Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.
Cost Function Optimisation for Unit Selection Speech Synthesis

Amelia C. Kelly

Submitted to the University of Dublin, Trinity College
in fulfillment of the thesis requirement for the degree of Doctor of Philosophy

December 2012
Josie was no longer young, and feared to turn thirty; she feared to work on her dissertation, lest she complete it.

Shirley Hazzard,
The Transit of Venus, 1980

There is nothing to writing. All you do is sit down at a typewriter and bleed.

variously attributed to
Ernest Hemingway
Red Smith
Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

[Signature]

Amelia C. Kelly
Acknowledgments

It is a pleasure to thank the many people who made this thesis possible.

I would like to thank my supervisor, Christer Gobi, the guidance and advice he provided throughout my Ph.D. research. I am grateful also to Ailbhe Ní Chasaide, Nick Campbell, Brian Ó Raghallaigh, and Pauline Welby for their encouragement and support, and all the help they provided me during my years at Trinity College. I would like to thank all my colleagues at TCD, especially John Kane for Matlab discussions, and for help with perceptual tests, my abair.ie teammates Neasa Ní Chiarain and Christoph Wendler; Amelie Dorn, Raya Kalaldeh, Irena Yanushevskaya, Maria O’Reilly, John Dalton, Emer Gilmartin, John Duggan and everyone at the Phonetics and Speech Laboratory. I am indebted also to Julie Berndsen and her group at UCD, particularly Peter Cahill and Mark Kane, and also to John Sheridan and John Healy of the UCD Electronic Engineering Department.

I would especially like to thank Harald Berthelsen for assistance with countless programming-related (and non-programming related) issues, and without whom this project could never have been completed. I would also like to thank my brother Damien, for his constant support and invaluable assistance throughout my research. Special thanks also to Paul Taylor and all at Phonetic Arts, for making my stay in Cambridge so memorable.

This thesis would have been impossible to finish if it weren’t for the support provided by friends and acquaintances far too numerous to list here. I would, however, like to thank my wonderful friends Ann-Marie Madigan and Conor Smyth. Thanks also to my sister Madeleine and my brother-in-law Sam, to my housemates in Rathdown Road – Maria Roy, Emmy Lugoye, Liam Gray and Peetsy Darcy, and to the rest of the Dublin crowd and founding members of Businnessmonnay Inc., Sinead, Ollie, Mikey, Roney the Pony, and John Ryan, to name but a few. Thanks especially to Ollie Kinnane, for his love, confidence, encouragement, and other magic, and for somehow managing to make the most unlikely years, the end-of-thesis years, the best of my life.

Finally, thanks to my parents Margaret-Mary and Eamonn Kelly for their love and support and encouragement. It is to them I would like to dedicate this thesis.

— Rathmines, December 2012
The primary focus of this thesis was the optimisation of the cost function of a unit selection speech synthesiser. This was achieved by:

1. investigating the signal processing techniques used to calculate numerical representations of speech signals,
2. developing an objective method of determining which numerical representation is suitable for a given signal, and
3. modifying the cost function of a unit selection speech synthesiser by using a suitable numerical representation for different types of speech signals.

The cost function of a unit selection speech synthesiser is responsible for which segments of the database are selected as the synthetic speech output. One of the components of the cost function is the spectral distance measure, which calculates the perceived difference between the numerical representations of potentially adjacent segments.

The mel-frequency cepstral coefficient (MFCC) is a popular numerical representation of speech. In this thesis, MFCC calculation is examined by varying three fundamental values (i) window length used to extract parameters, (ii) the manner of time-frequency analysis and (iii) the shape of the filter used to convert the frequency scale. We demonstrate that small changes in the values of some fundamental parameters can lead to the calculation of different MFCC values and hence differences in spectral distance measurements. By altering the values of (i) – (iii) over a closed range, we calculated 243 different MFCC representations for the same database unit.

Previous studies have attempted to ascertain which of any number of numerical representations best represents a speech signal. This was generally done by performing perceptual evaluations whereby a listener provides an opinion on the similarity of two signal segments. The conclusions reached by the many studies performed were contrasting and impossible to compare due to differences in the database used, the nature of the perceptual evaluation, and the manner in which the numerical representations are calculated.

In this study, we introduce a framework for objectively assessing the efficacy of a numerical representation of a speech signal. The scoring system was based on predictions about the spectral distances that would be calculated between speech signals. In order to bypass the
need for perceptual testing, the speech signals, whose numerical representations we would compare, were carefully chosen, and placed in three Categories:

**Category A:** naturally sequential segments of speech from the same signal, within the same phoneme.

**Category B:** one segment of interest compared to a generic model of a similar segment that is not of the same phoneme or signal.

