.1	-0.1	0.1	-27	-0.3	-0.4	-24	-16	-81	-37	-0.003	
AO	1.3	0.8	0.2	0.1	-0.4	0.4	-0.2	-0.5	-0.3	0.4	-79	-0.5	-0.1	-25	-15	35	-15	-0.009	
AA	1.5	0.6	0.0	0.1	-0.1	0.4	-0.1	-0.4	-0.4	0.2	-68	-0.3	0.0	-31	-15	-13	-34	-0.002	
UW	0.6	0.4	-0.1	-0.1	0.1	0.7	0.0	-0.5	-0.2	-0.1	-28	0.0	-0.1	-23	-5	-25	2	-0.016	
ER	1.4	0.6	0.0	0.0	-0.1	0.2	0.0	-0.3	-0.3	0.1	-62	-0.2	-0.1	-28	-18	-45	22	-0.002	
AY	1.5	0.9	0.4	-0.2	-0.2	-0.2	0.0	-0.3	-0.2	0.4	-88	-0.4	0.0	-24	-17	-12	-17	-0.004	
EY	1.1	0.4	-0.1	0.0	-0.5	0.0	0.0	0.0	0.0	0.2	-37	-0.4	0.7	-26	-5	-44	-80	-0.001	
AW	1.4	0.7	0.3	-0.2	-0.3	-0.4	0.1	-0.2	-0.2	0.4	-75	-0.4	-0.1	-23	-19	-4	-38	-0.008	
AX	0.9	0.3	0.4	0.0	-0.2	0.3	0.2	-0.1	-0.5	-0.2	-35	-0.2	-0.4	-20	-4	-80	-8	0.001	
IH	1.3	0.5	-0.1	0.1	-0.5	0.3	-0.1	-0.1	-0.1	0.0	-30	-0.5	0.0	-30	-8	-79	-55	-0.002	
AE	1.7	1.1	0.3	-0.4	-0.6	-0.1	-0.1	-0.2	0.1	0.4	-76	-0.4	-0.1	-23	-17	-49	1	-0.002	
AH	1.5	0.8	-0.2	-0.5	-0.4	0.8	0.3	-0.4	-0.3	0.1	-26	-0.5	-0.7	-31	-16	-13	-46	-0.009	
OY	1.0	-0.1	0.0	0.6	0.1	0.1	-0.3	-0.3	-0.2	-0.3	-21	-0.3	-0.3	-21	-4	-38	-105	0.000	
IY	0.6	0.3	0.1	0.3	0.0	0.2	-0.2	-0.3	-0.2	-0.1	-32	-0.3	0.0	-20	5	-79	-164	0.003	
OW	1.1	0.3	-0.1	0.4	-0.2	0.2	0.1	-0.2	-0.5	-0.1	-32	-0.3	-0.2	-25	-8	16	-57	-0.002	
AXR	0.7	0.2	0.2	0.2	0.2	-0.2	-0.3	-0.1	-0.2	0.0	-37	0.2	0.1	-12	-1	-27	-21	-0.010	

Table 31. Average differences in phoneme features between Loud and normal speech, speaker #1

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	0.2	1.6	-0.1	-0.5	0.2	-1.1	-0.3	0.0	0.7	0.4	12	-0.2	1.5	--	-119	-246	-108	-0.016
T	-1.6	-0.7	-1.1	-1.8	0.3	-0.1	0.5	0.8	0.7	0.6	124	0.2	0.6	--	-42	249	251	-0.014
K	-0.9	1.2	0.9	0.6	1.2	-1.1	-0.3	-0.7	-0.3	0.0	-62	0.4	0.7	--	-136	-95	-290	-0.022
B	2.5	3.7	0.2	-1.9	-0.2	-1.0	-0.7	0.2	0.5	1.4	-159	-1.0	2.1	79	-405	-196	102	0.003
D	2.7	1.3	-1.5	-1.0	-1.2	-0.9	-0.1	0.6	0.7	1.6	-79	-1.0	2.1	13	-143	-48	-91	0.021
G	-2.2	-0.8	0.0	-0.1	0.3	-0.3	-0.9	0.1	0.5	1.1	-20	0.6	1.4	-17	-47	-98	-249	0.001
DX	0.0	2.0	0.6	2.0	0.8	-2.0	-0.8	-0.4	-0.2	-0.2	-186	0.2	1.6	55	70	51	-211	-0.005
M	-2.7	2.5	0.3	0.6	1.5	0.3	-0.1	0.0	-1.2	-1.2	103	0.7	-1.5	47	93	89	-66	-0.006
N	-0.9	3.6	1.2	0.7	0.3	-1.0	-0.1	0.2	-0.8	-0.5	-24	0.1	0.2	38	4	-56	-178	-0.013
NX	-1.6	2.8	0.8	0.6	1.8	-0.8	-0.4	-0.3	-0.9	-0.7	-32	0.5	0.0	27	-92	-82	-224	-0.038
S	-0.3	-2.0	-2.7	-2.0	-0.8	-0.7	0.5	1.1	2.0	1.8	414	-0.2	2.4	--	289	335	501	-0.012
Z	-1.3	1.4	0.2	-0.1	-0.8	-0.8	-0.9	0.1	0.9	1.2	24	0.0	2.1	21	-448	-559	-431	-0.010
CH	-1.8	-2.4	-2.8	-2.2	-0.4	1.5	0.6	1.7	0.3	1.0	270	0.2	-0.5	--	281	-20	148	-0.006
TH	-2.4	-1.4	-2.2	-1.6	0.3	0.1	0.1	0.7	1.2	1.1	135	0.5	1.2	--	-99	-84	-243	-0.020
F	-0.8	-0.1	-0.7	-0.3	0.5	-0.1	-0.2	0.3	0.9	-0.5	83	0.2	0.2	--	-66	24	-159	-0.020
SH	-0.2	-1.7	-3.2	-2.7	-0.3	2.0	0.5	1.8	-0.7	1.4	232	-0.2	-1.0	--	761	480	484	-0.006
JH	-0.6	-0.6	-1.6	-1.5	0.4	0.9	0.8	1.3	-1.0	0.2	73	0.1	-1.1	-82	-104	45	101	-0.003
V	-0.3	2.5	1.3	1.4	1.8	-0.5	-0.9	-0.8	-0.8	-1.3	-65	0.3	-0.5	28	16	-104	-171	-0.014
L	-3.7	-0.1	1.8	1.2	3.4	-0.2	-0.6	-1.1	-1.4	-1.4	-10	1.5	-1.2	63	23	-447	-369	-0.001
R	-5.1	-1.0	1.5	1.4	1.0	0.2	0.3	-0.7	-0.8	-0.9	101	1.4	-1.2	58	37	-29	-33	0.006
Y	-3.2	0.3	0.4	0.8	3.2	0.0	-0.6	-0.6	-1.1	-1.2	105	1.2	-1.1	38	27	-164	-741	0.027
HH	2.4	2.2	1.1	0.0	-0.3	-1.7	0.2	-0.6	0.2	0.5	-128	-0.7	1.7	-7	-43	106	62	-0.024
EL	-1.8	1.1	2.4	1.4	3.6	-1.2	-0.9	-1.3	-1.2	-1.4	-74	1.1	-0.2	42	10	-598	-360	0.006
W	-3.5	2.1	3.9	0.0	0.6	-0.3	-0.1	-0.7	-0.8	-0.8	-118	0.7	-0.6	39	105	329	223	-0.001
EH	-5.0	-1.4	1.6	1.9	4.0	-0.2	-0.2	-0.9	-1.9	-2.1	81	2.0	-1.9	44	62	22	-59	0.017
AO	-5.3	-2.2	1.3	1.6	4.1	-0.1	-0.7	-1.1	-1.4	-1.5	52	2.1	-1.3	45	99	-289	-160	0.017
AA	-5.9	-1.6	0.9	1.4	3.2	0.6	0.0	-0.7	-1.5	-1.9	141	2.0	-2.4	53	26	-90	-97	0.021
UW	-4.2	-0.2	1.5	0.8	2.7	0.4	-0.1	-0.6	-1.3	-1.7	102	1.4	-1.9	36	47	25	-104	0.013
ER	-4.3	-0.4	1.9	2.2	2.0	0.1	-0.3	-1.2	-1.3	-1.4	8	1.5	-1.5	48	42	-7	-174	0.001
AY	-5.4	-1.2	1.3	1.8	4.4	-0.1	-1.0	-0.7	-1.7	-1.8	53	2.1	-1.5	45	47	-108	-197	0.016
EY	-4.8	0.1	2.0	1.3	3.7	-0.2	-0.5	-0.6	-1.8	-1.9	55	1.8	-1.7	49	37	-12	-59	0.001
AW	-5.5	-0.9	0.6	1.5	4.1	0.2	-0.4	-0.7	-1.7	-1.9	93	2.0	-2.0	48	41	28	49	0.023
AX	-4.2	0.0	1.2	2.1	3.6	-0.4	-1.2	-0.8	-1.4	-1.6	21	1.7	-1.0	49	13	-119	-48	-0.008
IH	-5.7	-0.6	1.1	1.6	3.9	-0.2	-0.3	-0.5	-1.7	-1.8	116	2.1	-1.6	47	22	10	-54	0.014
AE	-5.8	-1.5	0.8	2.3	5.0	-0.4	-0.4	-1.1	-2.0	-2.1	87	2.4	-1.8	48	55	56	18	0.030
AH	-5.5	-1.0	1.4	2.2	3.7	0.1	0.6	-1.1	-2.2	-2.5	145	2.1	-2.8	52	18	-376	-405	0.004
OY	-4.4	0.7	1.6	1.3	3.4	-0.3	0.9	-1.6	-1.9	-1.9	49	1.6	-2.1	47	-32	-492	-715	0.039
IY	-4.8	-0.5	1.1	1.3	3.7	-0.1	0.2	-1.1	-1.5	-1.8	103	1.8	-1.8	49	59	-34	-209	0.007
OW	-4.5	-1.0	1.2	0.4	1.4	0.3	0.7	-0.3	-0.8	-1.1	145	1.3	-1.6	49	41	-192	-148	-0.006
AXR	-2.5	0.1	2.3	2.4	2.8	-0.1	-1.3	-1.2	-1.5	-1.6	-45	1.3	-1.3	-4	44	-48	-444	-0.011

Table 32. Average differences in phoneme features between Lombard and normal speech, speaker #1

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.7	-0.6	-0.6	0.5	0.4	0.5	0.5	0.3	-0.4	-0.9	118	0.5	-1.3	--	57	-130	-93	-0.015
T	-2.0	-1.2	-1.2	-2.1	-1.1	0.0	1.6	1.2	1.0	0.7	229	0.1	0.4	--	-69	-9	93	-0.008
K	0.3	0.8	0.4	0.4	0.2	-0.7	1.1	-0.3	-0.3	-0.9	15	0.0	-0.5	--	-181	-44	-240	-0.021
B	3.5	3.5	0.5	-1.7	-0.7	-0.8	-0.5	0.3	0.6	0.9	-151	-1.3	1.6	88	-435	-121	-123	0.013
D	2.6	1.5	-0.9	-0.4	-1.5	-0.7	0.0	0.5	0.5	1.1	-55	-1.0	1.6	42	-134	-200	-213	0.009
G	0.0	-0.5	-1.0	-1.2	-0.3	-0.3	0.5	0.6	0.7	0.5	53	-0.2	0.7	3	14	299	499	0.007
DX	0.5	1.6	-0.1	2.8	0.9	-1.2	-1.1	-0.7	-0.7	-0.5	-228	0.2	0.6	50	57	133	-82	-0.008
M	-2.6	3.0	0.9	0.9	0.8	1.3	-0.4	0.5	-1.9	-1.8	124	0.6	-2.7	51	104	161	93	-0.009
N	-0.5	3.5	1.1	0.5	-0.1	-0.6	-0.2	0.4	-0.7	-0.5	-20	-0.1	0.0	43	-19	-77	-86	-0.010
NX	-0.1	5.3	1.0	0.8	0.2	-0.8	0.1	0.1	-1.2	-1.0	-65	-0.2	-0.6	40	-106	-73	47	-0.018
S	-0.7	-2.0	-2.2	-2.0	-1.8	-1.6	1.6	1.6	1.6	2.3	495	-0.2	3.0	--	107	121	239	-0.007
Z	-1.5	2.2	1.1	0.3	-1.6	-1.6	0.3	0.3	0.2	1.4	64	-0.1	2.2	30	-530	-796	-669	-0.009
CH	-2.7	-3.1	-2.5	-2.7	-1.8	1.9	2.3	1.7	0.6	0.5	391	0.2	-1.6	--	97	138	353	-0.013
TH	-1.4	-0.8	-2.1	-0.8	-0.4	0.3	0.7	0.9	0.8	0.1	155	0.2	0.0	--	17	30	-154	-0.025
F	-0.4	0.3	-0.5	0.4	0.7	0.2	0.3	0.2	-0.5	-0.9	51	0.2	-1.0	--	-43	-158	-94	-0.020
SH	-0.1	-0.6	-1.8	-2.5	-1.5	2.1	1.3	0.9	0.1	0.8	195	-0.5	-1.4	--	102	116	126	-0.008
JH	-2.0	-1.4	-1.9	-1.7	-1.4	1.8	1.8	0.9	0.3	0.1	246	0.1	-1.8	-84	-36	-82	87	-0.017
V	-0.8	2.2	1.6	2.2	0.8	0.5	0.2	-1.1	-1.6	-2.1	35	0.4	-2.5	40	8	-105	-180	-0.011
L	-3.3	-0.1	1.4	0.7	1.5	2.4	0.3	-1.2	-1.5	-2.0	120	1.0	-4.0	64	36	-259	-259	0.003
R	-5.1	-1.8	1.6	1.9	0.3	2.5	0.6	-1.3	-1.5	-1.7	178	1.5	-3.9	55	43	-88	98	0.008
Y	-2.8	0.5	0.5	0.8	0.2	1.1	1.5	-1.0	-1.2	-1.2	116	0.7	-2.7	35	25	-287	-545	0.028
HH	3.2	1.8	1.2	0.3	-0.4	-1.2	0.5	-0.6	-0.1	-0.4	-70	-0.8	0.4	-10	45	202	353	-0.025
EL	-1.4	1.8	2.7	1.6	1.8	1.1	-0.7	-1.6	-1.7	-1.8	-85	0.7	-2.7	46	2	-585	-440	0.003
W	-3.8	1.4	4.0	0.3	1.1	0.6	0.3	-1.1	-1.5	-1.4	-15	1.0	-2.1	41	112	169	181	0.001
EH	-4.8	-1.8	1.6	1.6	2.1	2.3	0.8	-1.2	-2.2	-2.5	145	1.7	-4.6	44	79	-22	-74	0.033
AO	-4.6	-1.8	1.2	1.6	2.3	0.8	-0.3	-0.9	-1.3	-1.3	53	1.6	-2.0	43	103	16	48	0.030
AA	-5.2	-1.7	0.6	1.3	2.2	3.2	-0.2	-1.0	-1.9	-2.4	183	1.7	-4.9	54	45	-120	-84	0.044
UW	-4.1	-0.6	1.8	1.2	1.6	1.5	1.1	-1.3	-1.7	-2.2	148	1.3	-3.8	41	55	-36	42	0.030
ER	-4.0	-0.6	2.0	2.1	0.9	2.0	-0.5	-1.3	-1.5	-1.7	49	1.3	-3.3	48	56	-5	-60	0.010
AY	-5.0	-1.3	0.9	1.5	2.5	2.7	-0.3	-1.2	-2.0	-2.2	110	1.7	-4.4	47	67	-101	-176	0.037
EY	-4.7	-0.2	2.7	1.9	1.7	1.5	0.7	-1.5	-2.2	-2.3	46	1.5	-3.9	48	73	-129	-116	0.012
AW	-5.6	-1.7	0.5	2.0	2.4	3.3	-0.1	-1.5	-2.2	-2.4	151	1.9	-5.1	54	65	61	47	0.032
AX	-4.2	-0.2	2.4	2.1	1.9	2.3	0.1	-1.6	-2.4	-2.6	78	1.5	-4.6	48	38	-107	33	-0.010
IH	-5.3	-0.9	0.7	1.1	1.5	2.8	1.2	-0.9	-2.2	-2.4	198	1.5	-5.1	49	22	-16	-23	0.025
AE	-5.1	-1.5	0.8	1.5	2.9	2.6	0.3	-1.7	-2.2	-2.4	127	1.8	-4.7	51	93	65	194	0.053
AH	-4.3	-1.2	1.2	2.1	2.1	2.8	1.0	-1.8	-2.5	-3.1	190	1.6	-5.7	44	62	-236	-281	0.019
OY	-4.4	0.1	1.3	1.4	1.4	3.2	1.1	-2.0	-2.3	-2.5	145	1.3	-5.5	47	-32	-455	-733	0.051
IY	-5.4	-1.5	1.5	2.0	1.0	2.2	1.8	-1.5	-2.1	-2.4	203	1.6	-4.8	52	73	-249	-327	0.021
OW	-4.8	-1.9	1.4	0.8	1.0	2.2	1.2	-0.8	-1.7	-2.0	228	1.4	-4.2	51	73	-305	-256	0.009
AXR	-1.9	0.7	2.0	2.3	2.0	1.3	-1.5	-1.3	-1.7	-1.8	-25	0.9	-2.5	-5	28	-63	-381	0.001

Table 33. Average differences in phoneme features between Lombard and loud speech, speaker #1

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.9	-2.3	-0.5	1.1	0.1	1.6	0.8	0.3	-1.1	-1.4	106	0.7	-2.8	--	175	116	15	0.001
T	-0.3	-0.5	-0.1	-0.3	-1.4	0.1	1.2	0.4	0.2	0.1	104	-0.2	-0.2	--	-27	-258	-158	0.005
K	1.1	-0.4	-0.4	-0.2	-1.0	0.3	1.4	0.4	0.1	-0.9	77	-0.4	-1.2	--	-45	50	50	0.001
B	1.0	-0.2	0.3	0.2	-0.5	0.2	0.2	0.1	0.1	-0.5	9	-0.3	-0.5	10	-29	76	-225	0.010
D	-0.1	0.2	0.6	0.5	-0.4	0.2	0.2	-0.1	-0.2	-0.4	24	0.0	-0.6	29	9	-152	-121	-0.012
G	2.3	0.3	-1.0	-1.1	-0.6	0.0	1.4	0.5	0.3	-0.5	73	-0.7	-0.7	20	61	397	748	0.006
DX	0.5	-0.4	-0.7	0.9	0.1	0.8	-0.3	-0.3	-0.5	-0.4	-42	0.0	-1.0	-5	-13	83	128	-0.003
M	0.1	0.5	0.6	0.3	-0.7	0.9	-0.3	0.5	-0.6	-0.5	22	-0.1	-1.2	4	11	73	159	-0.003
N	0.4	-0.1	-0.1	-0.2	-0.4	0.4	-0.2	0.2	0.1	0.0	4	-0.2	-0.2	5	-23	-21	92	0.003
NX	1.5	2.5	0.3	0.2	-1.6	0.1	0.5	0.4	-0.3	-0.4	-33	-0.7	-0.6	13	-14	10	271	0.019
S	-0.5	0.1	0.5	0.1	-1.1	-0.9	1.1	0.5	-0.3	0.5	80	-0.1	0.6	--	-182	-214	-282	0.005
Z	-0.2	0.8	0.9	0.4	-0.9	-0.8	1.2	0.2	-0.7	0.2	41	-0.1	0.1	9	-82	-237	-239	0.000
CH	-0.9	-0.7	0.3	-0.4	-1.4	0.5	1.7	0.0	0.3	-0.5	122	0.0	-1.1	--	-184	158	205	-0.007
TH	1.0	0.6	0.2	0.8	-0.7	0.2	0.6	0.2	-0.4	-1.0	20	-0.3	-1.2	--	117	114	88	-0.005
F	0.4	0.4	0.2	0.7	0.2	0.3	0.5	-0.2	-1.4	-0.4	-31	0.0	-1.2	--	23	-182	64	0.000
SH	0.1	1.1	1.4	0.2	-1.2	0.0	0.8	-1.0	0.8	-0.6	-37	-0.3	-0.4	--	-658	-384	-358	-0.002
JH	-1.4	-0.8	-0.3	-0.2	-1.8	0.9	1.0	-0.4	1.3	-0.1	174	0.0	-0.6	-2	67	-127	-14	-0.014
V	-0.5	-0.3	0.3	0.8	-0.9	1.0	1.1	-0.3	-0.8	-0.8	100	0.1	-2.0	11	-9	-2	-9	0.002
L	0.4	0.0	-0.4	-0.5	-1.9	2.6	0.9	-0.1	-0.2	-0.5	130	-0.5	-2.7	1	13	188	110	0.004
R	-0.1	-0.7	0.1	0.5	-0.6	2.3	0.3	-0.6	-0.7	-0.8	77	0.0	-2.7	-3	6	-58	131	0.002
Y	0.4	0.2	0.1	0.1	-3.0	1.1	2.1	-0.4	-0.1	0.0	11	-0.6	-1.6	-3	-2	-123	196	0.002
HH	0.8	-0.4	0.1	0.3	-0.1	0.6	0.3	0.0	-0.3	-1.0	58	-0.1	-1.3	-3	88	96	291	-0.001
EL	0.5	0.7	0.4	0.2	-1.8	2.3	0.2	-0.4	-0.5	-0.4	-11	-0.4	-2.4	4	-9	13	-80	-0.004
W	-0.3	-0.7	0.1	0.3	0.5	0.9	0.4	-0.4	-0.6	-0.6	103	0.2	-1.5	2	8	-160	-42	0.002
EH	0.2	-0.4	-0.1	-0.3	-2.0	2.5	1.0	-0.4	-0.3	-0.4	65	-0.4	-2.7	0	17	-44	-14	0.015
AO	0.7	0.3	0.0	0.0	-1.8	0.9	0.4	0.2	0.1	0.1	1	-0.5	-0.7	-1	4	305	208	0.014
AA	0.7	-0.1	-0.3	0.0	-1.0	2.6	-0.2	-0.4	-0.4	-0.6	42	-0.3	-2.6	1	18	-30	12	0.023
UW	0.1	-0.4	0.3	0.4	-1.1	1.1	1.2	-0.7	-0.4	-0.5	46	-0.1	-1.8	5	8	-61	146	0.018
ER	0.4	-0.2	0.0	-0.1	-1.1	2.0	-0.1	-0.2	-0.2	-0.3	41	-0.3	-1.8	-1	14	3	114	0.009
AY	0.4	-0.2	-0.4	-0.2	-1.8	2.8	0.7	-0.5	-0.3	-0.4	56	-0.4	-2.9	2	20	7	21	0.021
EY	0.1	-0.3	0.7	0.6	-2.0	1.7	1.2	-0.9	-0.4	-0.4	-9	-0.3	-2.2	-1	36	-118	-57	0.011
AW	-0.1	-0.8	-0.1	0.5	-1.7	3.1	0.2	-0.8	-0.5	-0.5	58	-0.1	-3.1	6	24	33	-2	0.009
AX	0.0	-0.3	1.2	0.1	-1.7	2.6	1.3	-0.8	-0.9	-1.0	56	-0.2	-3.6	-2	25	13	81	-0.002
IH	0.5	-0.3	-0.3	-0.5	-2.4	3.0	1.5	-0.4	-0.5	-0.6	81	-0.5	-3.5	2	0	-26	31	0.011
AE	0.7	0.0	0.0	-0.8	-2.0	2.9	0.7	-0.6	-0.2	-0.3	40	-0.6	-2.9	3	38	9	176	0.023
AH	1.2	-0.2	-0.2	-0.1	-1.6	2.7	0.5	-0.7	-0.3	-0.6	45	-0.5	-2.9	-7	44	140	124	0.015
OY	0.1	-0.6	-0.3	0.1	-2.0	3.4	0.3	-0.4	-0.5	-0.6	96	-0.3	-3.4	-1	0	36	-18	0.012
IY	-0.5	-1.0	0.4	0.7	-2.7	2.3	1.6	-0.4	-0.6	-0.7	100	-0.2	-3.0	3	15	-216	-118	0.014
OW	-0.4	-0.8	0.2	0.4	-0.4	1.9	0.4	-0.4	-0.9	-0.9	83	0.1	-2.6	1	32	-114	-108	0.015
AXR	0.6	0.6	-0.3	-0.1	-0.8	1.4	-0.2	-0.1	-0.2	-0.2	20	-0.3	-1.3	-1	-16	-16	63	0.012

Table 34. Average differences in phoneme features between Loud and normal speech, speaker #2

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-0.8	-1.4	-1.5	-0.4	0.0	0.5	0.6	0.5	0.2	-0.1	120	0.3	-0.6	--	104	181	181	-0.003
T	-0.1	0.7	1.2	1.1	-0.2	-0.6	0.0	0.0	-0.5	-0.5	3	-0.2	0.2	--	31	-169	-181	0.004
K	1.8	2.5	1.5	1.0	0.6	0.0	0.3	-0.3	-2.5	-0.7	-184	-0.3	-0.9	--	-155	0	-63	0.004
B	0.1	-1.3	-0.6	0.9	1.2	1.5	0.2	-0.9	-0.8	-1.3	99	0.5	-1.7	59	79	-369	-522	0.005
D	-1.3	0.4	-0.1	0.0	1.4	1.1	0.0	-0.8	-0.6	-0.7	24	0.9	-1.3	86	95	238	214	0.000
G	0.8	0.4	0.7	-0.6	-0.3	-0.8	0.5	0.7	-0.4	0.4	-109	-0.6	1.3	33	-75	-61	-334	-0.007
DX	-1.6	-1.0	0.6	2.0	2.6	-0.1	-1.0	-1.0	-1.3	-1.1	-72	1.0	-0.6	59	40	91	-47	-0.001
M	-1.3	0.3	-0.3	0.2	0.8	0.5	-0.3	-0.2	-0.4	-0.2	-31	0.5	-0.3	46	-18	1	98	-0.006
N	-0.9	0.6	0.1	0.0	1.8	0.1	-0.5	-0.6	-0.3	-0.4	-5	0.6	0.1	54	-11	159	188	0.001
NX	-1.5	0.4	0.5	0.5	1.8	1.4	-0.4	-1.1	-1.0	-1.1	41	0.8	-1.0	36	15	49	-102	-0.006
S	-1.2	-0.3	-0.1	-0.3	-1.3	-1.5	0.8	1.3	0.5	0.8	153	0.0	2.4	--	39	-95	2	-0.002
Z	-2.1	-1.6	0.5	0.7	-1.2	-1.2	0.5	0.7	0.7	0.5	241	0.2	1.8	11	91	-498	-58	-0.010
CH	0.6	1.8	2.5	1.3	-1.2	-0.7	0.4	-0.1	-0.7	-0.7	-89	0.4	0.0	--	40	-18	12	-0.003
TH	-0.7	-0.3	0.5	1.3	0.7	0.4	-0.3	-0.3	-1.2	-0.6	-15	0.3	-1.0	--	63	-235	-179	0.023
F	0.3	0.8	1.2	0.7	0.1	-0.3	-0.3	-0.4	-0.3	-0.4	-51	-0.2	-0.2	--	19	108	80	-0.020
SH	-1.9	-0.7	-0.3	-0.1	-0.8	1.0	0.2	1.4	-0.6	-0.3	86	0.4	-0.7	--	104	216	143	0.017
JH	-2.1	-1.1	-0.3	-0.4	-0.5	0.2	0.7	0.6	0.2	0.2	133	0.6	0.1	8	36	48	-10	-0.018
V	-0.5	0.6	1.3	0.9	0.4	0.3	-0.5	-0.5	-0.8	-0.5	-111	0.1	-0.3	44	8	-164	-215	-0.003
L	-1.7	-0.6	0.2	0.0	1.0	2.9	-0.3	-1.2	-0.8	-1.1	110	0.9	-1.1	47	5	-28	30	0.000
R	-0.8	-0.5	1.3	1.1	1.4	1.0	-1.1	-1.0	-1.0	-0.9	-46	0.6	0.3	52	26	-56	-723	0.010
Y	-1.3	-1.0	0.2	-0.5	2.7	2.1	-0.7	-1.0	-0.8	-1.2	39	0.6	-1.6	31	1	103	51	-0.001
HH	-1.7	-2.2	-1.1	-0.9	-0.2	0.4	0.6	0.5	0.3	0.8	146	0.4	-0.2	61	127	451	315	0.005
EL	-1.0	-0.1	0.9	0.3	1.5	1.0	-0.9	-0.9	-0.8	-0.5	-42	0.9	-0.3	31	11	-31	35	0.003
W	0.0	-0.2	1.0	1.4	0.7	0.8	-0.5	-0.9	-1.1	-0.9	-71	-0.1	-0.2	25	28	-90	44	0.042
EH	-1.8	-0.9	0.4	0.4	2.9	2.0	-0.5	-1.5	-1.3	-1.5	47	1.2	-1.0	48	35	84	101	0.013
AO	-1.4	-1.1	-0.3	-0.1	2.3	2.4	-0.5	-0.4	-1.3	-1.6	119	1.6	-1.9	41	39	33	217	0.006
AA	-2.0	-1.4	-0.4	0.2	2.1	1.8	-0.4	-0.4	-0.7	-1.5	132	0.9	-2.5	53	45	47	123	0.019
UW	-1.8	-0.6	0.9	1.3	2.5	1.5	-0.9	-1.2	-1.5	-1.6	23	1.1	-1.1	43	18	-11	-23	0.025
ER	-1.7	-1.1	0.5	1.3	2.0	1.1	-0.6	-1.2	-1.1	-1.3	47	1.2	-1.1	51	36	87	-238	0.003
AY	-1.7	-0.8	0.1	0.5	2.5	2.0	-0.4	-1.5	-1.3	-1.4	32	1.4	-1.9	46	29	40	145	0.010
EY	-1.2	0.6	1.5	0.3	2.4	2.1	-0.9	-1.8	-1.5	-1.3	-54	1.1	0.2	47	25	65	32	0.015
AW	-1.5	-0.7	0.8	1.3	2.0	1.9	-1.0	-1.8	-1.5	-0.8	-68	1.1	0.7	45	40	16	92	0.013
AX	-1.8	-0.6	0.7	0.5	2.0	2.0	-0.7	-1.3	-1.1	-1.2	27	1.0	-0.8	58	16	25	123	0.002
IH	-2.4	-1.4	0.2	0.5	3.4	2.6	-0.9	-1.4	-1.3	-2.0	89	1.9	-1.9	51	22	78	53	0.012
AE	-1.5	-1.0	0.0	0.6	2.2	1.9	-0.1	-1.3	-1.2	-1.5	65	0.9	-0.5	41	48	78	60	0.009
AH	-1.4	-0.3	0.9	1.2	2.1	1.4	-0.9	-1.6	-1.3	-1.2	-40	0.9	-0.3	53	41	42	169	0.013
OY	-2.0	-1.2	-0.9	-0.3	1.9	2.8	-0.1	-0.9	-1.0	-1.2	116	1.1	-2.1	44	21	281	232	0.037
IY	-2.4	-1.2	0.1	0.2	1.9	2.1	-0.4	-0.5	-0.9	-1.5	83	1.0	-1.8	45	6	13	-22	0.019
OW	-1.9	-1.2	0.2	0.9	2.0	2.7	-0.6	-1.3	-1.5	-1.7	80	1.2	-1.7	43	43	26	193	0.021
AXR	-1.3	-0.2	1.6	2.3	2.7	0.6	-1.3	-1.6	-1.6	-1.6	-43	1.1	-0.6	37	32	-9	-745	0.017

Table 35. Average differences in phoneme features between Lombard and normal speech, speaker #2

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-6.0	-6.7	-4.4	-1.4	-0.7	0.6	3.4	3.6	0.7	-1.1	350	1.8	-3.2	--	138	156	115	-0.014
T	-3.1	-3.7	-1.3	-0.4	-1.5	-1.3	2.2	3.1	-0.3	0.4	145	0.6	0.6	--	207	175	207	-0.008
K	0.9	0.9	0.0	0.1	-0.8	-0.7	1.8	1.8	-2.2	-0.4	-81	-0.2	-0.2	--	-148	0	182	0.001
B	2.6	-0.8	-2.7	-1.7	-1.3	-0.4	1.4	1.5	1.2	0.4	-95	-0.3	0.6	80	-148	-266	-124	0.023
D	3.0	0.8	-0.8	0.5	-1.7	-1.4	1.1	1.8	-0.3	-0.4	-93	-1.0	-1.4	22	225	-60	271	-0.003
G	2.9	-0.3	-0.5	0.1	-0.5	-1.7	1.5	0.7	-0.9	0.5	-140	-0.5	1.7	-14	139	286	353	0.002
DX	3.3	1.4	1.1	0.8	0.8	-0.3	-0.1	-0.2	-1.6	-1.0	-166	-0.7	-1.4	57	-27	156	150	-0.002
M	4.8	1.6	-0.1	0.9	-0.4	-1.6	-0.2	-0.1	-0.3	0.0	-254	-1.2	0.0	38	-64	-122	98	0.010
N	5.1	1.4	-0.7	0.8	-0.8	-1.2	0.2	0.1	-0.1	-0.1	-187	-1.5	0.4	48	-77	-152	-173	0.011
NX	4.5	1.0	0.1	1.1	-1.0	-0.8	0.2	-0.1	-0.3	-0.3	-122	-1.3	-0.4	29	-62	-244	-267	0.024
S	-4.7	-4.5	-1.5	-0.7	-1.9	-1.7	2.6	3.8	-0.5	1.0	270	0.7	0.7	--	237	210	352	-0.010
Z	1.7	-2.2	-0.4	0.5	-1.3	-2.6	1.6	2.7	-1.3	0.5	67	-1.1	1.4	35	205	-25	34	-0.011
CH	-3.8	-3.7	-1.3	-0.9	-2.5	-0.4	2.5	3.6	-0.9	0.6	170	1.0	0.5	--	373	264	268	0.007
TH	-3.8	-3.5	0.1	1.3	-0.3	-0.9	1.0	2.5	-1.1	-1.1	84	1.4	-2.5	--	201	19	-17	0.014
F	-0.7	-0.2	2.9	2.4	-0.3	-1.1	0.2	1.2	-1.8	-2.1	-176	-0.2	-1.2	--	178	320	354	-0.007
SH	-7.5	-7.6	-4.4	-3.0	-2.4	1.6	3.1	4.7	-0.5	1.6	388	1.0	-1.1	--	445	498	690	0.000
JH	-2.1	-1.9	1.3	0.8	-2.0	-1.2	2.1	2.3	-1.3	-0.5	-3	1.2	0.2	64	242	55	95	0.026
V	3.5	2.4	2.0	1.7	-0.6	-0.6	-0.2	-0.3	-1.2	-1.2	-203	-1.6	-1.7	37	-26	-97	-125	-0.005
L	2.2	2.0	1.2	0.7	-0.9	1.4	-0.2	-0.8	-0.9	-0.7	-89	-1.1	-0.3	42	6	23	-37	0.006
R	2.4	1.4	1.0	1.9	0.7	0.1	-0.9	-0.9	-1.2	-1.0	-126	0.3	-0.4	43	2	-3	-897	0.003
Y	3.4	1.5	1.4	0.1	0.4	-0.4	0.0	-0.2	-0.8	-0.8	-88	-0.8	-1.6	41	-7	105	11	0.014
HH	1.1	-1.9	-1.2	-0.2	-2.0	-0.8	1.1	1.5	0.6	0.4	74	-0.3	1.2	67	279	207	273	0.002
EL	3.3	2.6	1.1	0.6	0.0	-0.2	-0.5	-0.7	-0.6	-0.5	-185	-0.7	0.1	24	-26	14	-32	0.034
W	3.5	1.0	1.6	1.4	-0.1	0.2	-0.1	-0.8	-1.3	-1.0	-152	-1.2	-1.2	38	-11	-507	-331	0.022
EH	2.7	2.0	1.3	1.0	0.7	0.5	-0.2	-0.9	-1.5	-1.3	-85	-0.5	-0.8	45	29	-37	27	0.023
AO	2.9	2.5	0.5	0.6	0.4	1.3	-1.0	-0.7	-1.3	-0.9	-101	-0.7	-0.3	36	31	34	95	0.015
AA	2.7	2.2	0.8	1.1	-0.1	0.9	-0.6	-1.1	-1.0	-0.8	-99	-0.9	-0.6	44	33	48	-38	0.011
UW	1.5	1.3	1.3	1.4	1.2	0.5	-0.2	-1.0	-1.7	-1.5	-91	-0.1	-2.2	31	4	64	-38	-0.003
ER	2.4	1.3	0.4	1.4	0.4	0.0	-0.7	-0.7	-0.8	-0.8	-89	0.1	-0.1	44	5	35	-293	0.010
AY	3.2	2.6	1.1	0.9	0.1	0.5	-0.2	-1.2	-1.2	-0.8	-126	-1.1	-1.1	41	37	46	42	0.014
EY	4.2	3.7	2.7	0.9	0.2	-0.1	-0.2	-1.3	-1.9	-0.6	-247	-0.4	0.6	35	9	95	125	0.021
AW	3.7	2.4	1.3	1.3	-0.3	0.4	-1.0	-1.2	-1.1	-0.1	-207	-0.8	1.4	38	37	19	42	0.022
AX	2.3	1.8	1.8	1.5	0.5	0.7	-0.6	-0.9	-1.5	-1.3	-110	-0.7	-1.5	55	27	-37	30	0.007
IH	2.3	1.0	1.0	1.5	0.8	0.9	-0.7	-1.1	-1.4	-1.2	-82	-0.2	-0.5	42	22	-46	-62	0.019
AE	3.2	1.9	0.8	1.1	-0.2	0.5	0.2	-1.1	-1.0	-0.9	-61	-1.0	-0.8	34	24	-62	-5	0.035
AH	3.0	2.4	1.1	1.5	0.2	1.1	-0.3	-1.5	-1.8	-0.9	-143	-1.0	-0.7	43	54	52	-82	0.003
OY	2.7	0.8	-0.5	0.6	0.2	2.0	0.0	-0.9	-1.2	-1.2	-11	-0.7	-2.2	40	41	393	10	0.032
IY	2.6	1.5	1.6	1.0	-0.5	-0.7	-0.2	0.0	-0.8	-0.6	-142	-0.4	-0.7	44	-9	-20	49	0.015
OW	2.0	1.1	0.2	1.0	0.0	1.2	-0.3	-0.7	-1.2	-1.0	-55	-0.6	-1.4	37	15	-335	-157	0.009
AXR	3.2	1.9	1.9	2.1	0.4	-0.4	-0.7	-0.9	-1.3	-1.3	-155	0.1	-1.0	42	1	-75	-161	0.044

Table 36. Average differences in phoneme features between Lombard and loud speech, speaker #2

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-5.4	-5.3	-2.9	-1.0	-0.7	0.2	2.8	3.0	0.5	-1.1	230	1.6	-2.7	--	34	-25	-66	-0.011
T	-3.0	-4.4	-2.5	-1.5	-1.4	-0.7	2.1	3.1	0.2	0.9	142	0.8	0.4	--	175	344	389	-0.012
K	-0.9	-1.6	-1.5	-0.8	-1.4	-0.7	1.5	2.2	0.2	0.3	103	0.1	0.7	--	7	-1	245	-0.003
B	2.5	0.5	-2.1	-2.6	-2.5	-1.9	1.2	2.4	2.0	1.7	-194	-0.8	2.3	21	-227	103	397	0.018
D	4.2	0.5	-0.7	0.5	-3.0	-2.4	1.1	2.6	0.3	0.3	-117	-1.9	-0.1	-64	129	-298	57	-0.003
G	2.1	-0.8	-1.2	0.7	-0.2	-0.9	1.1	0.0	-0.5	0.2	-30	0.1	0.3	-47	214	347	687	0.009
DX	4.9	2.4	0.5	-1.2	-1.8	-0.3	0.9	0.9	-0.3	0.1	-94	-1.7	-0.9	-2	-67	65	197	-0.001
M	6.1	1.3	0.2	0.8	-1.1	-2.0	0.1	0.1	0.1	0.2	-223	-1.7	0.2	-8	-46	-123	-2	0.015
N	6.1	0.8	-0.9	0.8	-2.6	-1.2	0.7	0.6	0.3	0.3	-183	-2.1	0.3	-6	-67	-310	-362	0.009
NX	5.9	0.6	-0.4	0.6	-2.8	-2.2	0.6	1.0	0.7	0.8	-164	-2.0	0.5	-7	-76	-293	-165	0.030
S	-3.5	-4.2	-1.5	-0.4	-0.6	-0.2	1.7	2.5	-1.0	0.2	117	0.7	-1.7	--	198	306	350	-0.008
Z	3.8	-0.5	-0.8	-0.2	-0.1	-1.4	1.1	2.0	-1.9	0.0	-175	-1.3	-0.4	24	115	473	93	-0.001
CH	-4.5	-5.5	-3.8	-2.2	-1.3	0.3	2.2	3.7	-0.2	1.3	259	0.6	0.5	--	333	282	256	0.010
TH	-3.1	-3.2	-0.4	-0.1	-1.0	-1.3	1.3	2.9	0.1	-0.5	100	1.1	-1.5	--	138	253	161	-0.009
F	-1.0	-1.0	1.7	1.6	-0.4	-0.8	0.5	1.6	-1.5	-1.8	-126	0.0	-1.0	--	160	212	274	0.013
SH	-5.6	-6.9	-4.1	-2.8	-1.5	0.6	2.9	3.3	0.1	1.9	301	0.6	-0.4	--	341	282	546	-0.018
JH	0.0	-0.8	1.6	1.2	-1.5	-1.4	1.4	1.7	-1.5	-0.7	-137	0.5	0.2	56	206	7	105	0.044
V	3.9	1.9	0.7	0.8	-0.9	-0.8	0.3	0.2	-0.4	-0.7	-92	-1.8	-1.5	-7	-34	67	91	-0.001
L	3.9	2.6	1.0	0.7	-1.9	-1.6	0.1	0.4	-0.1	0.4	-199	-2.0	0.8	-5	1	51	-67	0.006
R	3.2	2.0	-0.3	0.7	-0.8	-0.9	0.2	0.1	-0.2	-0.1	-80	-0.2	-0.7	-9	-25	53	-173	-0.007
Y	4.7	2.5	1.1	0.7	-2.3	-2.4	0.8	0.8	0.0	0.4	-127	-1.4	0.1	10	-8	1	-41	0.015
HH	2.8	0.3	-0.2	0.8	-1.7	-1.2	0.5	1.0	0.3	-0.4	-71	-0.7	1.4	5	151	-244	-43	-0.004
EL	4.2	2.7	0.2	0.4	-1.5	-1.2	0.4	0.2	0.2	0.0	-143	-1.6	0.4	-7	-37	45	-67	0.030
W	3.5	1.2	0.5	-0.1	-0.7	-0.6	0.4	0.1	-0.1	-0.2	-81	-1.1	-1.0	13	-40	-417	-375	-0.020
EH	4.6	2.9	0.9	0.6	-2.2	-1.4	0.3	0.7	-0.2	0.3	-131	-1.7	0.2	-3	-6	-121	-74	0.010
AO	4.3	3.6	0.7	0.8	-1.9	-1.0	-0.4	-0.3	0.0	0.7	-219	-2.3	1.7	-5	-8	0	-122	0.009
AA	4.7	3.6	1.1	0.9	-2.1	-0.9	-0.2	-0.6	-0.2	0.7	-231	-1.8	2.0	-10	-11	1	-161	-0.008
UW	3.2	2.0	0.4	0.2	-1.2	-1.1	0.7	0.3	-0.2	0.1	-113	-1.2	-1.1	-13	-14	75	-15	-0.027
ER	4.1	2.5	-0.1	0.1	-1.6	-1.1	-0.1	0.5	0.4	0.5	-137	-1.1	1.0	-7	-31	-52	-55	0.006
AY	4.9	3.3	1.0	0.4	-2.4	-1.5	0.2	0.3	0.1	0.6	-158	-2.5	0.8	-6	8	6	-103	0.004
EY	5.4	3.1	1.2	0.6	-2.2	-2.2	0.7	0.5	-0.4	0.7	-192	-1.6	0.4	-12	-16	30	93	0.006
AW	5.2	3.2	0.5	0.1	-2.3	-1.6	0.0	0.6	0.4	0.7	-139	-1.9	0.7	-7	-3	4	-50	0.009
AX	4.1	2.4	1.1	0.9	-1.5	-1.3	0.0	0.3	-0.4	-0.1	-137	-1.7	-0.7	-3	11	-62	-93	0.006
IH	4.8	2.5	0.8	1.1	-2.7	-1.7	0.2	0.3	0.0	0.8	-171	-2.1	1.5	-9	0	-124	-115	0.007
AE	4.7	2.9	0.8	0.5	-2.5	-1.4	0.3	0.2	0.2	0.6	-126	-1.9	-0.3	-7	-24	-139	-65	0.026
AH	4.4	2.6	0.2	0.2	-1.9	-0.3	0.6	0.1	-0.4	0.2	-103	-1.9	-0.4	-9	13	9	-251	-0.010
OY	4.7	2.0	0.5	0.9	-1.8	-0.9	0.1	0.0	-0.2	0.1	-128	-1.8	-0.1	-4	20	113	-222	-0.005
IY	5.0	2.7	1.5	0.8	-2.4	-2.8	0.2	0.5	0.0	1.0	-225	-1.4	1.1	-1	-14	-33	71	-0.004
OW	3.9	2.3	0.0	0.1	-2.0	-1.5	0.3	0.6	0.3	0.7	-135	-1.8	0.3	-6	-28	-361	-350	-0.012
AXR	4.5	2.1	0.3	-0.1	-2.3	-1.0	0.6	0.7	0.4	0.3	-112	-1.1	-0.4	4	-31	-66	584	0.027

Table 37. Average differences in phoneme features between Loud and normal speech, speaker #3

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-2.9	-1.6	-1.4	-0.7	0.6	0.9	0.7	0.1	-0.1	0.1	94	0.8	-0.9	--	-51	-269	-249	-0.014
T	-1.6	-1.7	-1.4	-0.7	-0.4	0.1	1.4	0.4	0.4	0.3	232	0.3	0.1	--	-11	3	46	-0.008
K	-0.6	-0.6	-0.4	0.5	0.4	-0.1	0.4	-0.1	-0.7	0.0	21	0.3	-0.3	--	-9	-1	-87	-0.013
B	1.7	1.6	-1.6	-2.7	-0.1	-0.4	0.0	0.0	1.1	2.0	-36	-0.9	2.6	0	-359	65	15	0.005
D	-0.2	0.5	-0.9	-0.5	-0.2	0.1	0.7	0.1	-0.2	0.3	110	-0.2	-0.2	0	-131	-61	-110	-0.008
G	-0.7	-1.1	-0.3	0.0	0.4	-1.2	0.7	0.1	0.2	0.4	28	0.7	1.9	0	-8	70	197	-0.004
DX	1.3	2.6	0.2	0.2	-1.0	-1.3	-0.4	0.3	0.3	0.8	-99	-0.4	1.8	5	-43	-151	69	-0.001
M	-0.7	2.4	2.3	-0.6	0.3	0.4	0.3	-0.4	-0.8	-0.8	37	0.7	-1.4	12	73	158	9	-0.017
N	-0.4	2.9	2.2	0.0	0.7	-0.3	0.2	-0.7	-1.0	-0.7	-24	0.6	-0.6	0	-144	31	-103	-0.008
NX	0.1	5.2	3.2	-0.2	0.2	-0.7	-0.4	-0.9	-0.7	-0.3	-145	0.3	0.5	11	-845	-253	-329	-0.019
S	-0.8	0.0	0.3	0.7	-0.1	-2.0	0.0	-0.4	0.4	1.3	136	0.4	3.1	--	-54	-200	176	-0.025
Z	1.7	1.2	-0.6	-0.4	-1.5	-2.1	0.9	0.0	0.7	1.9	197	-0.7	3.2	0	-34	87	379	-0.032
CH	-0.9	-0.4	0.3	0.5	-0.9	-1.0	0.7	0.3	-0.4	1.0	82	0.6	-1.2	--	160	184	386	-0.018
TH	-1.7	-0.5	0.1	0.5	1.2	0.1	-0.7	-0.5	0.0	0.0	-23	0.7	0.2	--	-18	-93	-66	-0.007
F	-1.0	-1.0	-0.9	0.0	0.1	0.3	0.3	0.3	-0.1	0.0	38	0.3	-0.8	--	5	-153	-226	-0.010
SH	-1.8	-1.5	-0.2	0.0	-0.6	-0.2	1.6	1.0	-0.3	-0.5	116	0.5	-3.6	--	109	-206	-195	-0.007
JH	-1.1	-0.8	0.3	1.0	-1.0	-0.7	1.4	-0.3	-0.2	0.1	25	0.7	-1.0	0	91	20	10	-0.028
V	0.4	2.3	0.6	0.4	0.1	-0.4	-0.1	-0.3	-0.6	-0.2	-59	-0.3	-0.1	12	-145	-420	-399	-0.010
L	-1.2	2.4	3.8	1.2	0.6	0.0	-0.7	-1.1	-1.4	-0.8	-167	0.2	-1.0	13	42	-1049	-774	-0.007
R	-2.3	1.5	1.5	3.1	1.1	0.2	-0.5	-1.5	-1.6	-1.4	-77	1.3	-1.9	0	21	10	-243	-0.006
Y	-0.7	2.1	1.4	1.1	0.3	-0.3	-0.8	-0.7	-0.4	-0.2	-123	0.6	0.1	-2	8	-121	-37	0.001
HH	2.5	1.0	-0.3	-0.7	0.1	0.9	0.4	0.1	-0.9	-0.5	41	-0.7	-1.2	0	-45	103	109	-0.014
EL	-1.4	3.0	3.7	0.9	0.2	0.6	-0.5	-1.0	-1.4	-1.0	-129	0.1	-1.5	9	26	-907	-480	-0.010
W	0.7	4.3	4.0	-0.1	-0.6	-0.2	-0.3	-0.6	-0.9	-0.6	-236	-0.1	-0.7	0	-19	-472	-232	0.007
EH	-2.7	1.6	2.3	2.4	2.1	0.2	-0.8	-1.4	-2.1	-1.5	-54	0.5	-2.3	9	-17	-102	-60	0.012
AO	-3.2	0.5	3.0	2.7	1.4	0.3	-0.4	-1.1	-1.9	-1.8	-24	0.1	-2.6	0	-7	-261	-239	0.014
AA	-2.9	0.9	2.5	2.6	2.2	0.0	-0.7	-1.5	-1.8	-1.5	-59	0.8	-1.5	12	-27	-229	-487	0.004
UW	-2.0	2.2	1.9	2.0	2.2	0.0	-0.9	-1.4	-1.7	-1.3	-99	1.0	-1.3	10	35	10	26	0.007
ER	-2.3	2.1	3.3	3.1	1.2	-0.1	-0.5	-1.6	-2.0	-1.7	-79	0.5	-1.9	13	43	-51	-324	-0.009
AY	-3.1	1.0	2.8	2.7	2.8	0.1	-0.9	-1.6	-2.2	-1.9	-35	0.8	-2.4	10	-11	-198	-121	0.001
EY	-2.3	3.2	3.1	2.4	2.4	0.2	-1.2	-1.8	-2.0	-1.9	-96	1.7	-2.3	10	26	-58	-379	0.012
AW	-3.7	0.5	2.4	2.5	3.2	1.4	-0.5	-1.9	-2.7	-2.5	23	0.8	-4.3	5	-19	-304	-315	0.009
AX	-2.0	2.5	2.5	2.4	2.2	0.2	-0.6	-1.6	-2.3	-1.7	-93	0.4	-2.7	8	-6	-177	-192	-0.010
IH	-2.7	2.4	2.7	3.0	2.5	0.2	-0.8	-1.9	-2.4	-2.0	-56	1.0	-2.7	0	13	-83	-175	0.006
AE	-3.0	1.2	3.2	3.1	2.9	0.0	-1.4	-1.7	-2.1	-2.1	-52	0.8	-2.1	14	-7	19	69	0.025
AH	-2.4	1.6	3.6	3.1	2.2	-0.2	-1.4	-1.6	-2.1	-1.6	-139	0.2	-1.4	8	16	97	111	0.009
OY	-3.4	2.2	3.4	2.8	3.4	0.5	-1.1	-1.9	-2.8	-2.5	1	1.6	-3.5	2	45	-563	-320	0.004
IY	-1.7	1.9	1.2	1.0	1.4	0.3	-0.5	-0.9	-1.2	-0.8	-60	0.9	-1.4	9	14	-3	174	-0.003
OW	-2.4	1.5	2.8	2.3	1.6	0.2	-0.8	-1.4	-1.6	-1.4	-68	0.0	-1.3	8	57	-80	-154	0.004
AXR	-1.7	2.3	1.9	2.9	1.1	-0.4	0.2	-1.5	-1.8	-1.6	-22	0.8	-1.4	0	-49	-149	24	0.030

Table 38. Average differences in phoneme features between Lombard and normal speech, speaker #3

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-4.3	-2.0	-1.1	0.5	1.3	2.0	1.6	0.0	-1.4	-2.0	209	1.4	-3.8	--	31	-248	-222	-0.021
T	-3.1	-3.0	-1.6	-1.7	-0.7	0.9	1.9	2.0	0.3	-0.4	416	0.5	-2.5	--	108	161	221	-0.010
K	-0.6	-0.9	-1.1	-0.1	0.2	0.5	0.7	0.7	-0.6	-0.5	128	0.2	-1.4	--	215	59	224	-0.025
B	-0.4	1.0	-0.8	-1.3	0.0	0.4	0.2	-0.1	0.5	0.5	-4	-0.5	0.7	0	-166	110	87	0.009
D	-1.3	0.2	-1.6	-0.8	0.1	0.5	0.2	0.9	0.0	0.0	159	0.1	0.4	0	-185	81	-77	-0.007
G	-0.6	-1.7	-2.0	0.2	0.0	0.0	1.1	1.2	-0.8	-0.1	152	0.9	-1.9	0	2	196	221	-0.001
DX	1.2	2.9	0.2	1.3	-0.5	-0.9	-0.7	-0.2	-0.3	0.2	-146	-0.1	1.0	28	-79	-190	70	0.000
M	-1.9	3.5	3.0	2.1	2.2	0.1	-2.1	-1.7	-1.3	-1.4	-84	1.4	0.1	28	106	-32	-294	-0.006
N	-0.8	4.4	2.6	1.4	1.5	-0.5	-1.4	-1.2	-1.2	-0.8	-148	0.9	-0.1	0	-114	-261	-464	-0.010
NX	-0.6	6.6	4.2	0.6	1.2	-0.6	-1.6	-1.4	-1.2	-0.7	-243	0.8	0.3	28	-837	-395	-885	-0.015
S	-1.0	-0.2	0.2	0.1	-0.5	-1.7	-0.3	1.5	0.5	0.5	172	0.3	-1.4	--	-22	-188	298	-0.019
Z	0.3	0.6	-0.6	-0.7	-2.0	-1.7	0.3	1.6	0.9	1.6	265	-0.3	1.2	0	-213	52	315	-0.035
CH	-2.1	-0.4	1.0	0.6	0.1	-0.1	0.3	1.4	-1.4	-0.8	8	1.1	-3.6	--	17	64	-32	-0.011
TH	-1.7	-0.6	0.7	-0.3	-0.3	0.2	0.6	0.8	-0.3	-0.5	103	0.4	-2.4	--	50	79	101	-0.012
F	-1.3	-0.8	0.0	0.5	-0.2	-0.1	0.7	0.7	-0.4	-0.7	92	0.3	-1.5	--	55	-109	-67	-0.006
SH	-4.0	-3.2	-1.5	-1.8	-0.5	1.1	2.1	2.8	0.3	-1.4	322	0.8	-3.6	--	102	6	96	-0.006
JH	-3.4	-1.7	0.1	0.2	-0.2	0.6	1.8	1.6	-1.2	-1.6	107	1.2	-5.2	0	124	45	-41	-0.010
V	0.8	3.1	0.7	1.3	1.0	-0.3	-1.1	-0.9	-1.0	-0.3	-163	-0.1	-0.1	27	-249	-642	-581	-0.007
L	-2.6	1.8	3.8	0.1	1.6	2.4	-1.4	-1.4	-1.7	-1.4	-97	1.1	-0.5	34	50	-1100	-622	-0.003
R	-4.4	-0.7	2.3	5.2	2.1	2.4	-1.5	-2.6	-2.9	-2.7	18	2.8	-3.5	0	63	-121	-1295	0.003
Y	-2.7	1.2	0.9	0.7	2.9	3.2	-1.7	-1.6	-1.9	-1.7	-28	1.2	-2.0	20	15	-95	-68	0.001
HH	2.5	1.6	0.9	-0.1	0.3	0.2	-0.1	-0.2	-0.7	-0.7	-53	-0.5	-0.7	0	30	100	250	-0.015
EL	-3.2	2.3	4.5	1.2	2.0	3.6	-1.6	-2.3	-2.6	-2.3	-38	1.3	-2.7	27	50	-879	-429	-0.011
W	-1.1	3.3	5.5	2.5	0.8	1.4	-1.7	-2.1	-2.2	-2.1	-184	-0.1	-1.5	0	109	-28	-137	0.008
EH	-5.4	-0.3	1.7	2.0	3.3	3.4	-1.3	-1.6	-2.7	-2.6	93	2.0	-5.0	28	3	-194	-86	0.018
AO	-5.2	-1.6	1.7	2.1	2.0	3.6	-0.1	-1.6	-2.5	-2.7	161	0.9	-5.7	0	34	-202	-253	0.021
AA	-5.4	-1.5	1.5	2.1	3.8	3.3	-1.1	-1.9	-2.7	-2.8	120	2.2	-4.6	31	8	-312	-608	0.012
UW	-4.2	0.9	2.6	3.2	3.6	2.8	-1.4	-2.1	-3.3	-3.3	26	1.8	-3.9	30	43	-116	-207	0.010
ER	-4.4	0.6	3.4	5.4	2.4	2.0	-1.5	-2.9	-3.2	-3.1	-25	2.3	-3.9	32	71	-191	-969	0.000
AY	-5.3	-0.6	2.0	2.1	4.3	3.4	-1.4	-2.1	-3.1	-2.8	77	2.2	-5.5	28	11	-408	-125	0.015
EY	-4.5	2.2	3.0	2.5	3.7	2.8	-2.2	-1.9	-3.0	-3.0	-26	2.6	-4.3	28	47	-67	-342	0.023
AW	-5.4	-0.8	1.9	2.5	3.9	3.3	-1.0	-2.1	-3.1	-3.0	112	2.0	-5.8	22	-1	-371	-312	0.023
AX	-4.6	1.0	2.1	2.9	4.5	3.5	-1.5	-2.6	-3.7	-3.4	20	2.3	-6.1	26	10	-296	-211	-0.008
IH	-5.0	0.6	2.1	3.1	5.2	3.4	-2.1	-2.5	-3.7	-3.4	61	2.5	-4.5	0	29	-95	-256	0.008
AE	-3.8	0.5	2.7	2.4	3.6	2.1	-2.1	-1.7	-2.5	-2.3	-18	1.9	-3.9	30	10	-36	-32	0.025
AH	-5.3	-0.8	2.3	3.2	4.3	3.5	-1.3	-2.4	-3.7	-3.4	91	1.7	-5.8	26	59	-87	-112	0.012
OY	-5.2	0.4	3.6	4.3	5.3	2.2	-1.9	-2.9	-3.9	-3.8	66	2.4	-4.5	21	111	-651	-375	-0.001
IY	-4.4	0.8	1.0	1.5	4.8	3.8	-2.5	-1.6	-3.0	-2.7	27	2.0	-2.9	30	29	13	-241	0.007
OW	-4.6	-0.6	2.9	3.7	2.5	3.0	-0.9	-2.5	-3.0	-3.0	76	0.9	-4.2	27	140	277	51	0.012
AXR	-3.9	3.0	4.8	6.7	1.8	0.9	-2.1	-2.9	-3.5	-3.2	-111	1.9	-3.1	0	-12	-331	-965	0.019

Table 39. Average differences in phoneme features between Lombard and loud speech, speaker #3

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.4	-0.5	0.3	1.2	0.7	1.1	0.9	-0.1	-1.3	-2.1	115	0.6	-2.9	--	83	21	28	-0.006
T	-1.4	-1.4	-0.2	-1.0	-0.3	0.8	0.5	1.6	-0.1	-0.7	184	0.3	-2.6	--	119	159	175	-0.003
K	0.1	-0.3	-0.7	-0.6	-0.2	0.6	0.3	0.9	0.1	-0.6	106	0.0	-1.1	--	224	60	311	-0.012
B	-2.1	-0.6	0.8	1.4	0.1	0.8	0.2	-0.1	-0.7	-1.4	32	0.4	-1.9	0	193	44	72	0.004
D	-1.1	-0.3	-0.7	-0.2	0.3	0.4	-0.6	0.7	0.3	-0.3	50	0.3	0.6	0	-54	142	33	0.000
G	0.1	-0.6	-1.7	0.3	-0.3	1.2	0.3	1.1	-1.0	-0.6	124	0.3	-3.8	0	10	126	24	0.003
DX	-0.1	0.3	0.0	1.1	0.6	0.4	-0.4	-0.5	-0.6	-0.6	-47	0.3	-0.8	22	-36	-40	0	0.002
M	-1.2	1.2	0.8	2.7	1.9	-0.3	-2.4	-1.3	-0.5	-0.6	-121	0.7	1.5	16	33	-190	-303	0.012
N	-0.4	1.5	0.4	1.4	0.7	-0.2	-1.6	-0.5	-0.2	-0.1	-124	0.3	0.5	0	30	-292	-361	-0.002
NX	-0.7	1.4	1.1	0.8	0.9	0.1	-1.2	-0.5	-0.5	-0.4	-98	0.5	-0.2	17	9	-143	-556	0.003
S	-0.2	-0.3	-0.2	-0.5	-0.3	0.2	-0.4	1.9	0.1	-0.8	36	-0.1	-4.6	--	32	13	122	0.006
Z	-1.4	-0.6	0.1	-0.3	-0.5	0.4	-0.7	1.6	0.2	-0.3	68	0.3	-1.9	0	-179	-35	-64	-0.003
CH	-1.1	0.0	0.8	0.1	1.0	1.0	-0.4	1.1	-1.0	-1.8	-74	0.6	-2.4	--	-143	-120	-419	0.007
TH	0.0	-0.1	0.6	-0.7	-1.5	0.1	1.3	1.4	-0.3	-0.5	126	-0.3	-2.5	--	68	172	167	-0.006
F	-0.2	0.2	0.9	0.5	-0.3	-0.3	0.4	0.4	-0.3	-0.7	54	0.0	-0.7	--	50	44	159	0.004
SH	-2.2	-1.8	-1.4	-1.8	0.1	1.3	0.6	1.8	0.6	-0.9	206	0.3	0.0	--	-7	211	291	0.001
JH	-2.3	-0.9	-0.2	-0.8	0.8	1.2	0.4	1.9	-0.9	-1.7	82	0.4	-4.2	0	33	25	-51	0.018
V	0.5	0.8	0.1	0.9	0.9	0.0	-1.0	-0.6	-0.5	-0.1	-104	0.2	0.0	15	-103	-222	-182	0.003
L	-1.4	-0.6	0.0	-1.1	1.0	2.4	-0.6	-0.3	-0.4	-0.6	70	0.9	0.5	21	7	-51	152	0.004
R	-2.0	-2.2	0.8	2.2	1.0	2.2	-1.0	-1.1	-1.3	-1.4	95	1.5	-1.7	19	42	-131	-1052	0.009
Y	-2.1	-0.9	-0.5	-0.4	2.6	3.6	-0.9	-0.9	-1.5	-1.5	95	0.6	-2.1	22	7	27	-31	0.000
HH	0.0	0.6	1.2	0.6	0.2	-0.7	-0.5	-0.3	0.2	-0.3	-94	0.1	0.5	0	74	-3	140	-0.002
EL	-1.8	-0.8	0.8	0.2	1.8	2.9	-1.0	-1.2	-1.2	-1.3	91	1.2	-1.3	18	25	28	51	-0.001
W	-1.8	-1.1	1.5	2.6	1.5	1.6	-1.4	-1.5	-1.4	-1.5	52	0.0	-0.8	16	128	445	95	0.001
EH	-2.7	-1.9	-0.6	-0.5	1.3	3.2	-0.5	-0.2	-0.6	-1.1	147	1.6	-2.7	19	20	-91	-26	0.006
AO	-2.0	-2.1	-1.2	-0.5	0.6	3.3	0.2	-0.5	-0.6	-0.9	186	0.8	-3.1	20	40	59	-14	0.008
AA	-2.5	-2.4	-1.0	-0.4	1.6	3.4	-0.4	-0.4	-0.9	-1.3	180	1.4	-3.1	19	36	-83	-121	0.008
UW	-2.3	-1.3	0.7	1.2	1.4	2.7	-0.5	-0.8	-1.6	-2.0	126	0.8	-2.7	20	8	-125	-234	0.003
ER	-2.0	-1.5	0.2	2.3	1.3	2.1	-1.0	-1.3	-1.2	-1.4	54	1.8	-2.0	19	28	-140	-645	0.009
AY	-2.2	-1.6	-0.8	-0.6	1.6	3.3	-0.5	-0.6	-0.8	-0.9	112	1.4	-3.1	18	22	-210	-5	0.013
EY	-2.2	-0.9	0.1	0.2	1.3	2.6	-1.0	-0.1	-1.0	-1.1	70	0.9	-1.9	18	21	-9	36	0.011
AW	-1.7	-1.4	-0.5	0.0	0.7	1.8	-0.5	-0.2	-0.4	-0.5	88	1.2	-1.5	17	19	-66	4	0.014
AX	-2.6	-1.5	-0.4	0.5	2.3	3.3	-0.9	-0.9	-1.4	-1.7	114	1.9	-3.4	17	16	-119	-19	0.001
IH	-2.2	-1.8	-0.6	0.1	2.7	3.2	-1.4	-0.6	-1.3	-1.4	117	1.4	-1.8	21	16	-13	-80	0.002
AE	-0.8	-0.7	-0.5	-0.7	0.7	2.0	-0.8	0.0	-0.4	-0.2	34	1.1	-1.7	16	16	-55	-101	0.000
AH	-3.0	-2.4	-1.3	0.1	2.1	3.8	0.1	-0.8	-1.6	-1.8	230	1.5	-4.4	18	43	-184	-222	0.002
OY	-1.8	-1.8	0.2	1.5	1.9	1.7	-0.9	-1.0	-1.1	-1.3	65	0.9	-1.1	19	67	-88	-56	-0.006
IY	-2.7	-1.1	-0.1	0.6	3.4	3.4	-2.0	-0.7	-1.8	-2.0	87	1.0	-1.6	21	15	15	-415	0.010
OW	-2.2	-2.1	0.1	1.4	0.9	2.8	-0.1	-1.1	-1.3	-1.5	144	0.8	-2.8	18	83	358	205	0.008
AXR	-2.2	0.7	2.9	3.8	0.7	1.3	-2.3	-1.4	-1.6	-1.6	-89	1.0	-1.6	15	37	-181	-989	-0.012

Table 40. Average differences in phoneme features between Loud and normal speech, speaker #4

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	0.0	-1.6	-2.6	-0.6	-0.9	1.4	2.9	0.8	-0.3	-1.3	282	-0.1	-4.2	--	78	13	197	-0.009
T	-1.7	-1.6	-0.2	1.5	-0.7	0.3	1.1	0.3	-1.0	-0.8	66	0.3	-2.0	--	90	300	19	-0.003
K	0.8	0.3	0.4	1.2	-0.9	1.0	0.8	-0.4	-1.8	-0.2	-36	-0.2	-1.1	--	125	-294	-52	-0.004
B	-0.9	-0.7	-1.7	1.0	0.6	1.6	2.0	0.6	-1.7	-2.7	173	0.5	-4.8	36	-249	-440	-652	-0.014
D	2.4	1.6	-0.6	1.2	-0.8	-0.1	0.7	0.2	-1.1	-0.6	-27	-0.7	-0.6	37	26	2	-535	0.000
G	-0.9	0.1	0.4	1.6	3.3	0.7	0.0	-0.5	-2.9	-2.2	-76	1.7	-3.2	5	245	153	292	-0.006
DX	1.1	0.7	-2.9	0.0	1.3	0.6	0.7	0.1	-0.8	-0.7	103	1.0	-2.0	84	-84	58	-20	-0.002
M	-3.6	0.0	-0.4	1.0	0.9	-0.8	-0.1	0.6	-0.4	-0.3	115	0.4	0.5	43	12	-137	-256	0.006
N	-3.6	-0.2	-0.6	0.0	0.7	0.4	0.0	0.3	-0.1	-0.2	87	1.0	-0.5	72	21	120	32	-0.001
NX	-4.6	-1.4	-0.9	-0.8	1.0	1.1	0.2	0.5	0.1	-0.2	136	1.3	-0.7	27	45	16	-481	-0.015
S	-1.0	1.2	3.1	3.1	0.2	-1.2	-1.3	-1.9	-0.9	0.0	-411	0.4	1.1	--	91	-329	-417	0.012
Z	-1.5	-1.2	1.8	2.6	0.9	-1.9	-1.8	-1.0	-0.4	1.0	-178	0.3	2.0	37	201	163	177	0.010
CH	-0.5	0.4	2.6	2.0	-2.0	-1.8	0.0	0.0	-0.5	0.9	-62	0.2	-0.7	--	-152	-503	-65	-0.010
TH	-1.2	-1.2	-0.4	1.0	0.6	1.3	1.4	-0.2	-1.9	-1.4	-67	0.6	-4.4	--	204	180	290	-0.012
F	-0.9	-0.1	1.4	1.2	-1.3	0.0	0.8	-0.1	-1.3	0.2	-114	0.0	-0.9	--	73	-30	88	0.012
SH	-1.5	-0.6	-0.6	-0.7	-2.0	0.2	1.2	1.0	0.2	0.9	106	0.0	-1.7	--	-13	198	234	-0.005
JH	-1.1	0.9	2.5	2.1	-1.3	-1.2	0.8	-0.1	-1.1	-0.4	-114	0.6	-2.4	31	131	-276	-251	0.008
V	0.1	1.6	-2.5	1.0	-0.7	1.5	0.7	-0.1	-0.9	-0.5	76	0.0	-1.8	46	-63	23	-68	0.007
L	-1.1	-0.3	-0.5	1.4	0.7	0.8	0.4	-0.3	-1.2	-1.2	87	-0.6	-2.3	84	30	128	58	0.002
R	-1.1	-1.3	-0.4	1.4	0.4	1.2	0.6	0.4	-1.4	-1.8	143	1.3	-3.3	85	23	-9	-343	0.013
Y	-0.7	-0.8	-0.5	1.0	-0.3	1.8	-0.1	1.1	-1.1	-1.7	124	-0.3	-4.7	68	-5	-58	-45	0.013
HH	-1.1	-1.2	-2.0	0.7	0.2	1.3	0.9	0.2	-1.0	-0.7	161	0.7	-2.1	31	156	232	192	0.012
EL	0.1	-0.2	-1.7	1.9	0.9	1.1	0.6	-0.6	-1.2	-1.7	172	-0.1	-3.2	61	21	149	-38	0.007
W	-0.2	1.1	-0.2	1.3	1.5	1.9	-0.1	-1.1	-1.7	-1.9	106	-0.1	-2.7	79	-3	69	-146	0.013
EH	-2.2	-2.1	-0.7	0.5	-0.1	2.7	0.4	0.1	-1.1	-1.3	203	0.6	-2.4	92	79	58	23	0.029
AO	-0.4	-0.6	-1.7	0.3	0.1	0.4	0.0	0.3	0.4	-0.3	110	0.3	-2.0	98	67	141	115	0.026
AA	-2.2	-2.2	-2.5	-0.1	0.0	2.0	0.9	0.8	0.0	-1.3	273	0.9	-5.4	101	84	105	48	0.034
UW	-0.8	-1.4	0.7	1.7	0.8	0.7	1.0	0.6	-2.4	-2.2	117	-0.4	-3.2	76	28	63	-18	0.051
ER	-1.6	-1.7	-1.7	0.6	0.1	2.0	0.2	0.7	-0.7	-1.3	225	1.1	-2.8	83	43	103	-520	0.016
AY	-0.7	-0.5	-2.2	0.2	-1.3	3.4	1.0	0.3	-0.8	-1.3	228	-0.2	-4.2	80	105	109	102	0.029
EY	-1.0	-0.9	0.2	0.3	-0.7	2.1	0.3	0.4	-1.1	-0.9	109	0.1	-3.4	83	33	44	108	0.038
AW	-1.9	-1.6	-2.9	0.3	-0.9	4.7	0.8	-0.1	-0.9	-1.4	291	0.8	-4.2	81	101	119	103	0.046
AX	-2.9	-1.9	-2.0	0.6	1.9	2.7	0.1	-0.2	-1.1	-1.9	235	1.8	-3.5	71	19	177	128	0.023
IH	-2.1	-1.9	-0.3	0.1	0.0	2.5	0.3	0.2	-1.0	-1.0	141	0.5	-2.8	100	25	109	98	0.016
AE	-0.1	-0.2	-1.3	0.4	-0.7	3.7	0.1	-0.3	-1.1	-1.2	148	-0.2	-2.8	86	67	37	116	0.033
AH	-2.7	-1.8	-0.5	1.7	0.4	1.4	-0.4	0.6	-1.1	-1.4	150	0.9	-2.7	100	75	126	54	0.034
OY	-2.2	-1.1	-1.2	1.5	0.4	2.8	0.9	-0.6	-1.5	-2.2	196	-0.7	-4.7	89	57	623	92	0.029
IY	-0.2	0.8	0.8	0.6	-1.3	1.7	0.7	0.0	-1.1	-1.0	45	0.1	-2.7	73	-18	91	48	0.028
OW	-1.9	-2.1	-1.8	1.4	0.4	1.3	0.9	0.3	-0.8	-1.6	216	0.3	-3.8	81	65	95	-13	0.036
AXR	-1.1	-0.1	-0.3	1.6	0.8	2.2	-0.2	-0.7	-1.5	-1.9	112	0.9	-3.2	62	12	-46	-853	0.042

Table 41. Average differences in phoneme features between Lombard and normal speech, speaker #4

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.0	-1.8	-1.4	1.1	1.2	2.0	2.4	-0.6	-1.4	-3.0	198	0.8	-5.8	--	126	-325	-8	-0.006
T	-1.9	-1.4	-0.8	-1.2	-2.6	0.3	1.7	1.9	0.7	0.4	313	-0.4	-1.0	--	14	-278	2	0.003
K	-0.6	0.4	1.8	1.0	-1.0	-0.9	0.8	-0.3	-0.4	-0.1	-42	0.1	-0.6	--	0	-239	-193	-0.005
B	-1.3	-0.7	0.6	1.5	0.6	1.1	0.4	-1.0	-1.2	-1.3	-11	0.3	-1.7	1	73	-34	-809	-0.011
D	-0.3	-0.3	1.1	0.7	-0.7	0.5	0.3	-0.3	-0.6	-0.4	-2	-0.6	-0.7	6	-28	48	38	0.003
G	-1.7	-1.0	2.8	0.8	0.6	0.4	-1.1	0.0	-1.4	-0.5	-244	0.7	-0.9	23	46	12	249	-0.002
DX	-0.8	-1.1	0.4	0.4	-0.1	1.0	-0.2	-0.5	-0.1	-0.1	-63	0.5	0.0	7	24	-38	232	0.002
M	-2.0	0.7	2.1	2.0	2.2	-1.4	-1.0	-0.8	-1.0	-0.8	-90	0.4	0.6	9	94	-12	53	0.001
N	-2.2	0.9	2.2	1.0	1.6	-0.6	-0.8	-0.7	-0.7	-0.6	-104	1.1	0.1	16	83	94	-35	0.005
NX	-2.9	0.3	3.2	0.8	-0.1	0.1	-0.4	-0.5	-0.6	-0.4	-89	1.2	0.1	5	118	-61	-412	0.008
S	-1.9	-0.8	-0.1	-0.7	-1.5	-0.5	0.6	0.8	0.8	1.1	282	0.0	0.6	--	-4	-194	-65	0.026
Z	-2.3	-2.8	-0.7	-1.4	-1.5	0.1	1.1	1.3	1.0	1.0	374	0.3	-1.0	19	55	47	380	0.010
CH	-0.2	-0.4	-0.2	-0.4	-1.0	0.0	0.8	-0.1	0.0	1.0	80	-0.1	-1.5	--	47	55	406	0.012
TH	-0.4	-0.2	0.1	0.0	-0.8	0.0	1.1	0.5	-0.3	-0.5	-8	-0.4	-1.5	--	48	31	132	0.001
F	-1.1	-0.5	0.3	-0.2	-0.7	0.4	0.6	0.5	-0.6	0.0	54	0.2	-1.0	--	28	52	41	0.002
SH	-1.1	0.3	0.6	-0.8	-1.3	0.5	0.6	0.4	0.0	0.7	34	0.3	-0.7	--	-109	46	190	0.013
JH	-3.2	-1.3	0.5	0.1	-0.6	0.3	1.2	0.3	-0.1	-0.4	124	1.3	-1.8	2	64	-138	-270	0.057
V	-0.5	0.3	2.1	0.8	-1.2	0.6	0.2	-0.1	-0.7	-0.6	2	-1.1	-0.8	-2	35	-81	135	0.009
L	-2.1	-1.7	0.7	1.2	0.2	1.3	-0.4	-0.7	-0.7	-0.5	9	-0.5	-0.8	-4	40	101	162	-0.004
R	-1.3	-1.6	1.2	0.9	0.2	0.2	-0.3	-0.2	-0.4	-0.4	11	1.2	-0.5	9	31	-70	74	0.014
Y	-1.8	-0.9	1.7	1.0	-0.5	0.6	-0.5	0.0	-0.4	-0.6	-11	0.5	0.9	6	49	-73	-161	-0.002
HH	-1.7	-0.7	0.1	0.2	-0.9	0.6	0.7	0.2	-0.4	0.1	130	0.2	-0.7	0	-1	-77	9	0.011
EL	-1.4	-1.6	1.3	1.3	-0.2	0.5	-0.1	-0.4	-0.6	-0.5	3	-0.4	0.0	-4	33	-10	-48	0.013
W	-1.6	-2.0	0.7	2.1	1.1	0.2	-0.5	-1.0	-0.8	-0.8	-2	0.0	-0.5	11	44	-81	-460	-0.010
EH	-1.9	-1.7	1.4	0.4	-1.5	0.2	0.7	0.2	-0.1	0.0	22	-0.4	0.3	7	30	-67	-3	0.011
AO	-1.0	-0.9	0.4	0.8	0.0	0.4	0.1	-0.3	-0.7	0.0	-22	0.7	-0.1	7	9	1	-173	-0.002
AA	-2.8	-2.8	0.0	0.8	0.0	0.9	0.3	0.0	-0.4	-0.5	73	1.7	-1.1	10	29	65	-60	0.017
UW	-0.9	-0.6	1.1	0.7	-0.2	2.1	-0.5	-0.3	-1.2	-0.8	-6	-0.7	-1.1	4	33	27	-29	0.014
ER	-1.7	-1.9	0.4	0.6	0.2	1.0	0.0	-0.3	-0.7	-0.4	51	1.2	0.0	11	23	6	-70	0.002
AY	-1.5	-1.4	0.5	0.4	-1.0	0.5	0.7	0.2	-0.2	-0.1	43	0.0	-1.5	2	29	-18	-56	0.008
EY	-1.0	-0.9	1.7	1.2	-1.5	0.6	-0.1	-0.5	-0.3	0.1	-48	-0.3	1.0	5	38	-142	-29	0.019
AW	-1.7	-1.5	0.2	0.1	-1.5	1.3	1.1	0.2	-0.1	-0.4	87	0.7	-2.3	9	26	-7	-4	0.017
AX	-2.9	-2.1	1.6	1.0	0.1	0.9	-0.1	-0.5	-0.6	-0.6	13	0.6	0.0	8	44	2	104	0.011
IH	-1.9	-1.6	1.5	0.8	-1.1	1.3	0.0	-0.3	-0.4	-0.3	-3	-0.3	0.2	10	38	-93	29	0.003
AE	-0.9	-0.8	0.6	0.2	-1.5	0.9	0.8	0.1	-0.1	-0.3	54	-0.4	-1.2	8	10	-39	17	0.020
AH	-2.5	-2.3	0.6	0.7	-0.8	1.0	0.7	0.1	-0.3	-0.6	103	0.6	-1.6	9	33	3	-110	0.014
OY	-2.6	-2.3	0.3	1.6	0.4	1.1	-0.2	-0.5	-0.7	-1.0	57	-0.5	-1.4	4	41	295	-257	0.018
IY	-1.7	-0.4	2.5	1.7	-0.8	0.8	-0.4	-0.5	-0.8	-0.8	-25	0.5	-0.6	11	58	-85	-116	0.008
OW	-1.7	-2.2	0.3	1.8	-0.1	0.7	0.2	-0.4	-0.7	-0.8	47	0.2	-1.0	5	41	95	-133	0.006
AXR	-2.7	-1.6	1.8	1.3	0.2	0.6	-0.4	-0.7	-0.7	-0.6	-32	1.5	-0.6	11	25	-137	-251	0.009

Table 42. Average differences in phoneme features between Lombard and loud speech, speaker #4

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.0	-0.2	1.1	1.7	2.1	0.6	-0.5	-1.5	-1.1	-1.6	-84	0.8	-1.6	--	47	-338	-203	0.002
T	-0.2	0.2	-0.7	-2.7	-2.0	0.0	0.6	1.6	1.7	1.1	247	-0.7	1.0	--	-76	22	-17	0.006
K	-1.4	0.1	1.4	-0.1	-0.1	-1.9	0.0	0.2	1.4	0.1	-8	0.3	0.5	--	-125	54	-141	-0.001
B	-0.3	0.0	2.3	0.5	0.0	-0.6	-1.7	-1.6	0.5	1.4	-184	-0.2	3.1	-35	322	405	-156	0.003
D	-2.8	-1.9	1.7	-0.6	0.1	0.6	-0.5	-0.4	0.6	0.2	25	0.1	-0.2	-30	-54	45	573	0.002
G	-0.9	-1.0	2.4	-0.8	-2.8	-0.3	-1.1	0.5	1.6	1.7	-168	-0.9	2.3	18	-198	-140	-43	0.004
DX	-1.9	-1.8	3.3	0.4	-1.4	0.3	-0.9	-0.7	0.6	0.5	-165	-0.5	2.0	-77	108	-96	251	0.005
M	1.6	0.6	2.4	1.0	1.2	-0.6	-0.9	-1.4	-0.6	-0.5	-205	0.0	0.1	-34	82	124	308	-0.006
N	1.4	1.0	2.7	1.0	0.9	-1.0	-0.7	-1.1	-0.6	-0.4	-190	0.1	0.6	-56	62	-28	-67	0.006
NX	1.7	1.6	4.2	1.6	-1.1	-1.0	-0.6	-1.0	-0.6	-0.2	-225	-0.1	0.8	-22	73	-78	69	0.023
S	-0.9	-1.9	-3.2	-3.8	-1.7	0.7	2.0	2.6	1.7	1.1	692	-0.4	-0.5	--	-96	135	352	0.013
Z	-0.9	-1.6	-2.5	-4.0	-2.3	2.1	2.9	2.3	1.4	0.0	552	0.0	-3.0	-18	-146	-116	204	0.000
CH	0.3	-0.8	-2.8	-2.4	1.0	1.8	0.9	-0.1	0.4	0.2	142	-0.3	-0.8	--	199	558	471	0.022
TH	0.7	1.0	0.5	-1.0	-1.4	-1.3	-0.3	0.7	1.6	0.9	59	-1.0	2.9	--	-156	-128	-158	0.013
F	-0.2	-0.4	-1.1	-1.4	0.7	0.5	-0.2	0.6	0.7	-0.2	168	0.2	-0.1	--	-45	82	-47	-0.010
SH	0.4	0.8	1.2	-0.1	0.7	0.3	-0.7	-0.6	-0.3	-0.2	-72	0.4	1.0	--	-96	-151	-44	0.018
JH	-2.1	-2.2	-2.0	-2.0	0.7	1.5	0.4	0.4	1.0	0.0	237	0.7	0.7	-28	-67	138	-19	0.049
V	-0.6	-1.3	4.6	-0.2	-0.5	-0.9	-0.4	0.0	0.2	-0.1	-74	-1.1	1.0	-48	98	-104	203	0.002
L	-1.0	-1.3	1.3	-0.2	-0.5	0.5	-0.7	-0.4	0.5	0.7	-78	0.1	1.6	-88	10	-27	104	-0.006
R	-0.2	-0.3	1.6	-0.5	-0.1	-0.9	-0.9	-0.7	1.0	1.4	-132	-0.1	2.9	-76	7	-61	418	0.001
Y	-1.1	-0.1	2.2	0.1	-0.2	-1.2	-0.4	-1.1	0.6	1.1	-134	0.8	5.6	-62	54	-16	-116	-0.015
HH	-0.6	0.4	2.1	-0.5	-1.1	-0.7	-0.2	0.1	0.5	0.8	-31	-0.5	1.4	-31	-157	-309	-184	-0.002
EL	-1.4	-1.4	3.0	-0.6	-1.1	-0.6	-0.7	0.2	0.6	1.2	-169	-0.3	3.1	-66	13	-159	-10	0.006
W	-1.3	-3.1	0.9	0.8	-0.4	-1.7	-0.4	0.1	0.9	1.0	-108	0.2	2.2	-68	47	-150	-313	-0.023
EH	0.2	0.4	2.1	-0.1	-1.4	-2.5	0.3	0.1	1.0	1.3	-182	-1.0	2.6	-85	-49	-125	-26	-0.017
AO	-0.6	-0.3	2.1	0.5	0.0	-0.1	0.1	-0.6	-1.1	0.3	-132	0.4	1.9	-91	-58	-139	-288	-0.029
AA	-0.6	-0.6	2.5	0.9	0.0	-1.1	-0.6	-0.8	-0.3	0.8	-200	0.8	4.3	-91	-54	-40	-108	-0.017
UW	-0.1	0.9	0.5	-1.1	-1.0	1.4	-1.5	-0.9	1.2	1.4	-123	-0.4	2.0	-72	6	-36	-12	-0.038
ER	-0.1	-0.2	2.0	0.0	0.2	-1.0	-0.2	-1.0	0.0	0.9	-174	0.1	2.9	-72	-20	-96	450	-0.013
AY	-0.8	-1.0	2.7	0.2	0.3	-2.9	-0.3	-0.2	0.6	1.1	-185	0.3	2.7	-78	-77	-128	-158	-0.021
EY	0.0	0.0	1.5	0.9	-0.9	-1.5	-0.4	-0.9	0.8	0.9	-157	-0.4	4.4	-78	5	-186	-137	-0.019
AW	0.1	0.1	3.1	-0.2	-0.7	-3.3	0.3	0.4	0.8	1.1	-204	-0.2	1.9	-72	-75	-126	-107	-0.029
AX	0.1	-0.2	3.6	0.4	-1.8	-1.8	-0.3	-0.3	0.5	1.3	-222	-1.2	3.5	-62	25	-175	-24	-0.012
IH	0.2	0.4	1.9	0.7	-1.1	-1.2	-0.4	-0.5	0.6	0.7	-144	-0.8	3.0	-89	13	-203	-69	-0.012
AE	-0.7	-0.6	1.9	-0.3	-0.8	-2.7	0.7	0.4	1.0	1.0	-94	-0.3	1.6	-79	-57	-76	-99	-0.012
AH	0.2	-0.4	1.1	-1.0	-1.2	-0.4	1.1	-0.6	0.8	0.8	-47	-0.3	1.1	-91	-41	-123	-165	-0.020
OY	-0.4	-1.2	1.5	0.1	0.1	-1.7	-1.1	0.1	0.8	1.2	-139	0.2	3.3	-85	-17	-328	-349	-0.011
IY	-1.5	-1.3	1.6	1.1	0.5	-1.0	-1.1	-0.5	0.3	0.3	-70	0.5	2.1	-62	75	-176	-164	-0.020
OW	0.1	-0.2	2.0	0.4	-0.5	-0.6	-0.7	-0.6	0.0	0.8	-169	0.0	2.8	-75	-23	0	-120	-0.029
AXR	-1.6	-1.5	2.2	-0.2	-0.7	-1.6	-0.2	0.0	0.9	1.3	-144	0.6	2.6	-52	13	-91	602	-0.033

Table 43. Average differences in phoneme features between Loud and normal speech, speaker #5

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-0.8	-0.6	-1.6	-1.4	0.4	0.7	0.3	0.3	0.4	0.5	56	-0.1	-0.3	--	-71	139	103	-0.011
T	-2.4	-2.0	-1.6	-0.6	-0.7	-0.5	0.7	1.0	1.2	0.7	212	0.3	0.8	--	-44	-32	236	0.001
K	-1.3	0.3	0.7	1.6	1.1	0.1	-0.1	-0.8	-1.1	-1.0	-2	0.6	-1.5	--	-41	-5	-369	-0.008
B	0.3	0.8	0.1	-0.7	0.0	-0.5	-1.1	0.0	0.5	1.5	-106	-0.5	2.3	56	-40	-354	-156	-0.003
D	0.3	0.2	-1.6	0.4	-0.1	-0.2	0.2	-0.4	0.2	0.5	-40	0.5	0.7	62	-17	-180	-535	-0.005
G	0.2	2.3	2.6	0.9	0.4	-0.4	0.0	-0.7	-1.3	-0.7	-97	-0.5	-0.3	70	-76	-50	-74	-0.013
DX	-0.8	-0.3	-1.4	1.3	2.5	1.8	-0.9	-1.0	-1.4	-1.4	25	1.5	-0.7	80	34	84	-106	-0.004
M	-1.0	-0.7	-0.9	1.4	0.7	0.4	0.4	-0.5	-0.7	-0.9	88	-0.3	-1.4	53	-17	-390	-801	0.001
N	0.3	-0.1	-0.7	1.2	1.2	0.1	-0.1	-0.6	-0.7	-0.8	-24	0.3	-0.8	52	-17	-58	-177	0.003
NX	1.0	-0.1	0.2	0.9	3.0	0.4	-0.6	-1.4	-1.2	-1.4	-23	0.4	-1.0	33	11	43	-302	0.002
S	0.1	-1.2	-1.4	-0.6	-0.4	-1.0	-0.2	0.8	1.3	0.9	250	0.0	2.1	--	19	-28	-16	0.008
Z	2.2	1.0	1.0	0.7	-1.5	-3.0	0.0	1.2	0.8	0.6	46	-1.0	2.1	45	-168	-957	-636	0.021
CH	-1.0	-1.2	-1.2	-1.1	-0.9	1.0	1.0	0.1	0.4	0.6	176	-0.2	-0.8	--	98	-76	-127	-0.017
TH	-2.0	-1.9	-1.8	-0.6	0.8	0.1	-0.3	0.0	1.0	0.6	98	0.7	0.5	--	-46	321	629	-0.013
F	-1.2	-2.4	-3.3	-2.9	-0.6	0.1	0.7	1.7	2.2	1.3	279	0.0	1.6	--	-166	161	84	-0.007
SH	-2.8	-3.1	-2.1	-1.6	-1.0	1.4	1.6	0.8	0.6	0.5	286	0.5	-1.8	--	335	74	-101	-0.005
JH	-0.6	-1.4	-0.8	-0.9	-0.5	0.9	0.6	0.1	0.3	0.3	122	0.0	-0.4	3	-195	225	243	-0.001
V	-0.2	-0.6	-1.5	-0.4	1.1	0.9	-0.5	0.1	0.0	-0.2	26	0.8	-0.2	53	-55	-128	-28	0.002
L	-0.7	0.0	0.7	0.9	1.7	1.6	-0.4	-1.4	-1.3	-1.5	-26	0.3	-1.8	62	6	-126	-419	0.001
R	-1.4	-1.2	0.6	0.6	0.1	0.4	0.6	-0.4	-0.5	-0.5	33	0.8	-1.7	70	23	-55	15	0.011
Y	-1.2	-2.2	-2.1	-0.4	0.1	1.4	1.1	-0.1	-0.1	-0.2	153	0.0	-2.3	65	-20	24	349	0.035
HH	-2.0	-0.9	0.0	-0.2	0.6	0.5	-0.9	0.0	0.4	0.2	44	0.2	0.8	53	-7	-43	-162	0.009
EL	-2.1	-2.4	-0.2	1.4	1.4	2.8	0.0	-0.9	-1.7	-2.0	139	0.9	-3.8	35	53	110	-40	0.015
W	-1.1	-0.8	-1.6	-0.2	-0.2	-0.3	0.4	0.4	0.5	0.5	16	0.0	0.3	57	14	620	437	0.042
EH	-3.8	-4.0	-1.5	0.5	3.2	2.0	1.9	-0.6	-1.6	-2.5	266	2.4	-5.8	64	68	163	-85	0.015
AO	-3.8	-3.6	-2.3	0.6	0.4	1.5	2.5	0.3	-0.7	-1.7	276	2.1	-4.1	79	127	1	-19	0.014
AA	-2.4	-2.7	-1.2	0.9	0.5	1.3	1.6	-0.2	-0.8	-1.6	196	1.8	-4.0	71	93	40	-48	0.011
UW	-1.7	-1.3	0.9	1.3	1.0	0.0	0.7	-0.7	-1.0	-1.0	39	0.3	-2.9	65	16	-173	-212	0.003
ER	-2.0	-2.2	-0.2	1.5	0.5	1.1	0.6	-0.6	-0.9	-1.2	86	1.8	-2.0	74	65	68	4	0.013
AY	-3.6	-4.0	-3.1	0.8	2.2	2.5	2.2	-0.5	-1.3	-2.4	296	2.4	-6.0	59	119	66	35	0.012
EY	-2.8	-2.9	-0.7	0.2	1.8	1.1	2.2	-0.2	-1.4	-1.9	199	1.1	-6.3	64	46	45	359	0.026
AW	-2.2	-3.0	-2.2	1.4	1.5	1.9	1.3	-0.5	-1.1	-2.2	235	2.2	-4.3	56	113	187	27	0.006
AX	-2.2	-2.0	-1.5	0.9	1.8	1.6	0.5	-0.9	-0.9	-1.3	126	1.9	-2.7	41	45	195	5	-0.001
IH	-3.5	-3.2	-1.0	-0.3	2.6	2.4	1.7	-0.7	-1.6	-2.0	208	2.2	-6.0	75	53	173	37	0.005
AE	-4.1	-4.4	-2.7	0.4	2.7	2.2	2.0	-0.6	-1.3	-2.0	273	2.2	-5.5	66	98	101	-122	0.025
AH	-4.7	-4.5	-2.4	1.6	2.6	1.5	2.1	-0.4	-1.6	-2.3	265	2.9	-6.3	68	114	224	-152	0.017
OY	-1.7	-1.9	-0.5	0.4	1.9	2.2	1.0	-1.0	-1.6	-1.7	136	1.5	-4.7	59	71	769	139	0.015
IY	-1.9	-1.9	-0.9	0.0	1.9	2.3	1.8	-1.3	-1.5	-1.7	126	0.8	-6.0	57	5	20	51	0.001
OW	-2.0	-2.2	-0.2	0.9	1.2	1.3	0.7	-0.9	-1.0	-1.1	96	1.0	-2.5	62	81	275	144	0.013
AXR	-2.9	-2.2	0.7	2.7	1.8	2.4	-0.6	-1.9	-1.8	-1.9	80	2.4	-1.8	29	29	34	-38	0.054