**Category C:** two segments from the same phoneme and signal, which are separated by a small distance so that they are not naturally consecutive.

The scoring system was then used to rank the 243 different MFCC representations previously defined. The aims of the experiment were to ascertain whether the same numerical representations ranked in a similar fashion for all signals examined and across all speakers, and to investigate whether there was a relationship between certain characteristics of a signal and the values (i) – (iii) used in the calculation of the higher ranking MFCC representations for that signal.

The signals examined belonged to the following Groups:

**Group 1:** non-turbulent, periodic sounds – vowels and diphthongs.

**Group 2:** turbulent, periodic sounds – voiced fricatives.

**Group 3:** turbulent, non-periodic sounds – voiceless fricatives.

and were examined for three American English speakers of the Arctic database – RMS (male), BDL (male), and SLT (female). It was concluded that the highest scoring numerical representation may be database specific, and more than one numerical representation may need to be used in a cost function in order to improve output synthesis.

In order to demonstrate the benefits of a tailored cost function, an evaluation procedure was designed to test the effects of using a different numerical representation for each sound group. The cost function was tailored to each by choosing a high scoring representation for each of the three sound Groups, based on the results calculated for the RMS database. Listeners compared synthesis created using the original cost function with that made using the modified cost function, using a natural utterance as a reference. The results significantly favoured the synthesis from the modified version of the cost function, with a 65% preference, compared to 35% for the original synthesiser.

The research presented in this thesis very strongly demonstrates:

1. The importance of carefully chosen values for fundamental parameters when calculating numerical speech representations, and the impact of these choices on the selection of segments by a unit selection cost function.

2. The benefit of tailoring the cost function for different types of signals contained in databases used for unit selection speech synthesis.

3. The contribution of an objective framework for assessing the suitability of numerical representations of speech signals.
Contents

List of Tables i
List of Figures ii

1 Introduction 1
  1.1 Thesis Overview .................................................. 1
  1.2 Background ......................................................... 4
    1.2.1 Unit selection speech synthesis ......................... 4
    1.2.2 Challenges remaining ..................................... 6
  1.3 Research Question ............................................... 9
    1.3.1 Objectives ...................................................... 9
  1.4 Contributions of the Thesis ................................... 10
    1.4.1 List of Publications ....................................... 11
  1.5 Thesis Organisation .............................................. 12

2 Unit Selection Speech Synthesis 14
  2.1 Text-to-Speech ..................................................... 14
    2.1.1 Front-end processing .................................... 15
    2.1.2 Back-end processing .................................... 18
  2.2 Unit Selection Speech Synthesis ............................. 23
    2.2.1 The cost function .......................................... 26
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Component cost functions for the default multisyn target cost</td>
<td>30</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary of perceptual test and feature extraction methods used in previous</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>studies of join cost</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Target utterances used for synthesis test</td>
<td>110</td>
</tr>
<tr>
<td>5.1</td>
<td>American English phone set used the the Arctic prompt list in Festival and</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>IPA notation</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Units extracted from the RMS, BDL and SLT Arctic database for testing</td>
<td>131</td>
</tr>
<tr>
<td>6.1</td>
<td>Component cost functions for the synthesiser's target cost</td>
<td>174</td>
</tr>
<tr>
<td>6.2</td>
<td>List of target and join sub costs used in the synthesiser</td>
<td>175</td>
</tr>
<tr>
<td>6.3</td>
<td>Percentage of Group 1–3 signal segments in the test stimuli</td>
<td>177</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Basic structure of a cascade formant synthesiser ................................. 20
2.2 Basic structure of a parallel formant synthesiser .................................. 20
2.3 Target cost and join cost ................................................................. 26
2.4 Viterbi algorithm ............................................................................. 31
2.5 Flowchart of the join cost calculation .................................................. 34
2.6 Detection and false alarm rate for detecting discontinuities in a signal ... 42
2.7 Probability density functions for continuous and discontinuous test stimuli, from Pantazis (2006). ................................................................. 43