Table 44. Average differences in phoneme features between Lombard and normal speech, speaker #5

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)											COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	Lo		Hi	1		2	3		
P	4.8	2.7	0.6	1.0	2.4	1.2	-0.2	-2.1	-1.2	-2.6	70	-0.9	-3.2	--	-49	60	-120	-0.009	
T	1.6	2.9	2.3	2.2	1.2	-1.3	-0.9	-2.8	0.5	-0.8	-26	-0.7	0.7	--	-55	-362	-274	0.011	
K	1.2	1.1	-0.8	0.3	1.6	-0.3	0.4	-2.2	0.5	-0.4	105	-0.1	-0.1	--	15	31	-293	0.003	
B	2.3	1.0	1.4	1.5	2.6	0.8	-0.7	-1.9	-1.4	-2.2	8	0.0	-2.2	16	-45	8	-439	-0.002	
D	-0.4	-1.1	-1.0	0.6	2.1	0.6	0.1	-2.1	0.4	-0.7	93	0.4	-1.1	54	24	-67	-296	0.002	
G	2.7	1.2	0.6	0.3	0.4	-0.5	-0.4	-1.1	0.1	0.2	16	-0.2	1.4	0	58	474	-265	-0.007	
DX	0.2	-2.0	-2.7	0.2	1.4	0.6	0.2	-0.5	0.2	-0.3	176	1.3	-0.9	24	24	73	79	0.000	
M	1.2	-1.1	-2.7	-1.1	-1.2	0.8	0.9	0.4	0.6	0.9	125	-0.6	-0.1	8	-50	-161	-584	-0.011	
N	1.4	-0.5	-2.4	0.6	0.5	0.3	0.2	-0.4	-0.1	0.0	53	0.0	-0.5	12	-40	-378	-440	-0.001	
NX	2.6	-0.4	-1.9	-0.1	2.1	1.2	-0.3	-1.1	-0.6	-0.8	26	-0.4	-0.3	14	-44	61	-95	0.020	
S	3.0	4.6	3.6	1.8	1.0	-1.5	-0.5	-4.1	0.3	-0.1	-255	-0.9	3.9	--	-165	-343	-453	0.025	
Z	1.7	1.8	3.4	1.9	-0.1	-3.3	-0.2	-2.9	1.3	1.1	-51	-0.8	5.9	36	-109	-690	-410	0.036	
CH	2.9	4.9	3.8	1.8	1.5	-1.3	-0.9	-3.6	0.1	-0.8	-166	0.0	-0.5	--	-85	-242	-399	-0.007	
TH	1.0	-0.4	-2.0	0.0	1.3	0.1	-0.4	-2.1	1.3	0.8	123	0.2	0.5	--	4	293	298	0.011	
F	1.3	0.3	-1.8	-2.8	-0.5	-0.6	0.1	0.1	2.4	1.9	217	-1.1	2.8	--	-97	113	94	0.015	
SH	1.5	4.5	4.7	2.4	0.0	-2.0	0.4	-3.2	-0.3	-0.4	-78	0.7	1.2	--	-403	-414	-383	0.011	
JH	3.4	4.8	4.0	2.0	1.0	-2.0	-0.2	-3.2	-0.4	-0.7	-173	-0.1	1.7	71	-412	-316	-263	0.034	
V	1.3	0.3	-1.7	-1.9	-0.3	-0.1	0.2	0.1	1.2	1.2	7	0.1	1.3	17	-84	-183	152	0.005	
L	1.2	-0.1	-0.7	0.3	3.1	0.7	-0.2	-1.5	-0.9	-1.2	-6	1.5	-1.4	22	-27	-70	-86	0.006	
R	0.0	-1.6	-1.6	1.1	2.8	0.9	-0.1	-1.0	-1.2	-1.4	118	1.8	-2.0	11	-2	-36	-795	0.002	
Y	-0.2	-2.1	-0.9	1.3	3.3	0.5	0.1	-1.6	-0.9	-1.7	36	-0.2	-1.8	3	3	-135	-358	-0.006	
HH	1.8	1.3	0.1	0.3	1.7	0.4	-0.5	-2.1	0.1	-0.6	-55	-0.4	-0.8	39	-167	-367	-415	0.012	
EL	1.1	-0.6	-1.7	-0.2	2.0	1.9	-0.2	-0.9	-0.8	-1.0	82	1.0	-1.3	10	0	-24	-129	0.005	
W	-0.1	-1.1	-1.9	-0.1	1.0	0.9	0.2	-0.2	-0.2	-0.4	57	0.6	-0.7	18	15	455	152	0.018	
EH	-0.1	-1.7	-2.4	0.0	2.3	0.9	0.7	-0.2	-0.3	-1.4	213	1.6	-3.0	14	7	-30	-181	0.008	
AO	0.7	-1.1	-1.9	0.2	0.4	-0.1	0.9	0.1	-0.1	-0.3	90	0.6	-1.2	5	36	-103	-173	-0.006	
AA	0.8	-1.7	-2.1	0.1	1.0	0.1	1.2	0.0	-0.2	-0.8	163	0.9	-1.9	9	28	36	73	0.012	
UW	-0.3	-1.7	-0.8	1.4	2.3	1.2	0.1	-1.0	-1.3	-1.8	113	1.0	-3.1	15	4	-193	-258	-0.013	
ER	0.5	-1.6	-2.1	0.9	1.3	1.0	0.6	-0.4	-0.8	-1.4	150	1.3	-2.4	10	4	-8	-373	0.006	
AY	1.0	-0.9	-2.2	-0.1	1.2	0.2	0.5	0.0	-0.1	-0.6	116	0.7	-1.3	7	16	-22	-48	-0.001	
EY	-1.0	-3.3	-3.1	-0.4	1.5	-0.3	0.7	1.5	0.5	-0.8	250	0.3	-1.7	4	-1	-81	-167	0.009	
AW	1.5	-0.7	-1.5	0.4	1.3	-0.5	-0.1	-0.2	0.0	-0.5	54	0.5	-0.1	4	20	39	-98	0.019	
AX	0.0	-1.4	-1.9	0.6	1.9	0.8	0.4	-0.7	-0.4	-1.3	128	1.5	-1.9	11	2	34	42	0.005	
IH	-0.8	-2.7	-2.4	0.1	2.3	1.3	0.8	0.0	-0.7	-1.5	221	1.5	-3.3	7	11	1	3	0.004	
AE	0.5	-1.0	-2.0	-0.5	1.3	0.7	0.5	0.1	0.0	-0.7	146	0.6	-1.5	16	33	-29	-22	0.017	
AH	-0.4	-1.9	-2.8	-0.5	1.5	1.2	1.3	0.0	-0.3	-1.0	212	1.1	-3.6	7	12	152	28	-0.002	
OY	0.3	-2.6	-1.6	0.8	1.6	1.9	0.3	-0.8	-1.0	-1.5	138	0.8	-2.2	7	55	444	86	0.010	
IY	-0.3	-1.4	-2.2	-0.3	2.7	0.8	0.9	-0.6	-0.7	-1.1	119	-0.1	-3.3	15	-11	-106	-386	0.012	
OW	0.0	-1.3	-1.1	0.9	2.5	1.4	0.0	-1.1	-1.3	-1.5	92	1.3	-2.1	10	21	213	-21	0.015	
AXR	-0.2	-1.1	-1.2	2.3	2.4	0.7	-0.2	-1.2	-1.5	-1.8	110	1.8	-2.2	9	-6	16	-574	0.019	

Table 45. Average differences in phoneme features between Lombard and loud speech, speaker #5

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	5.7	3.3	2.2	2.4	2.0	0.5	-0.5	-2.3	-1.6	-3.1	14	-0.8	-2.9	--	22	-80	-223	0.003
T	4.0	4.9	3.9	2.8	2.0	-0.8	-1.7	-3.8	-0.6	-1.5	-238	-1.0	0.0	--	-11	-330	-510	0.010
K	2.6	0.9	-1.5	-1.2	0.5	-0.4	0.5	-1.4	1.6	0.6	107	-0.7	1.3	--	56	36	76	0.011
B	2.0	0.2	1.3	2.2	2.7	1.3	0.4	-1.8	-1.9	-3.7	114	0.5	-4.5	-40	-6	362	-283	0.001
D	-0.7	-1.4	0.6	0.2	2.2	0.8	-0.1	-1.7	0.2	-1.3	133	-0.1	-1.7	-7	41	113	239	0.008
G	2.5	-1.1	-2.0	-0.6	0.0	-0.2	-0.4	-0.4	1.5	0.9	113	0.3	1.8	-70	134	524	-192	0.006
DX	1.0	-1.7	-1.3	-1.0	-1.2	-1.2	1.1	0.6	1.6	1.0	152	-0.1	-0.1	-56	-10	-11	185	0.004
M	2.1	-0.4	-1.8	-2.5	-1.9	0.4	0.5	0.9	1.3	1.8	36	-0.2	1.4	-44	-32	229	217	-0.013
N	1.0	-0.5	-1.8	-0.6	-0.7	0.2	0.3	0.2	0.5	0.8	77	-0.3	0.3	-40	-23	-319	-263	-0.004
NX	1.6	-0.3	-2.1	-0.9	-0.9	0.8	0.3	0.3	0.6	0.7	50	-0.8	0.6	-20	-55	18	207	0.018
S	2.9	5.8	5.0	2.4	1.4	-0.5	-0.3	-4.9	-1.0	-1.1	-504	-0.9	1.8	--	-183	-314	-437	0.017
Z	-0.5	0.8	2.4	1.2	1.3	-0.3	-0.2	-4.1	0.5	0.5	-97	0.2	3.8	-9	59	267	226	0.015
CH	3.9	6.2	5.0	2.9	2.4	-2.3	-1.9	-3.7	-0.3	-1.4	-342	0.1	0.3	--	-183	-166	-273	0.009
TH	3.1	1.5	-0.3	0.6	0.4	-0.1	-0.1	-2.1	0.3	0.2	25	-0.5	0.0	--	50	-28	-331	0.024
F	2.5	2.7	1.5	0.1	0.1	-0.7	-0.5	-1.6	0.3	0.6	-62	-1.1	1.2	--	69	-48	10	0.023
SH	4.4	7.6	6.8	4.0	1.0	-3.5	-1.2	-4.0	-0.9	-1.0	-364	0.2	3.0	--	-737	-488	-282	0.016
JH	3.9	6.2	4.8	2.9	1.5	-2.9	-0.8	-3.3	-0.7	-0.9	-294	0.0	2.1	69	-216	-540	-506	0.035
V	1.5	0.9	-0.1	-1.4	-1.3	-1.0	0.7	0.0	1.2	1.5	-19	-0.7	1.6	-36	-29	-55	180	0.004
L	1.9	-0.2	-1.4	-0.6	1.4	-0.9	0.2	-0.2	0.4	0.3	20	1.2	0.4	-40	-33	56	333	0.004
R	1.4	-0.3	-2.2	0.5	2.7	0.6	-0.8	-0.6	-0.7	-0.9	85	1.1	-0.3	-59	-24	19	-810	-0.009
Y	1.1	0.1	1.2	1.7	3.2	-0.9	-1.0	-1.6	-0.8	-1.5	-117	-0.3	0.5	-62	23	-159	-707	-0.041
HH	3.8	2.2	0.1	0.5	1.1	-0.1	0.4	-2.0	-0.3	-0.8	-100	-0.6	-1.6	-14	-161	-323	-253	0.003
EL	3.3	1.8	-1.6	-1.6	0.6	-0.8	-0.2	0.0	0.9	0.9	-57	0.1	2.5	-24	-53	-133	-89	-0.011
W	1.0	-0.3	-0.3	0.1	1.2	1.2	-0.2	-0.6	-0.7	-0.9	42	0.6	-1.0	-39	1	-165	-285	-0.024
EH	3.6	2.3	-0.9	-0.5	-0.9	-1.1	-1.2	0.4	1.3	1.1	-53	-0.8	2.8	-49	-60	-193	-96	-0.007
AO	4.4	2.5	0.4	-0.4	0.0	-1.6	-1.6	-0.2	0.5	1.5	-186	-1.5	2.9	-74	-91	-104	-154	-0.020
AA	3.2	1.0	-0.9	-0.7	0.5	-1.2	-0.5	0.2	0.6	0.8	-34	-0.9	2.0	-62	-65	-5	121	0.001
UW	1.4	-0.3	-1.7	0.1	1.3	1.2	-0.6	-0.3	-0.3	-0.8	73	0.7	-0.2	-49	-12	-20	-45	-0.016
ER	2.5	0.5	-1.9	-0.5	0.9	0.0	0.0	0.2	0.1	-0.1	65	-0.5	-0.3	-64	-61	-76	-377	-0.007
AY	4.6	3.1	0.9	-0.9	-1.0	-2.2	-1.7	0.5	1.2	1.8	-180	-1.7	4.7	-52	-103	-88	-83	-0.013
EY	1.8	-0.4	-2.4	-0.6	-0.4	-1.5	-1.5	1.7	1.9	1.0	51	-0.8	4.6	-61	-46	-126	-526	-0.017
AW	3.7	2.4	0.6	-1.0	-0.2	-2.4	-1.4	0.3	1.1	1.8	-181	-1.7	4.2	-52	-93	-148	-126	0.013
AX	2.2	0.6	-0.4	-0.3	0.1	-0.8	-0.1	0.2	0.5	0.0	1	-0.4	0.7	-31	-43	-161	37	0.006
IH	2.7	0.5	-1.4	0.4	-0.4	-1.1	-1.0	0.7	0.9	0.4	12	-0.7	2.7	-68	-42	-172	-35	-0.001
AE	4.7	3.4	0.6	-1.0	-1.3	-1.5	-1.5	0.7	1.3	1.2	-127	-1.6	4.0	-50	-65	-130	100	-0.007
AH	4.3	2.6	-0.4	-2.1	-1.1	-0.3	-0.8	0.4	1.4	1.3	-53	-1.8	2.6	-61	-102	-72	180	-0.020
OY	2.0	-0.7	-1.1	0.4	-0.2	-0.4	-0.8	0.3	0.6	0.2	1	-0.7	2.5	-51	-16	-326	-53	-0.005
IY	1.5	0.5	-1.3	-0.3	0.7	-1.5	-0.9	0.6	0.9	0.6	-7	-0.9	2.6	-42	-16	-126	-437	0.011
OW	2.0	0.9	-0.9	0.0	1.3	0.0	-0.7	-0.2	-0.2	-0.4	-4	0.3	0.4	-52	-60	-62	-165	0.001
AXR	2.7	1.1	-1.9	-0.4	0.6	-1.6	0.4	0.7	0.3	0.1	30	-0.6	-0.3	-21	-35	-18	-536	-0.035

Table 46. Average differences in phoneme features between Loud and normal speech, speaker #6

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-3.4	-2.7	-0.1	0.5	1.8	0.3	2.8	0.6	-1.7	-2.7	228	1.7	-4.3	--	58	18	54	-0.014
T	-0.8	-1.5	-1.7	-2.2	0.4	-2.0	2.3	1.3	1.1	0.5	314	0.2	0.8	--	-64	168	270	-0.007
K	3.0	0.1	-1.3	-0.7	1.8	-0.6	0.4	-0.6	-0.5	0.2	-123	-0.5	-0.1	--	31	163	262	-0.007
B	1.3	2.4	1.1	0.1	1.3	-0.8	-0.6	-0.6	-0.7	0.0	-159	0.2	0.7	25	-339	-257	-147	-0.005
D	3.5	1.0	-1.5	-0.3	1.3	-2.5	0.8	-0.5	0.1	0.9	-152	-0.1	2.8	32	-120	91	69	0.001
G	0.7	-0.1	-2.8	-2.3	1.5	-0.3	0.9	-0.1	0.3	1.3	-25	0.1	0.8	24	-71	32	196	0.000
DX	-0.4	0.9	-0.5	0.1	2.9	-1.2	-0.9	-0.2	0.1	-0.7	2	0.5	0.6	57	46	48	-85	-0.006
M	0.8	0.8	-3.2	-1.4	-1.4	-1.2	0.8	1.4	1.5	1.5	-35	-0.5	1.3	49	-7	460	493	-0.001
N	0.4	0.2	-2.8	-0.9	0.3	-1.2	0.8	0.3	0.8	0.9	-14	-0.1	1.1	51	-31	170	46	-0.002
NX	-0.1	1.4	-1.4	-1.0	0.2	-0.8	0.7	0.3	0.5	0.4	-32	0.1	0.3	57	-32	40	-265	-0.004
S	0.3	-0.4	-1.2	-1.0	1.4	-3.6	1.2	0.8	1.0	0.8	252	-0.3	1.6	--	24	172	186	-0.022
Z	1.6	1.0	-1.3	-0.1	2.4	-3.5	0.7	-0.1	0.2	0.3	-23	-0.3	1.8	40	-101	48	-158	-0.015
CH	-1.0	0.0	-0.4	-1.6	1.4	-3.9	2.0	1.0	0.9	0.6	249	0.0	4.3	--	-226	49	39	-0.018
TH	-0.1	2.6	1.3	-0.4	1.3	-1.3	0.0	-0.3	-0.6	0.0	-132	-0.1	-0.2	--	-93	24	-315	-0.036
F	-0.4	3.1	2.2	-0.9	1.4	-1.0	0.0	-0.2	-1.1	0.0	-216	0.3	-0.5	--	-268	-6	-118	-0.028
SH	-0.3	-1.5	-2.7	-2.2	0.4	-2.8	3.2	0.9	0.9	1.3	418	-0.6	0.6	--	472	416	887	-0.013
JH	0.5	0.7	-0.7	-0.5	0.9	-3.0	1.6	0.6	0.3	0.3	110	-0.4	2.3	-35	-321	-187	-199	-0.011
V	-0.7	1.6	-0.9	-0.1	3.0	-1.7	-0.4	-0.3	0.3	-0.5	8	2.4	1.6	59	-41	50	111	-0.004
L	-0.4	0.2	-0.8	-0.5	2.2	0.6	-0.3	-0.4	0.0	-1.2	103	1.7	-2.2	35	20	41	161	0.007
R	-1.4	-0.2	0.2	-0.2	3.8	-0.9	-0.8	-1.0	-0.2	-0.6	9	0.0	0.0	59	21	81	232	0.002
Y	-0.7	0.4	-0.9	-1.2	2.8	0.5	-0.7	-0.7	-0.3	0.0	-23	0.0	0.3	22	-20	55	-484	0.010
HH	0.3	-0.1	-1.4	-0.2	2.8	-0.4	0.2	-0.3	-0.5	-0.9	80	0.8	-0.7	21	-34	51	117	-0.018
EL	-0.9	-0.3	0.0	1.0	2.3	0.6	-1.2	-1.5	0.3	-1.2	42	1.0	-2.1	54	40	114	78	0.005
W	-1.4	-0.4	1.2	0.2	1.1	-1.2	-0.5	0.5	0.4	-0.7	84	0.6	-0.8	37	33	532	558	0.002
EH	-0.6	-0.5	0.5	0.0	3.0	-1.6	-0.8	0.1	0.5	-1.0	80	2.3	0.8	52	57	40	73	0.030
AO	-0.5	-0.6	-1.2	-0.1	2.5	-1.0	0.9	1.6	-1.2	-1.8	152	1.7	-1.2	38	82	115	183	0.031
AA	-0.8	-0.4	-1.0	0.2	2.0	-0.4	-0.5	1.2	-0.2	-1.6	148	1.1	0.7	57	86	132	169	0.020
UW	-0.6	0.8	1.1	0.4	2.2	-1.9	-0.5	-0.8	-0.1	0.1	-133	0.6	1.3	52	30	-8	143	0.023
ER	-1.4	-0.2	-0.1	0.3	2.6	-1.6	-1.0	0.1	0.9	-1.0	128	0.2	1.1	58	32	136	229	0.014
AY	-2.1	-0.9	-1.1	-0.4	2.4	0.8	0.0	0.8	-0.5	-1.9	217	1.5	-1.0	56	81	75	120	0.025
EY	-1.5	-0.8	1.3	-0.4	2.5	-1.8	0.1	-0.5	0.7	-0.8	82	1.2	-0.5	56	50	-2	4	0.049
AW	-2.0	-0.4	-1.3	-0.3	2.2	-0.2	-0.1	1.1	0.3	-1.8	254	1.3	0.9	60	53	98	133	0.017
AX	-0.5	-0.1	-0.4	0.1	3.0	-0.3	-0.8	-1.0	0.0	-0.7	26	2.3	-0.9	54	22	134	122	0.006
IH	-1.8	-0.3	-0.4	-0.5	3.5	-1.1	0.3	-0.8	-0.2	-0.7	42	2.2	-0.4	59	7	35	119	0.012
AE	-2.4	-1.3	-1.2	-0.5	1.7	0.9	0.4	0.8	-0.2	-1.5	208	1.4	-1.2	54	82	29	74	0.033
AH	-1.0	-0.8	-0.5	0.5	3.0	-0.6	-0.1	-0.1	-0.5	-1.3	111	2.6	-0.5	53	50	65	93	0.015
OY	-0.8	0.3	0.3	-0.1	2.2	-0.3	-1.2	-0.4	0.0	-0.2	-41	1.6	0.6	45	26	81	183	-0.005
IY	-0.7	1.3	1.6	-0.1	2.4	-2.4	-0.5	-0.5	0.1	-0.1	-131	0.8	0.4	62	21	27	-339	-0.010
OW	-1.1	-0.5	0.1	0.4	2.9	-1.3	-0.8	-0.5	0.7	-1.1	112	2.0	-0.6	56	39	184	311	0.009
AXR	-1.3	-0.1	-0.1	0.1	3.4	-0.5	-0.6	-1.1	-0.1	-0.8	28	0.2	-0.1	30	5	12	-218	0.019

Table 47. Average differences in phoneme features between Lombard and normal speech, speaker #6

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-2.4	-1.6	0.5	1.4	2.7	1.2	2.8	-0.1	-2.6	-4.5	227	1.6	-7.2	--	61	-23	-135	-0.011
T	0.5	-0.2	0.6	-0.9	0.2	-1.6	1.7	0.3	-0.1	0.1	111	0.0	0.7	--	-2	67	152	-0.002
K	2.7	0.1	-0.3	-0.4	1.6	0.7	0.5	-1.0	-1.1	-0.8	-45	-0.4	-2.1	--	71	144	206	-0.008
B	1.3	1.8	2.3	1.3	1.9	-0.3	-0.6	-0.8	-1.8	-1.6	-81	0.5	-1.7	20	-261	-377	-351	-0.010
D	2.7	0.3	-1.7	-1.0	2.2	-0.6	1.1	-0.8	-0.8	0.1	-35	0.3	0.3	2	-106	-63	74	-0.003
G	0.7	0.2	-1.3	-1.1	1.1	0.0	1.2	-0.4	-0.3	0.0	-30	0.1	-1.1	51	-8	-21	-41	0.009
DX	1.2	1.3	0.1	0.3	2.8	-0.2	-0.1	-1.0	-1.3	-1.1	-92	0.4	-1.1	15	10	0	-137	-0.005
M	2.3	1.2	-2.1	-0.7	-1.1	-1.5	0.7	0.9	0.9	0.9	-120	-0.8	0.6	13	-35	205	142	0.002
N	1.5	1.5	-1.3	0.0	1.2	-1.2	0.7	-0.4	-0.2	-0.2	-74	0.0	0.3	11	-23	27	-63	0.000
NX	1.3	1.6	-1.2	-0.8	0.3	-0.7	0.6	0.1	0.1	0.2	-96	-0.2	0.3	20	-15	19	-71	0.011
S	1.1	1.2	0.4	-0.6	1.3	-2.6	1.0	-0.2	-0.4	0.7	-14	-0.3	0.7	--	-57	343	-4	-0.001
Z	2.4	1.3	0.1	0.0	2.3	-2.2	0.8	-0.8	-1.3	0.3	-162	-0.4	-0.7	14	-82	-1	-30	-0.011
CH	0.5	0.9	-0.3	-1.9	0.8	-3.5	2.4	0.8	0.2	0.9	187	-0.5	2.7	--	-281	264	399	-0.008
TH	2.4	2.3	0.8	-0.6	1.5	-0.7	1.0	-0.3	-1.5	-0.8	-85	-0.3	-2.4	--	-65	-205	-300	-0.016
F	0.9	2.5	2.7	-0.6	1.8	-1.7	0.5	-0.2	-1.1	-0.9	-158	0.0	-1.0	--	-243	-207	-326	-0.014
SH	0.9	0.7	-0.4	-1.4	0.4	-1.9	2.9	0.1	-0.2	0.1	91	-0.6	-0.5	--	7	372	180	-0.002
JH	1.9	2.0	0.5	-0.2	0.8	-3.1	1.4	0.1	0.0	-0.2	-12	-0.6	3.1	-20	-361	-172	-345	0.004
V	0.4	2.0	-0.2	-0.3	2.8	-1.2	1.1	-0.7	-0.7	-1.5	47	1.9	-0.1	12	-46	-53	67	-0.001
L	0.2	0.1	0.0	-0.8	2.5	0.9	1.5	-0.8	-1.7	-1.7	95	1.6	-3.4	14	26	-29	243	0.004
R	0.4	0.0	0.2	-0.9	3.2	0.1	0.7	-0.9	-0.9	-1.4	106	-0.5	-2.7	18	24	68	481	-0.001
Y	1.6	1.7	-0.1	-0.9	1.6	-1.1	0.1	-0.4	-0.6	0.5	-159	-0.6	0.9	18	-12	102	-250	-0.004
HH	2.6	2.8	1.2	0.3	1.7	-0.2	-0.2	-0.9	-1.1	-1.6	-33	-0.1	-1.0	29	-53	-446	-379	-0.009
EL	1.2	0.7	-0.3	0.5	2.6	0.4	0.8	-1.2	-1.8	-1.6	17	0.9	-2.0	16	14	433	441	0.010
W	1.6	0.1	1.2	0.0	1.5	-0.9	0.8	-0.3	-0.9	-1.1	18	0.0	-1.8	10	10	-474	-193	-0.009
EH	0.5	-0.2	-0.5	-1.0	1.8	-0.3	1.2	0.3	-0.5	-1.5	160	1.9	-1.3	15	31	-48	-7	0.019
AO	1.4	0.6	-1.0	-0.1	1.6	2.0	1.0	-1.4	-2.3	-0.8	-43	0.7	-4.4	15	61	58	167	0.012
AA	2.0	1.1	-0.9	0.0	0.9	1.3	1.1	-0.7	-1.9	-1.0	-21	-0.1	-2.2	18	53	80	114	0.014
UW	0.5	1.5	0.8	-0.2	2.1	-0.8	0.8	-1.6	-0.7	-0.7	-137	0.7	0.1	15	13	-63	165	0.010
ER	1.5	0.7	-0.1	-0.1	1.9	-0.4	0.7	-0.7	-1.0	-0.8	20	-0.3	-1.1	17	20	90	570	0.007
AY	0.6	0.5	-1.2	-0.2	2.2	1.1	1.4	-0.7	-2.0	-1.3	68	1.1	-2.7	16	59	186	118	0.015
EY	-0.2	-0.2	0.0	-1.1	1.6	-0.5	1.6	-0.5	0.1	-1.2	142	0.7	-1.7	16	28	-65	-37	0.037
AW	0.5	0.4	-1.6	-0.6	2.0	0.7	1.8	-0.1	-1.8	-1.3	121	1.2	-1.9	11	37	62	10	0.001
AX	0.8	0.4	-1.0	-0.5	2.2	0.7	1.1	-0.3	-1.5	-1.4	91	2.0	-2.6	20	5	22	25	0.007
IH	-0.1	-0.4	-1.1	-1.1	2.1	0.2	2.1	-0.2	-0.7	-1.7	173	1.3	-3.4	18	4	-73	50	0.008
AE	0.5	0.7	-0.4	-0.1	1.6	1.1	-0.6	0.0	-1.2	-0.8	7	0.8	-1.4	20	52	-67	-10	0.020
AH	0.9	-0.1	-1.3	-0.3	1.9	1.5	1.3	-0.2	-2.1	-1.7	114	1.6	-3.0	14	23	40	-13	0.003
OY	0.9	-0.5	-0.9	0.1	2.2	0.3	0.9	-0.1	-0.9	-2.1	174	1.2	-4.1	15	18	78	145	-0.011
IY	-0.2	1.5	0.5	-0.2	2.5	-1.1	1.8	-1.4	-0.7	-1.4	-15	0.3	-3.1	16	4	-86	-391	0.004
OW	0.6	-0.6	-0.8	0.8	2.3	-0.2	1.2	-0.5	-1.3	-1.8	143	1.2	-2.7	19	32	179	151	0.006
AXR	1.0	-0.2	0.8	0.6	3.3	-0.7	-0.2	-1.4	-1.1	-1.0	-38	-0.3	-0.2	12	31	-1	-73	0.041

Table 48. Average differences in phoneme features between Lombard and loud speech, speaker #6

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	1.0	1.1	0.7	1.0	0.9	0.9	-0.1	-0.7	-0.9	-1.8	-1	-0.1	-2.9	--	3	-42	-189	0.004
T	1.4	1.3	2.3	1.3	-0.3	0.4	-0.6	-1.0	-1.1	-0.4	-203	-0.2	-0.1	--	61	-101	-118	0.005
K	-0.3	0.0	1.0	0.3	-0.2	1.4	0.2	-0.4	-0.6	-1.0	78	0.0	-2.0	--	40	-19	-56	-0.001
B	0.0	-0.6	1.2	1.3	0.6	0.6	0.0	-0.2	-1.1	-1.5	77	0.3	-2.3	-6	78	-120	-203	-0.005
D	-0.8	-0.7	-0.2	-0.6	0.9	1.9	0.3	-0.2	-0.9	-0.8	117	0.3	-2.4	-29	14	-153	4	-0.003
G	0.0	0.3	1.4	1.2	-0.3	0.3	0.3	-0.2	-0.6	-1.4	-5	0.0	-1.9	27	63	-53	-236	0.008
DX	1.5	0.4	0.6	0.1	-0.1	1.0	0.8	-0.8	-1.4	-0.4	-94	-0.1	-1.7	-43	-36	-48	-52	0.001
M	1.5	0.3	1.1	0.7	0.4	-0.3	-0.1	-0.5	-0.6	-0.6	-85	-0.2	-0.7	-36	-28	-255	-351	0.004
N	1.0	1.3	1.4	0.9	0.9	-0.1	-0.1	-0.7	-1.0	-1.1	-80	0.1	-0.8	-40	8	-143	-109	0.002
NX	1.4	0.2	0.2	0.2	0.1	0.1	-0.1	-0.2	-0.3	-0.2	-63	-0.3	0.1	-38	17	-21	194	0.015
S	0.8	1.6	1.6	0.5	-0.1	1.0	-0.2	-1.0	-1.4	0.0	-266	0.1	-0.9	--	-81	172	-190	0.021
Z	0.9	0.3	1.4	0.0	-0.1	1.3	0.1	-0.7	-1.5	0.0	-138	-0.1	-2.5	-27	20	-49	128	0.004
CH	1.5	0.9	0.1	-0.4	-0.6	0.5	0.4	-0.2	-0.7	0.3	-62	-0.5	-1.7	--	-55	215	360	0.011
TH	2.5	-0.3	-0.5	-0.2	0.2	0.6	1.0	0.0	-0.9	-0.8	48	-0.2	-2.1	--	28	-229	15	0.020
F	1.3	-0.7	0.5	0.3	0.4	-0.7	0.4	0.1	0.0	-0.9	58	-0.3	-0.5	--	24	-200	-208	0.014
SH	1.2	2.2	2.2	0.7	0.0	0.9	-0.3	-0.9	-1.1	-1.3	-327	0.0	-1.1	--	-465	-44	-707	0.011
JH	1.4	1.3	1.2	0.3	-0.1	-0.1	-0.2	-0.5	-0.3	-0.4	-122	-0.2	0.8	15	-40	15	-147	0.015
V	1.1	0.4	0.7	-0.2	-0.2	0.6	1.5	-0.4	-0.9	-1.0	39	-0.5	-1.6	-46	-5	-103	-45	0.003
L	0.7	-0.1	0.8	-0.3	0.3	0.4	1.8	-0.4	-1.7	-0.5	-8	-0.1	-1.2	-21	6	-70	82	-0.003
R	1.8	0.2	-0.1	-0.7	-0.6	0.9	1.5	0.0	-0.7	-0.8	96	-0.5	-2.7	-40	3	-12	249	-0.003
Y	2.3	1.3	0.8	0.3	-1.2	-1.6	0.8	0.3	-0.3	0.5	-136	-0.6	0.6	-4	8	47	234	-0.014
HH	2.3	2.9	2.6	0.5	-1.1	0.2	-0.3	-0.6	-0.6	-0.7	-113	-0.9	-0.3	8	-19	-497	-496	0.009
EL	2.1	1.0	-0.4	-0.4	0.3	-0.2	2.0	0.4	-2.1	-0.4	-24	-0.1	0.1	-38	-27	319	362	0.004
W	2.9	0.5	0.0	-0.2	0.4	0.3	1.3	-0.7	-1.3	-0.4	-67	-0.6	-1.1	-27	-23	-1006	-751	-0.012
EH	1.1	0.3	-1.0	-1.1	-1.2	1.3	2.1	0.3	-0.9	-0.4	81	-0.4	-2.1	-37	-25	-88	-79	-0.011
AO	1.9	1.2	0.2	-0.1	-0.9	3.0	0.1	-2.9	-1.1	1.0	-195	-1.1	-3.2	-23	-21	-57	-16	-0.018
AA	2.8	1.5	0.1	-0.1	-1.1	1.7	1.6	-1.9	-1.7	0.6	-170	-1.2	-2.9	-39	-33	-52	-55	-0.006
UW	1.2	0.7	-0.2	-0.6	-0.1	1.2	1.3	-0.8	-0.6	-0.7	-4	0.1	-1.2	-37	-17	-55	22	-0.013
ER	2.9	0.9	0.0	-0.4	-0.7	1.2	1.7	-0.8	-1.9	0.2	-108	-0.5	-2.2	-42	-13	-46	342	-0.008
AY	2.7	1.4	-0.1	0.1	-0.2	0.3	1.4	-1.5	-1.6	0.6	-149	-0.4	-1.7	-40	-22	111	-2	-0.009
EY	1.3	0.7	-1.3	-0.7	-0.9	1.3	1.5	0.0	-0.5	-0.4	60	-0.5	-1.2	-41	-22	-63	-41	-0.011
AW	2.4	0.8	-0.3	-0.3	-0.3	0.9	1.9	-1.3	-2.2	0.5	-132	-0.1	-2.8	-48	-16	-35	-123	-0.016
AX	1.3	0.5	-0.5	-0.7	-0.8	1.0	1.9	0.7	-1.5	-0.7	65	-0.3	-1.7	-34	-17	-112	-97	0.001
IH	1.6	-0.1	-0.8	-0.6	-1.4	1.3	1.7	0.5	-0.5	-1.0	131	-0.9	-3.0	-40	-3	-108	-69	-0.004
AE	2.8	2.1	0.8	0.4	-0.2	0.1	-1.0	-0.8	-0.9	0.7	-202	-0.7	-0.3	-34	-30	-97	-84	-0.013
AH	1.9	0.7	-0.7	-0.7	-1.1	2.1	1.5	0.0	-1.6	-0.4	3	-1.0	-2.5	-38	-27	-26	-107	-0.012
OY	1.7	-0.8	-1.2	0.2	0.0	0.6	2.1	0.3	-0.8	-2.0	216	-0.4	-4.6	-30	-9	-3	-38	-0.006
IY	0.5	0.2	-1.2	-0.1	0.0	1.3	2.3	-0.8	-0.9	-1.2	115	-0.5	-3.5	-46	-17	-113	-53	0.015
OW	1.8	0.0	-0.8	0.4	-0.6	1.2	2.0	0.0	-2.1	-0.7	31	-0.8	-2.1	-37	-7	-5	-160	-0.003
AXR	2.3	-0.1	0.9	0.6	-0.1	-0.2	0.4	-0.3	-1.0	-0.2	-66	-0.4	0.0	-17	26	-13	145	0.022

Table 49. Average differences in phoneme features between Loud and normal speech, speaker #7

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.9	-1.8	-2.0	-0.8	0.6	0.8	1.3	1.3	-0.4	-1.0	191	0.4	-3.1	--	-7	-9	22	-0.014
T	-2.6	-0.9	-0.2	-0.9	-1.4	-0.7	0.0	1.3	1.0	1.5	158	0.4	1.8	--	15	-41	-100	0.002
K	-1.1	0.6	0.4	0.4	0.7	-0.3	-0.3	0.3	-0.5	-0.3	-8	0.6	-0.7	--	31	161	-28	-0.004
B	-4.0	-1.1	-0.2	0.9	0.5	1.0	0.9	0.4	-0.9	-1.3	190	0.6	-2.4	65	57	-266	-182	-0.008
D	-3.1	-2.0	-0.5	0.0	-0.8	0.4	0.2	0.8	0.4	0.4	181	0.5	0.1	26	-134	70	153	-0.005
G	-2.9	-1.5	-1.4	-1.6	0.6	0.1	1.5	1.0	-0.4	0.4	184	0.2	-1.0	-14	-88	128	52	-0.004
DX	-3.3	0.0	0.7	1.9	2.0	-0.5	-1.4	-0.6	-0.4	-0.6	-33	1.4	0.2	80	29	46	72	-0.002
M	-5.1	-3.6	-0.7	0.3	-0.8	-0.1	0.4	0.9	0.7	0.9	232	0.8	0.4	94	32	-641	-635	0.001
N	-4.0	-2.1	-1.9	-0.1	0.1	-0.4	-0.1	1.0	0.8	1.1	171	1.0	1.5	85	1	-151	-195	0.001
NX	-4.3	-2.4	-3.1	-1.6	-0.5	-0.4	0.5	1.6	1.5	2.0	201	0.6	2.0	76	-15	-33	-200	0.013
S	-0.8	-0.2	-1.0	-1.1	-1.7	-1.6	-0.1	1.1	1.3	2.7	354	-0.2	4.0	--	25	62	66	0.009
Z	-1.6	-0.3	-0.6	-0.3	-1.1	-2.6	-0.7	1.4	1.3	2.6	156	0.4	3.8	89	-69	-438	-360	0.003
CH	-1.5	-1.0	-1.9	-2.6	-2.2	0.4	1.1	1.8	1.5	1.6	356	-0.2	0.8	--	-46	360	305	-0.007
TH	-1.7	-0.6	0.1	0.6	-0.1	-0.2	-0.8	0.4	0.5	0.1	42	0.4	0.7	--	56	-214	-187	-0.017
F	-1.0	-1.2	-1.8	-1.1	-0.4	0.1	0.1	0.8	1.2	0.7	129	0.2	1.2	--	12	-119	-51	0.001
SH	-2.8	-1.9	-2.3	-3.6	-0.6	0.2	0.8	2.0	1.9	1.5	363	0.1	2.0	--	-103	122	131	-0.017
JH	-0.9	0.8	-0.9	-2.2	-1.5	0.6	0.4	1.1	0.8	1.2	185	-0.6	0.6	18	-348	79	-35	0.002
V	-3.5	-2.0	-0.5	1.2	1.1	1.9	-0.4	-0.7	-0.4	-1.0	142	1.3	-2.1	78	40	51	-5	0.013
L	-2.5	1.1	1.8	0.4	-0.4	0.4	-0.8	0.1	0.1	-0.5	14	0.6	0.0	59	35	-159	56	0.008
R	-3.2	-2.0	0.9	1.6	0.8	1.4	-0.8	-0.7	-0.7	-1.0	85	1.3	-0.6	98	65	35	-455	0.008
Y	-3.0	-0.7	1.4	1.1	0.3	0.8	-2.0	0.7	0.2	-0.9	67	1.6	-1.1	91	27	-101	58	0.018
HH	0.7	1.9	0.6	0.5	-0.1	-1.1	-0.1	0.6	-0.6	-0.2	-28	-0.2	0.3	90	-77	-352	-484	-0.005
EL	-3.7	-0.6	0.9	0.1	0.8	2.4	-0.3	-0.9	-0.3	-1.1	111	1.3	-1.2	55	46	-30	204	0.026
W	-3.6	-0.6	0.5	0.9	0.5	-0.1	-0.4	0.2	-0.2	-0.3	30	0.7	0.3	76	30	-233	208	0.024
EH	-3.3	-1.0	1.7	1.6	1.1	0.9	-1.4	-1.0	-0.2	-0.8	19	0.3	1.1	89	74	57	66	0.027
AO	-3.0	-0.7	1.2	0.9	1.1	1.2	-1.4	-0.8	-0.3	-0.5	-32	0.8	0.9	85	69	65	118	0.043
AA	-4.8	-2.3	1.0	1.7	2.1	1.4	-1.3	-1.1	-0.7	-1.1	61	1.3	0.0	98	107	79	-76	0.035
UW	-2.7	-0.9	1.5	1.2	0.8	0.5	-1.0	0.1	-0.3	-1.3	108	0.3	-1.8	84	55	116	19	0.023
ER	-3.9	-1.4	0.8	1.7	1.3	1.0	-1.2	-1.1	-0.4	-0.5	41	0.6	0.2	106	81	110	-139	0.011
AY	-4.4	-2.0	1.2	1.7	2.0	1.0	-2.0	-1.2	0.1	-0.8	30	1.4	1.5	85	110	78	97	0.047
EY	-4.5	-1.8	1.7	0.8	1.6	0.9	-0.7	-0.6	-0.2	-1.2	93	2.2	-1.6	88	59	-35	-355	0.054
AW	-4.2	-1.4	0.8	2.1	2.0	1.5	-2.6	-0.8	-0.2	-1.0	21	0.5	1.2	85	121	139	84	0.048
AX	-3.3	-0.6	1.9	1.3	1.3	2.2	-1.3	-1.4	-0.7	-1.4	56	0.5	-1.3	84	80	58	92	0.007
IH	-4.5	-2.1	0.9	0.8	1.7	1.1	-0.4	-0.8	-0.2	-1.1	123	1.2	-1.2	100	63	100	-24	0.013
AE	-3.9	-1.5	1.4	1.4	1.5	0.8	-1.2	-0.9	-0.2	-0.7	37	1.1	0.5	90	71	30	-10	0.024
AH	-4.5	-1.5	1.6	1.7	1.8	2.4	-1.6	-0.8	-1.3	-1.7	96	0.3	-1.1	99	154	149	-78	0.028
OY	-3.5	0.6	2.9	1.1	0.7	2.2	-1.9	-1.3	-0.9	-0.8	-36	0.6	1.9	68	114	-79	37	0.015
IY	-2.5	-1.6	0.9	-0.4	0.9	0.4	-0.6	0.4	0.5	-0.7	139	1.3	-1.2	90	51	21	-156	0.025
OW	-3.9	-1.7	1.0	1.1	1.6	1.6	-1.1	-0.8	-0.6	-1.2	103	1.2	-1.0	97	72	-22	14	0.021
AXR	-1.9	1.1	1.8	1.6	0.9	1.7	-1.8	-1.2	-0.9	-0.9	-49	0.0	1.3	52	56	30	-230	0.033

Table 50. Average differences in phoneme features between Lombard and normal speech, speaker #7

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	2.0	0.2	-1.2	-0.5	-0.1	-0.1	-0.6	0.8	0.2	0.4	9	-0.5	0.2	--	72	236	280	-0.014
T	-0.6	-0.3	0.0	-1.0	-0.9	-0.8	-0.7	0.7	1.7	1.2	121	0.0	4.0	--	-49	91	83	0.006
K	0.1	1.7	0.6	-0.2	1.2	-1.2	-1.3	0.4	0.3	0.0	-21	0.1	0.9	--	-182	-123	-80	-0.003
B	1.3	3.9	0.0	-0.6	-0.8	-1.1	-1.0	-0.3	0.8	1.6	-171	-0.9	2.9	26	-177	-141	99	-0.005
D	-1.2	-0.2	-1.1	-1.7	-1.1	-0.4	0.4	0.5	1.5	1.6	64	-0.2	1.6	29	-284	-109	-207	-0.001
G	-1.3	-0.3	0.4	-0.6	2.6	-0.4	0.3	0.0	-1.1	-0.6	88	0.3	-0.7	22	80	481	280	-0.008
DX	-1.1	1.3	-1.0	0.5	1.1	-1.9	-0.6	0.6	0.3	0.3	-26	0.5	1.4	56	-9	108	68	0.002
M	-2.6	-0.3	0.5	2.9	0.8	0.4	-1.2	-0.9	-0.9	-1.0	-8	0.3	0.2	56	1	-930	-872	-0.001
N	-1.1	2.4	0.8	0.0	0.5	-0.7	-0.9	0.0	-0.1	0.2	-144	0.5	1.2	47	-11	234	128	0.008
NX	-1.7	2.1	1.4	0.8	0.7	-1.2	-0.7	-0.2	-0.2	-0.2	-198	0.6	1.2	50	-28	-58	-240	0.007
S	0.4	0.0	-1.0	-1.6	-2.4	-3.4	-0.1	1.2	3.5	3.3	512	-0.7	9.2	--	27	-26	150	0.020
Z	1.0	-1.0	-1.0	-1.1	-1.6	-4.3	-0.8	1.3	3.5	3.4	301	-0.6	9.0	40	-86	-319	-451	-0.003
CH	0.2	-0.2	-2.0	-2.9	-2.6	-0.8	0.9	1.6	3.3	1.5	439	-0.6	6.3	--	183	48	95	-0.011
TH	0.5	0.8	0.2	0.6	-0.1	-1.4	-2.3	-0.2	1.6	1.3	-67	-0.1	5.4	--	-9	48	153	-0.024
F	-0.2	-1.2	-1.8	-0.9	-0.5	-0.3	-0.7	0.2	1.9	1.5	29	0.3	3.8	--	14	-139	-118	-0.014
SH	-2.1	-1.9	-2.2	-3.4	0.0	0.0	-0.1	2.4	2.8	0.5	364	-0.1	5.4	--	-48	105	-75	-0.005
JH	0.1	1.5	-0.1	-2.5	-1.2	0.2	-0.1	1.5	1.3	0.4	188	-0.8	2.2	81	-471	-53	-106	0.018
V	-0.7	-0.1	-1.5	0.5	2.8	0.9	-1.1	-0.7	-0.5	-0.9	52	1.1	-0.2	50	-30	97	30	0.005
L	-1.3	0.6	1.0	0.6	0.8	-0.4	-0.9	0.4	-0.4	-0.6	-22	0.6	1.1	32	32	-116	-56	0.006
R	-0.7	-0.2	0.9	1.1	0.3	-0.8	-0.4	-0.1	0.0	-0.3	-2	0.6	0.2	53	30	28	-341	0.013
Y	-0.9	2.0	1.5	0.6	0.8	-0.8	-0.9	-0.2	-0.1	-0.4	-88	0.9	1.1	71	36	-9	112	0.007
HH	1.3	1.6	0.9	0.4	0.1	-1.5	-1.0	0.8	-0.1	0.1	-56	-0.1	1.3	85	-41	-339	-395	0.002
EL	-1.5	0.4	1.3	0.1	1.2	0.4	-0.7	-0.3	-0.6	-0.5	-54	1.0	0.2	44	37	73	272	0.021
W	-1.4	1.2	1.4	1.3	1.3	-1.1	-1.1	-0.3	-0.4	-0.5	-125	0.6	1.1	48	-6	-144	-115	0.007
EH	-2.0	-0.4	1.0	0.8	2.7	-0.9	-1.2	-0.1	-0.5	-0.8	12	1.5	1.0	57	65	81	102	0.014
AO	-2.1	0.1	0.4	0.7	2.0	-0.4	-1.1	0.1	-0.6	-0.6	-17	1.4	0.7	62	85	115	224	0.022
AA	-2.6	-1.4	-0.4	1.1	3.0	0.1	-0.8	0.0	-0.9	-1.3	85	1.4	-0.6	49	101	140	-32	0.026
UW	-1.6	0.6	0.2	0.5	1.8	-0.7	-0.7	-0.2	-0.2	-0.4	-10	0.7	0.6	49	25	143	82	0.013
ER	-1.4	0.5	0.8	0.8	1.0	-0.9	-0.8	0.0	-0.2	-0.2	-48	-0.2	0.6	60	40	163	-68	0.006
AY	-2.6	-0.7	0.5	1.0	3.5	-0.5	-1.7	-0.1	-0.8	-0.9	3	2.2	1.2	50	116	197	177	0.023
EY	-3.5	-1.4	0.5	-0.3	1.8	-0.4	0.1	1.1	-0.6	-0.8	140	1.4	-1.4	57	48	25	-209	0.039
AW	-2.8	-0.8	0.4	1.5	2.9	-0.3	-1.6	0.0	-0.8	-1.2	32	1.5	0.7	55	122	105	122	-0.004
AX	-2.2	-0.4	1.1	1.0	2.9	1.2	-1.2	-1.0	-1.3	-1.5	18	1.3	-0.8	49	57	98	129	0.016
IH	-2.4	-0.1	0.9	0.2	2.4	-0.7	-0.9	0.1	-0.4	-0.7	21	1.1	-0.1	54	42	130	-8	0.007
AE	-1.9	0.1	0.7	0.2	2.1	-0.4	-0.7	-0.1	-0.5	-0.6	12	1.4	0.4	50	80	57	75	0.010
AH	-2.9	0.0	1.1	1.0	2.7	0.5	-1.3	-0.5	-1.1	-1.2	30	1.3	0.0	52	106	100	-89	-0.001
OY	-2.9	1.6	2.3	1.5	2.6	0.7	-1.7	-1.4	-1.4	-1.4	-55	1.2	0.5	58	49	153	23	0.024
IY	-0.7	1.0	0.2	-0.7	1.0	-0.5	-0.7	0.3	0.5	-0.2	4	0.3	0.9	56	19	-22	-354	0.018
OW	-0.9	0.1	0.3	0.3	1.4	-0.6	-0.5	-0.2	-0.1	-0.2	-34	0.8	0.7	52	17	1	27	0.023
AXR	-2.4	0.3	0.9	0.6	3.9	1.3	-2.9	-1.2	-1.0	-0.9	-29	2.0	2.9	38	59	38	-53	0.023

Table 51. Average differences in phoneme features between Lombard and loud speech, speaker #7

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	3.8	2.0	0.8	0.3	-0.6	-0.9	-2.0	-0.5	0.5	1.4	-183	-0.9	3.2	--	79	245	259	0.000
T	2.0	0.7	0.2	-0.1	0.5	-0.1	-0.7	-0.6	0.8	-0.4	-37	-0.3	2.2	--	-84	131	183	0.005
K	1.2	1.1	0.3	-0.6	0.5	-0.8	-0.9	0.1	0.8	0.3	-13	-0.5	1.5	--	-213	-285	-52	0.001
B	5.3	4.9	0.2	-1.5	-1.2	-2.1	-1.9	-0.7	1.7	3.0	-361	-1.5	5.3	-39	-234	125	280	0.004
D	1.9	1.7	-0.5	-1.7	-0.3	-0.8	0.2	-0.3	1.1	1.1	-96	-0.7	1.5	3	-149	-179	-361	0.004
G	1.6	1.3	1.8	0.9	2.0	-0.5	-1.2	-1.0	-0.7	-1.0	-96	0.1	0.3	36	168	353	228	-0.004
DX	2.1	1.3	-1.7	-1.4	-0.8	-1.4	0.8	1.2	0.7	0.9	7	-0.9	1.2	-23	-38	62	-4	0.004
M	2.5	3.4	1.3	2.6	1.6	0.5	-1.5	-1.7	-1.6	-1.9	-241	-0.5	-0.2	-39	-31	-289	-237	-0.002
N	3.0	4.6	2.8	0.2	0.4	-0.4	-0.7	-0.9	-0.9	-0.9	-315	-0.4	-0.3	-38	-12	385	324	0.007
NX	2.6	4.5	4.5	2.4	1.2	-0.7	-1.3	-1.8	-1.7	-2.1	-399	0.0	-0.8	-27	-13	-25	-40	-0.006
S	1.2	0.2	-0.1	-0.4	-0.8	-1.9	0.0	0.0	2.2	0.6	158	-0.5	5.2	--	2	-88	84	0.011
Z	2.7	-0.7	-0.4	-0.8	-0.5	-1.7	-0.2	-0.1	2.2	0.8	145	-1.0	5.2	-49	-17	119	-91	-0.005
CH	1.7	0.8	-0.1	-0.3	-0.3	-1.2	-0.2	-0.2	1.8	-0.1	83	-0.4	5.5	--	229	-312	-210	-0.004
TH	2.1	1.4	0.1	0.0	0.1	-1.1	-1.5	-0.6	1.1	1.2	-108	-0.5	4.7	--	-65	262	340	-0.007
F	0.8	0.0	0.0	0.2	-0.1	-0.4	-0.9	-0.6	0.8	0.8	-100	0.1	2.6	--	2	-20	-66	-0.015
SH	0.7	0.0	0.1	0.2	0.5	-0.2	-0.9	0.4	0.9	-1.0	1	-0.2	3.4	--	54	-17	-206	0.012
JH	1.0	0.8	0.8	0.3	0.3	-0.4	-0.5	0.3	0.5	-0.8	3	-0.2	1.5	63	-123	-132	-71	0.016
V	2.8	1.8	-1.0	-0.7	1.7	-1.0	-0.6	0.0	-0.1	0.1	-90	-0.2	1.9	-28	-70	46	36	-0.008
L	1.2	-0.4	-0.8	0.2	1.2	-0.8	-0.1	0.3	-0.5	-0.1	-36	0.0	1.2	-27	-3	44	-112	-0.002
R	2.5	1.8	-0.1	-0.5	-0.5	-2.2	0.4	0.5	0.6	0.7	-87	-0.8	0.8	-45	-35	-7	113	0.005
Y	2.1	2.7	0.1	-0.5	0.5	-1.6	1.1	-0.9	-0.3	0.4	-155	-0.8	2.2	-20	8	93	54	-0.011
HH	0.6	-0.3	0.3	0.0	0.2	-0.4	-0.9	0.1	0.5	0.3	-28	0.1	1.0	-4	37	14	90	0.008
EL	2.3	1.0	0.4	0.0	0.5	-2.0	-0.4	0.6	-0.2	0.6	-164	-0.2	1.4	-11	-9	104	69	-0.005
W	2.2	1.9	1.0	0.4	0.8	-1.0	-0.7	-0.5	-0.2	-0.2	-156	-0.1	0.9	-27	-36	89	-323	-0.017
EH	1.3	0.6	-0.7	-0.8	1.6	-1.8	0.2	0.9	-0.2	0.0	-7	1.2	-0.1	-32	-10	24	36	-0.013
AO	0.9	0.9	-0.7	-0.2	0.9	-1.6	0.3	1.0	-0.3	-0.1	15	0.6	-0.2	-23	17	51	106	-0.021
AA	2.2	0.9	-1.4	-0.6	0.9	-1.4	0.4	1.2	-0.2	-0.2	24	0.1	-0.6	-49	-6	60	44	-0.008
UW	1.1	1.5	-1.3	-0.7	1.0	-1.1	0.2	-0.3	0.1	0.9	-118	0.4	2.5	-35	-29	27	63	-0.010
ER	2.5	1.9	0.0	-0.9	-0.3	-1.9	0.4	1.1	0.3	0.3	-89	-0.8	0.4	-46	-42	53	71	-0.004
AY	1.8	1.3	-0.6	-0.7	1.5	-1.6	0.3	1.1	-0.8	-0.1	-27	0.9	-0.3	-35	6	119	80	-0.024
EY	1.1	0.5	-1.3	-1.1	0.3	-1.3	0.8	1.7	-0.4	0.4	47	-0.8	0.2	-31	-11	60	146	-0.015
AW	1.5	0.6	-0.5	-0.6	1.0	-1.8	1.0	0.8	-0.6	-0.2	11	1.0	-0.6	-29	1	-34	38	-0.052
AX	1.1	0.2	-0.7	-0.3	1.6	-1.0	0.1	0.4	-0.7	0.0	-38	0.8	0.5	-35	-23	40	37	0.009
IH	2.1	2.0	0.0	-0.7	0.7	-1.8	-0.5	0.9	-0.1	0.4	-102	-0.1	1.1	-46	-21	30	16	-0.007
AE	2.0	1.6	-0.8	-1.2	0.6	-1.1	0.5	0.8	-0.3	0.1	-25	0.3	-0.1	-40	9	27	85	-0.013
AH	1.6	1.5	-0.5	-0.7	0.8	-1.9	0.3	0.3	0.2	0.5	-66	0.9	1.1	-47	-48	-49	-11	-0.029
OY	0.6	1.0	-0.6	0.4	1.9	-1.5	0.2	-0.1	-0.5	-0.6	-19	0.6	-1.4	-10	-65	231	-14	0.009
IY	1.9	2.6	-0.7	-0.3	0.1	-1.0	-0.1	-0.1	0.0	0.6	-136	-0.9	2.1	-34	-32	-43	-198	-0.007
OW	3.0	1.8	-0.8	-0.8	-0.2	-2.2	0.6	0.6	0.4	1.0	-137	-0.4	1.7	-44	-55	23	13	0.002
AXR	-0.5	-0.8	-0.8	-1.0	3.1	-0.4	-1.1	0.2	0.0	0.0	20	2.0	1.6	-14	2	7	176	-0.010

Table 52. Average differences in phoneme features between Loud and normal speech, speaker #8

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-0.5	1.9	1.2	1.1	2.6	0.2	0.5	-1.0	-1.1	-3.3	134	0.6	-3.0	--	-97	-207	-375	-0.014
T	-1.1	1.0	-0.4	-1.7	-0.6	-0.9	1.3	0.6	1.8	-0.2	298	0.1	0.2	--	-78	-13	69	-0.016
K	-1.3	1.3	0.1	0.3	1.8	-1.5	0.2	-0.7	0.6	-0.8	69	0.3	0.8	--	-30	116	364	-0.017
B	2.9	3.6	1.3	-0.5	-0.1	-0.7	-0.1	-0.2	-0.3	-0.2	-129	-0.9	0.5	60	43	139	230	-0.002
D	1.7	1.2	-1.6	-2.3	-0.2	0.1	0.9	0.6	0.7	0.2	115	0.0	0.3	19	-13	10	-356	-0.001
G	1.5	2.7	-0.1	-0.9	1.1	-1.1	0.1	-0.3	-0.3	0.4	-65	-0.7	1.4	43	-95	62	-101	0.002
DX	0.7	2.5	0.3	0.2	1.1	-0.6	0.1	-0.4	-0.5	-0.7	-112	-0.3	0.1	63	-15	-85	-12	-0.003
M	0.6	3.2	0.3	-0.3	0.0	0.2	0.5	-0.2	-0.7	-0.5	-49	-0.7	-0.8	61	-84	-75	-525	-0.017
N	1.1	4.0	1.4	0.4	0.2	-0.8	-0.1	-0.1	-0.8	-0.7	-155	-0.2	-0.2	67	-58	18	16	-0.006
NX	2.4	6.3	2.4	0.0	-0.3	-0.9	-0.5	-0.3	-0.7	-0.7	-254	-0.4	0.2	60	-102	77	126	-0.053
S	-1.5	0.5	0.8	-0.1	0.6	-1.6	-0.8	-1.9	2.3	1.3	169	0.4	5.8	--	-44	37	-355	-0.031
Z	2.0	3.1	2.1	1.5	0.2	-3.3	-1.0	-2.3	1.8	0.9	-68	-1.2	5.4	51	-20	-52	-321	-0.014
CH	-3.6	-2.0	-2.0	-2.3	0.2	-1.2	2.5	0.0	3.1	0.1	386	0.7	5.1	--	40	588	100	-0.021
TH	-2.0	0.4	-0.6	-1.3	-0.1	-1.5	1.7	0.8	1.6	-0.5	295	0.6	0.2	--	-17	-204	-551	-0.032
F	-1.7	0.4	0.4	-0.6	0.3	-1.3	0.5	0.6	1.6	-1.1	202	0.2	2.7	--	-73	-171	-443	-0.036
SH	-5.6	-3.5	-2.6	-4.2	-1.9	-0.3	4.1	2.0	3.5	0.2	650	0.3	2.8	--	-18	322	250	-0.028
JH	-0.7	-0.8	-1.2	-2.4	-0.2	-0.8	1.6	1.0	1.9	-0.1	277	-0.2	1.5	40	118	255	140	-0.028
V	1.1	2.8	-1.1	-0.3	0.2	-0.7	0.3	0.6	0.2	-0.7	85	0.2	-0.1	60	-83	-109	223	-0.007
L	-0.9	0.2	0.0	0.1	0.9	-0.1	1.4	0.1	-0.6	-1.5	142	-0.1	-3.0	72	21	163	229	0.006
R	-0.2	0.8	0.8	0.2	0.7	-0.5	0.7	-0.2	-0.6	-0.9	-7	0.0	-0.7	67	7	-55	536	0.005
Y	1.1	0.5	0.9	0.4	0.2	-1.0	0.1	-0.9	0.2	0.0	-107	-0.3	0.6	46	-4	-15	-468	0.002
HH	2.1	3.6	1.7	-0.7	-0.4	-2.5	-0.5	-0.1	1.4	0.5	-68	-1.0	3.4	-4	-92	173	213	-0.039
EL	-0.1	0.4	-0.2	-0.2	1.3	0.4	0.8	-0.4	-0.4	-1.4	175	-0.3	-1.9	58	13	39	134	0.013
W	1.0	2.3	1.6	1.5	0.4	-0.8	-0.5	-0.4	-0.5	-1.4	-64	-1.0	-0.6	44	-35	-218	-224	0.010
EH	-0.4	0.1	-0.4	-1.3	1.3	-0.7	0.8	1.0	0.8	-1.5	264	0.4	-0.3	64	41	-7	152	0.025
AO	-1.2	-0.1	-1.3	-0.7	2.4	-0.1	1.0	-0.2	0.6	-2.0	313	-0.3	-0.5	55	64	105	284	0.040
AA	-1.5	-1.1	-2.0	-1.7	3.0	0.5	1.3	0.1	0.5	-1.9	385	-0.3	-1.6	70	63	207	462	0.036
UW	-0.4	0.8	1.1	0.2	0.0	-0.3	1.0	0.4	-0.7	-1.4	92	0.1	-1.9	64	19	29	-2	0.035
ER	-1.0	-0.1	0.0	-1.4	1.8	-0.6	1.1	0.9	0.0	-1.6	292	-0.7	-1.2	74	30	145	418	0.007
AY	-0.8	0.1	-1.5	-1.8	1.4	-0.3	0.8	1.0	1.1	-1.4	325	-0.6	0.5	56	42	106	273	0.020
EY	-0.7	0.0	0.8	-0.8	-0.2	-1.4	1.2	1.4	0.5	-1.0	177	0.1	-2.4	62	22	51	-33	0.012
AW	-1.4	-0.1	-1.5	-1.0	1.2	-0.6	0.7	1.2	1.0	-1.4	326	-0.5	1.3	64	49	77	216	0.014
AX	-0.3	0.5	-0.5	-0.8	1.7	-0.4	0.6	0.5	0.1	-1.5	210	0.7	-1.9	73	10	69	155	-0.011
IH	-1.3	-0.3	0.4	-1.2	0.9	-0.8	1.5	1.7	-0.2	-1.6	251	1.0	-1.6	73	22	26	126	0.010
AE	-0.9	0.5	-1.3	-2.1	0.7	-0.3	1.2	1.5	0.8	-1.1	273	-0.2	-0.9	52	32	-33	23	0.013
AH	-1.9	-0.6	-0.8	-1.0	1.8	-0.7	0.4	1.5	0.9	-2.0	365	-0.2	0.6	63	66	75	211	0.036
OY	-1.5	-0.2	-0.5	-0.5	0.5	-1.2	1.9	1.7	0.2	-1.9	342	0.6	-2.6	72	30	159	318	0.020
IY	-0.3	1.1	0.9	0.0	0.2	-1.9	1.5	0.8	-0.4	-0.9	72	0.1	0.1	59	8	54	-159	0.001
OW	-1.1	-0.3	-0.2	-0.2	1.8	-0.5	0.8	0.0	0.3	-1.7	268	0.0	-1.1	67	34	150	325	0.015
AXR	1.6	2.3	1.5	0.1	0.2	-1.0	0.2	0.3	-0.5	-1.0	11	-1.1	-0.5	46	3	-55	854	0.091

Table 53. Average differences in phoneme features between Lombard and normal speech, speaker #8

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-1.0	1.4	1.3	1.1	3.0	-0.4	0.5	-0.6	-1.8	-2.6	49	0.8	-3.0	--	-15	-95	-506	-0.013
T	-0.9	0.4	-0.7	-2.1	-1.1	-1.2	1.4	0.1	2.2	0.9	210	-0.1	1.7	--	-65	45	77	-0.011
K	-0.6	1.9	0.9	-0.5	0.6	-1.0	-0.2	-0.5	0.9	-0.1	-30	-0.5	1.1	--	-103	135	279	-0.012
B	2.4	2.9	0.9	-0.5	0.0	-0.9	-0.2	-0.1	-0.1	0.1	-137	-0.7	0.8	41	325	170	232	-0.003
D	1.1	-1.1	-2.4	-2.7	0.0	0.6	0.6	0.7	0.6	1.3	108	0.0	1.1	36	-74	-116	-302	0.006
G	0.2	0.3	-0.7	-0.7	-0.3	-0.1	-0.3	0.5	0.4	0.7	26	-0.5	0.9	16	93	25	-340	0.004
DX	1.2	2.6	-0.4	-0.3	0.9	-0.6	-0.1	-0.2	-0.4	-0.1	-120	-0.2	0.3	52	-38	3	190	0.000
M	1.0	2.5	-2.2	-0.8	-0.5	0.3	0.4	0.4	0.0	0.4	-67	-0.5	-0.1	47	-83	-48	-345	-0.007
N	2.0	3.7	0.4	-0.3	0.1	-0.4	-0.2	-0.2	-0.4	-0.4	-146	-0.3	0.0	59	-53	77	-89	-0.004
NX	2.5	4.1	1.5	-0.2	-0.2	0.0	-0.3	-0.3	-0.7	-0.6	-155	-0.4	-0.3	45	-88	26	6	-0.025
S	-1.9	-0.6	0.3	0.0	1.1	-1.3	-0.2	-2.5	2.1	1.3	171	0.6	4.9	--	-18	-65	-492	-0.008
Z	1.1	0.3	0.1	0.0	0.6	-2.4	0.1	-2.0	2.3	1.0	118	-0.2	4.5	28	50	14	-530	-0.024
CH	-1.9	-1.4	-1.7	-1.9	-0.4	-1.1	1.4	-0.3	2.9	0.9	326	0.2	4.5	--	112	474	-129	-0.016
TH	-0.5	0.4	-0.7	-1.2	-0.7	-1.6	1.2	0.5	1.9	0.2	205	0.1	1.3	--	21	-86	-392	-0.013
F	-1.5	-0.3	0.1	-0.2	0.6	-1.3	0.6	-0.2	1.3	-0.5	119	0.5	1.6	--	-17	-185	-512	-0.012
SH	-2.6	-2.2	-1.8	-3.6	-1.3	-0.7	2.6	0.6	3.1	1.3	479	-0.1	2.9	--	480	418	58	-0.026
JH	-2.2	-1.7	-1.4	-2.5	-0.3	-0.4	2.0	0.4	2.4	0.0	366	0.3	1.9	26	154	443	227	-0.011
V	2.3	3.2	-1.3	-1.0	0.7	-1.2	0.2	0.1	0.4	0.1	-27	0.1	1.5	67	-111	-60	241	0.003
L	0.1	1.2	-0.9	-0.2	1.5	-0.2	0.3	-0.8	0.2	-0.6	17	0.4	-0.1	54	-40	73	142	-0.002
R	0.0	1.1	0.1	-0.5	1.7	-0.2	0.2	-0.2	-0.6	-0.7	-21	-0.7	-0.7	62	-13	-26	319	0.000
Y	1.5	2.0	1.8	0.4	1.9	-1.1	-1.0	-1.4	-0.3	-0.3	-228	0.0	0.1	36	14	87	-597	-0.002
HH	1.8	3.4	1.2	-0.5	0.3	-2.3	-0.4	0.1	1.5	-0.6	76	-0.7	2.2	53	-21	156	76	-0.032
EL	0.4	0.6	-1.7	-1.1	1.1	0.2	1.1	-0.7	0.7	-0.5	155	0.3	0.0	50	0	89	210	0.031
W	1.6	1.8	1.2	1.3	1.1	-1.1	-0.9	-0.7	-0.2	-0.9	-104	-0.6	0.3	38	-18	-52	114	-0.004
EH	0.3	0.9	-0.3	-1.2	1.3	0.3	0.5	-0.3	0.2	-0.9	129	0.1	-0.6	56	25	-14	137	0.016
AO	-0.7	0.3	-1.3	-0.3	2.7	0.2	0.0	-0.8	0.4	-1.4	209	-0.2	-0.1	53	64	266	441	0.027
AA	-0.2	0.0	-2.1	-0.9	3.1	0.5	0.1	-0.6	0.2	-1.3	231	-0.3	-1.1	56	60	253	453	0.029
UW	0.1	0.6	1.4	-0.1	0.9	-0.3	0.0	-0.2	-0.3	-0.9	-20	0.9	-0.6	63	28	27	90	-0.014
ER	-0.2	1.0	-0.3	-1.5	2.3	-0.1	0.2	-0.1	0.1	-0.9	144	-0.8	-0.9	62	15	140	693	0.001
AY	0.1	0.9	-1.0	-1.4	1.9	0.0	0.3	-0.6	0.8	-0.8	156	-0.5	0.6	59	52	196	388	0.014
EY	0.1	0.6	1.5	-0.5	1.0	-0.5	0.1	-0.6	0.3	-0.6	12	0.2	-0.8	55	20	107	-191	0.036
AW	-1.0	0.7	-1.9	-0.7	2.2	-0.6	0.1	0.0	1.1	-1.0	225	-0.6	1.8	67	41	418	277	0.004
AX	0.8	1.2	-1.2	-1.2	1.7	0.0	0.0	-0.1	0.3	-0.6	92	1.0	-0.1	59	-12	9	168	-0.009
IH	-0.3	0.7	0.4	-1.2	1.4	-0.3	0.6	0.1	-0.1	-0.9	87	1.1	-0.8	61	-3	11	4	0.002
AE	-0.6	0.6	-1.5	-2.2	1.7	0.6	1.4	-0.9	0.8	-0.8	167	0.6	-1.0	57	12	37	21	-0.004
AH	-0.2	1.1	-0.9	-0.7	2.4	-0.2	-0.1	-0.2	0.2	-1.1	146	-0.8	0.4	53	60	277	296	0.041
OY	0.0	1.2	0.2	0.4	1.5	-1.6	-0.1	0.2	0.2	-1.0	79	0.6	-0.1	64	15	215	231	0.030
IY	0.2	0.8	1.5	0.0	0.8	-1.4	0.3	-0.3	0.0	-0.4	-45	0.2	0.6	60	22	111	-160	0.006
OW	-0.5	0.2	-1.2	0.7	2.3	-0.6	-0.1	-0.4	0.2	-1.4	200	0.2	-0.3	59	29	437	551	0.022
AXR	1.3	1.6	-0.2	-1.2	1.7	-0.1	0.4	0.0	-0.4	-0.8	59	-0.9	-1.1	38	-2	109	643	0.041

Table 54. Average differences in phoneme features between Lombard and loud speech, speaker #8

Energies are expressed in dB, tilt is expressed in dB/octave, center of gravity, pitch, and formants are expressed in Hertz, and duration is expressed in seconds.

Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-25	25-5	5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	-0.4	-0.5	0.0	0.0	0.4	-0.6	0.0	0.3	-0.7	0.7	-85	0.2	-0.1	--	83	112	-131	0.001
T	0.2	-0.6	-0.3	-0.4	-0.5	-0.3	0.1	-0.4	0.4	1.2	-88	-0.2	1.6	--	13	58	8	0.005
K	0.7	0.6	0.7	-0.8	-1.2	0.5	-0.4	0.1	0.3	0.7	-99	-0.8	0.3	--	-73	19	-85	0.005
B	-0.5	-0.7	-0.4	0.0	0.2	-0.1	0.0	0.1	0.2	0.3	-8	0.2	0.3	-19	281	30	2	-0.001
D	-0.6	-2.3	-0.8	-0.4	0.1	0.5	-0.3	0.1	-0.1	1.1	-7	0.0	0.7	17	-62	-125	53	0.007
G	-1.3	-2.4	-0.7	0.2	-1.4	1.0	-0.4	0.8	0.7	0.4	91	0.2	-0.5	-27	188	-37	-240	0.003
DX	0.5	0.0	-0.7	-0.4	-0.2	0.0	-0.1	0.2	0.1	0.6	-8	0.2	0.2	-11	-23	88	202	0.003
M	0.4	-0.8	-2.5	-0.5	-0.5	0.1	-0.1	0.5	0.7	0.9	-17	0.1	0.7	-14	2	27	180	0.009
N	0.9	-0.3	-1.0	-0.7	0.0	0.5	0.0	-0.1	0.4	0.3	10	-0.1	0.2	-9	4	59	-105	0.002
NX	0.0	-2.2	-0.9	-0.2	0.1	0.9	0.1	-0.1	0.1	0.1	99	0.0	-0.5	-16	13	-52	-121	0.028
S	-0.4	-1.1	-0.5	0.1	0.5	0.2	0.5	-0.6	-0.2	0.0	2	0.2	-0.9	--	26	-102	-137	0.023
Z	-1.0	-2.8	-2.0	-1.5	0.4	0.9	1.1	0.3	0.5	0.2	185	1.0	-1.0	-23	70	66	-209	-0.010
CH	1.7	0.6	0.3	0.5	-0.6	0.2	-1.0	-0.3	-0.2	0.8	-60	-0.5	-0.6	--	72	-114	-229	0.005
TH	1.5	0.1	-0.1	0.1	-0.6	-0.1	-0.5	-0.3	0.4	0.6	-90	-0.5	1.1	--	38	119	159	0.018
F	0.3	-0.7	-0.3	0.4	0.3	0.1	0.1	-0.8	-0.3	0.6	-83	0.2	-1.1	--	55	-13	-68	0.024
SH	3.0	1.3	0.8	0.6	0.6	-0.5	-1.5	-1.4	-0.3	1.1	-172	-0.4	0.1	--	498	96	-192	0.002
JH	-1.5	-0.9	-0.2	-0.1	0.0	0.5	0.4	-0.6	0.5	0.1	90	0.4	0.3	-14	37	187	87	0.017
V	1.2	0.5	-0.2	-0.7	0.4	-0.4	-0.1	-0.5	0.2	0.8	-112	-0.1	1.6	7	-28	49	18	0.009
L	1.0	1.0	-0.9	-0.3	0.6	-0.1	-1.1	-0.8	0.8	0.9	-125	0.5	2.9	-17	-61	-90	-87	-0.008
R	0.2	0.3	-0.6	-0.7	1.0	0.3	-0.5	-0.1	0.0	0.2	-14	-0.7	0.0	-5	-20	29	-217	-0.006
Y	0.5	1.5	0.9	0.0	1.7	-0.1	-1.2	-0.5	-0.5	-0.4	-121	0.3	-0.5	-10	18	102	-129	-0.004
HH	-0.2	-0.2	-0.5	0.2	0.7	0.2	0.1	0.2	0.1	-1.2	144	0.3	-1.2	57	71	-16	-138	0.007
EL	0.5	0.2	-1.5	-0.9	-0.2	-0.3	0.3	-0.3	1.0	0.9	-20	0.7	1.9	-8	-13	50	75	0.018
W	0.7	-0.5	-0.4	-0.2	0.7	-0.3	-0.4	-0.3	0.4	0.4	-40	0.5	0.8	-6	17	165	338	-0.015
EH	0.8	0.8	0.0	0.1	0.0	1.0	-0.3	-1.2	-0.6	0.6	-136	-0.2	-0.4	-8	-16	-7	-15	-0.009
AO	0.5	0.4	0.0	0.5	0.3	0.3	-1.1	-0.5	-0.2	0.6	-104	0.1	0.4	-3	0	161	157	-0.013
AA	1.4	1.1	-0.1	0.8	0.1	0.0	-1.1	-0.7	-0.3	0.6	-154	0.0	0.4	-14	-3	45	-8	-0.007
UW	0.5	-0.2	0.3	-0.3	0.9	0.0	-1.0	-0.6	0.4	0.4	-112	0.9	1.3	-1	10	-2	92	-0.049
ER	0.8	1.1	-0.3	-0.1	0.5	0.5	-0.9	-1.0	0.1	0.7	-148	-0.1	0.3	-12	-15	-4	275	-0.005
AY	0.8	0.8	0.5	0.4	0.5	0.2	-0.5	-1.6	-0.2	0.6	-169	0.1	0.1	2	11	90	115	-0.006
EY	0.8	0.5	0.6	0.3	1.1	0.9	-1.1	-2.1	-0.2	0.4	-165	0.1	1.6	-7	-3	56	-159	0.024
AW	0.3	0.8	-0.4	0.3	1.0	0.0	-0.7	-1.2	0.1	0.4	-101	-0.1	0.4	3	-8	342	61	-0.011
AX	1.1	0.7	-0.7	-0.5	0.0	0.4	-0.6	-0.6	0.3	0.9	-118	0.2	1.8	-14	-22	-60	13	0.002
IH	1.0	1.0	0.0	0.1	0.6	0.6	-0.9	-1.6	0.1	0.7	-164	0.2	0.8	-12	-25	-15	-122	-0.008
AE	0.3	0.0	-0.2	-0.1	1.0	1.0	0.1	-2.4	0.0	0.3	-106	0.8	-0.1	5	-20	70	-3	-0.018
AH	1.7	1.7	-0.1	0.3	0.6	0.5	-0.5	-1.7	-0.8	0.9	-219	-0.6	-0.1	-11	-6	201	84	0.005
OY	1.4	1.4	0.7	0.9	0.9	-0.4	-2.0	-1.5	0.0	1.0	-263	0.0	2.4	-8	-16	57	-87	0.011
IY	0.5	-0.4	0.5	0.0	0.6	0.6	-1.2	-1.1	0.4	0.5	-118	0.1	0.5	1	14	57	-1	0.005
OW	0.6	0.5	-1.0	0.9	0.5	-0.1	-0.8	-0.4	-0.1	0.3	-68	0.2	0.8	-9	-4	287	226	0.007
AXR	-0.3	-0.6	-1.7	-1.3	1.5	0.9	0.1	-0.4	0.0	0.2	48	0.2	-0.7	-8	-5	165	-211	-0.050

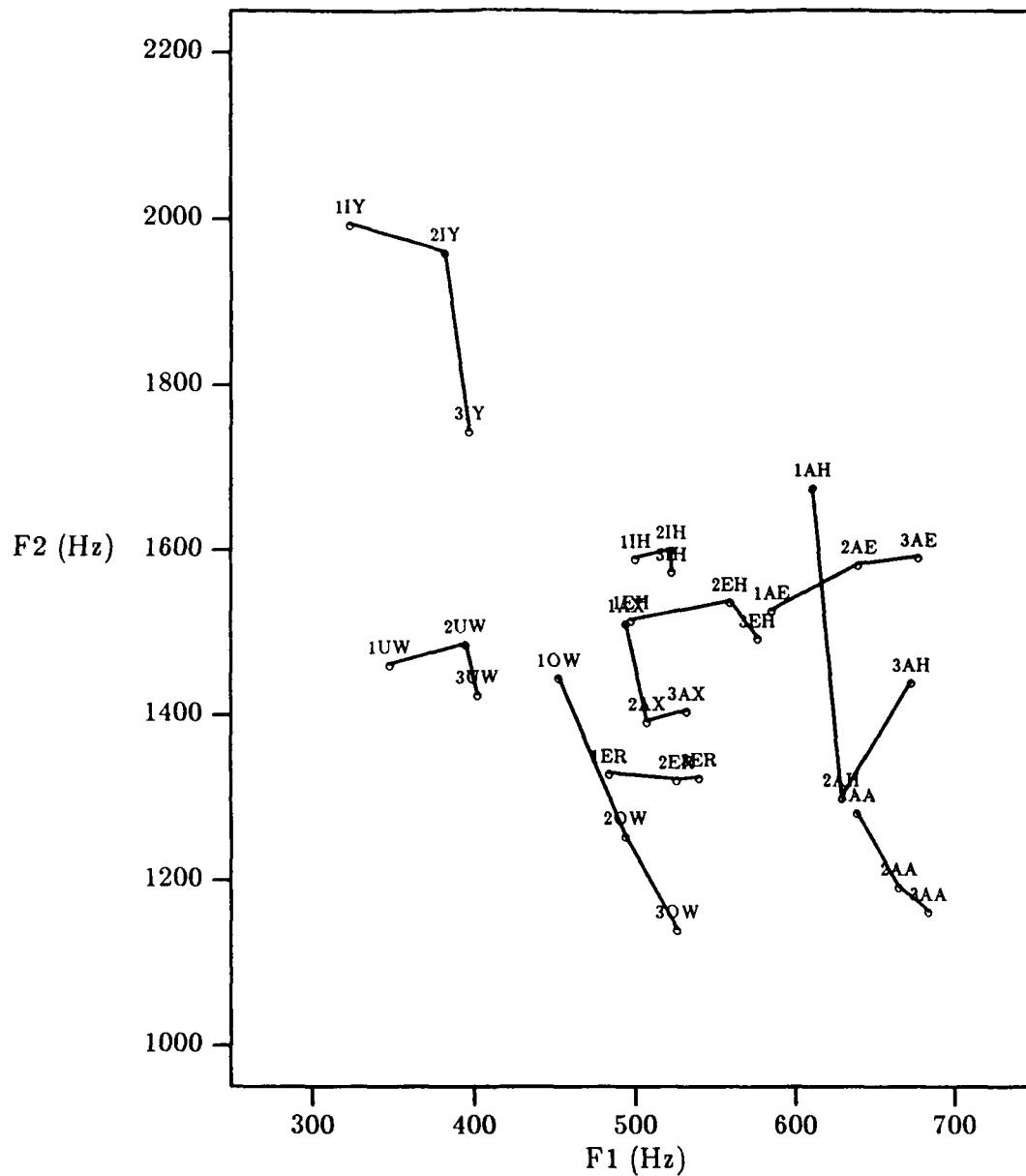


Figure 53. Average shifts of the first and second formants for selected vowels of Speaker #1

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

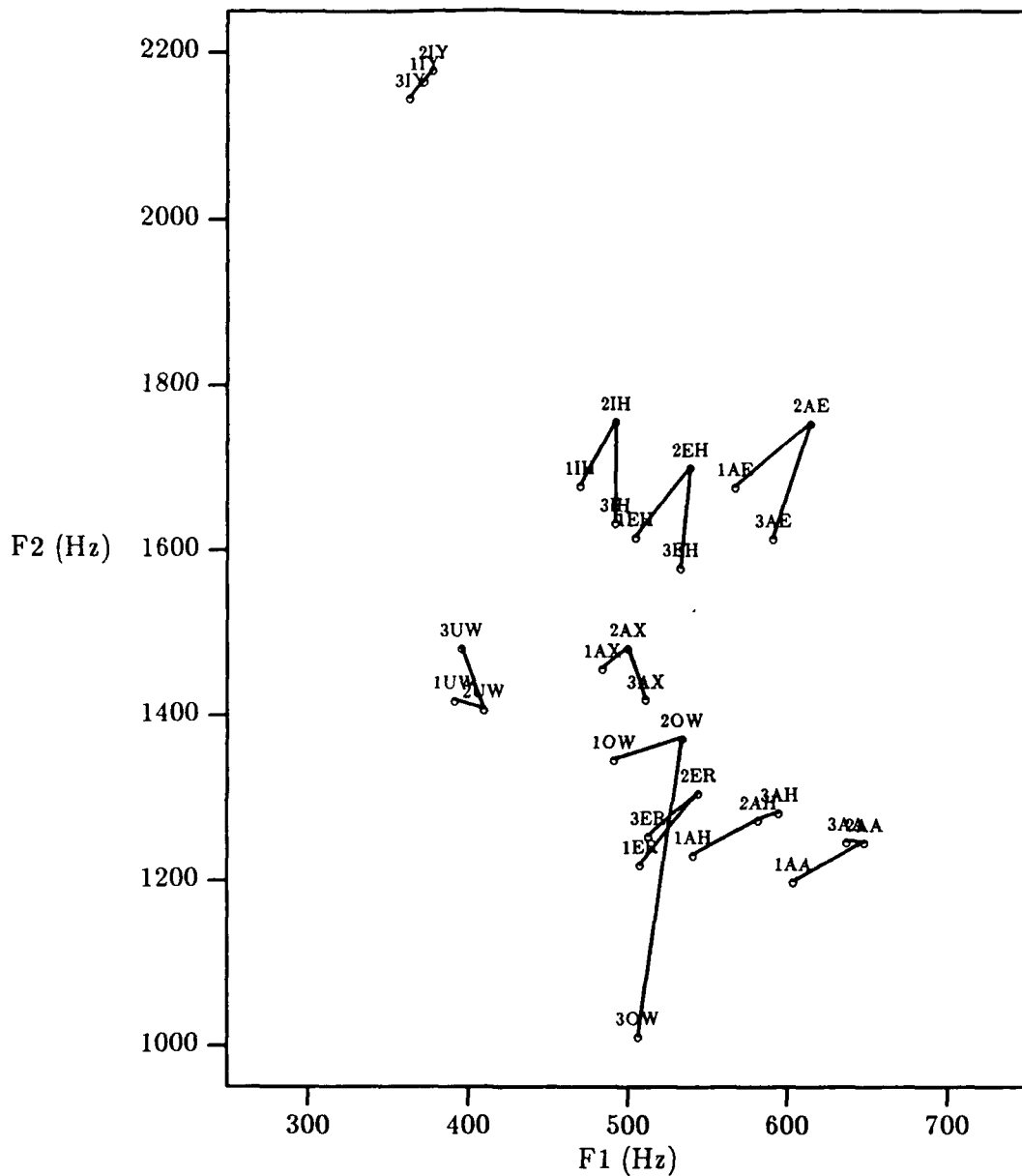


Figure 54. Average shifts of the first and second formants for selected vowels of Speaker #2

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

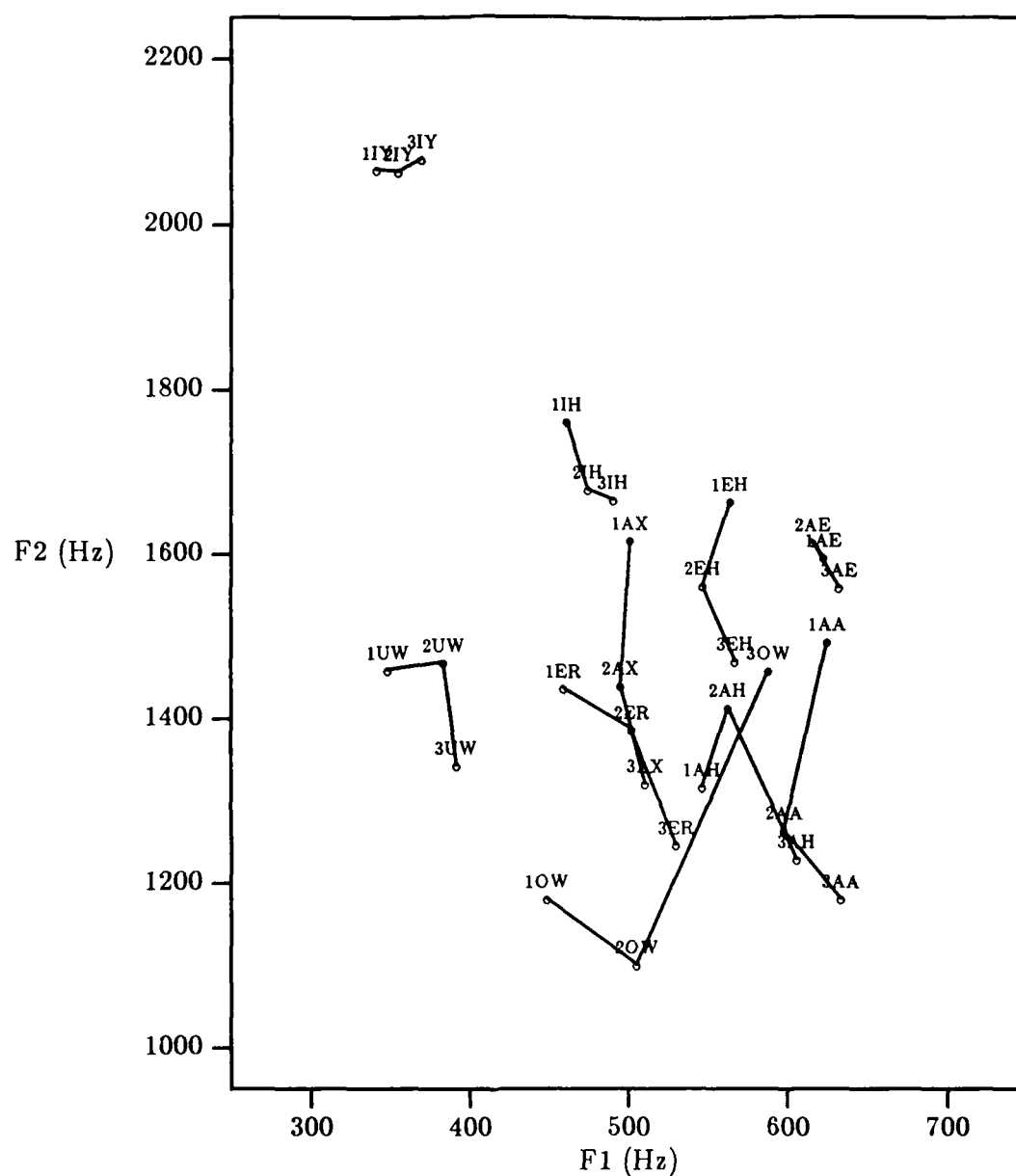


Figure 55. Average shifts of the first and second formants for selected vowels of Speaker #3

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

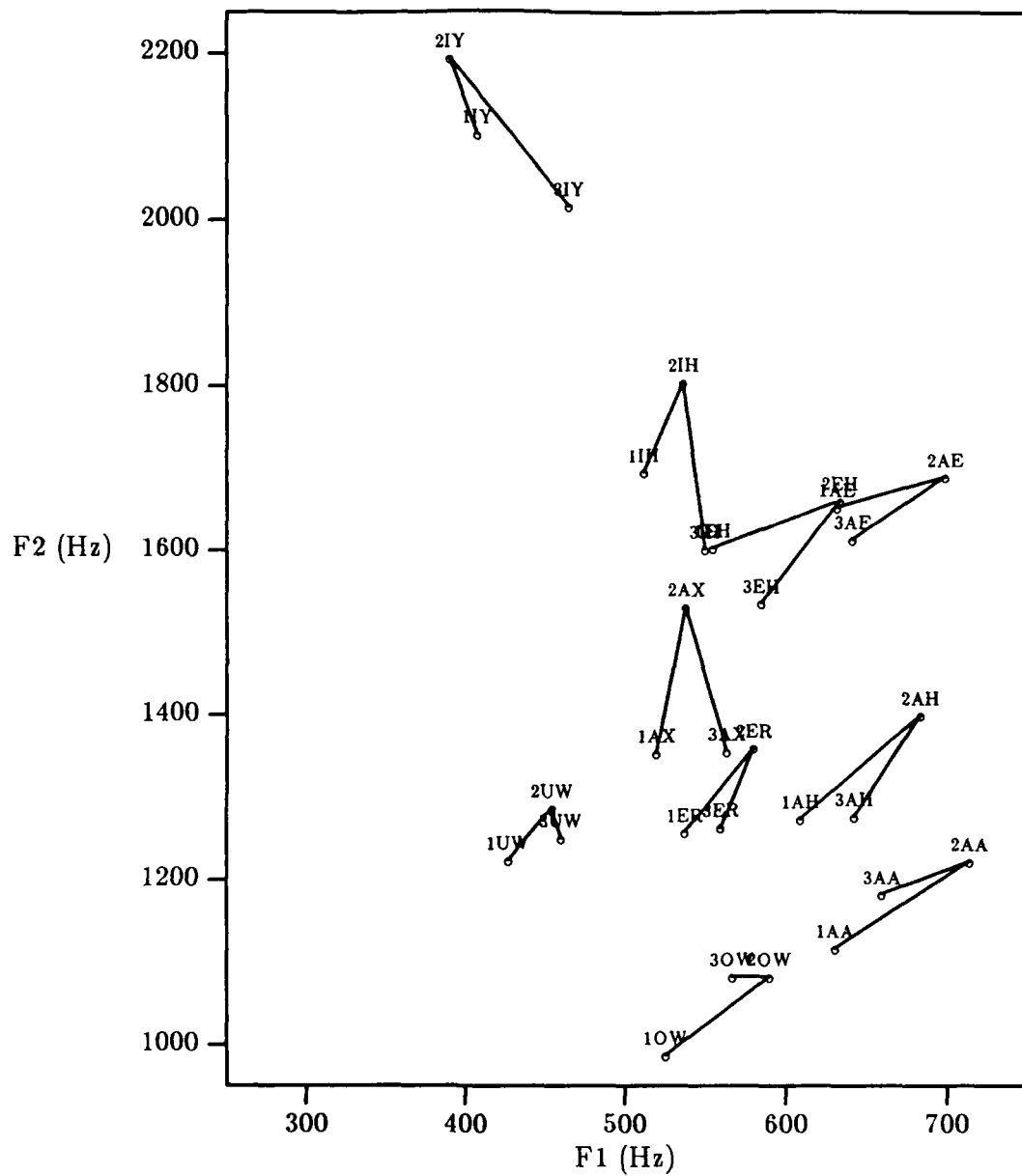


Figure 56. Average shifts of the first and second formants for selected vowels of Speaker #4

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

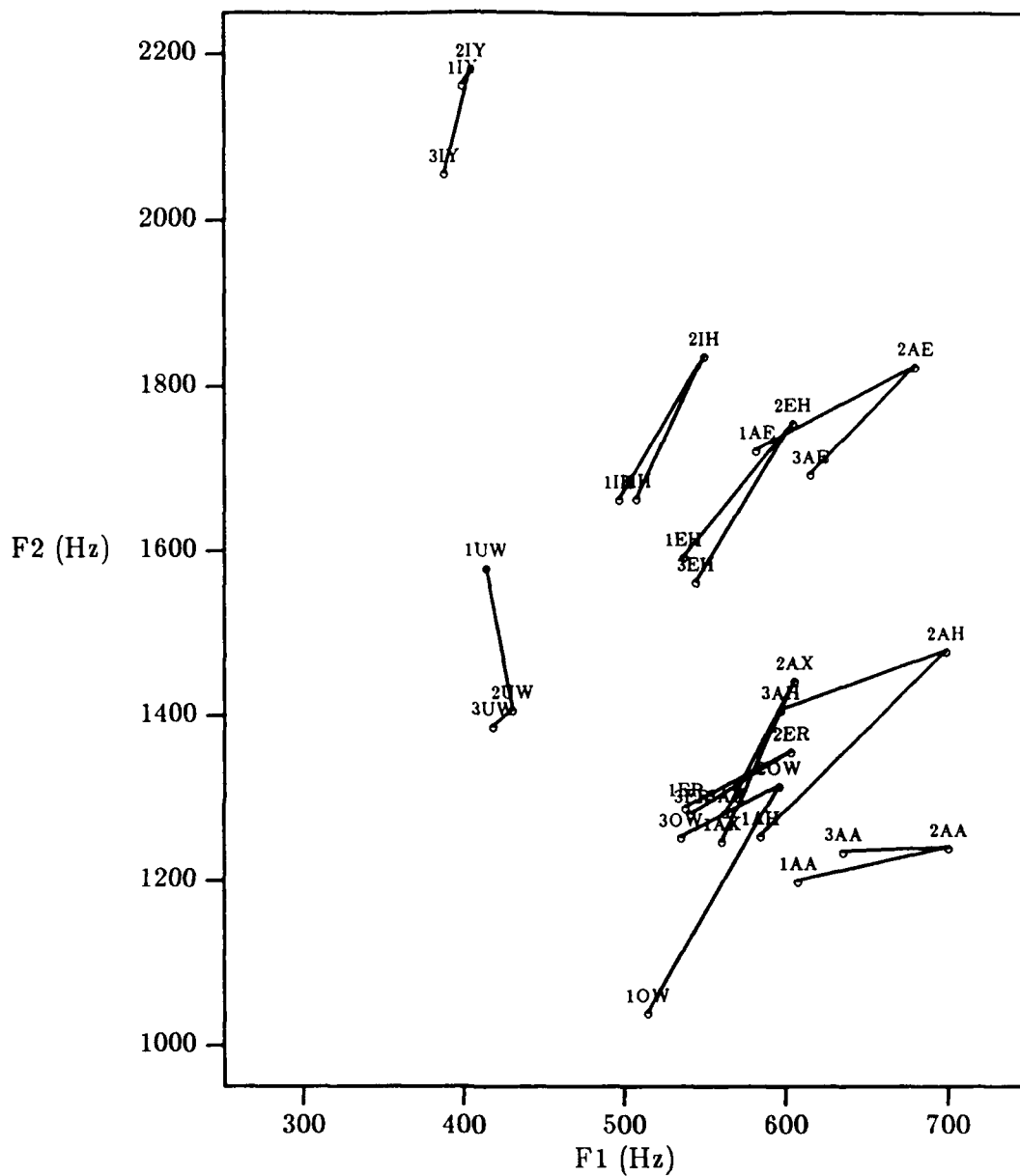


Figure 57. Average shifts of the first and second formants for selected vowels of Speaker #5

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

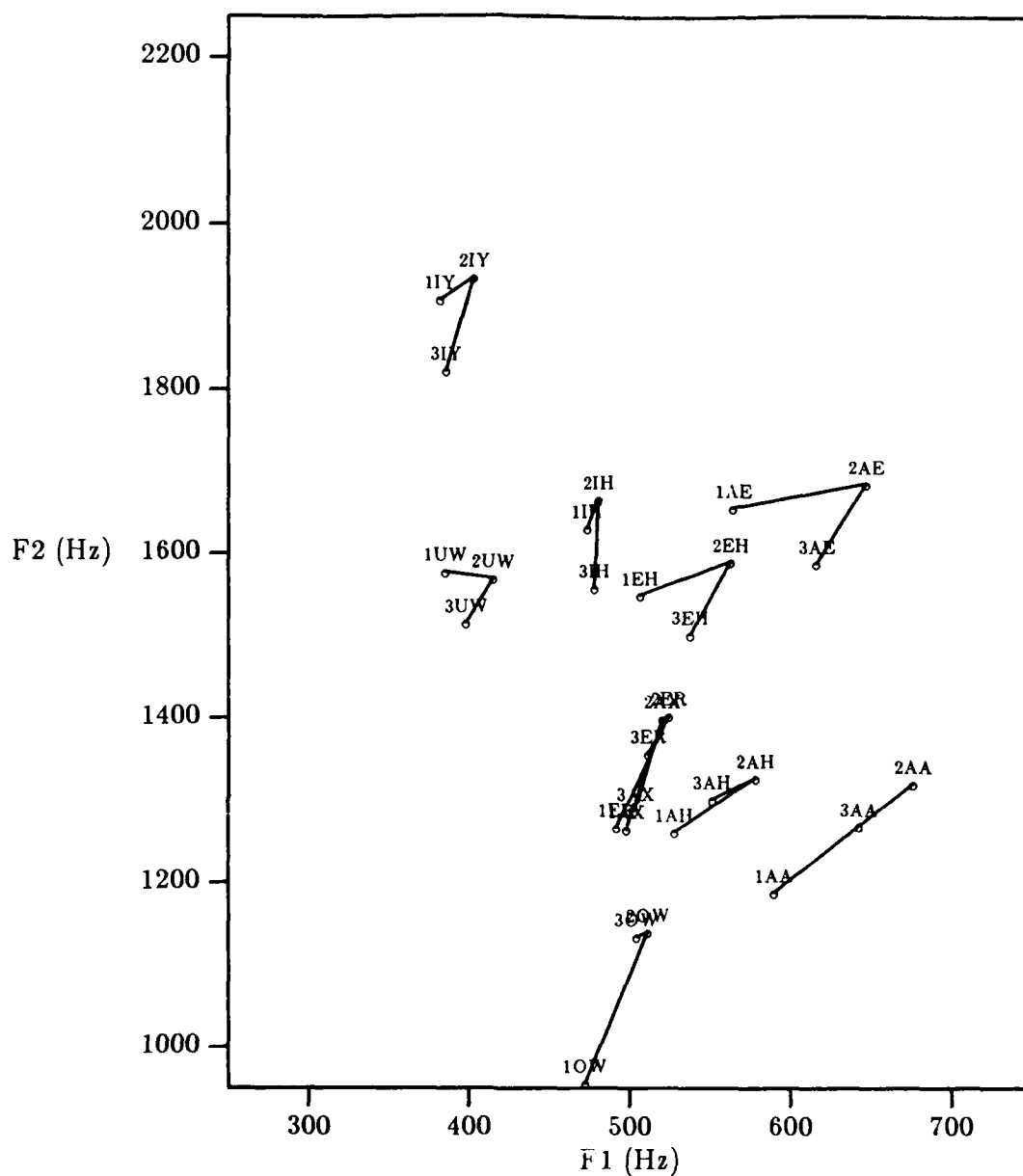


Figure 58. Average shifts of the first and second formants for selected vowels of Speaker #6

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

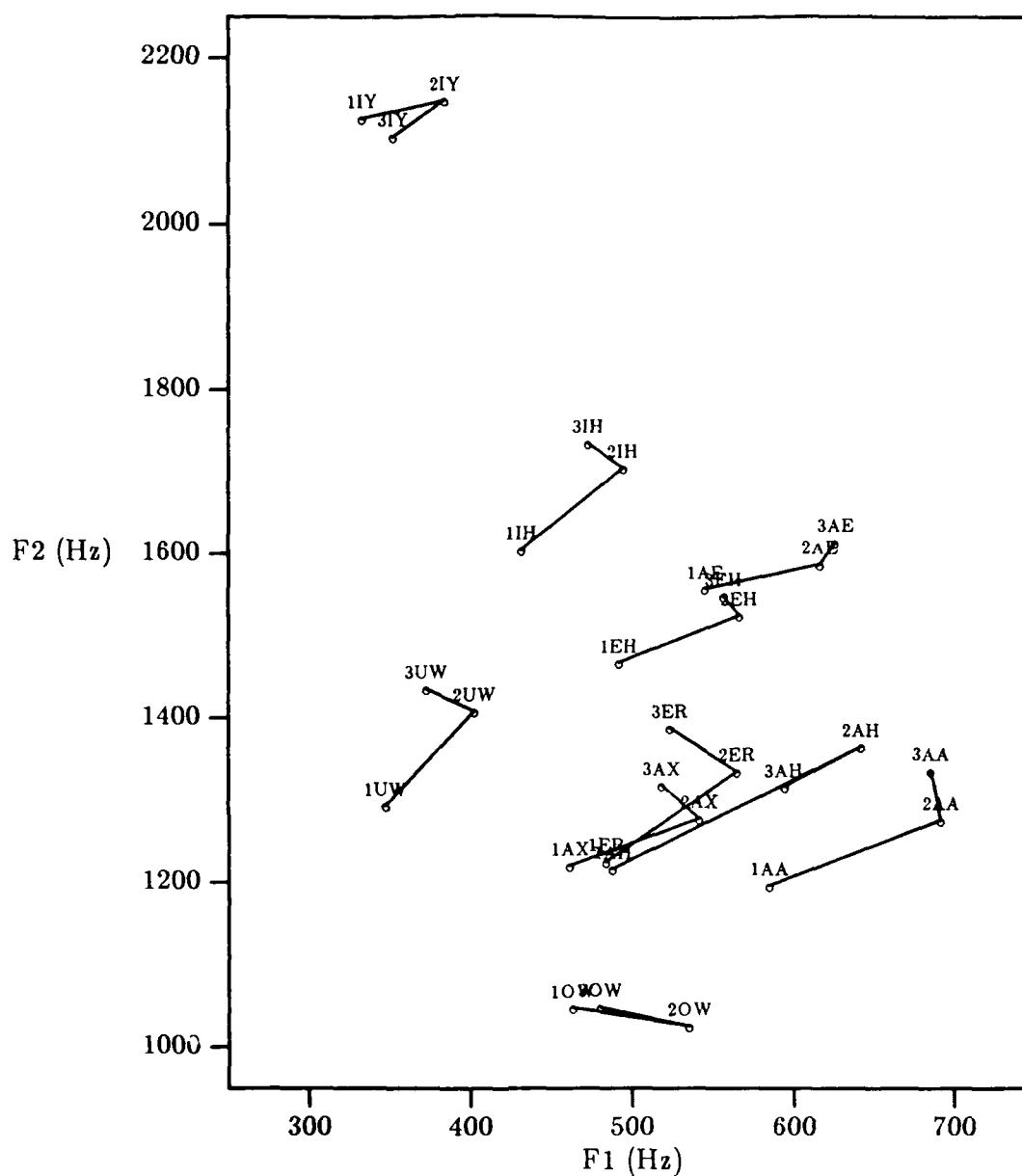


Figure 59. Average shifts of the first and second formants for selected vowels of Speaker #7

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

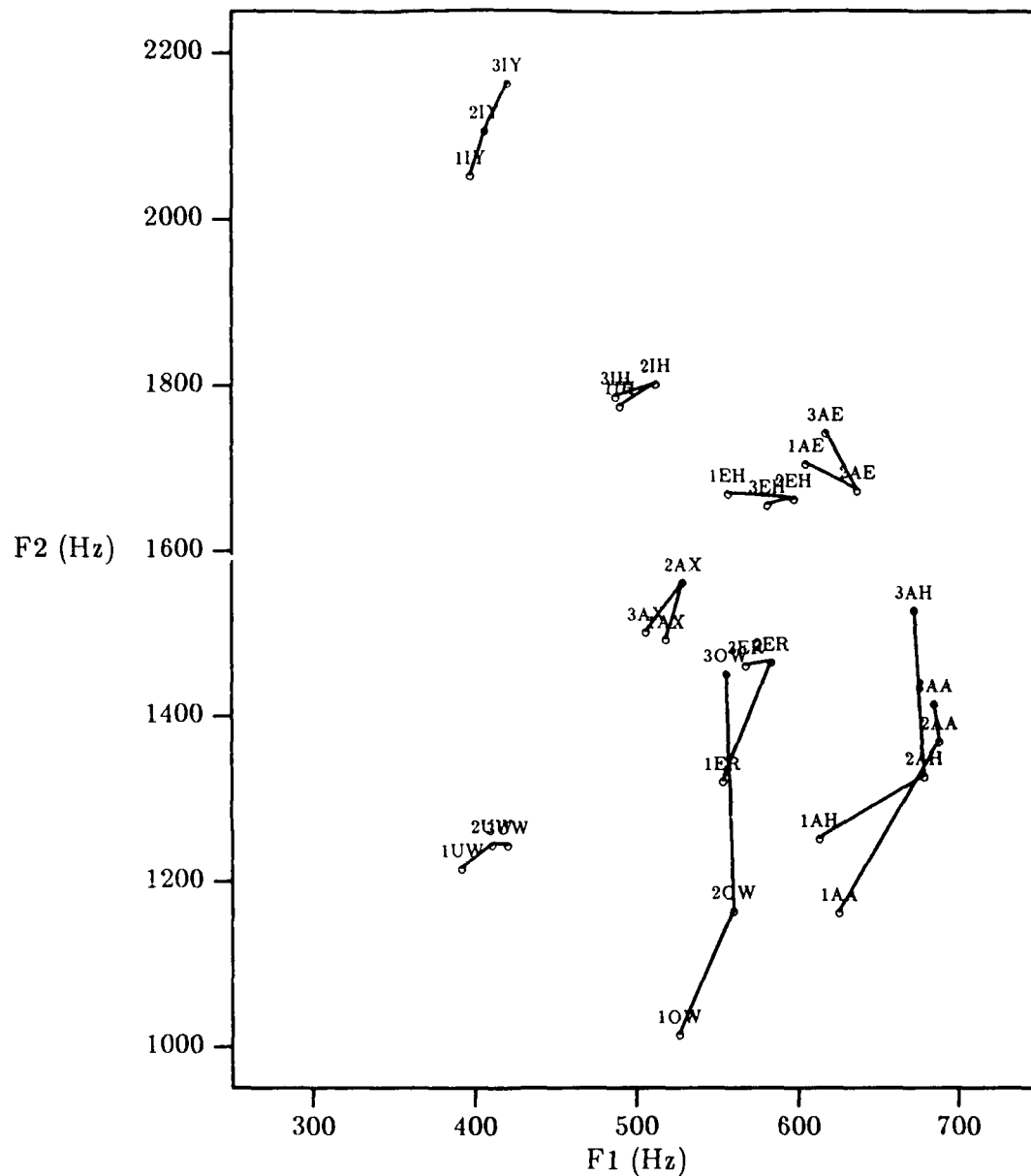


Figure 60. Average shifts of the first and second formants for selected vowels of Speaker #8

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and second formant frequencies for phoneme UW in loud speech.

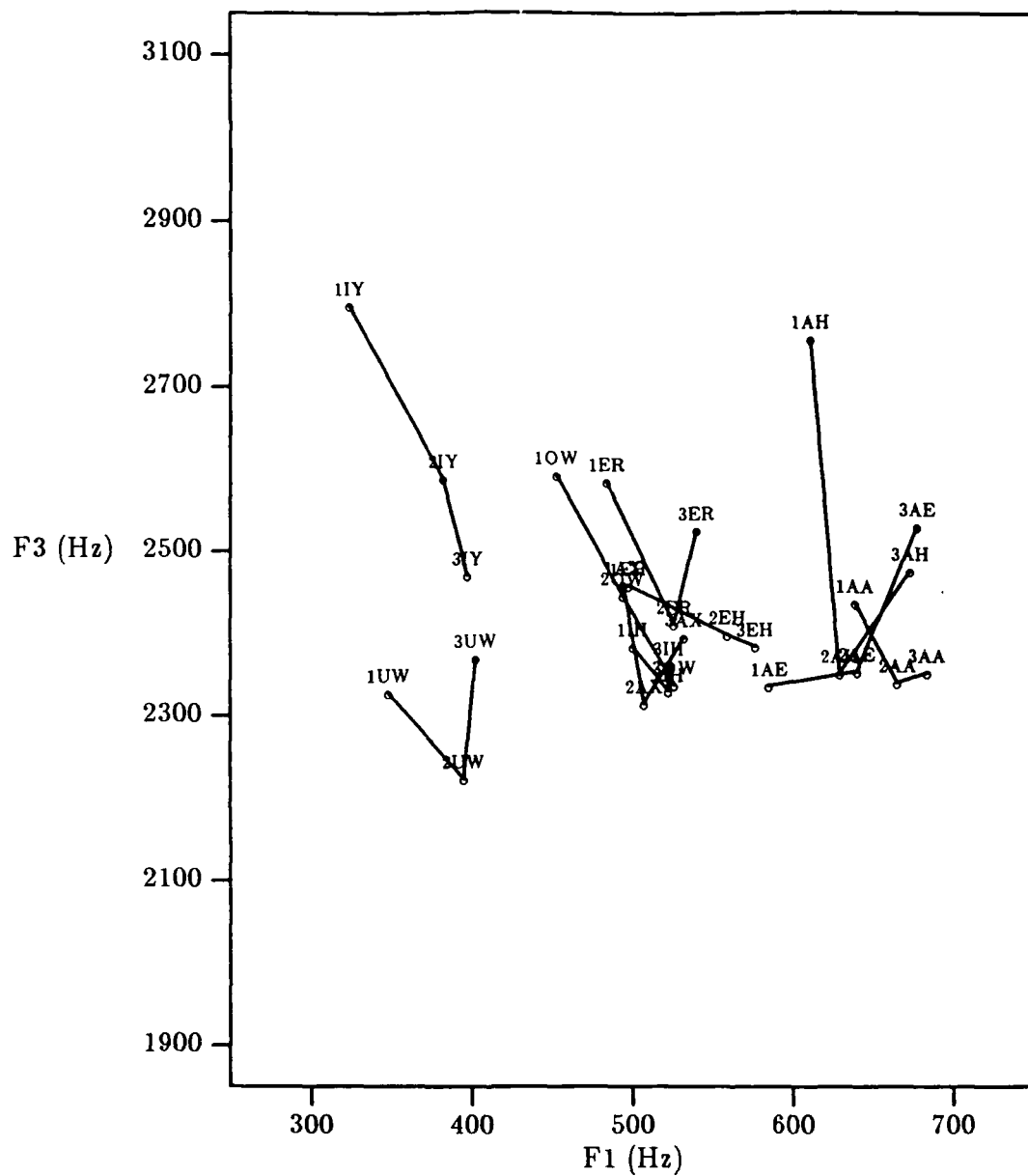


Figure 81. Average shifts of the first and third formants for selected vowels of Speaker #1

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

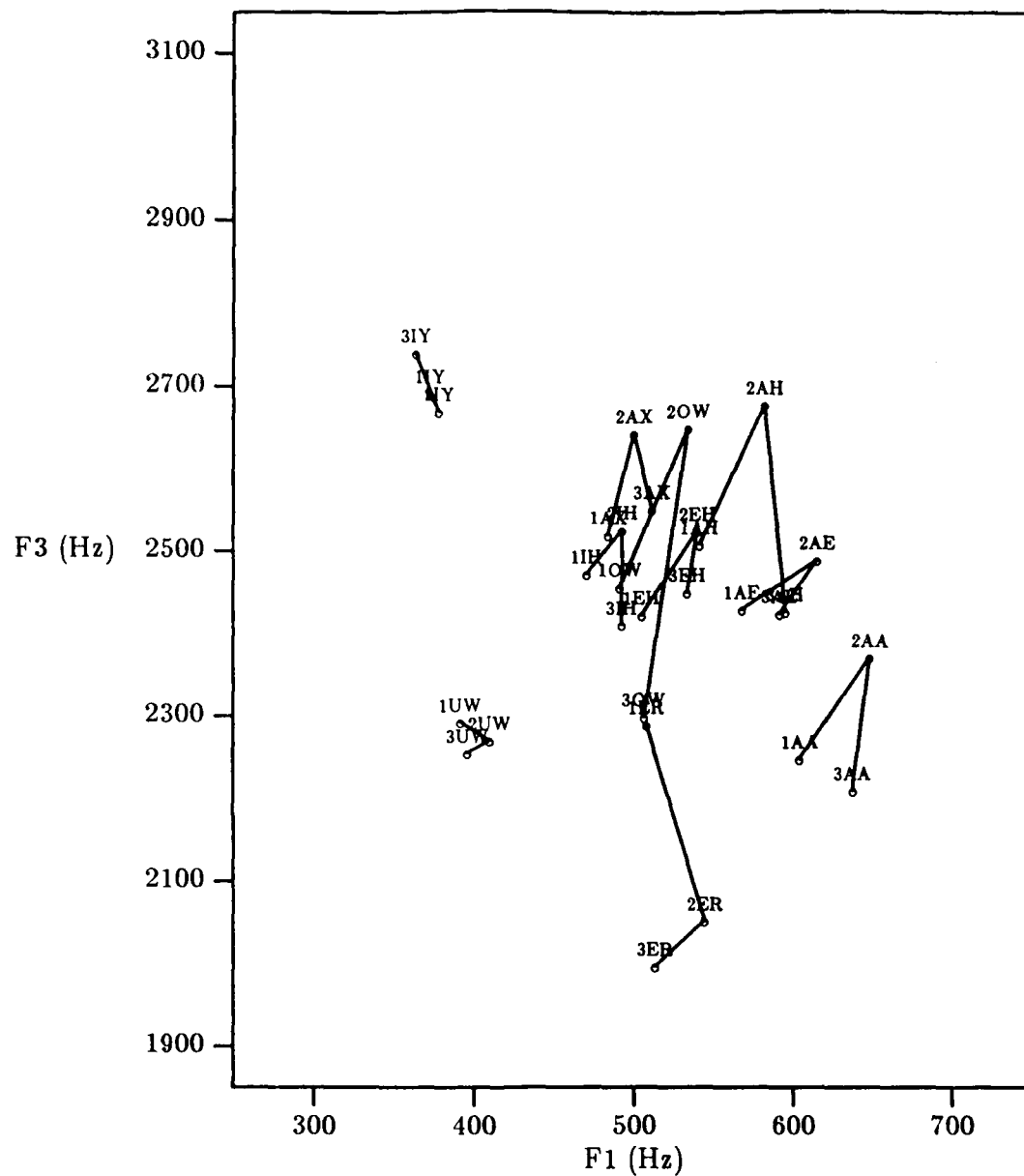


Figure 62. Average shifts of the first and third formants for selected vowels of Speaker #2

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

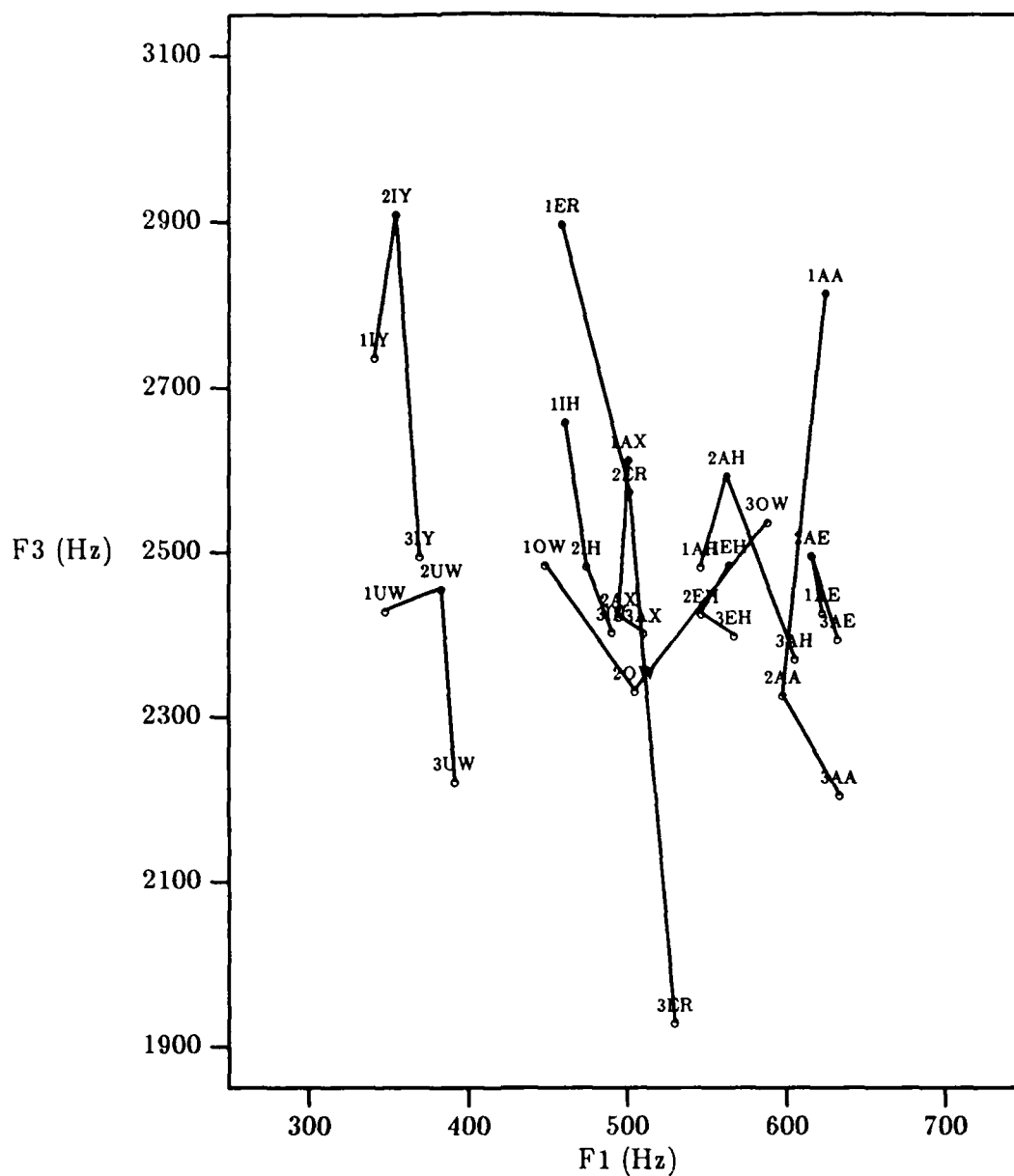


Figure 63. Average shifts of the first and third formants for selected vowels of Speaker #3

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

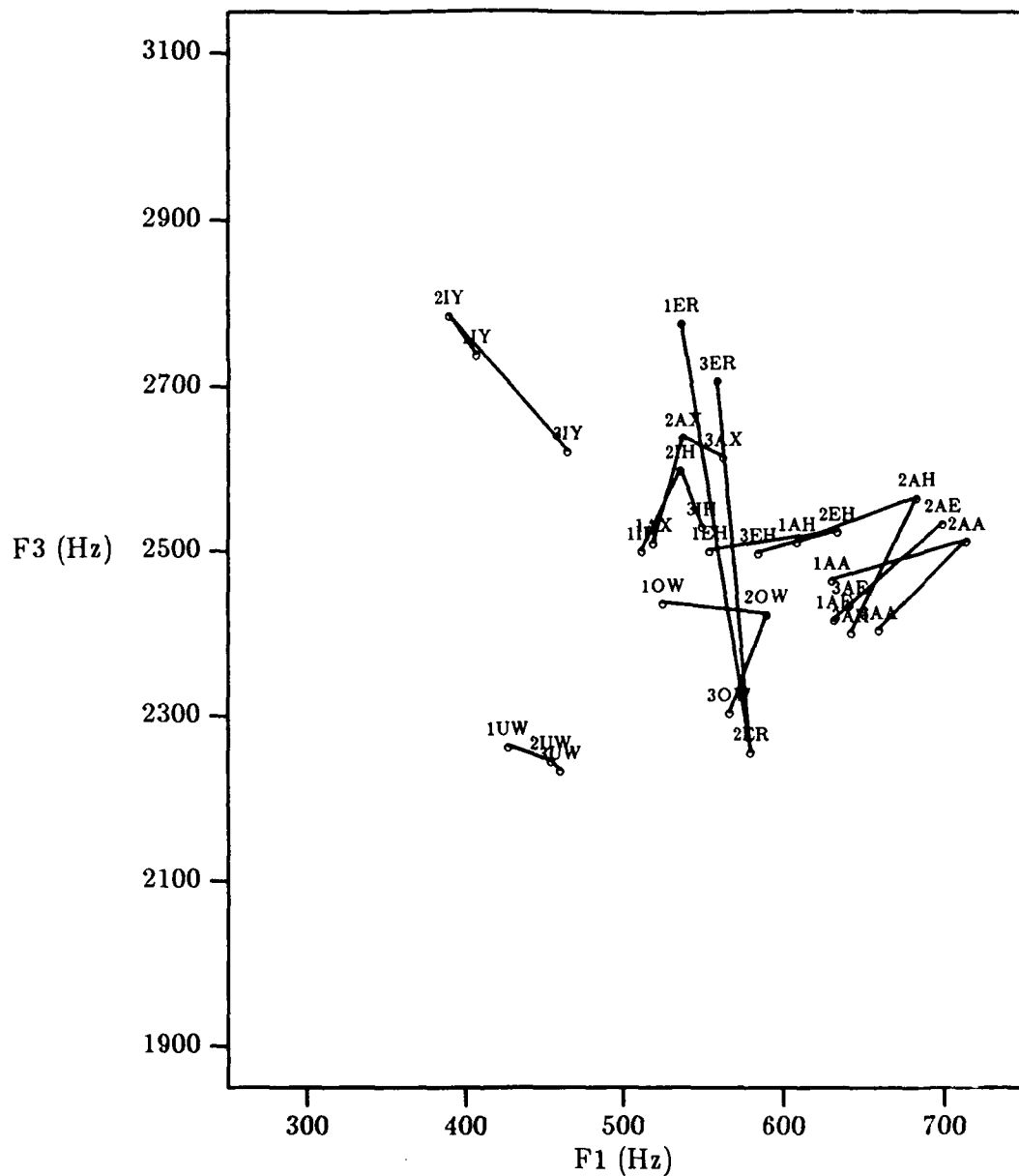


Figure 64. Average shifts of the first and third formants for selected vowels of Speaker #4

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

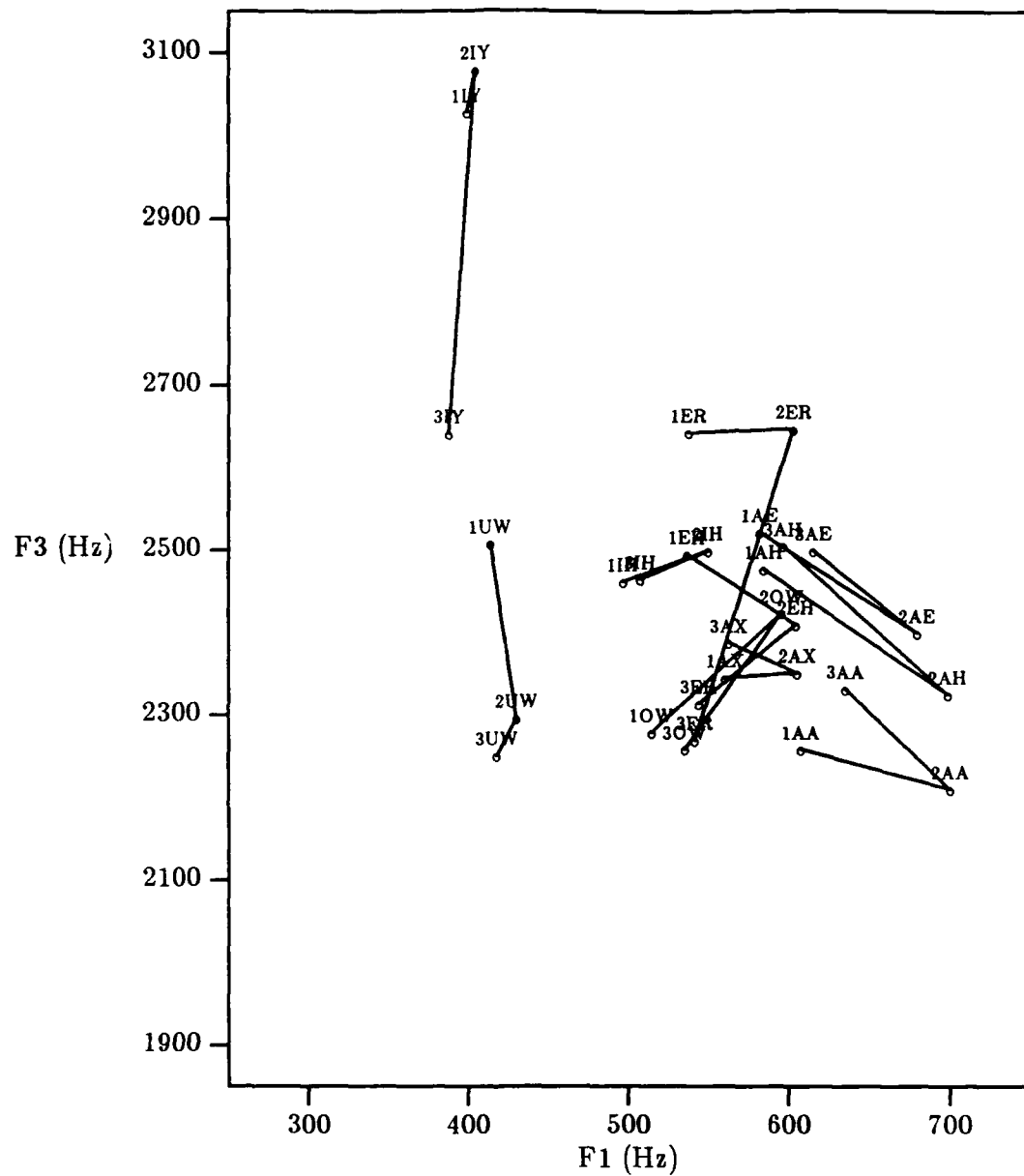


Figure 65. Average shifts of the first and third formants for selected vowels of Speaker #5

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

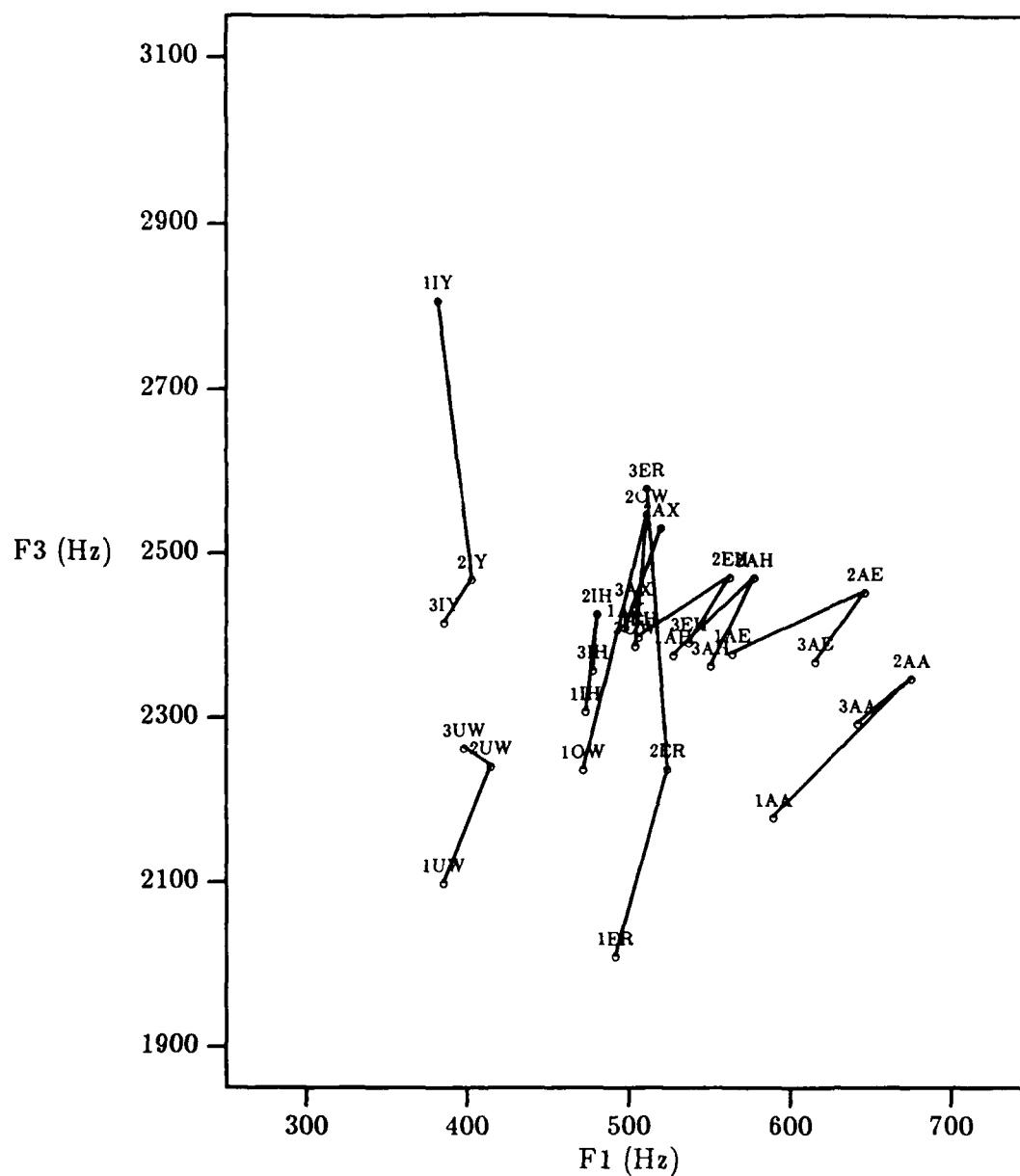


Figure 66. Average shifts of the first and third formants for selected vowels of Speaker #6

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

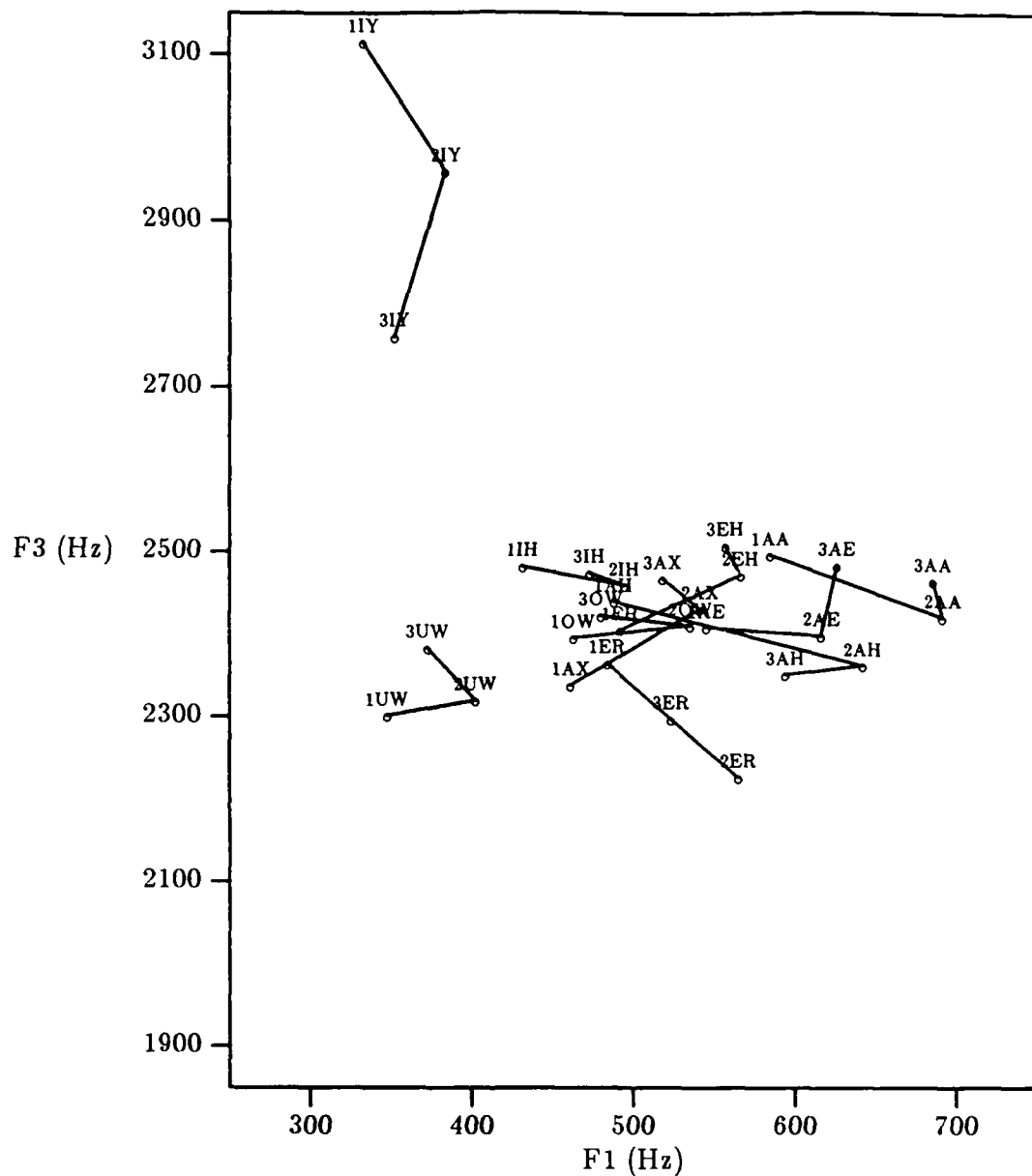


Figure 67. Average shifts of the first and third formants for selected vowels of Speaker #7

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

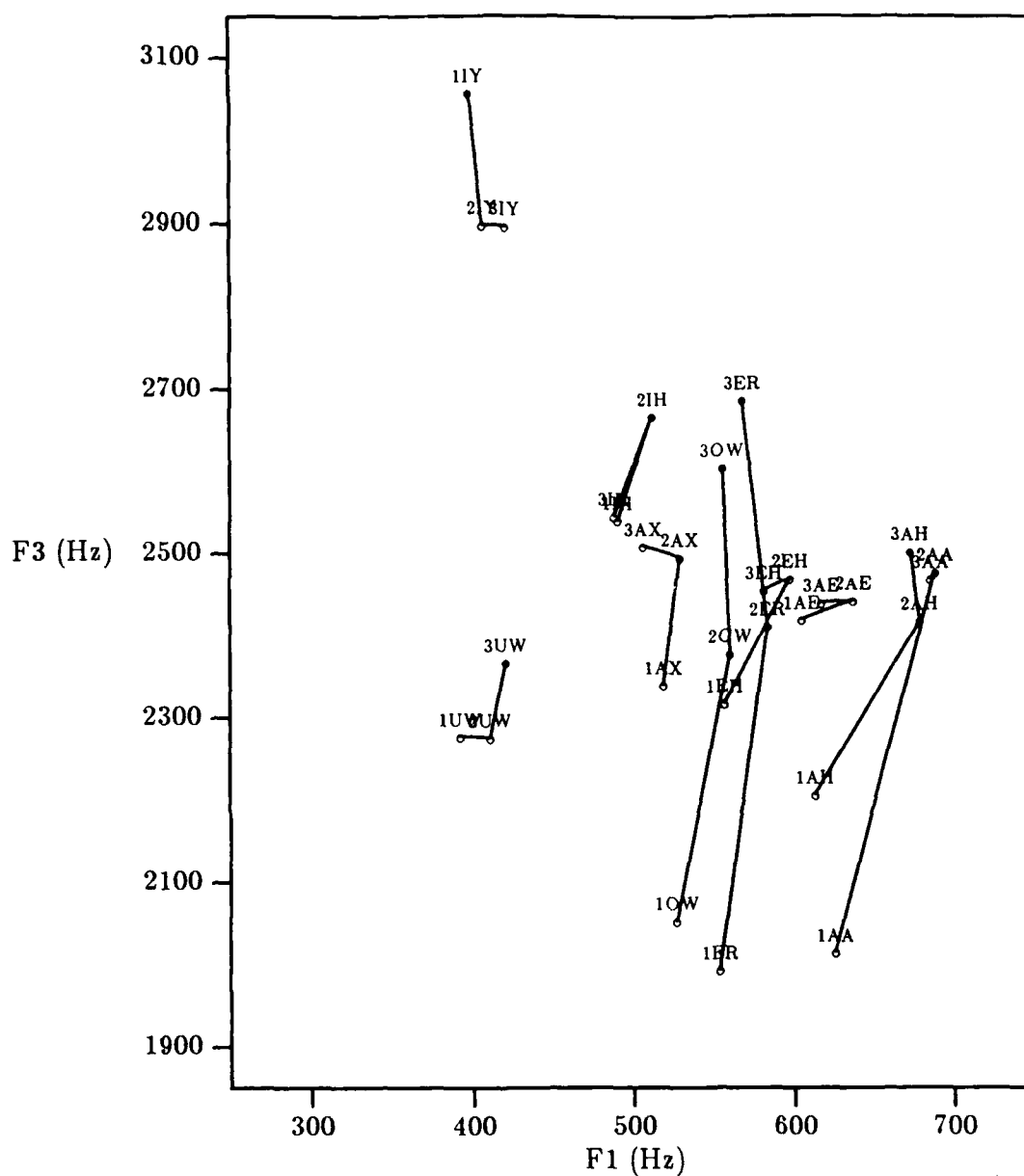


Figure 88. Average shifts of the first and third formants for selected vowels of Speaker #8

Speech condition is indicated by 1=normal, 2=loud, and 3=Lombard. Phoneme is indicated by ARPABET symbol. For example, the point 2UW indicates the average first and third formant frequencies for phoneme UW in loud speech.

Table 55. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #1. Level of significance: 0.01

KEY:																		
Δ indicates feature was higher than normal for both loud and Lombard																		
▼ indicates feature was lower than normal for both loud and Lombard																		
○ indicates feature was higher for loud and lower for Lombard																		
■ indicates feature was lower for loud and higher for Lombard																		
Phones	Energy in Frequency Bands (kHz)										C'OG	Tilt		Pitch	Formants			Dur
	0-0.25	0.25-0.5	0.5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P		○				■											▼	
T	▼			▼	▼	○		Δ		Δ	Δ				○	Δ	▼	
K							■							▼			▼	
B		Δ																
D																		
G																		
DX				Δ		▼	▼				▼					▼		
M	▼	Δ			Δ				▼	▼		Δ		Δ	Δ			
N		Δ	Δ			▼			▼			Δ		Δ				
NX		Δ												Δ				
S		▼	▼	▼	▼	▼		Δ	Δ	Δ	Δ	▼	Δ	Δ	Δ	Δ		
Z					▼	▼							Δ		▼			
CH	▼	▼	▼	▼	▼	Δ	Δ	Δ			Δ							
TH							Δ				Δ							
F									○				○					
SH			▼	▼		Δ	Δ	Δ								Δ		
JH				▼	○	Δ	Δ		■							Δ		
V				Δ						▼		▼	Δ					
L	▼				Δ	■			▼	▼		Δ	▼	Δ				
R	▼	▼	Δ	Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
Y	▼				Δ		■		▼	▼		Δ	▼	Δ			Δ	
HH						▼												
EL					Δ	■												
W	▼		Δ									Δ		Δ				
EH	▼	▼		Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ	Δ		Δ	
AO	▼	▼	Δ	Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ		Δ	
AA	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ			Δ	
UW	▼				Δ	Δ	■	▼	▼	▼	Δ	Δ	▼	Δ	Δ			
ER	▼		Δ	Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
AY	▼	▼	Δ	Δ	Δ	■	▼	▼	▼	▼	Δ	Δ	▼	Δ	Δ			
EY	▼		Δ	Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ			
AW	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ				
AX	▼		Δ	Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ				
IH	▼			Δ	Δ	■	■	▼	▼	▼	Δ	Δ	▼	Δ			Δ	
AE	▼	▼		Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ	Δ		Δ	
AH	▼	▼		Δ	Δ	Δ		▼	▼	▼	Δ	Δ	▼	Δ				
OY	▼		Δ	Δ	Δ	■		▼	▼	▼	Δ	Δ	▼	Δ		▼	Δ	
IY	▼	▼	Δ	Δ	Δ	■	Δ	▼	▼	▼	Δ	Δ	▼	Δ	Δ	▼		
OW	▼	▼			Δ	Δ	Δ	▼	▼	▼	Δ	Δ	▼	Δ	Δ			
AXR	▼		Δ	Δ	Δ	■	▼	▼	▼	▼			▼			▼		

Table 56. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #2, level of significance: 0.01

KEY:																			
Δ		indicates feature was higher than normal for both loud and Lombard																	
▼		indicates feature was lower than normal for both loud and Lombard																	
○		indicates feature was higher for loud and lower for Lombard																	
◻		indicates feature was lower for loud and higher for Lombard																	
Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur	
	0-25	25-50	50-75	75-100	100-150	150-200	200-300	300-400	400-600	600-800		Lo	Hi		1	2	3		
P	▼	▼	▼	▼				Δ	Δ	▼	Δ	Δ	▼			Δ	◻	◻	
T	▼	○	○	○	▼	▼		Δ	Δ		Δ	◻							
K								Δ	◻	▼									
B																			
D																			
G																			
DX	◻	◻			Δ					▼		○		Δ					
M	◻										▼	○		Δ	▼				
N	◻	Δ			○	○		◻	◻		▼	○		Δ	▼	○	○		
NX	◻				○	○						○			○	○			
S	▼	▼	▼		▼	▼		Δ	Δ	○	Δ	Δ	Δ		Δ			Δ	
Z	◻				▼	▼		Δ	Δ	○		○							
CH	○	○	○	○	▼			Δ	◻		◻	Δ							
TH	▼	▼						◻	◻			Δ				Δ			
F			Δ	Δ		▼			◻	▼	▼					Δ			
SH	▼	▼	▼	▼	▼			Δ	Δ	○	◻	Δ				Δ			
JH					▼			Δ	Δ			○				Δ			
V	◻											○		Δ					
L	◻	◻				Δ					○	○		Δ					
R	◻	◻	Δ	Δ	Δ	Δ	▼	▼	▼	▼	▼	○		Δ				▼	
Y	◻	◻				○					○			Δ		Δ			
HH					▼					Δ						Δ			
EL	◻																		
W	◻										▼	▼		Δ					
EH	◻	◻	Δ	Δ	Δ	Δ		▼	▼	▼	○	○		Δ	Δ	○	Δ		
AO	◻	◻	◻	◻	Δ	Δ			▼	▼	▼	○	○	Δ	Δ	Δ		Δ	
AA	◻	◻	◻		Δ	Δ			▼	▼	▼	○	○	○	Δ	Δ			
UW	◻	◻		Δ	Δ	Δ	▼	▼	▼	▼	▼	○	○	○	Δ	Δ			
ER	◻	◻		Δ	Δ	Δ	▼	▼	▼	▼	▼	○	○	Δ	Δ	Δ			
AY	◻	◻	Δ		Δ	Δ		▼	▼	▼	▼	○	○	○	Δ			Δ	
EY	◻	Δ	Δ		○	○		▼	▼	▼	▼	○	○	○	Δ	Δ			
AW	◻	◻	Δ		○	Δ		▼	▼	▼	▼	○	○	○	Δ	Δ			
AX	◻	◻			Δ	Δ		▼	▼	▼	▼	○	○	○	Δ	Δ			
IH	◻	◻			Δ	Δ	▼	▼	▼	▼	▼	○	○	○	Δ	Δ	Δ		Δ
AE	◻	◻			○			▼	▼	▼	▼		○	○	Δ	Δ			
AH	◻	◻		Δ	Δ			▼	▼	▼	▼		○	○	Δ	Δ	Δ		
OY	◻	◻			Δ	Δ		▼	▼	▼	▼	○	○	○	Δ	Δ			Δ
IY	◻	◻	Δ	Δ	○	○		▼	▼	▼	▼	○	○	○	Δ	Δ			
OW	◻	◻		Δ	Δ	Δ	▼	▼	▼	▼	▼	○	○	○	Δ	Δ	Δ		
AXR	◻	◻	Δ	Δ	Δ	○	▼	▼	▼	▼	▼	○	○	○	Δ		▼	▼	Δ

Table 57. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #3, level of significance: 0.01

KEY:																		
Δ indicates feature was higher than normal for both loud and Lombard																		
▼ indicates feature was lower than normal for both loud and Lombard																		
○ indicates feature was higher for loud and lower for Lombard																		
◻ indicates feature was lower for loud and higher for Lombard																		
Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-0.5	0.5-1	1-1.5	1.5-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	▼	▼		◻	Δ	Δ	Δ		▼	○	Δ	Δ	▼		◻			▼
T	▼	▼		▼		◻	Δ	Δ			Δ	Δ	○					
K																		
B																		
D																		
G																		
DY		Δ												Δ				
M	▼	Δ	Δ	◻	Δ			○	▼	▼		Δ		Δ	Δ	○	○	
N		Δ	Δ	◻	Δ	▼		○	▼	▼	▼	Δ		Δ		○	▼	
NX		Δ	Δ					▼	▼	▼				Δ	▼		▼	
S	▼			Δ		▼			◻		Δ	Δ	○			▼	Δ	
Z					▼	▼			Δ	Δ			Δ					▼
CH																		
TH					○			◻	◻				○					
F	▼							Δ										
SH	▼	▼		▼				Δ	Δ		Δ		▼					
JH	▼					◻		Δ	◻		○		▼					
V		Δ						▼			▼			Δ	▼	▼	▼	
L		Δ	Δ					▼						Δ		▼	▼	
R	▼	○	Δ	Δ	Δ		Δ	▼	▼	▼	◻	Δ	▼		Δ	○	▼	
Y	▼					◻		▼	▼	▼		Δ						
HH																		
EL	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼	Δ	▼	Δ		▼	▼	
W		Δ	Δ	◻		◻		▼	▼	▼	▼	▼			◻			
EH	▼	○	Δ	Δ	Δ	Δ		▼	▼	▼	▼	◻	Δ	▼		▼		
AO	▼	○	Δ	Δ	Δ	Δ			▼	▼	▼	◻	Δ	▼				
AA	▼	○	Δ	Δ	Δ	Δ	◻		▼	▼	▼	◻	Δ	▼			▼	
UW	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ			
ER	▼	Δ	Δ	Δ	Δ	Δ	◻	▼	▼	▼	▼		Δ	▼	Δ	Δ	▼	
AY	▼	○	Δ	Δ	Δ	Δ		▼	▼	▼	▼	◻	Δ	▼	Δ			
EY	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ	Δ		
AW	▼	○	Δ	Δ	Δ	Δ			▼	▼	▼		Δ	▼	Δ		▼	
AX	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ		▼	
IH	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼	◻	Δ	▼			▼	
AE	▼		Δ	Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ			
AH	▼	○	Δ	Δ	Δ	Δ	◻	▼	▼	▼	▼	◻	Δ	▼	Δ			
OY	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼		Δ	▼	Δ	Δ		
IY	▼	Δ	Δ	Δ	Δ	Δ		▼	▼	▼	▼	◻	Δ	▼	Δ	Δ	○	
OW	▼	○	Δ	Δ	Δ	Δ		▼	▼	▼	▼	◻	Δ	▼	Δ	Δ		
AXR	▼	Δ	Δ	Δ	Δ	Δ	◻	○	▼	▼	▼	▼	Δ	▼		▼	▼	Δ

Table 58. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #4, level of significance: 0.01

KEY:		indicates feature was higher than normal for both loud and Lombard																	
		indicates feature was lower than normal for both loud and Lombard																	
		indicates feature was higher for loud and lower for Lombard																	
		indicates feature was lower for loud and higher for Lombard																	
Phonemes	Energy in Frequency Bands (kHz)											COG	Tilt		Pitch	Formants			Dur
	0-25	25-50	50-100	100-200	200-300	300-400	400-500	500-600	600-700	700-800	Lo		Hi	1		2	3		
P			▼	■	■	Δ	Δ	○	▼	▼	Δ	■	▼			Δ			
T	▼	▼		○	▼		Δ	Δ	■		Δ	○				Δ	▼		
K			Δ			○		Δ	▼										
B						Δ	Δ	○											
D																			
G					Δ				▼			Δ							
DX	Δ	○											▼	Δ					
M					Δ			○								Δ			
N	▼	■	■		Δ	○	▼	○	▼		○	Δ		Δ		Δ			
NX	▼		■	■							○					Δ			
S	▼	○	Δ	○	○	▼	■	■	■	■	■	○		Δ	○	▼	▼		
Z			○	○	○		■	■	■	■						Δ			
CH			○			▼													
TH						Δ	Δ					○	▼			Δ			
F	▼			○	▼		Δ	Δ	▼		■								
SH																			
JH	▼		Δ	Δ		■			▼			Δ						Δ	
V			■									▼		○	■				
L																			
R	▼	▼	■	Δ		Δ	○		▼	▼	Δ	Δ	▼	○	Δ	Δ		Δ	
Y	▼												■		Δ				
HH																○			
EL	○	▼	■						▼	▼	Δ		▼	○					
W	▼	○											▼	Δ					
EH	▼	▼	■		▼	Δ	Δ		▼	■	Δ	○	■	Δ	Δ	○		Δ	
AO			■	Δ					○		Δ			Δ	Δ	Δ	○	○	
AA	▼	▼	▼			Δ	Δ	Δ	▼	▼	Δ	Δ	▼	Δ	Δ				
UW		▼		Δ		Δ	○	Δ	▼	▼	○	Δ	▼	Δ	Δ				
ER	▼	▼	■			Δ		○	▼	▼	Δ	Δ	■	Δ	Δ	Δ	▼		
AY	▼	▼	■		▼	Δ	Δ	Δ	▼	▼	Δ	Δ	▼	Δ	Δ	○	○		
EY			Δ		▼	Δ		○	▼	■	○		■	Δ	Δ	Δ			
AW			■			Δ			▼	▼	Δ		▼	Δ	Δ	○		Δ	
AX	▼	▼	■	Δ	Δ	Δ			▼	▼	○	Δ	■	Δ	Δ	○		Δ	
IH	▼	▼	■			Δ			▼	▼	○		■	Δ		○			
AE			■		▼	Δ			▼	▼	Δ			Δ	Δ		Δ		
AH	▼	▼		Δ					▼	▼	Δ			Δ	Δ	Δ			
OY	▼	▼	■	Δ		Δ	○	▼	▼	▼	Δ		▼	Δ	Δ	Δ		Δ	
IY	▼	○	Δ	Δ		Δ	○		▼	▼		Δ		Δ	■	○			
OW	▼	▼	■	Δ		Δ	Δ	○	▼	▼	Δ		▼	Δ	Δ			Δ	
AXR	▼	▼	■	Δ		Δ		▼	▼	▼	○	Δ	▼	Δ			▼		

Table 59. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #5, level of significance: 0.01

KEY:																			
△		indicates feature was higher than normal for both loud and Lombard																	
▼		indicates feature was lower than normal for both loud and Lombard																	
○		indicates feature was higher for loud and lower for Lombard																	
□		indicates feature was lower for loud and higher for Lombard																	
Phones		Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			dur
		0-25	25-50	50-75	75-100	100-150	150-200	200-300	300-400	400-500	500-700		Lo	Hi		1	2	3	
P		□	□	□	□	△	△		○	○	○			▼			▼	○	
T		□	□	□	□	□	▼	○	○	△	○	○	○		□				
K		□							▼	□									
B					□				▼	▼	○			○					
D									▼	▼									
G																			
DX					△				□				△		△				
M															△			▼	
N		△		▼		△					□			▼	△	▼	▼		
NX															△				
S		△	□	□	□	□	▼		○	△	○	○	○	△	△	○	▼	▼	△
Z				△		▼	▼		○				▼	△			▼	▼	
CH		□	□	□			○		○										
TH		□							○										
F		□	□	▼	▼				○	△	△	△	○	△					
SH		□	□	□	□		○		○			○			△				
JH			□	□	□		○		○										
V															△				
L						△													
R		□	▼	○	△	△		○	▼	▼	▼	△	△	▼	△			○	
Y			▼		△	△					▼		△		△			○	
HH						△									△				
EL		□	▼	▼			△			▼	▼	△		▼	△				
W						□									△				
EH		▼	▼	▼		△	△	△		▼	▼	△	△	▼	△	△	○		
AO		□	▼	▼			○	△			▼	△	△	▼	△	△			
AA		□	▼	▼				△			▼	△	△	▼	△	△			
UW		▼	▼	▼		△	△		▼	▼	▼	△	△	▼	△	△			
ER		□	▼	▼		△			▼	▼	▼	△	△	▼	△	△			
AY		□	▼	▼	○	△	△	△		▼	▼	△	△	▼	△	△			
EY		▼	▼	▼		△		△	□	□	▼	△	△	▼	△	△			
AW		□	▼	▼						□	▼	△	△	▼	△	△	△		
AX		□	▼	▼							▼	△	△	▼	△	△	△		
IH		▼	▼	▼		△	△	△	▼	▼	▼	△	△	▼	△	△	△		
AE		□	▼	▼		△	△	△		□	▼	△	△	▼	△	△	△		
AH		▼	▼	▼	○	△		△		▼	▼	△	△	▼	△	△	△		
OY		□	▼	▼		△			▼	▼	▼		○	▼	△	△	△		
IY		▼	▼	▼		△	△	△	▼	▼	▼			▼	△	△	△	○	
OW		□	▼	▼		△	△	△	▼	▼	▼	△	△	▼	△	△	△		
AXR		▼	▼	○	△	△	△	△	▼	▼	▼	△	△	▼	△	○	△	▼	△

Table 60. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #6, level of significance: 0.01

KEY:																		
△ indicates feature was higher than normal for both loud and Lombard																		
▼ indicates feature was lower than normal for both loud and Lombard																		
○ indicates feature was higher for loud and lower for Lombard																		
□ indicates feature was lower for loud and higher for Lombard																		
Phones	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants			Dur
	0-0.25	0.25-0.5	0.5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8		Lo	Hi		1	2	3	
P	▼	▼		△	△		△		▼	▼	△	△	▼					
T	□	▼	□	▼		▼	△	△	▼		△		▼					
K	△					□		▼	▼				○					
B										▼								
D																		
G			▼															
DX					△				○					△				
M	△		▼					△	△	△		▼		△	▼			
N	△	△	▼		△	▼	△	○	○	○			△	△	▼			
NX	□													△				
S		□	□	▼	△	▼	△	○	○	△	○	▼		△	▼	△		
Z					△	▼			○									
CH					△	▼							△					
TH	□	△			△	▼	△		▼								▼	
F	□	△	△	▼	△	▼			▼	▼	▼				▼		▼	
SH		□	▼	▼			△			△	△					△	△	
JH						▼							△					
V	□	△			△	▼				▼		△	○	△	▼			
L							□		▼	▼		△	▼	△				
R	□			▼	△		□	▼	▼	▼	△		▼	△		△	△	
Y	□	△			△						▼							
HH					△													
EL					△				○					△				
W	□				△		□		○			○		△				
EH	□		○	○	△	▼	□		○	▼	△	△	○	△	△	○	△	
AO	□	□	▼		△	□	△	○	▼	▼	○	△	▼	△	△	△		
AA	□	□	▼		△	□	□	○	▼	▼	○	○	○	△	△	△		
UW	□	△			△	▼	□	○	○	▼	▼			△		△		
ER	□				△	▼	□	○	○	▼	△		○	△		△		
AY	□	□	▼		△	△	△	○	▼	▼	△	△	▼	△	△	△		
EY	▼	▼	○		△	▼	△	○	○	▼	△	△	▼	△	△	△		
AW	□		▼		△		□	○	○	▼	△	△	○	△	△	△		
AX					△		□	○	○	▼	△	△	▼	△				
IH	▼		▼	▼	△	□	△	▼	▼	▼	△	△	▼	△		○	△	
AE	□	□			△				▼	▼	△	△		△				
AH	□				△	□			▼	▼	△	△		△	△	△		
OY	□	○	○		△		□			▼	□	△	○	△	△	△	△	
IY		△	△		△	▼	□	▼	○	▼	▼	△	○	△		○	▼	
OW	□				△	▼	□		○	▼	△	△	▼	△	△	△	△	
AXR	□				△			▼	▼	▼				△			△	

Table 61. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #7, level of significance: 0.01

KEY:																			
△		indicates feature was higher than normal for both loud and Lombard																	
▽		indicates feature was lower than normal for both loud and Lombard																	
○		indicates feature was higher for loud and lower for Lombard																	
□		indicates feature was lower for loud and higher for Lombard																	
Phonemes	Energy in Frequency Bands (kHz)										C0G	Tim		Pitch	Formants			Dur	
	0-0.5	0.5-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	Lo		Hi	1		2	3			
P	□	□	▽			○	○	△		□	△	○	□						
T	▽				▽	▽	▽	△	△	△	△	△	△				○		
K																			
B																			
D																			
G					△														
DX	▽											△		△					
M	▽	▽		△			○	○	○	○	○	○	△		△		▽	▽	
N	▽	□	□			▽	▽	△	○	△	○	○	△	△	△		□		
NX	▽	□	□	□			○	○	○	○	○	○			△				
S	□		▽	▽	▽	▽		△	△	△	△	△	▽	△	△				
Z					▽	▽			△	△	△	△			△				
CH				▽				△	△	△				△	△				
TH	□					▽	▽							△	△				
F		▽	▽				○			△				△	△				
SH				▽				△	△					△	△		▽		
JH				▽															
V	▽	▽			△	△						△			△				
L	▽														△				
R	▽	▽		△		○	▽	▽	▽	▽	○	△		△	△		△		▽
Y	▽	□										△		△	△				
HH						▽													
EL	▽					△					○	△		△	△				
W	▽	□					▽				○			△	△				
EH	▽		△	△	△	○	▽	▽		▽		△		△	△		△		△
AO	▽		△		△	○	▽	□		□		△		△	△		△		△
AA	▽	▽	○	△	△	△	▽	□	▽	▽		△		△	△		△		
UW	▽	□			△		▽	▽		▽		○		□	△		△		
ER	▽	□				○	▽	▽		▽		○			△		△		
AY	▽	▽		△	△	○	▽	▽		○			△		△		△		△
EY	▽	▽			△	○	▽	□				△		△	△		△		
AW	▽				△	○	▽	▽				△		△	△		△		
AX	▽				△	△	▽	▽	▽	▽		△		△	△		△		△
IH	▽	▽	△	△	△	○	▽	□		▽		△	△	▽	△		△		
AE	▽				△		▽	▽				△		△	△		△		
AH	▽			△	△	△	▽	▽	▽	▽				△	△		△		
OY	▽	△	△		△	△	▽	▽	▽	▽		△		△	△		△		
IY	▽	□									△	△		△	△			▽	
OW	▽	□			△	○	▽	▽	▽	▽	○	△		△	△		△		
AXR	▽		△	△	△	△	▽	▽	▽	▽		△		△	△		△		

Table 62. Significant differences in phoneme features for normal, loud, and Lombard speech for speaker #8, level of significance: 0.01

KEY:															
△ indicates feature was higher than normal for both loud and Lombard ▽ indicates feature was lower than normal for both loud and Lombard ○ indicates feature was higher for loud and lower for Lombard ■ indicates feature was lower for loud and higher for Lombard															
Phonemes	Energy in Frequency Bands (kHz)										COG	Tilt		Pitch	Formants
	0-25	25-50	50-100	100-200	200-400	400-800	800-1600	1600-3200	3200-6400	6400-12800		Lo	Hi		1 2 3
P					△			▽	▽	▽					
T				▽			▽	△		△	△				
K		△				▽				■					
B	△														
D				▽									△		
G															
DX	△	△												△	
M														△	▽
N	△	△	△			▽			▽	▽	▽			△	▽
NX	△	△									▽			△	▽
S	▽				△	▽		▽	△	△		△	△	△	▽
Z			△			▽		▽	△	△		▽	△	△	
CH	▽								△	△			△	?	△
TH				▽		▽	△		△	△	△				
F	▽					▽	△		△	△	△		△		■
SH	▽	▽	▽	▽			△		△	△	△		△	△	△
JH				▽			△		△	△				△	△
V	△	△				▽								△	▽
L					△	▽	△		▽	▽			▽	△	△
R		△												△	△
Y						▽								△	▽
HH						▽								△	
EL			▽						■	▽	△			△	△
W	△	△				▽								△	
EH		△		▽	△			○	△	▽	△			△	△
AO	▽		▽	▽	△		○			▽	△			△	△
AA	▽	▽	▽	▽	△		△			▽	△			△	△
UW					■		△			▽	△			△	△
ER	▽	■		▽	△		△	○		▽	△			△	△
AY	■	△	▽	▽	△		△	○	△	▽	△	▽		△	△
EY					■			○		▽	△		▽	△	△
AW			▽		△					▽	△			△	△
AX					△					▽	△			△	△
IH	▽	■		▽	△		▽	△	△	▽	△	△		△	△
AE			▽	▽	△		△	○		▽	△			△	△
AH		■			△					▽	△			△	△
OY	▽	■					▽	○	△	▽	△		▽	△	△
IY		△					▽	△		▽	△			△	△
OW	▽		▽		△		○			▽	△			△	△
AXR	△	△	△	○	△	▽				▽	△	▽	▽	△	△

Appendix M: Performance Curves for the Baseline System

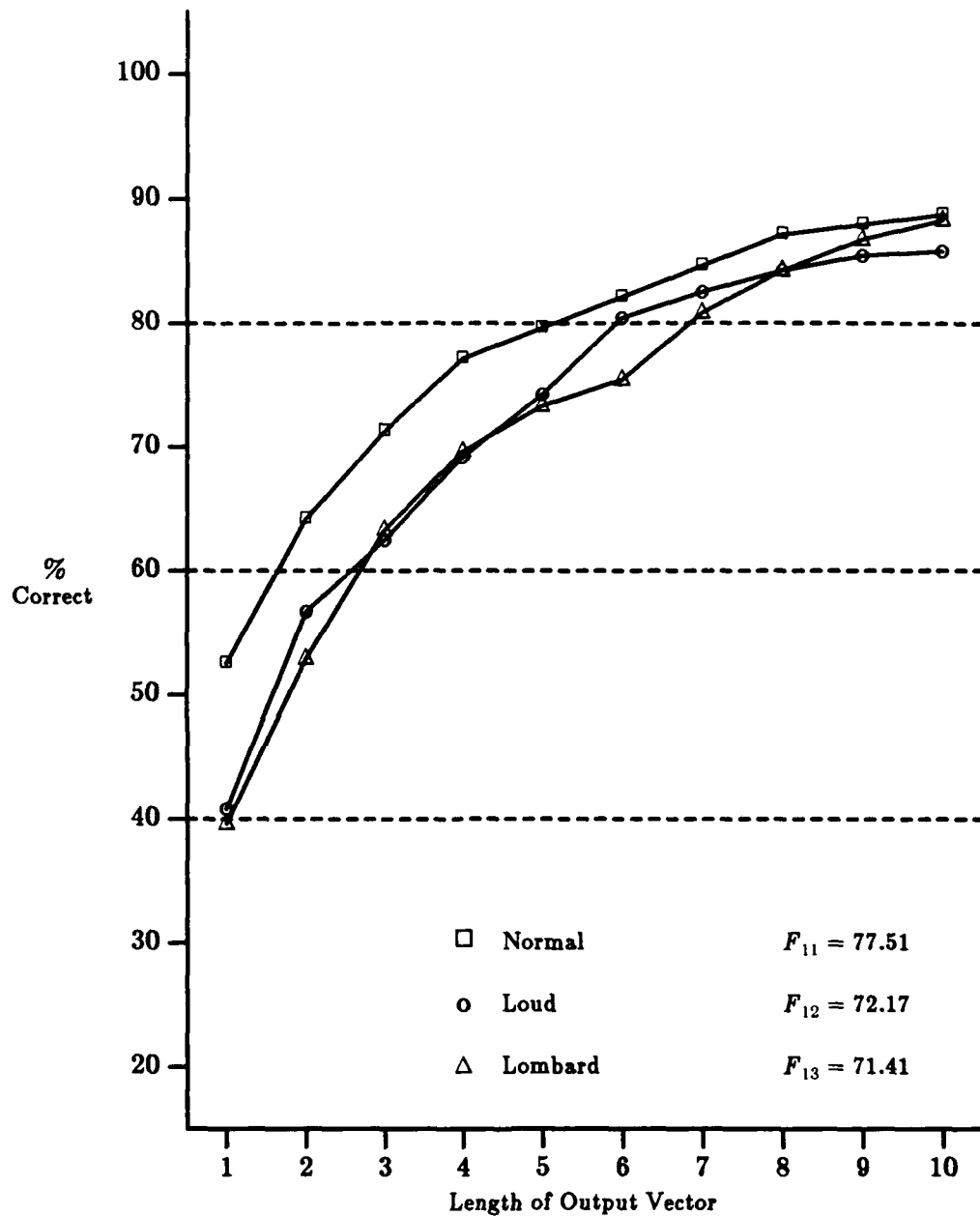


Figure 69. Recognition performance for baseline system, Speaker #1

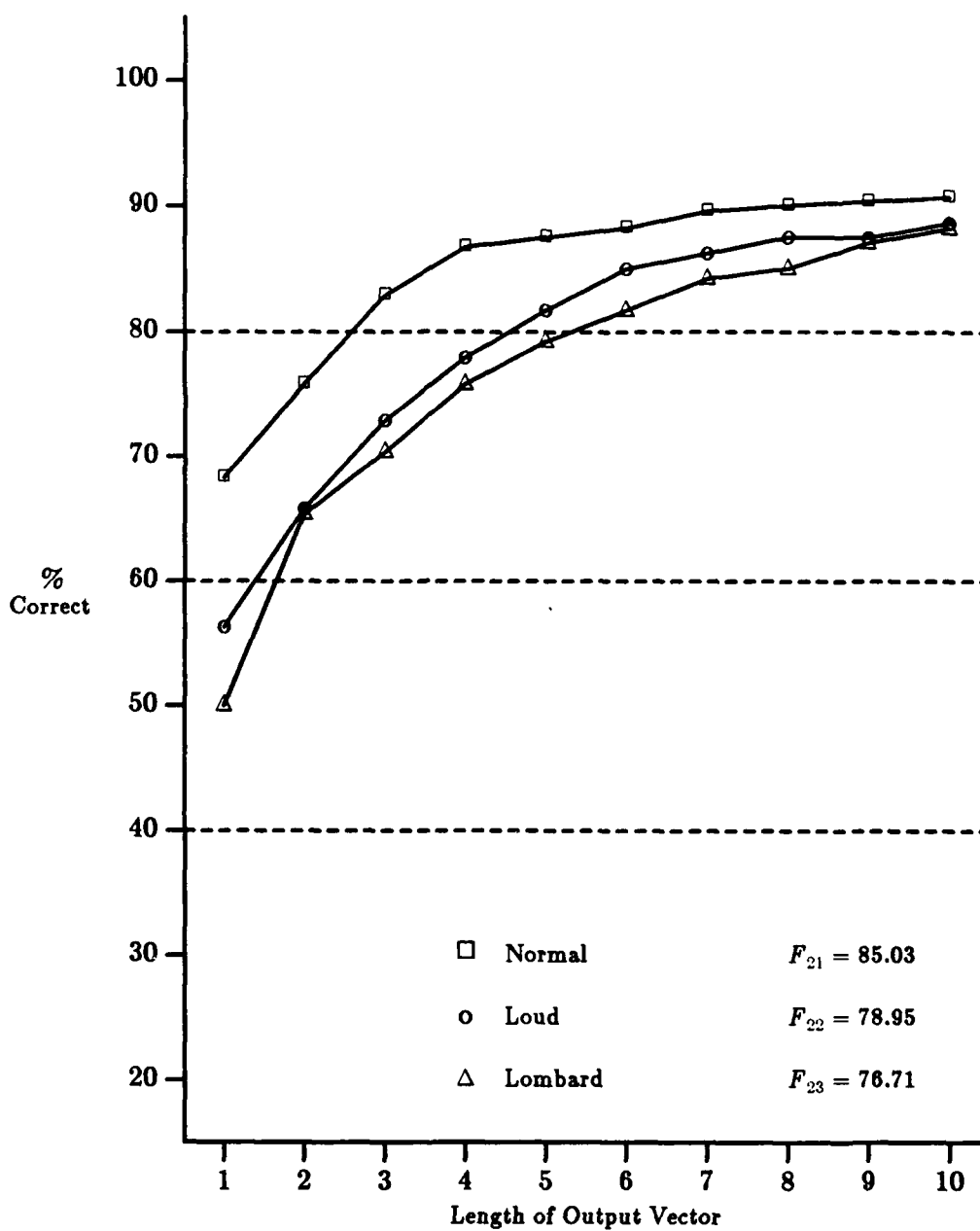


Figure 70. Recognition performance for baseline system, Speaker #2

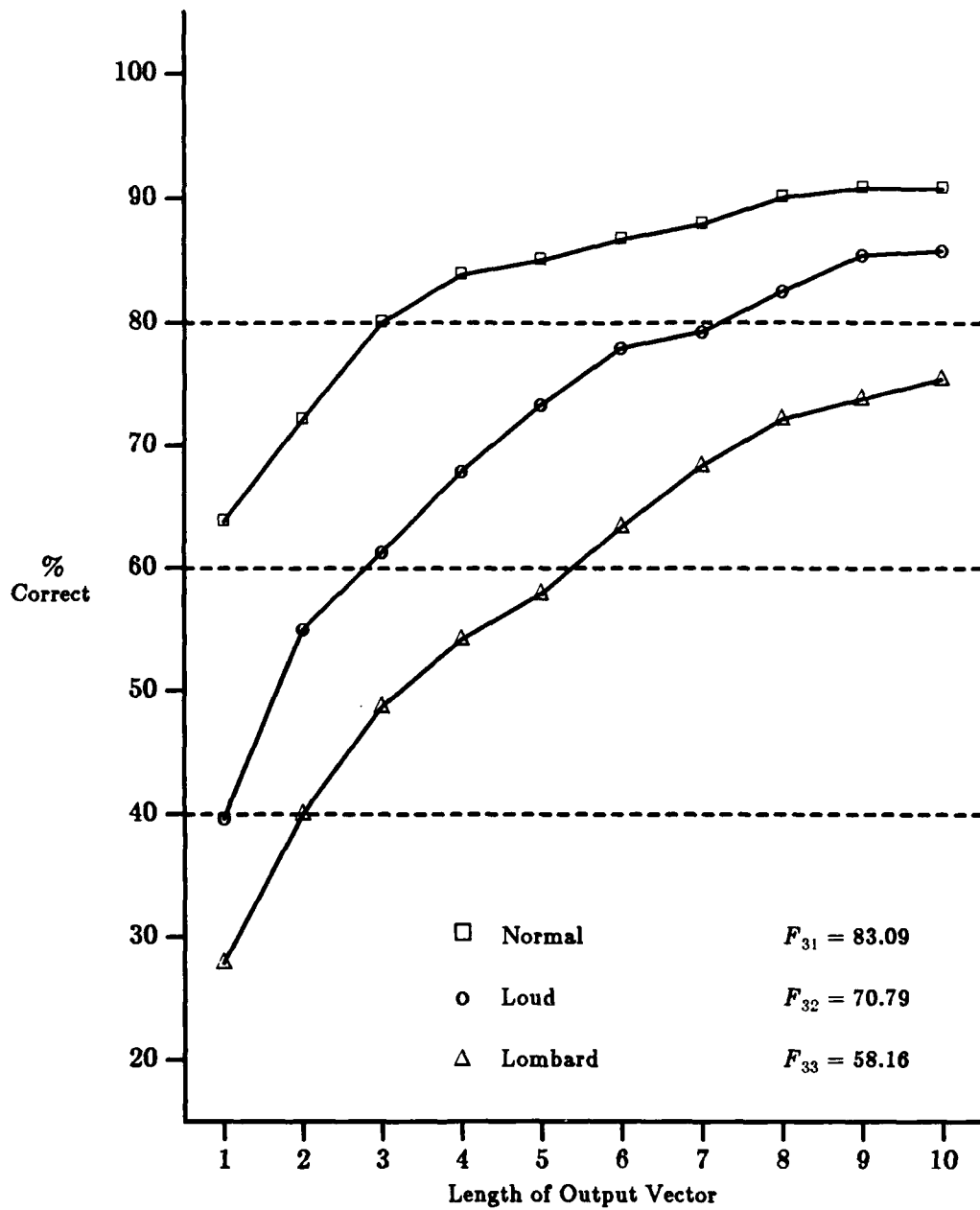


Figure 71. Recognition performance for baseline system, Speaker #3

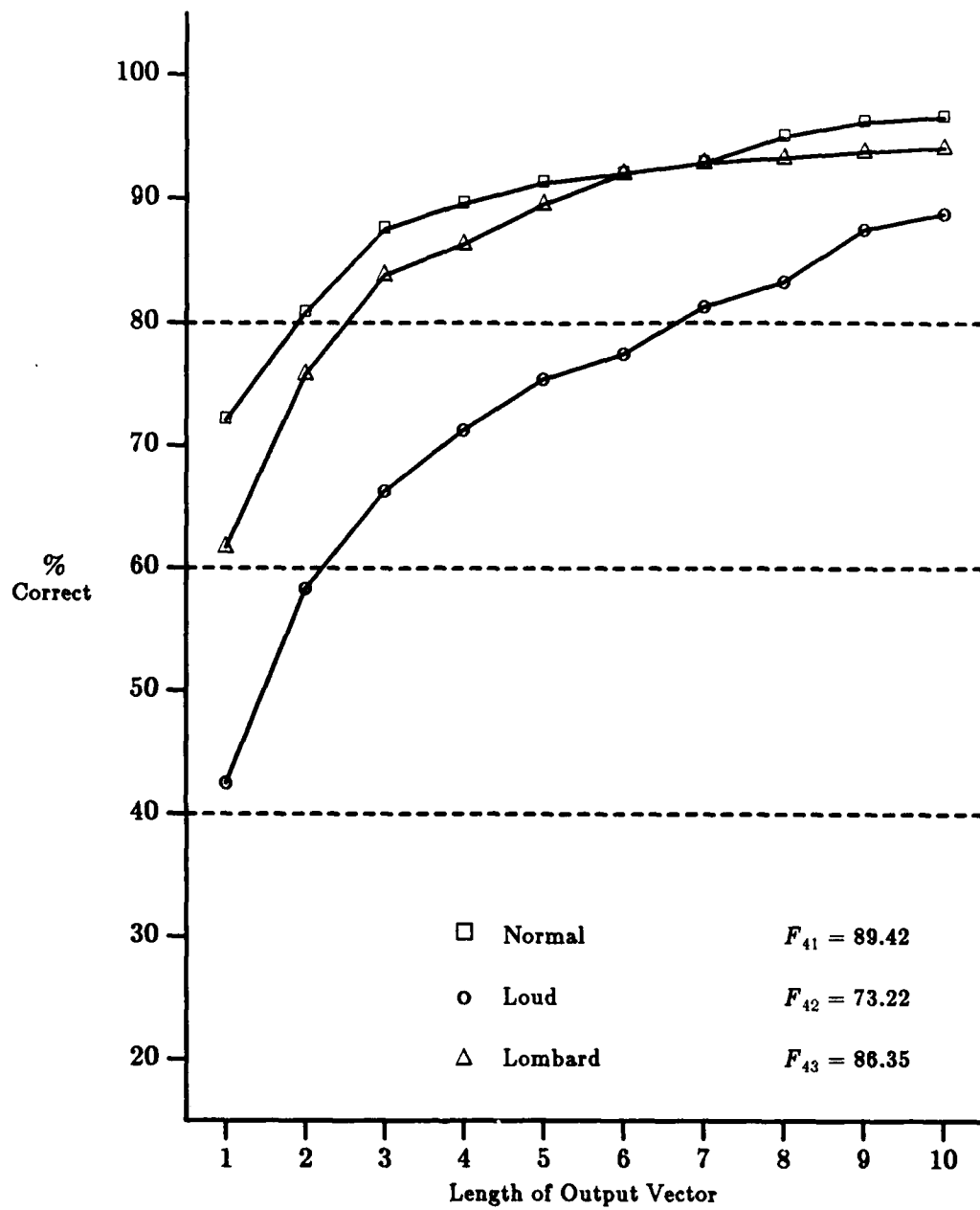


Figure 72. Recognition performance for baseline system, Speaker #4

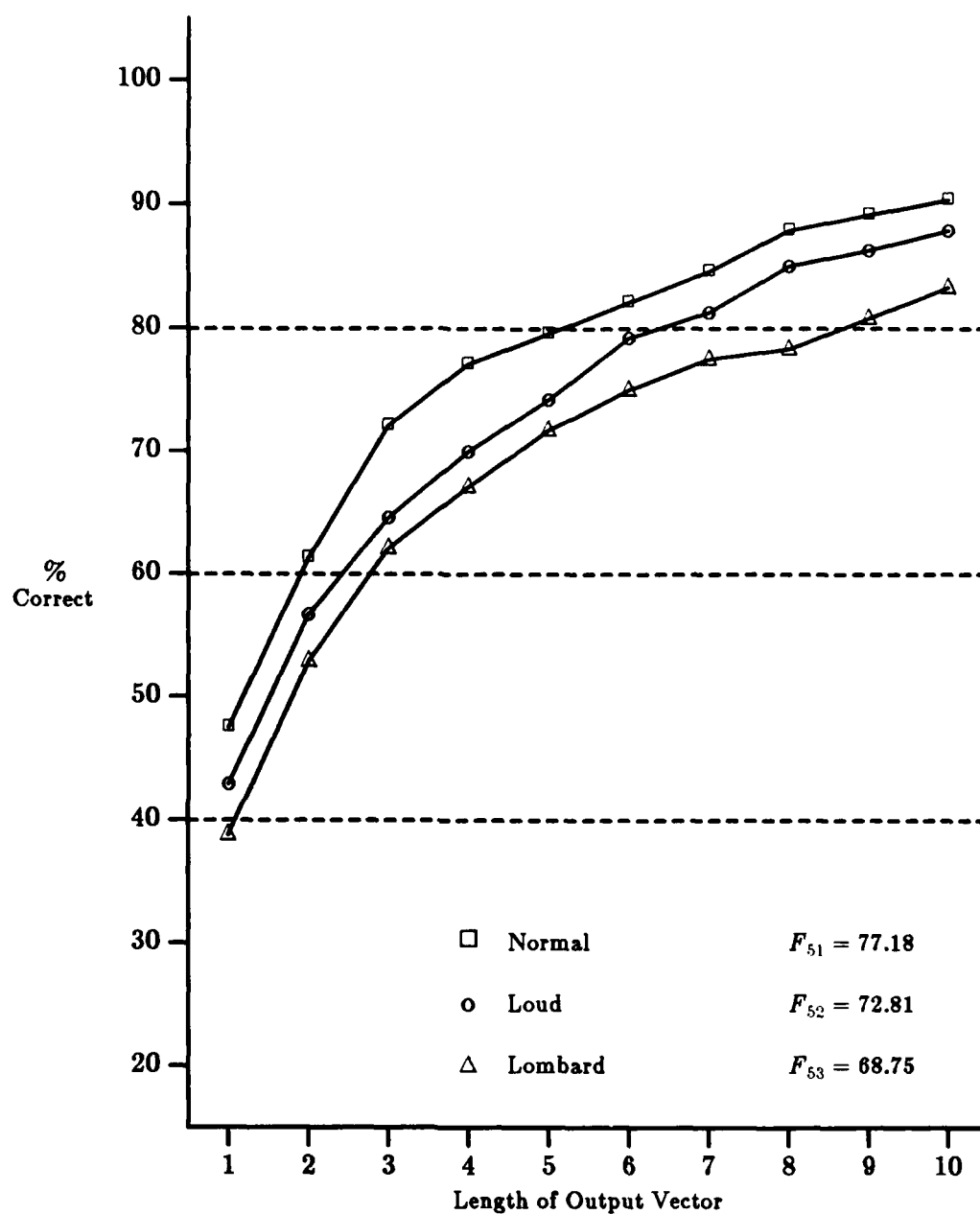


Figure 73. Recognition performance for baseline system, Speaker #5

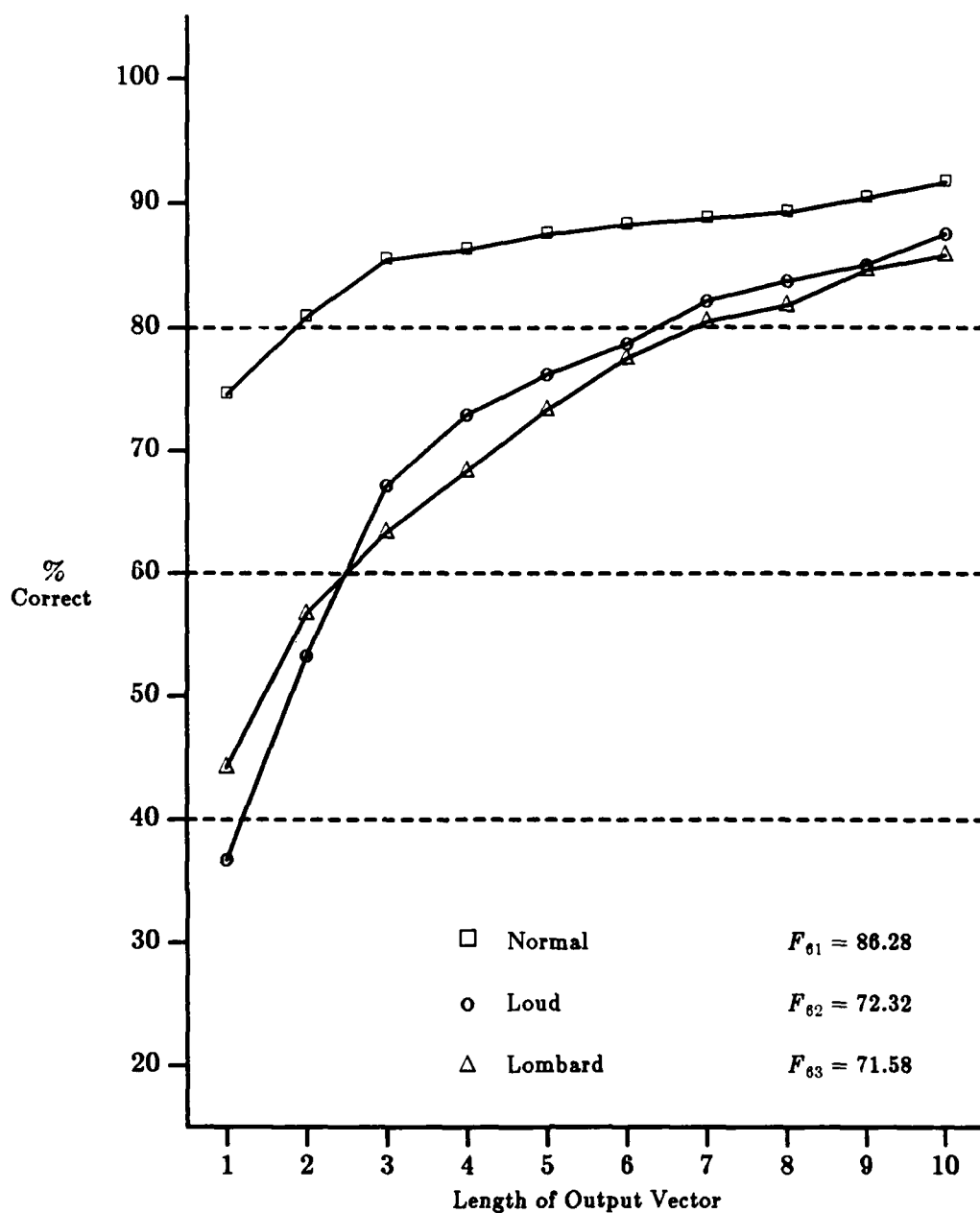


Figure 74. Recognition performance for baseline system, Speaker #6

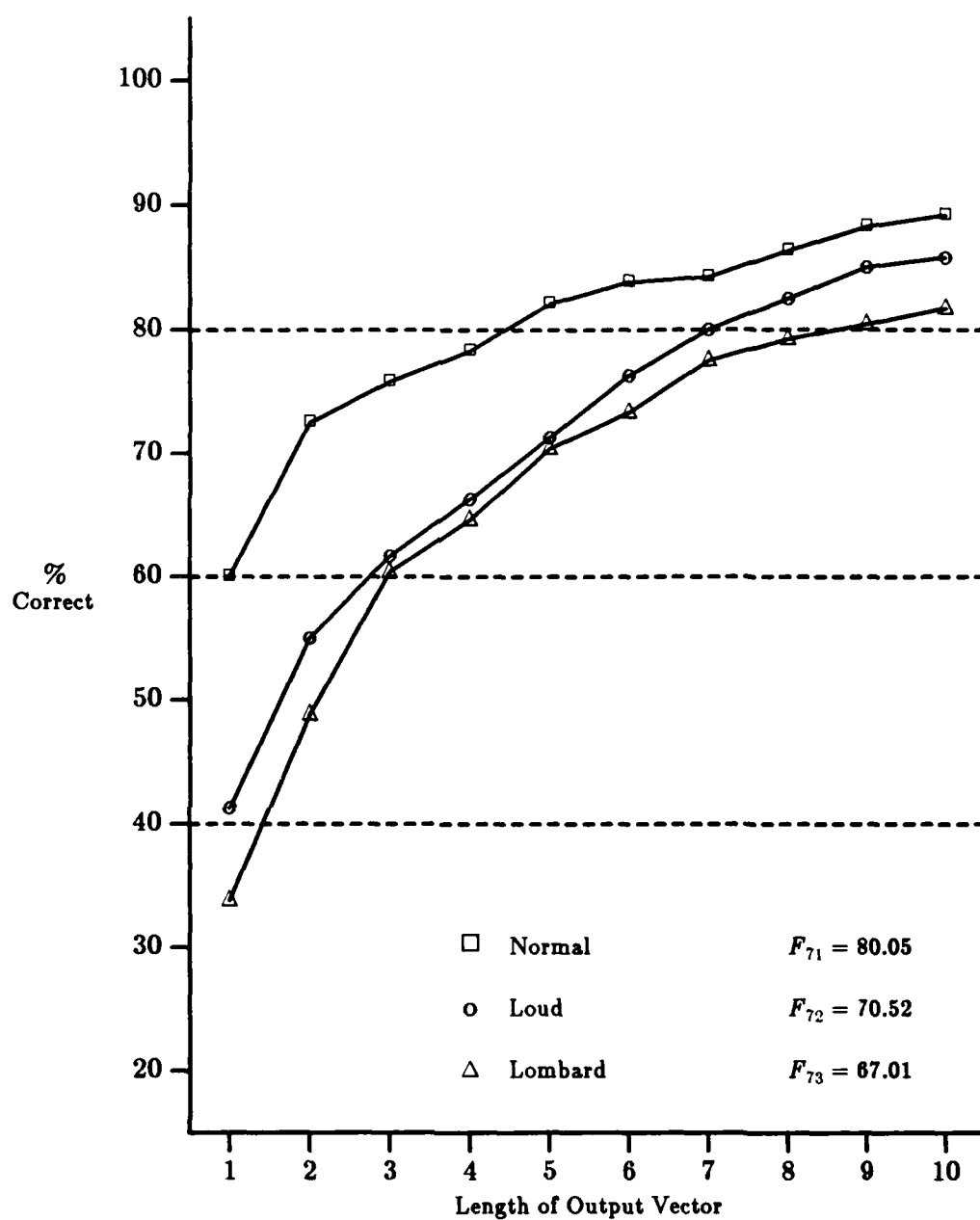


Figure 75. Recognition performance for baseline system, Speaker #7

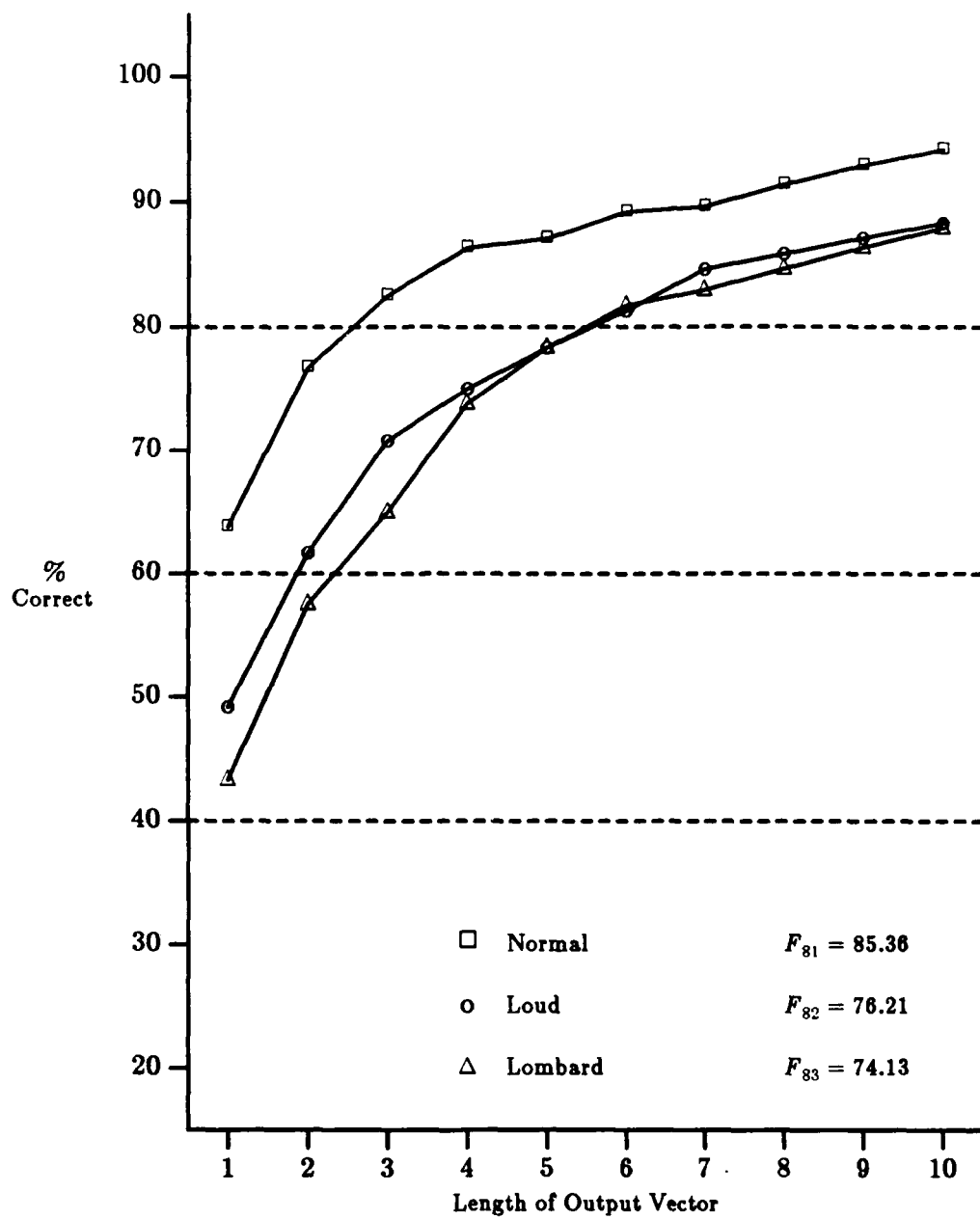


Figure 76. Recognition performance for baseline system, Speaker #8

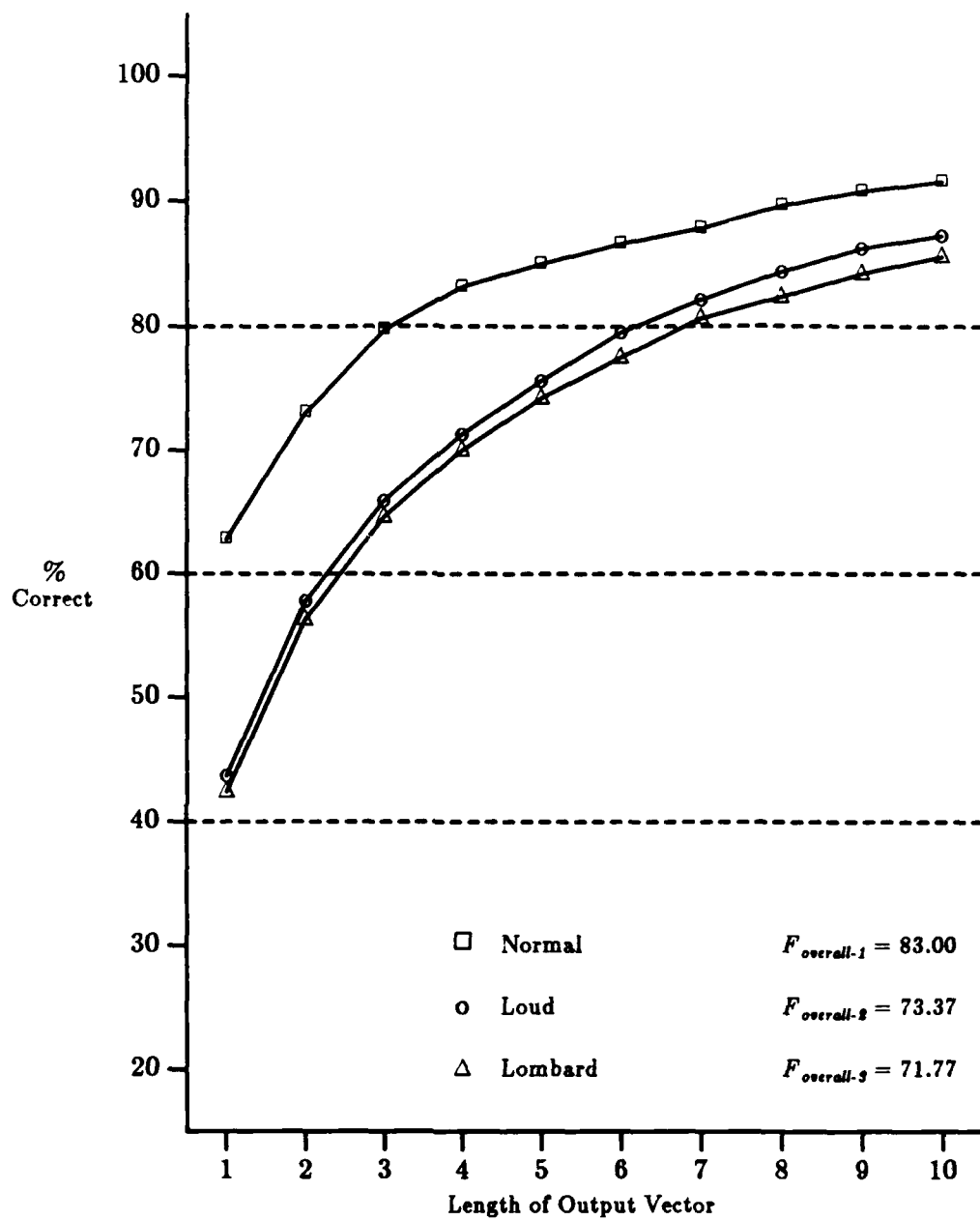


Figure 77. Recognition performance for baseline system, all phonemes, all speakers

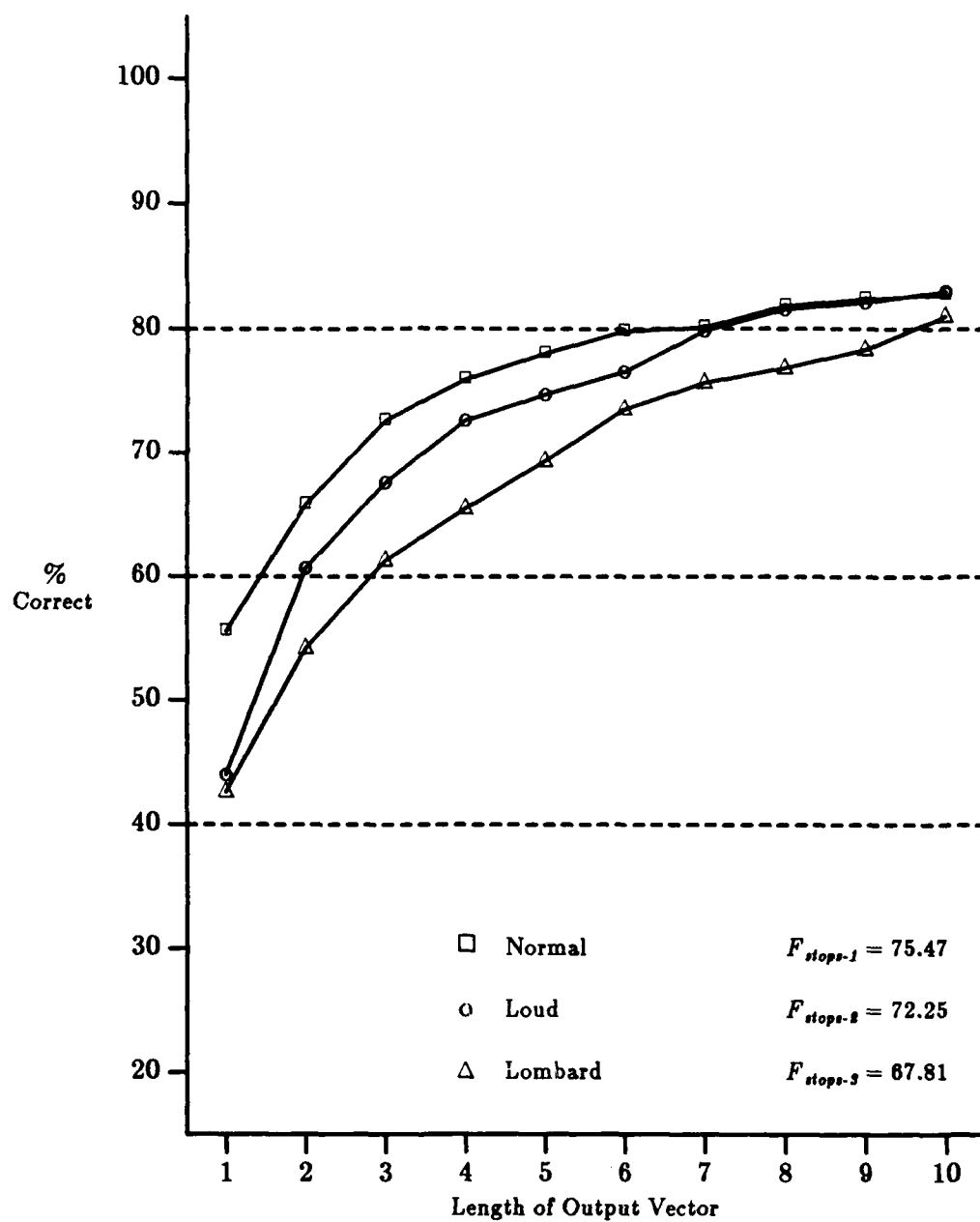


Figure 78. Recognition performance for baseline system, stops, all speakers

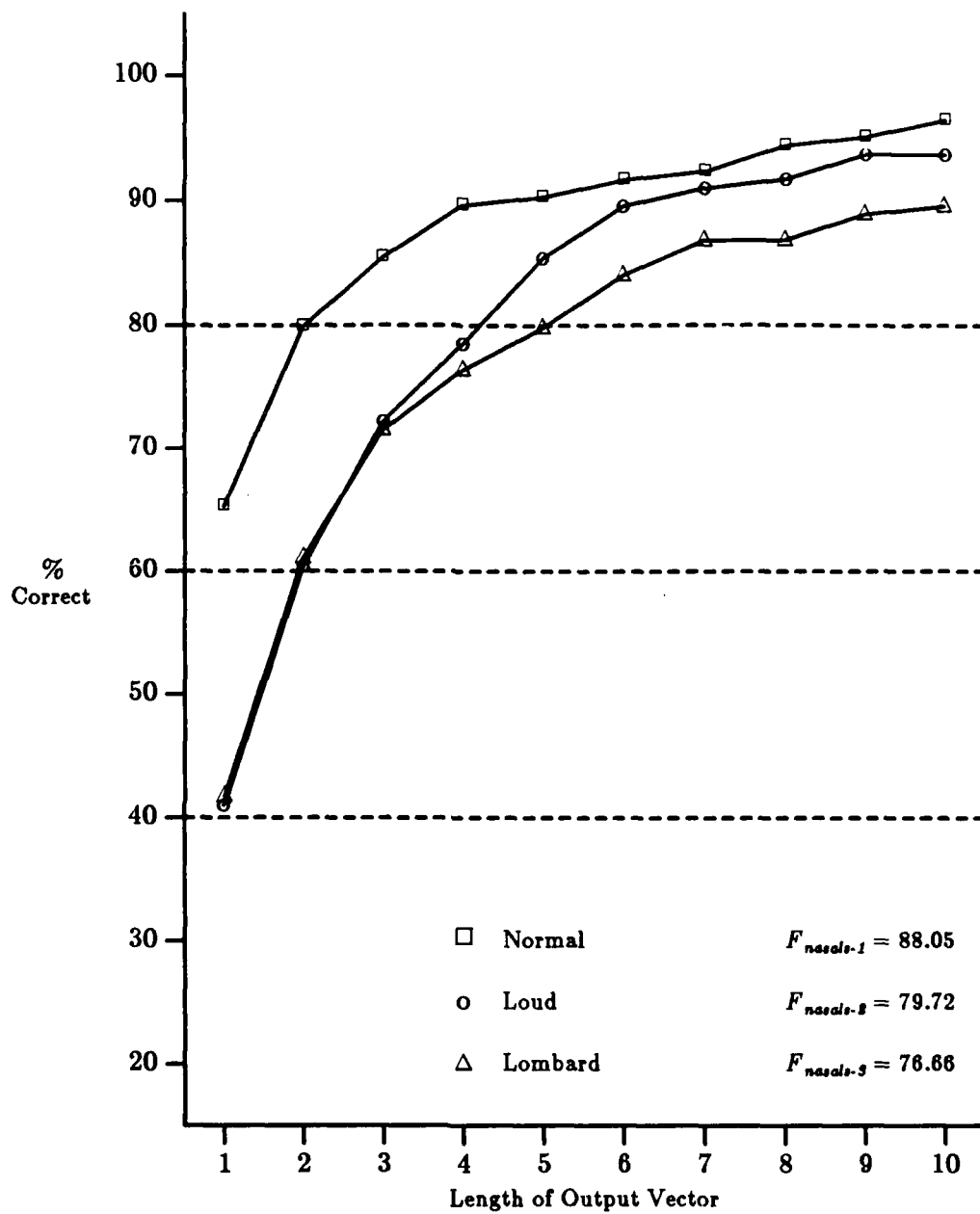


Figure 79. Recognition performance for baseline system, nasals, all speakers

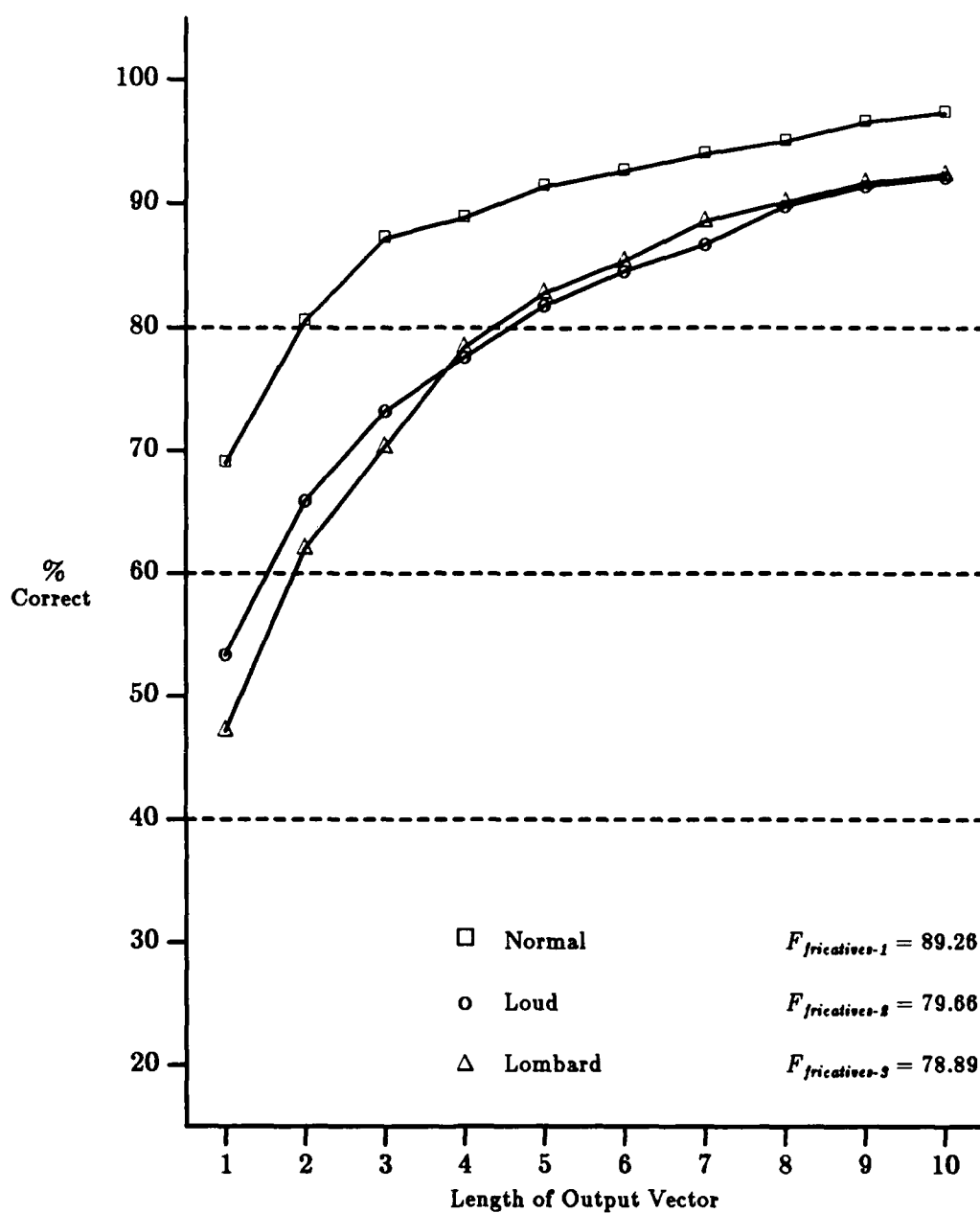


Figure 80. Recognition performance for baseline system, fricatives, all speakers

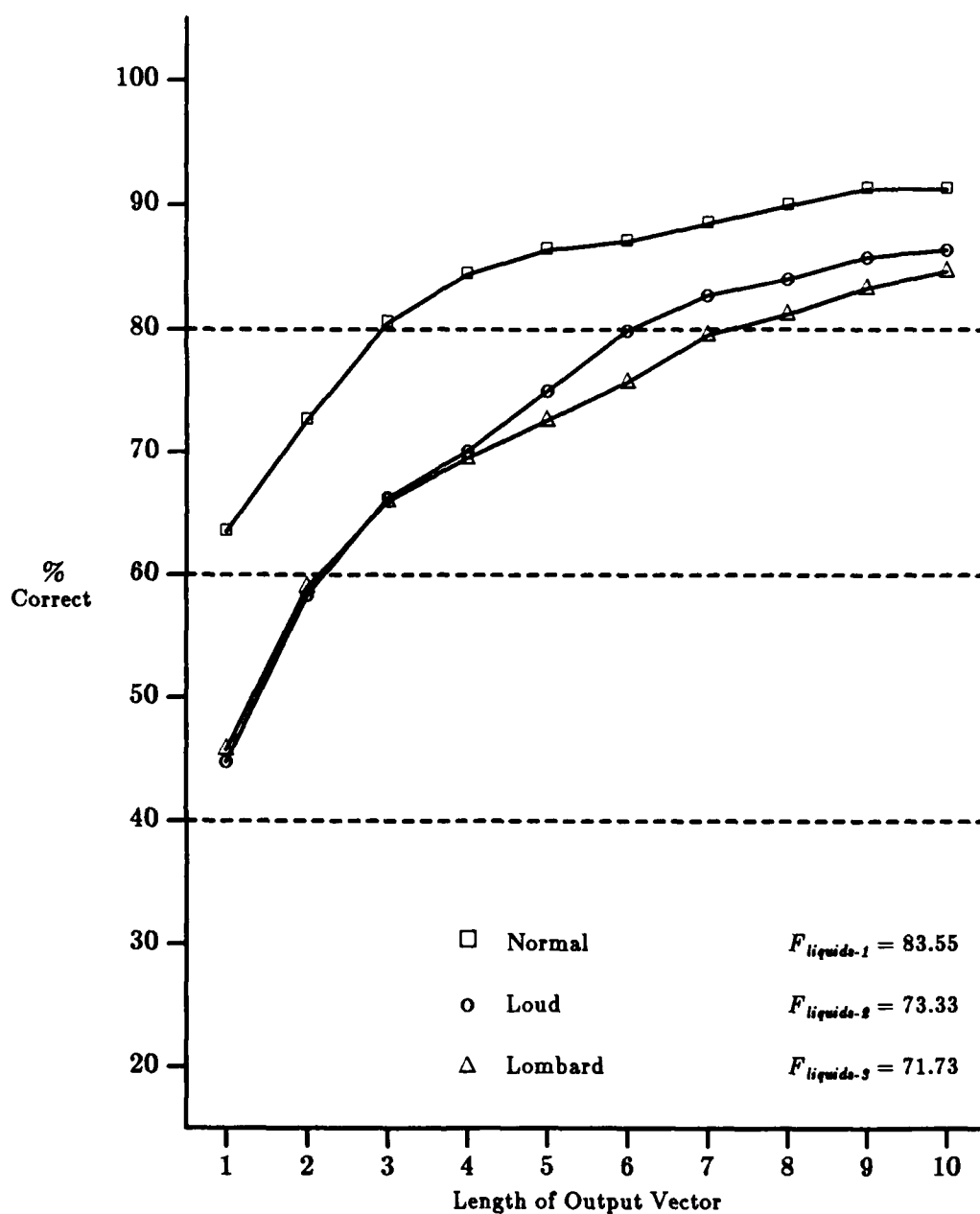


Figure 81. Recognition performance for baseline system, liquids, all speakers

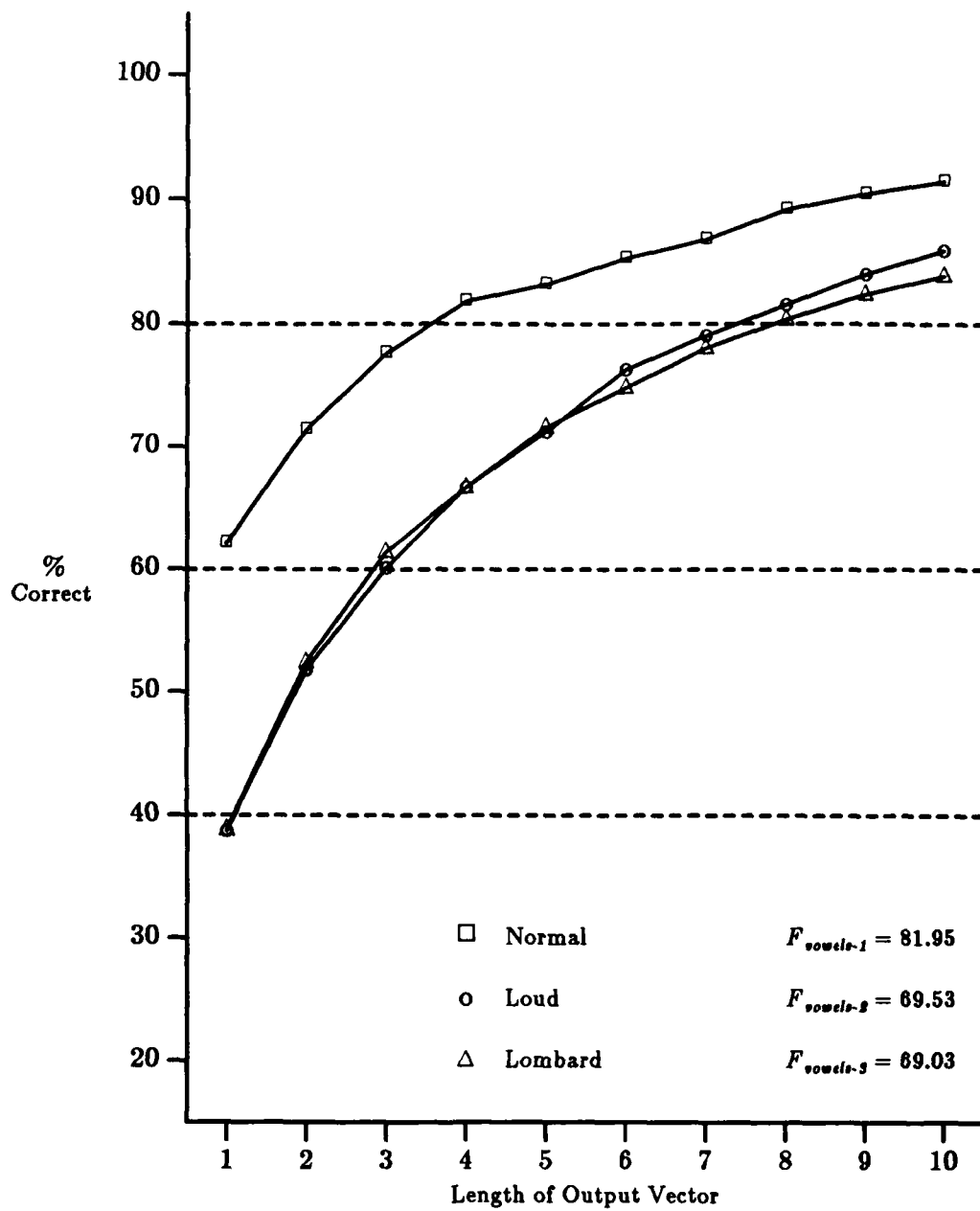


Figure 82. Recognition performance for baseline system, vowels, all speakers

Appendix N: Figure of Merit Comparisons

This appendix contains tables that compare the method of SCD (smallest cumulative distance) combined with the other metrics tested in this research to the baseline system described in Chapter 8.

Table 63. Performance of baseline system with smallest cumulative distance compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	77.44	-0.07
12	72.17	72.84	0.67
13	71.41	72.40	0.99
12-11	-5.34	-4.60	0.74
13-11	-6.10	-5.04	1.06
21	85.03	81.53	-3.50
22	78.95	76.45	-2.50
23	76.71	77.59	0.88
22-21	-6.08	-5.08	1.00
23-21	-8.32	-3.94	4.38
31	83.09	78.80	-4.29
32	70.79	72.75	1.96
33	58.16	65.97	7.81
32-31	-12.30	-6.05	6.25
33-31	-24.93	-12.83	12.10
41	89.42	86.28	-3.14
42	73.22	78.77	5.55
43	86.35	82.88	-3.47
42-41	-16.20	-7.51	8.69
43-41	-3.07	-3.40	-0.33
51	77.18	76.60	-0.58
52	72.81	72.57	-0.24
53	68.75	68.42	-0.33
52-51	-4.37	-4.03	0.34
53-51	-8.43	-8.18	0.25
61	86.28	86.05	-0.23
62	72.32	76.05	3.73
63	71.58	75.43	3.85
62-61	-13.96	-10.00	3.96
63-61	-14.70	-10.62	4.08
71	80.05	74.91	-5.14
72	70.52	75.03	4.51
73	67.01	72.17	5.16
72-71	-9.53	0.12	9.65
73-71	-13.04	-2.74	10.30
81	85.36	83.74	-1.62
82	76.21	78.05	1.84
83	74.13	78.82	4.49
82-81	-9.15	-5.69	3.46
83-81	-11.23	-5.12	6.11
Total Normal	82.99	80.67	-2.32
Total Loud	73.37	75.31	1.94
Total Lombard	71.76	74.18	2.42
Overall	76.04	76.72	0.68

Table 64. Performance of the cepstral measure with smallest cumulative distance compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	75.70	-1.81
12	72.17	69.98	-2.19
13	71.41	70.02	-1.39
12-11	-5.34	-5.72	-0.38
13-11	-6.10	-5.68	0.42
21	85.03	81.30	-3.73
22	78.95	75.96	-2.99
23	76.71	76.76	0.05
22-21	-6.08	-5.34	0.74
23-21	-8.32	-4.54	3.78
31	83.09	78.75	-4.34
32	70.79	71.87	1.08
33	58.16	65.54	7.38
32-31	-12.30	-6.88	5.42
33-31	-24.93	-13.21	11.72
41	89.42	80.33	-9.09
42	73.22	73.96	0.74
43	86.35	78.55	-7.80
42-41	-16.20	-8.37	9.83
43-41	-3.07	-1.78	1.29
51	77.18	70.68	-6.50
52	72.81	65.95	-6.86
53	68.75	63.13	-5.62
52-51	-4.37	-4.73	-0.36
53-51	-8.43	-7.55	0.88
61	86.28	83.92	-2.36
62	72.32	73.75	1.43
63	71.58	73.42	1.84
62-61	-13.96	-10.17	3.79
63-61	-14.70	-10.50	4.20
71	80.05	73.80	-6.25
72	70.52	74.12	3.60
73	67.01	70.43	3.42
72-71	-9.53	0.32	9.85
73-71	-13.04	-3.37	9.67
81	85.36	81.03	-4.33
82	76.21	74.93	-1.28
83	74.13	75.75	1.62
82-81	-9.15	-6.10	3.05
83-81	-11.23	-5.28	5.95
Total Normal	82.99	78.19	-4.80
Total Loud	73.37	72.57	-0.81
Total Lombard	71.76	71.70	-0.06
Overall	76.04	74.15	-1.89

Table 65. Performance of the likelihood ratio with smallest cumulative distance compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	76.84	-0.67
12	72.17	72.24	0.07
13	71.41	73.04	1.63
12-11	-5.34	-4.60	0.74
13-11	-6.10	-3.80	2.30
21	85.03	82.08	-2.95
22	78.95	77.55	-1.40
23	76.71	77.92	1.21
22-21	-6.08	-4.53	1.55
23-21	-8.32	-4.16	4.16
31	83.09	80.23	-2.86
32	70.79	75.34	4.55
33	58.16	69.12	10.96
32-31	-12.30	-4.89	7.41
33-31	-24.93	-11.11	13.82
41	89.42	85.43	-3.99
42	73.22	79.08	5.86
43	86.35	83.36	-2.99
42-41	-16.20	-6.35	9.85
43-41	-3.07	-2.07	1.00
51	77.18	77.41	0.23
52	72.81	74.38	1.57
53	68.75	70.16	1.41
52-51	-4.37	-3.03	1.34
53-51	-8.43	-7.25	1.18
61	86.28	85.04	-1.24
62	72.32	73.88	1.56
63	71.58	71.40	-0.18
62-61	-13.96	-11.16	2.80
63-61	-14.70	-13.64	1.06
71	80.05	74.29	-5.76
72	70.52	74.68	4.16
73	67.01	72.83	5.82
72-71	-9.53	0.39	9.92
73-71	-13.04	-1.46	11.58
81	85.36	82.23	-3.13
82	76.21	75.75	-0.46
83	74.13	76.33	2.20
82-81	-9.15	-6.48	2.67
83-81	-11.23	-5.90	5.33
Total Normal	82.99	80.44	-2.55
Total Loud	73.37	75.36	1.99
Total Lombard	71.76	74.27	2.51
Overall	76.04	76.69	0.65

Table 66. Performance of the spectral slope estimate with smallest cumulative distance compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	78.47	0.96
12	72.17	74.62	2.45
13	71.41	71.00	-0.41
12-11	-5.34	-3.85	1.49
13-11	-6.10	-7.47	-1.37
21	85.03	79.33	-5.70
22	78.95	74.68	-4.27
23	76.71	58.88	-17.83
22-21	-6.08	-4.65	1.43
23-21	-8.32	-20.45	-12.13
31	83.09	72.99	-10.10
32	70.79	65.32	-5.47
33	58.16	62.00	3.84
32-31	-12.30	-7.67	4.63
33-31	-24.93	-10.99	13.94
41	89.42	78.89	-10.53
42	73.22	62.17	-11.05
43	86.35	74.55	-11.80
42-41	-16.20	-16.72	-0.52
43-41	-3.07	-4.34	-1.27
51	77.18	74.66	-2.52
52	72.81	65.99	-6.82
53	68.75	53.35	-15.40
52-51	-4.37	-8.67	-4.30
53-51	-8.43	-21.31	-12.88
61	86.28	79.91	-6.37
62	72.32	58.51	-13.81
63	71.58	51.84	-19.74
62-61	-13.96	-21.40	-7.44
63-61	-14.70	-28.07	-13.37
71	80.05	71.75	-8.30
72	70.52	63.80	-6.72
73	67.01	56.92	-10.09
72-71	-9.53	-7.95	1.58
73-71	-13.04	-14.83	-1.79
81	85.36	78.33	-7.03
82	76.21	54.45	-21.76
83	74.13	45.92	-28.21
82-81	-9.15	-23.88	-14.73
83-81	-11.23	-32.41	-21.18
Total Normal	82.99	76.79	-6.20
Total Loud	73.37	64.94	-8.43
Total Lombard	71.76	59.31	-12.45
Overall	76.04	67.01	-9.03

Table 67. Performance of root power sums with smallest cumulative distance compared to baseline system

Key of table entries for speaker i			
Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
i1	F_{i1}	F'_{i1}	$F'_{i1} - F_{i1}$
i2	F_{i2}	F'_{i2}	$F'_{i2} - F_{i2}$
i3	F_{i3}	F'_{i3}	$F'_{i3} - F_{i3}$
i2-i1	$F_{i2} - F_{i1}$	$F'_{i2} - F'_{i1}$	$F'_{i2} - F'_{i1} - (F_{i2} - F_{i1})$
i3-i1	$F_{i3} - F_{i1}$	$F'_{i3} - F'_{i1}$	$F'_{i3} - F'_{i1} - (F_{i3} - F_{i1})$

Session (speaker, condition)	Baseline F	Test F	Difference, F_{Δ}
11	77.51	78.28	0.77
12	72.17	72.62	0.45
13	71.41	73.34	1.93
12-11	-5.34	-5.66	-0.32
13-11	-6.10	-4.94	1.16
21	85.03	83.09	-1.94
22	78.95	82.80	3.85
23	76.71	71.49	-5.22
22-21	-6.08	-0.29	5.79
23-21	-8.32	-11.60	-3.28
31	83.09	76.96	-6.13
32	70.79	70.46	-0.33
33	58.16	64.19	6.03
32-31	-12.30	-6.50	5.80
33-31	-24.93	-12.77	12.16
41	89.42	76.51	-12.91
42	73.22	66.97	-6.25
43	86.35	73.82	-12.53
42-41	-16.20	-9.54	6.66
43-41	-3.07	-2.69	0.38
51	77.18	69.87	-7.31
52	72.81	64.79	-8.02
53	68.75	57.63	-11.12
52-51	-4.37	-5.08	-0.71
53-51	-8.43	-12.24	-3.81
61	86.28	81.80	-4.48
62	72.32	63.46	-8.86
63	71.58	57.61	-13.97
62-61	-13.96	-18.34	-4.38
63-61	-14.70	-24.19	-9.49
71	80.05	74.89	-5.16
72	70.52	70.79	0.27
73	67.01	66.64	-0.37
72-71	-9.53	-4.10	5.43
73-71	-13.04	-8.25	4.79
81	85.36	78.87	-6.49
82	76.21	66.94	-9.27
83	74.13	61.41	-12.72
82-81	-9.15	-11.93	-2.78
83-81	-11.23	-17.46	-6.23
Total Normal	82.99	77.53	-5.46
Total Loud	73.37	69.85	-3.52
Total Lombard	71.76	65.77	-6.00
Overall	76.04	71.05	-4.99

Appendix O: Performance Curves for SDW-SCD

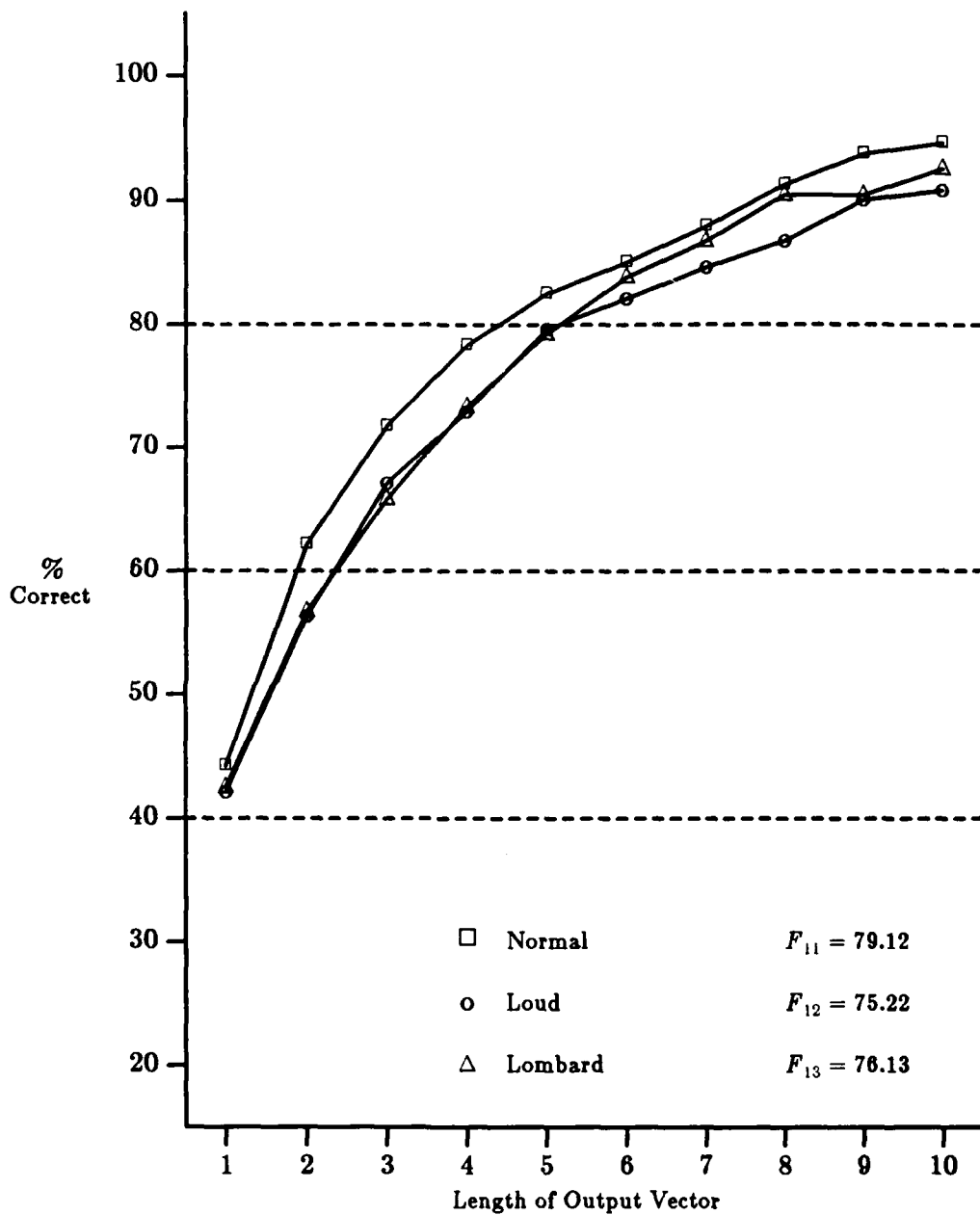


Figure 83. Recognition performance for SDW-SCD, Speaker #1

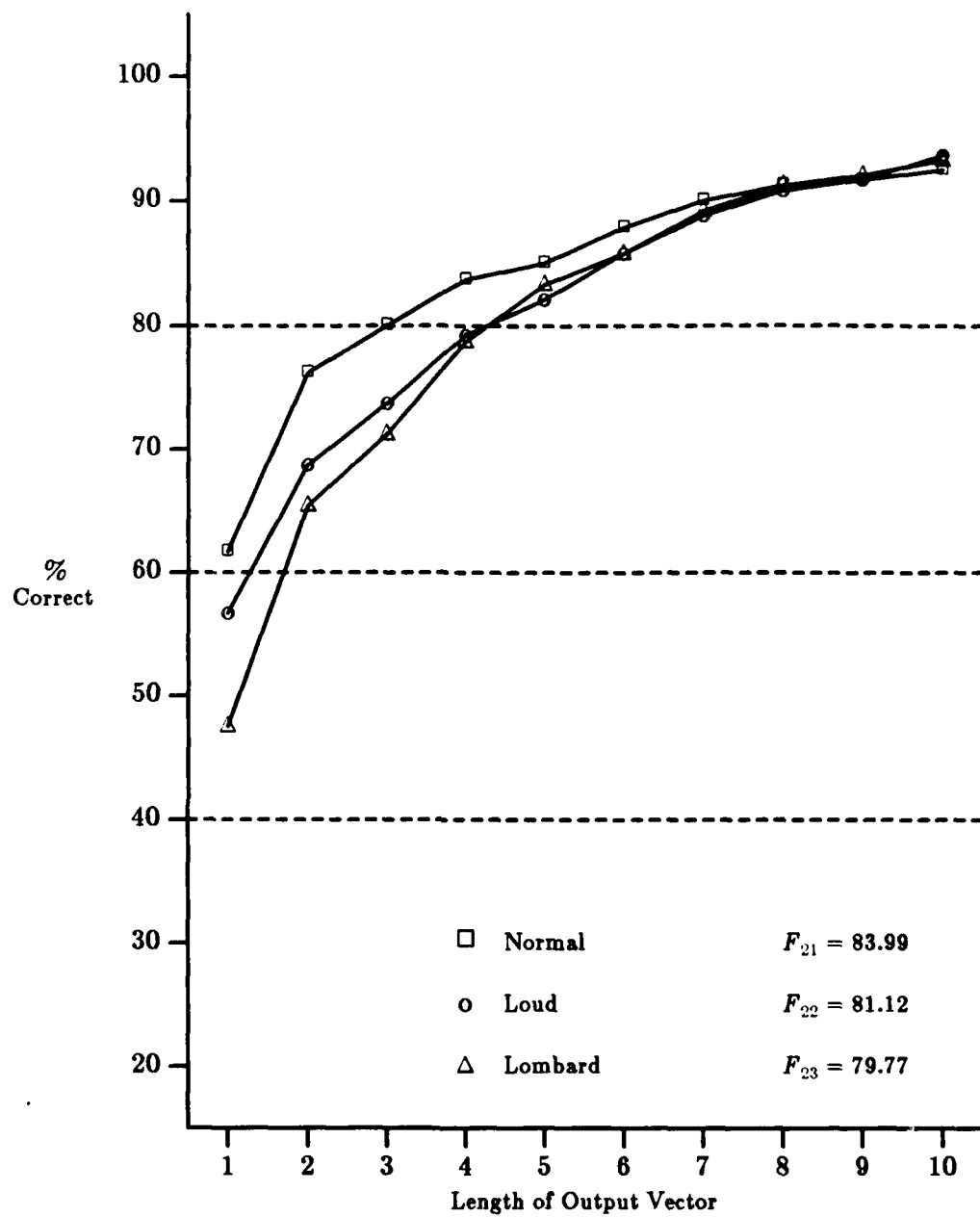


Figure 84. Recognition performance for SDW-SCD, Speaker #2

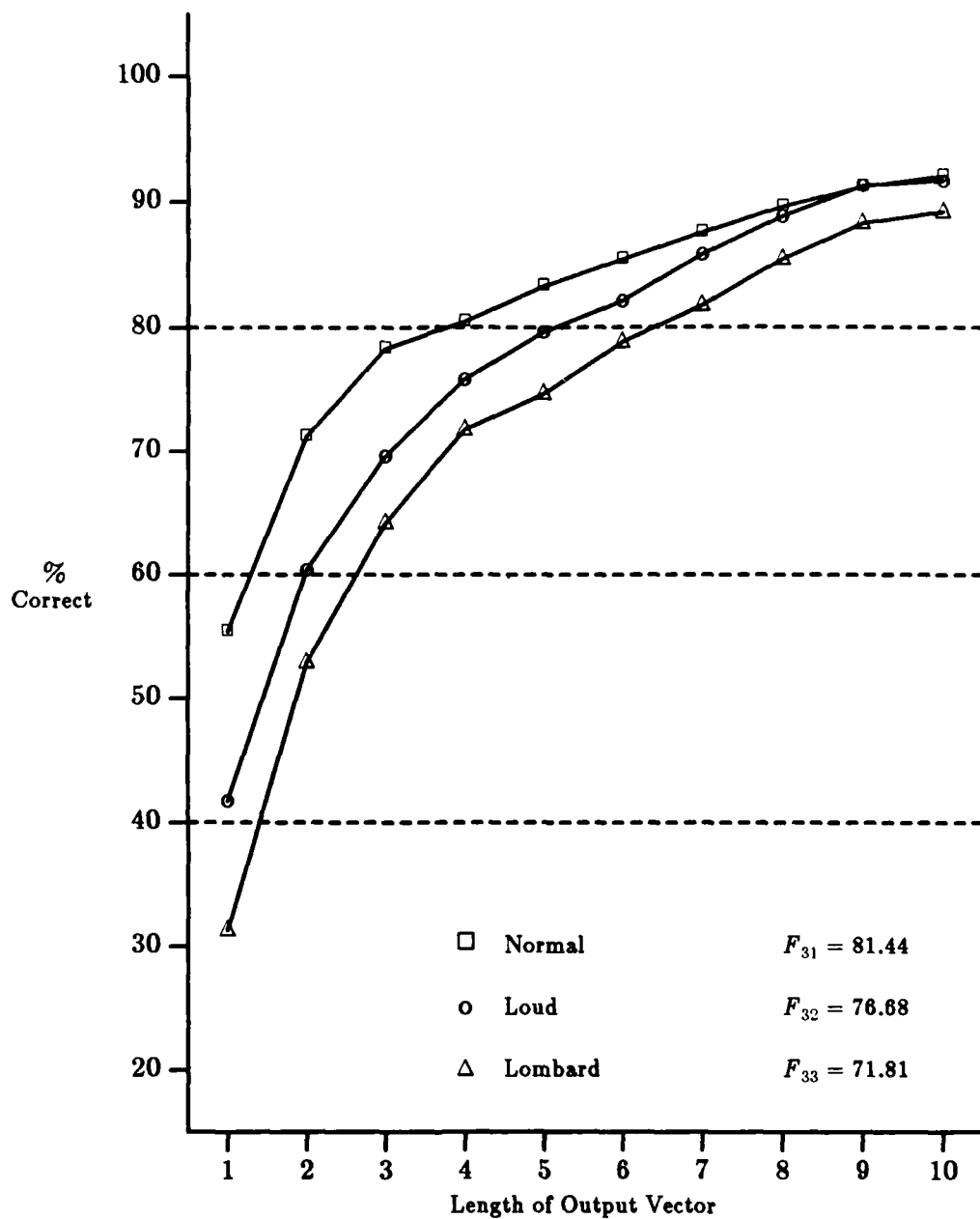


Figure 85. Recognition performance for SDW-SCD, Speaker #3

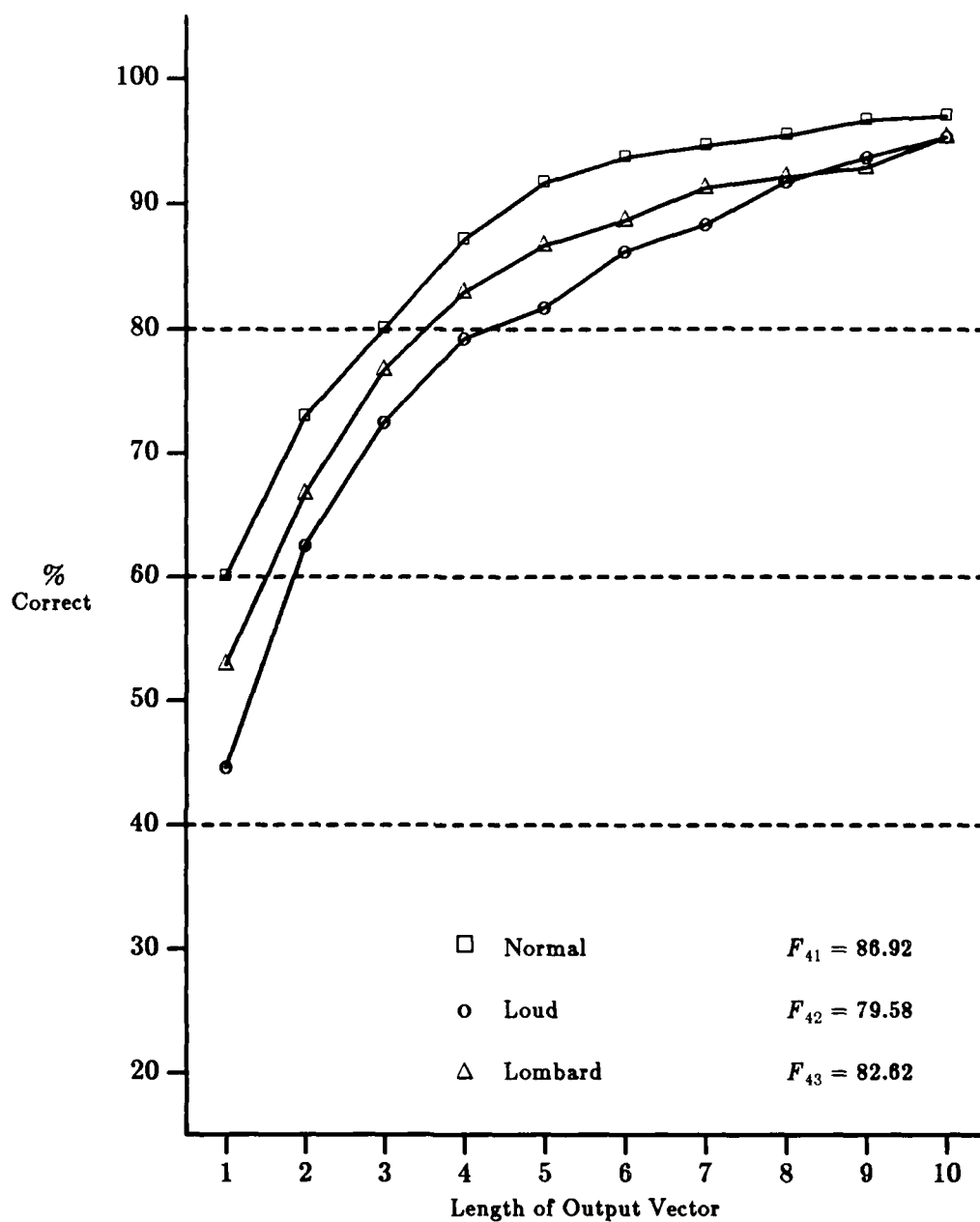


Figure 86. Recognition performance for SDW-SCD, Speaker #4

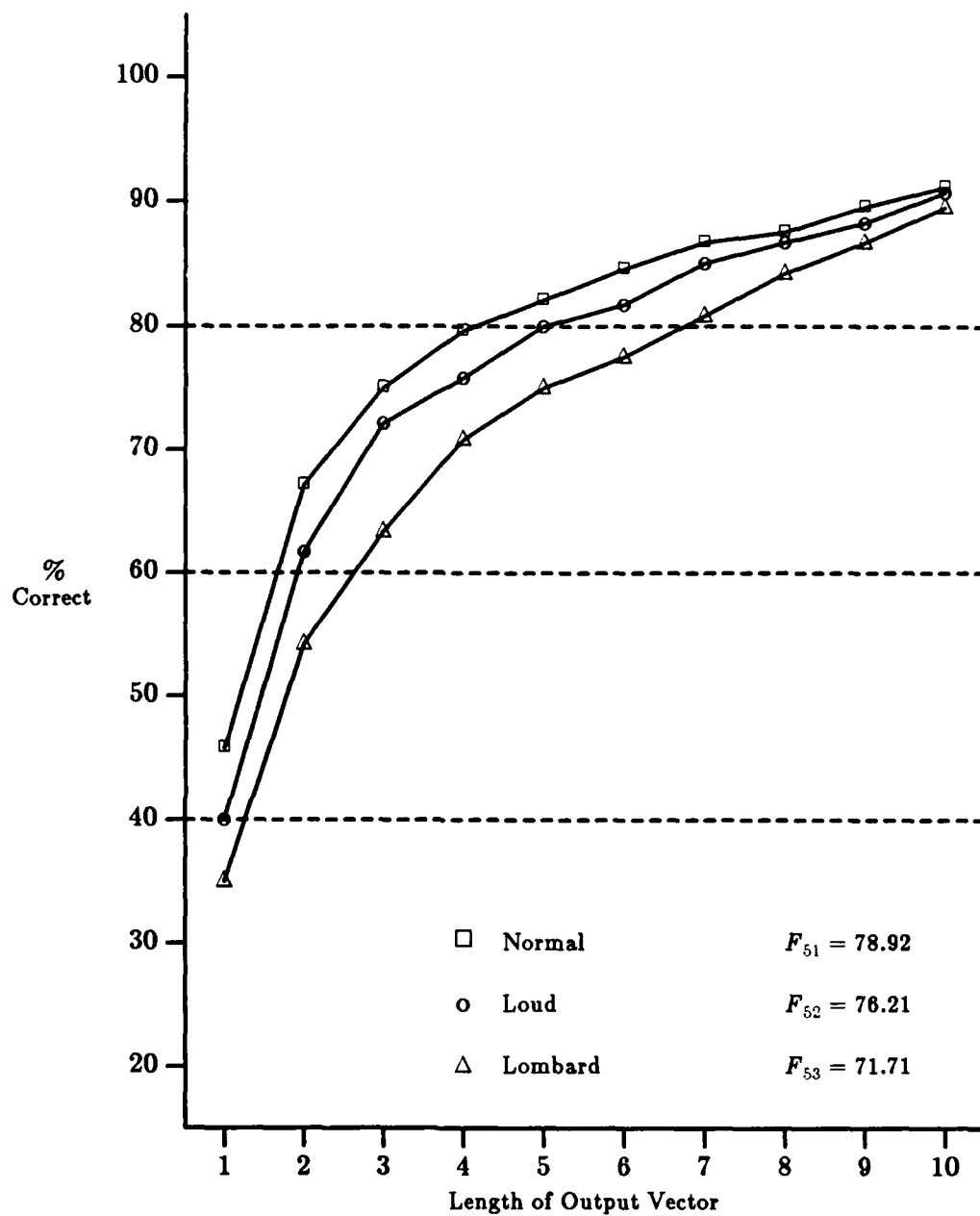


Figure 87. Recognition performance for SDW-SCD, Speaker #5

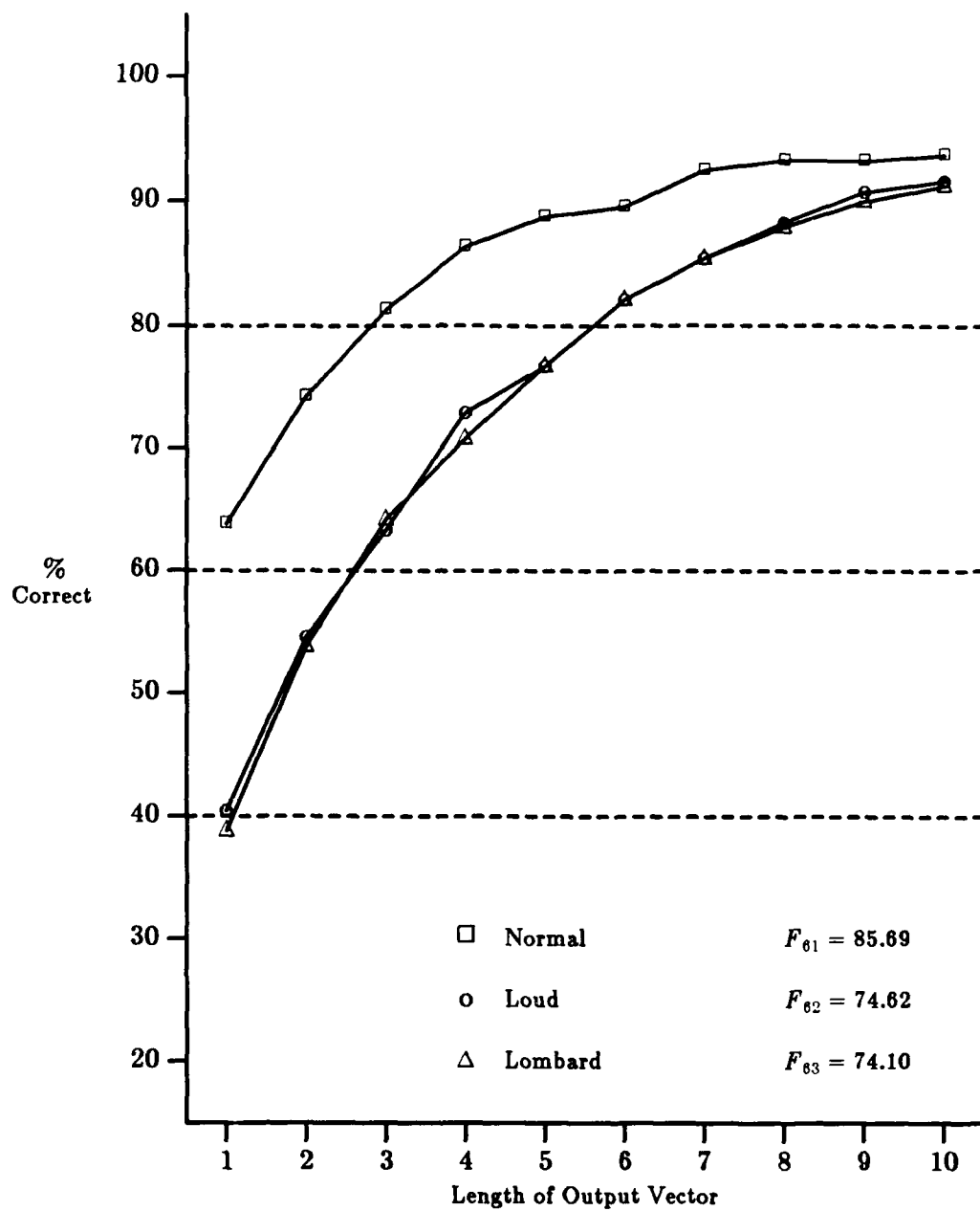


Figure 88. Recognition performance for SDW-SCD, Speaker #6

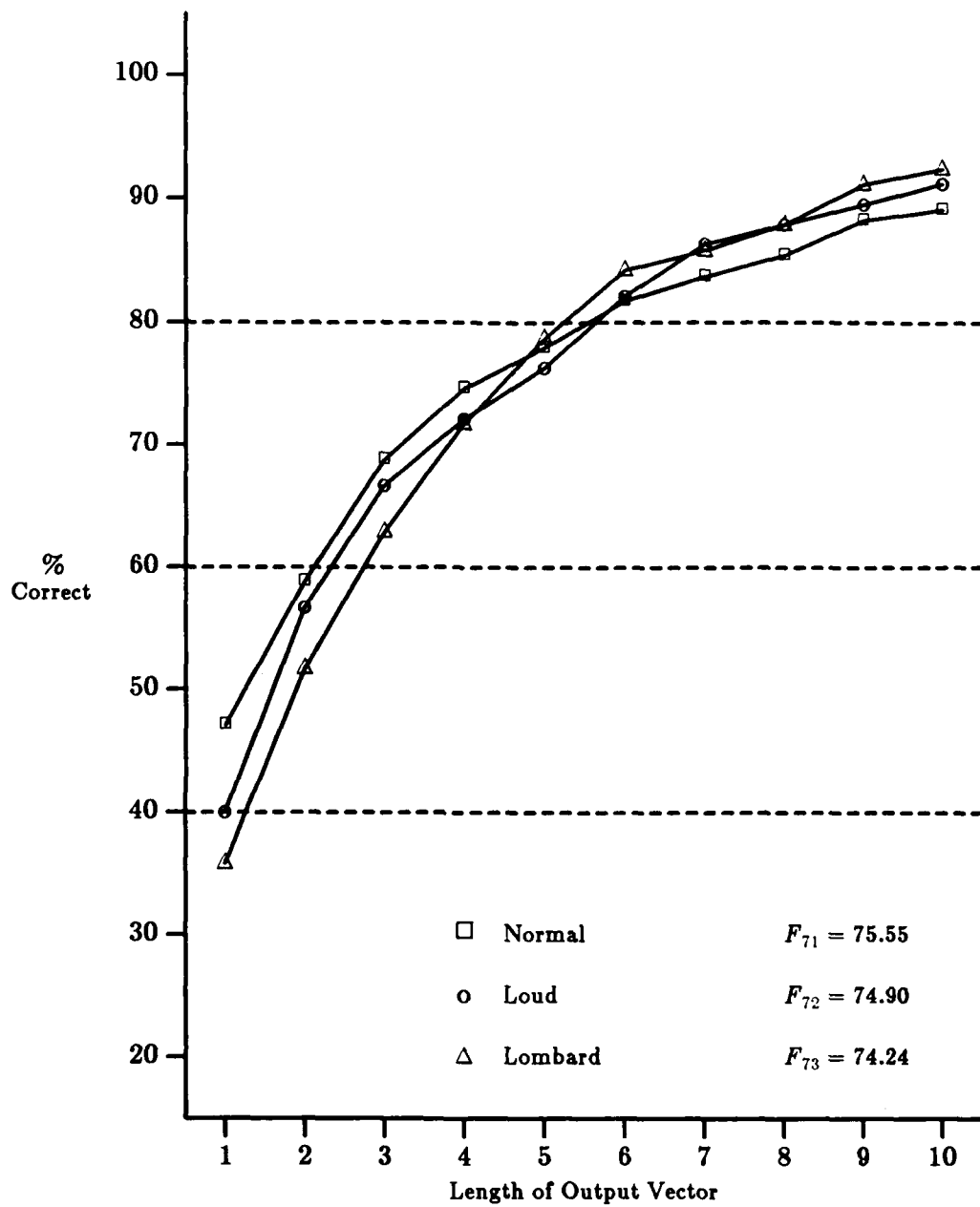


Figure 89. Recognition performance for SDW-SCD, Speaker #7

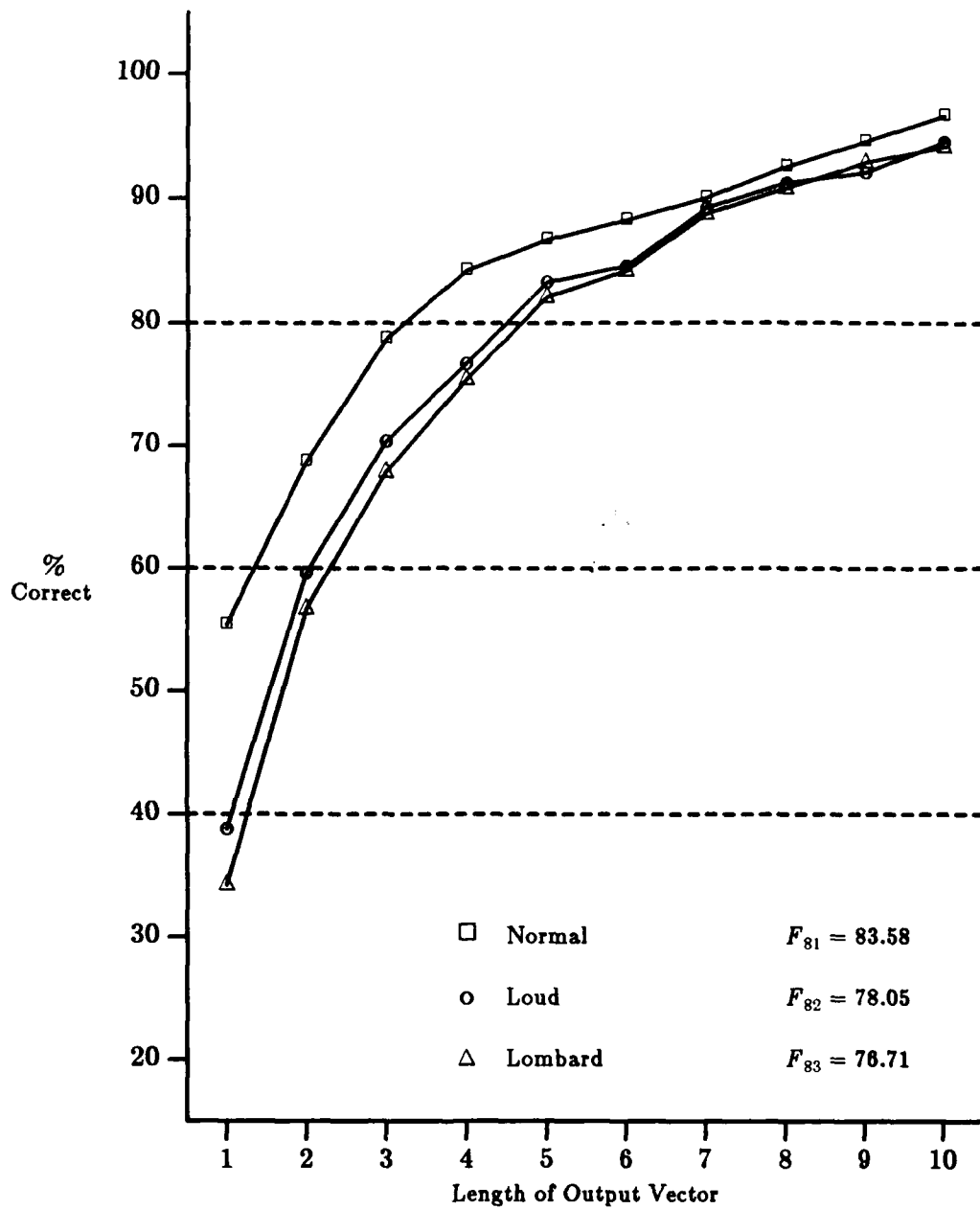


Figure 90. Recognition performance for SDW-SCD, Speaker #8

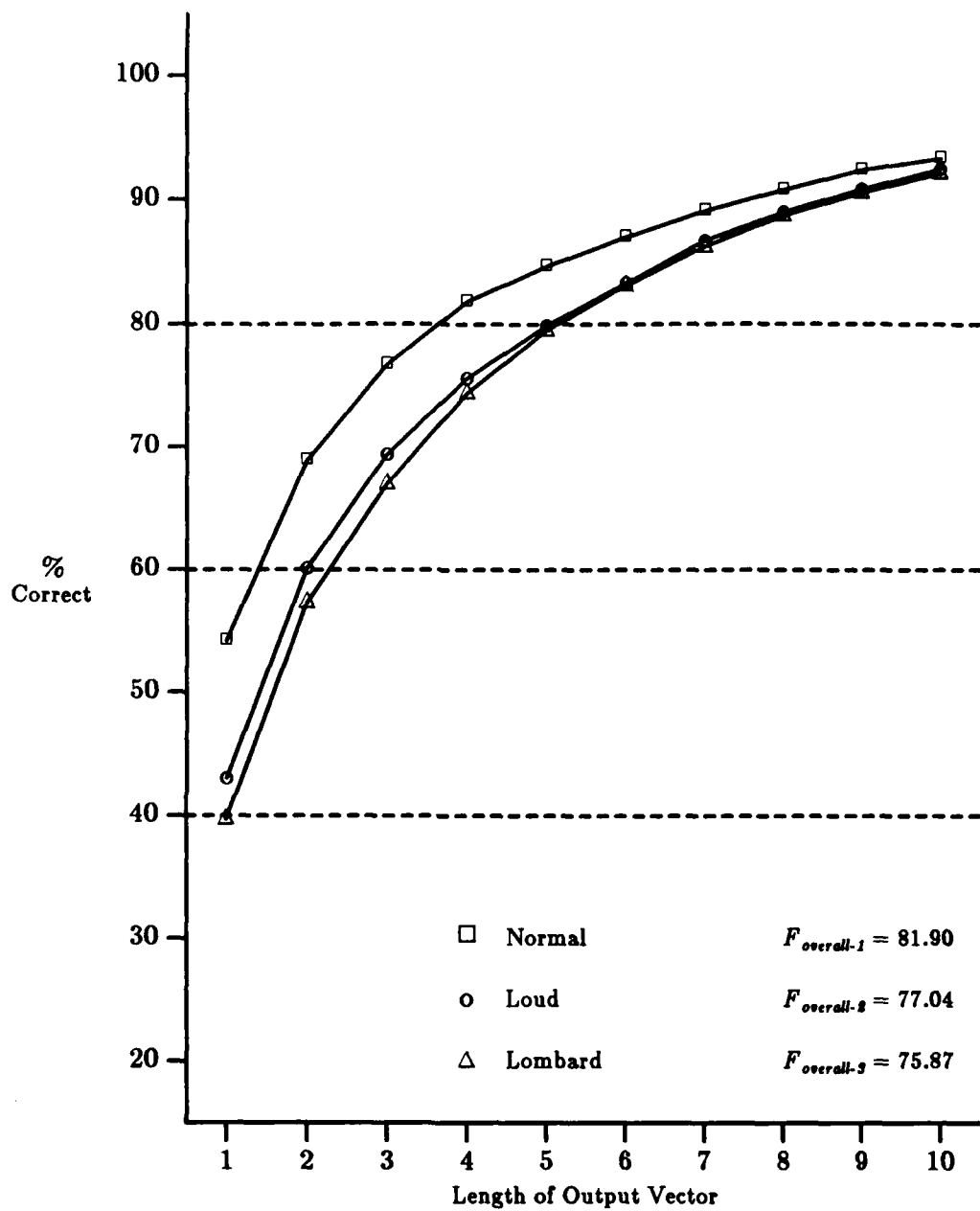


Figure 91. Recognition performance for SDW-SCD, all phonemes, all speakers

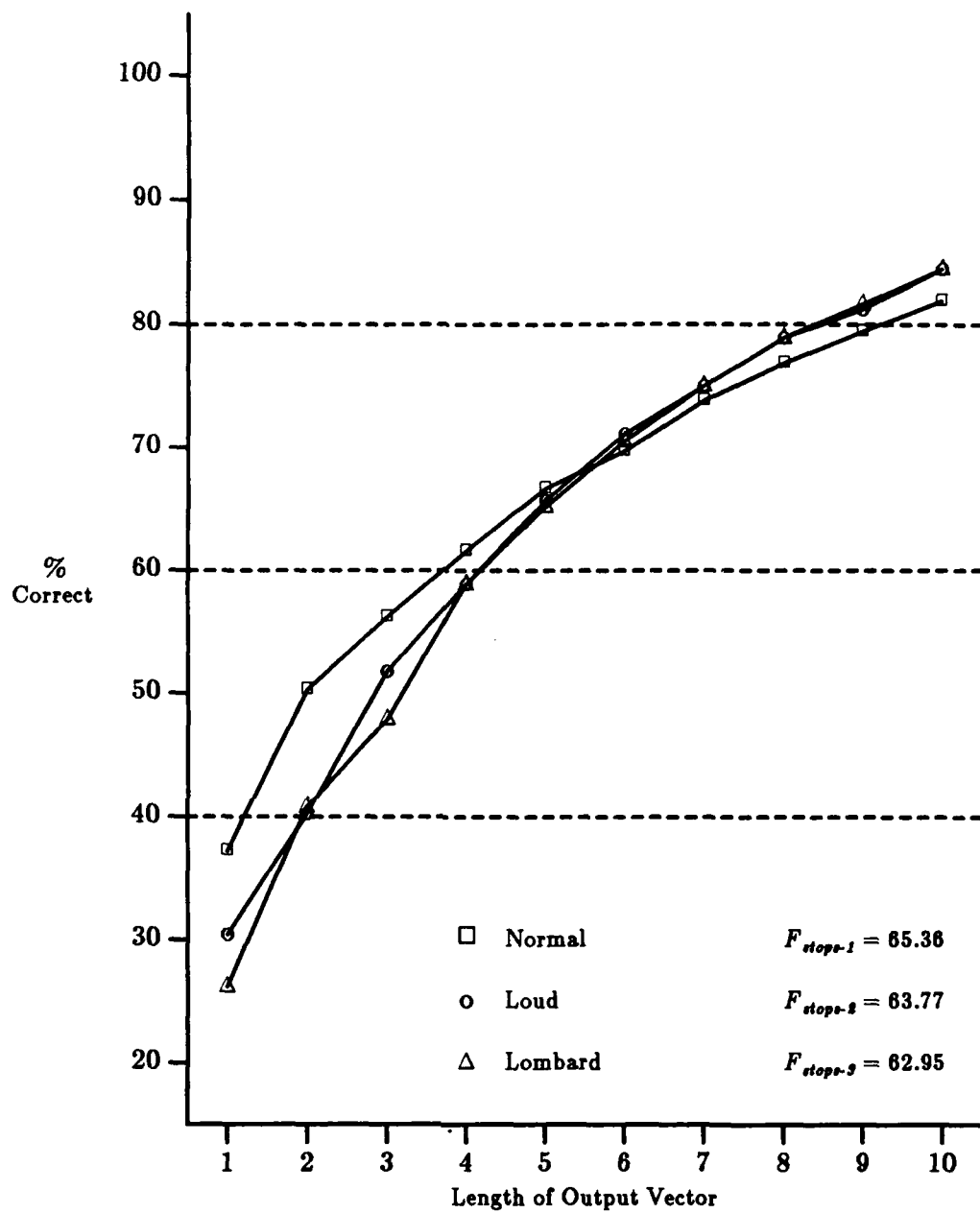


Figure 92. Recognition performance for SDW-SCD, stops, all speakers

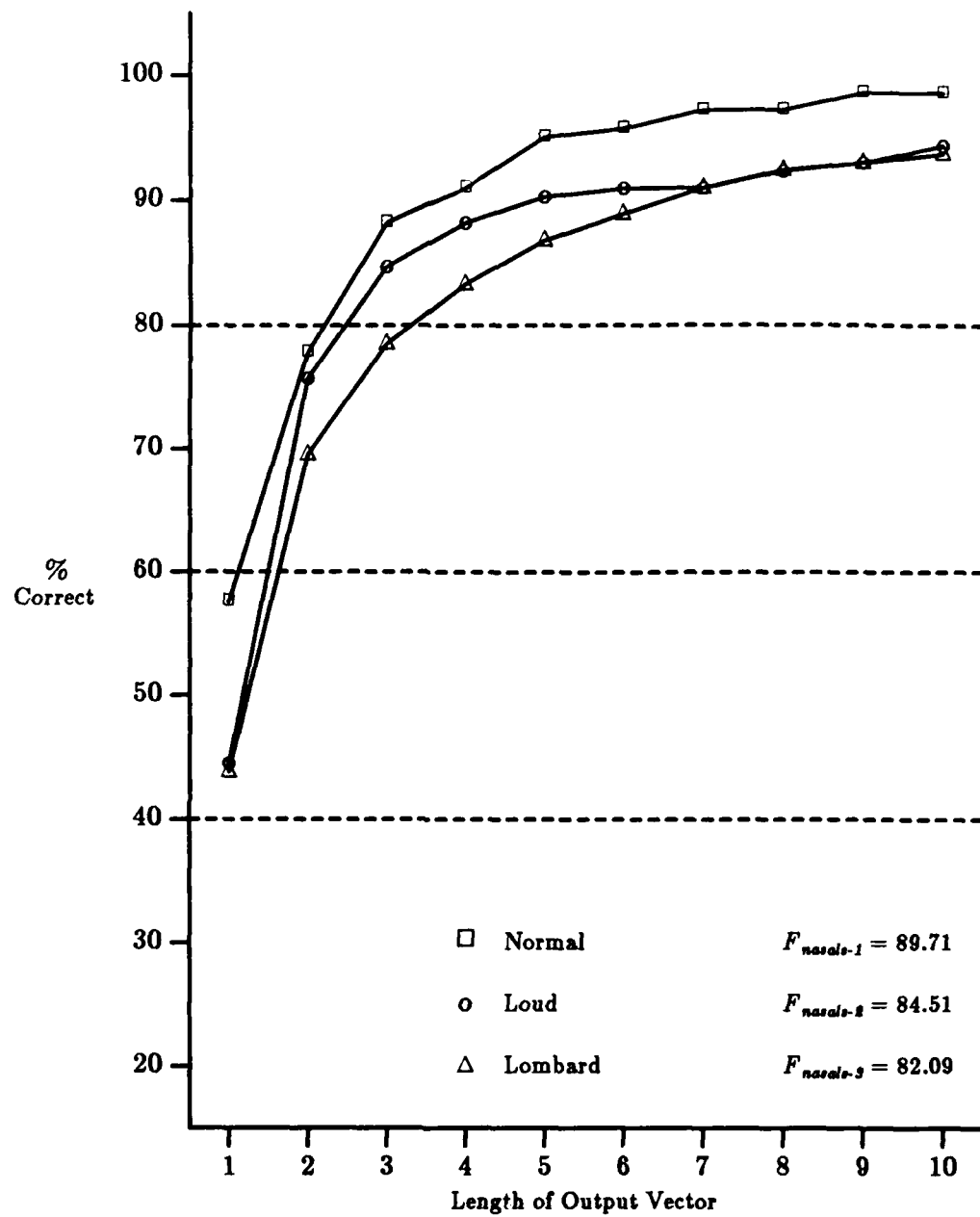


Figure 93. Recognition performance for SDW-SCD, nasals, all speakers

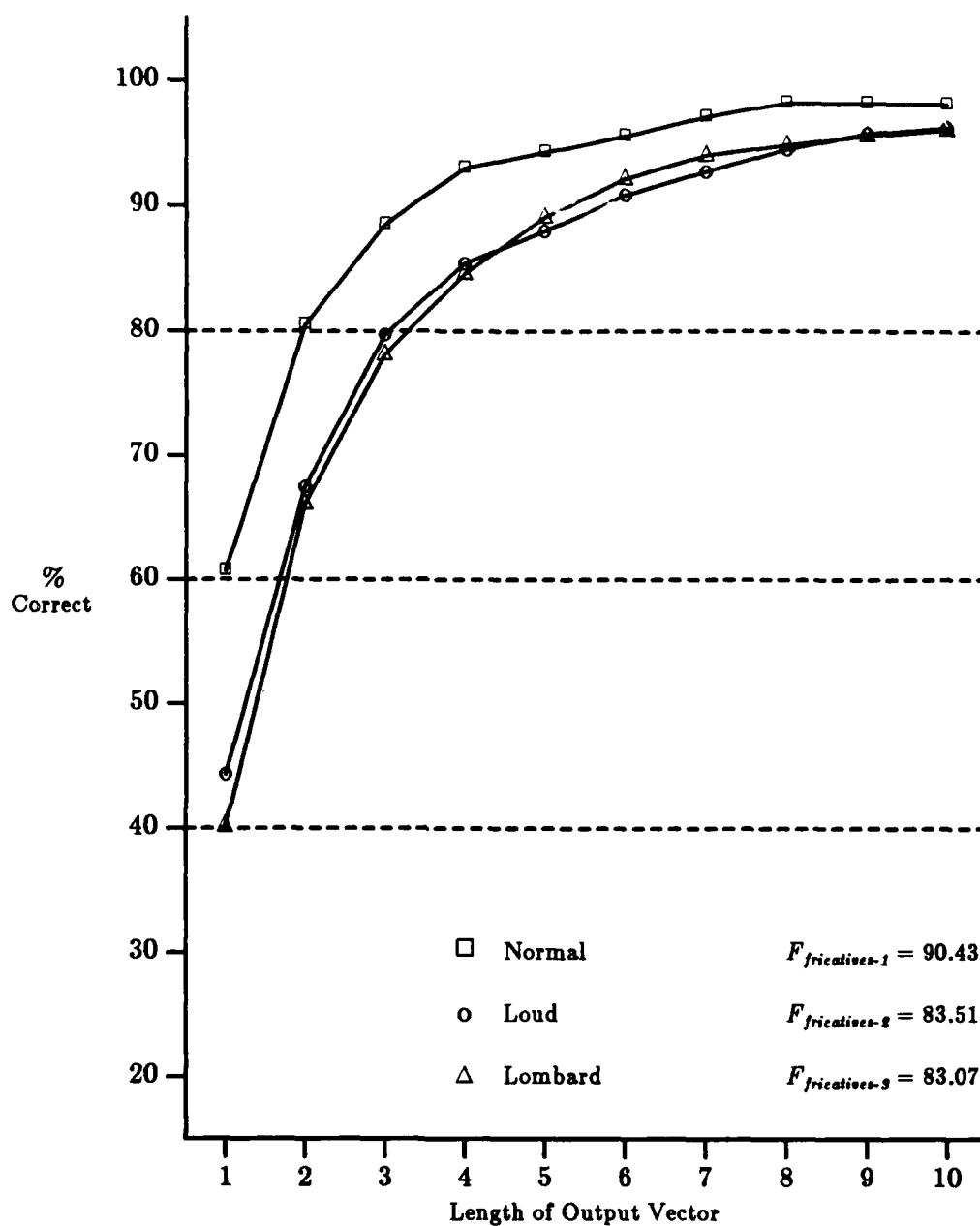


Figure 94. Recognition performance for SDW-SCD, fricatives, all speakers

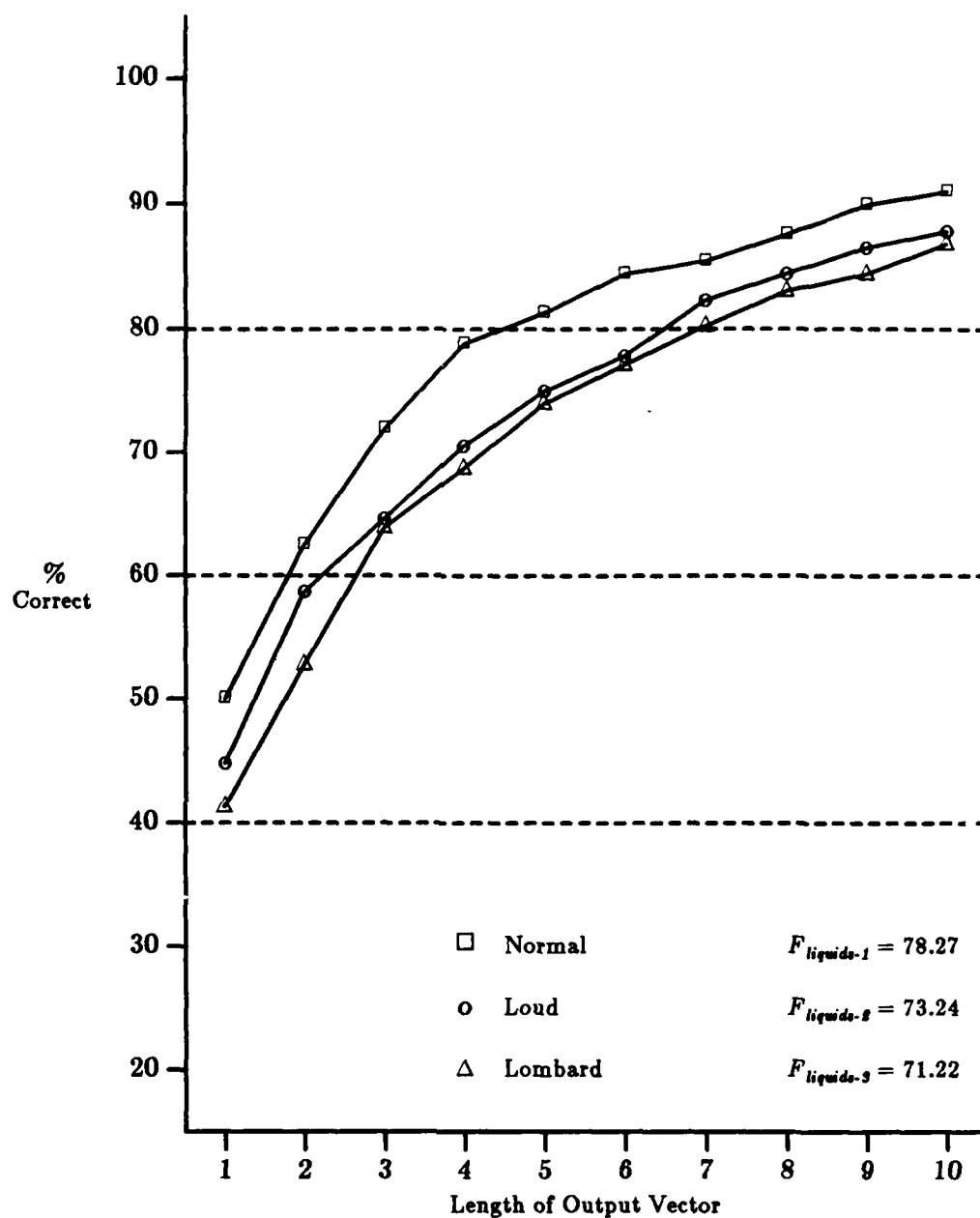


Figure 95. Recognition performance for SDW-SCD, liquids, all speakers

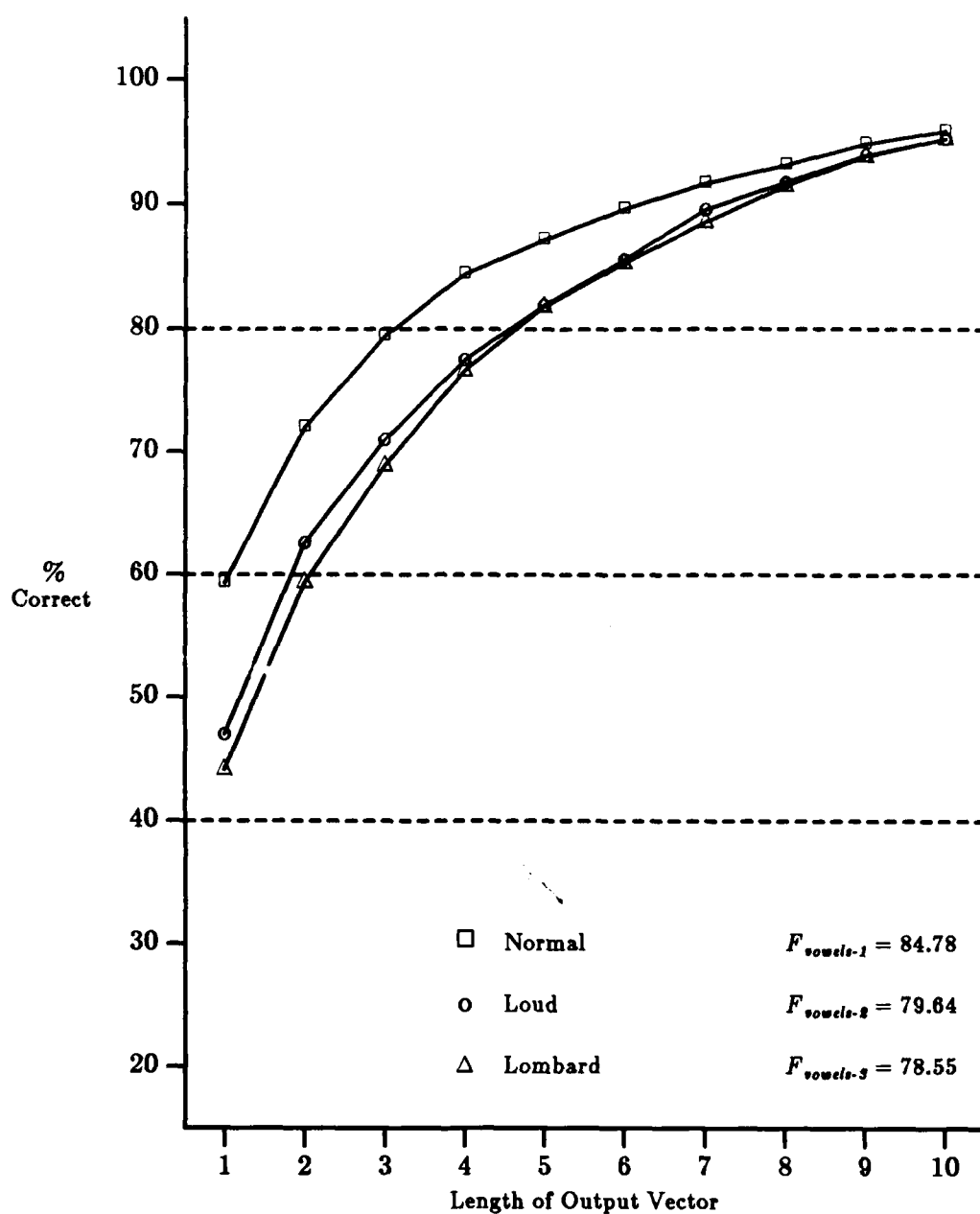


Figure 96. Recognition performance for SDW-SCD, vowels, all speakers

VITA

VITA

Bill J. Stanton, [REDACTED] is a distinguished graduate of the U. S. Air Force Academy, Class of 1973. As an undergraduate, his work included research in the security of defense communications systems. He completed Undergraduate Pilot Training with top honors in 1974 and Advanced Fighter Training in 1975. He served as an F4D Aircraft Commander and Wing Weapons and Tactics Officer at Royal Air Force Base Woodbridge, England from 1975 to 1978. During that time he participated in joint NATO Force exercises and taught laser weapon tactics. From 1978 to 1981 he served as an AT-38B fighter instructor pilot and academic instructor. His duties included aerial instruction in basic and advanced fighter maneuvers, surface attack tactics, low-level ingress techniques, and various tactical formations. Additionally, he was responsible for significant phases of the surface attack academic curriculum. He attended the Massachusetts Institute of Technology from 1981 to 1982, receiving a Master of Science in Electrical Engineering and Computer Science. His masters research involved designing and building an automatic thresholding system for a document scanner that used a linear charge-coupled sensing device. From 1983 to 1985, he was a member of the electrical engineering faculty at the U. S. Air Force Academy. During that time, he taught courses in advanced circuits, electronic devices, and speech processing. Additionally, he served as principal investigator of speech research under the sponsorship of the Rome Air Development Center, division chief of circuits and systems, Cadet Squadron 33 training officer, and T41 instructor pilot. In 1985, he won the William P. Clements Award for Outstanding Military Educator in Electrical Engineering. He is currently rated as a senior pilot in the U. S. Air Force with over 1400 hours of flying time. [REDACTED]

[REDACTED]

END

DATE

FILMED

DTIC

9-88