3.1 The relationship between mel and Hz frequencies ................................. 50
3.2 Mel-scaled filter bank analysis on log magnitude spectra ...................... 51
3.3 The cepstrum of a waveform ............................................................ 52
3.4 Triangular filter (Chakroborty and Saha, 2009) ................................... 54
3.5 Speech waveform and corresponding MFCC values for the first 250 frames ... 55
3.6 Hanning windowed segment of a speech signal .................................... 57
3.7 The fractional Fourier domain .......................................................... 58
3.8 Rectangular, triangular and Gaussian filter banks .................................. 60
3.9 Windowed signal consisting of two cosine functions and its Fourier transform, where $\beta = 0$ ms ............................................. 64
3.10 Windowed signal consisting of two cosine functions and its Fourier transform, where $\beta = 0.5$ ms .................. 65
3.11 FRFT representations of a sinusoid for various values of $\alpha$ ............. 69
3.12 The result of multiplying the FT of a signal consisting of sinusoids with a mel-scaled filter bank composed of (a) triangular filters, (b) rectangular filters, and (c) Gaussian filters .......................................................... 72
3.13 MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of triangular filters 74
3.14 MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of rectangular filters 75
3.15 MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of Gaussian filters 76
3.16 Pythagoras theorem applied to distances in two-dimensional space ........ 79
3.17 Pythagoras theorem applied to distances in three-dimensional space ...... 80
4.1 The Euclidean distances between Category A MFCC vectors calculated using a range of values of window length, $\alpha$, and filter type ...................... 90
4.2 The Euclidean distances between Category B MFCC vectors calculated using a range of values of window length, $\alpha$, and filter type ...................... 92
4.3 The Euclidean distances between Category C MFCC vectors calculated using a range of values of window length, $\alpha$, and filter type ...................... 94
4.4 The Euclidean distances between MFCC vectors from Categories A, B, and C, calculated using a range of window lengths, $\alpha$ values, and triangular filters 95
4.5 The Euclidean distances between MFCC vectors from Categories A, B, and C, calculated using a range of window lengths, $\alpha$ values, and rectangular filters 96
4.6 The Euclidean distances between MFCC vectors from Categories A, B, and C, calculated using a range of window lengths, $\alpha$ values, and Gaussian filters 97
4.7 Scores for MFCC representations of one example of the phoneme /\l/ ..... 101
4.8 Comparison of three numerical representations using the scoring system ... 107
4.9 Screen shot of target utterances in diphone form ................................. 110
4.10 Results of the synthesis-based evaluation ............................................. 112
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Frequency distribution of phonemes in the Arctic database, from Kominek and Black (2003)</td>
<td>128</td>
</tr>
<tr>
<td>5.2</td>
<td>Mean ( f_0 ) of Group 1 segments in the RMS database</td>
<td>132</td>
</tr>
<tr>
<td>5.3</td>
<td>Scores for MFCC representations of Group 1 segments in the RMS database</td>
<td>134</td>
</tr>
<tr>
<td>5.4</td>
<td>Mean ( f_0 ) of Group 2 segments in the RMS database</td>
<td>139</td>
</tr>
<tr>
<td>5.5</td>
<td>Scores for MFCC representations of Group 2 segments in the RMS database</td>
<td>140</td>
</tr>
<tr>
<td>5.6</td>
<td>Scores for MFCC representations of Group 3 segments in the RMS database</td>
<td>142</td>
</tr>
<tr>
<td>5.7</td>
<td>Mean ( f_0 ) of Group 1 segments in the BDL database</td>
<td>145</td>
</tr>
<tr>
<td>5.8</td>
<td>Scores for MFCC representations of Group 1 signals in the BDL database</td>
<td>146</td>
</tr>
<tr>
<td>5.9</td>
<td>Mean ( f_0 ) of Group 2 segments in the BDL database</td>
<td>149</td>
</tr>
<tr>
<td>5.10</td>
<td>Scores for MFCC representations of Group 2 signals in the BDL database</td>
<td>150</td>
</tr>
<tr>
<td>5.11</td>
<td>Scores for MFCC representations of Group 3 signals in the BDL database</td>
<td>152</td>
</tr>
<tr>
<td>5.12</td>
<td>Mean ( f_0 ) of Group 1 segments in the SLT database</td>
<td>155</td>
</tr>
<tr>
<td>5.13</td>
<td>Scores for MFCC representations of Group 1 signals in the SLT database</td>
<td>156</td>
</tr>
<tr>
<td>5.14</td>
<td>Mean ( f_0 ) of Group 2 segments in the SLT database</td>
<td>158</td>
</tr>
<tr>
<td>5.15</td>
<td>Scores for MFCC representations of Group 2 signals in the SLT database</td>
<td>159</td>
</tr>
<tr>
<td>5.16</td>
<td>Scores for MFCC representations of Group 3 signals in the SLT database</td>
<td>161</td>
</tr>
<tr>
<td>6.1</td>
<td>Example of a target utterance in XML</td>
<td>173</td>
</tr>
<tr>
<td>6.2</td>
<td>Screen shot of part of the online listening test</td>
<td>178</td>
</tr>
<tr>
<td>6.3</td>
<td>Preference for V2 synthesis (modified cost function) compared to V1 synthesis (baseline)</td>
<td>179</td>
</tr>
<tr>
<td>6.4</td>
<td>Preference for utterances synthesised using V2 (modified synthesiser) com­ pared to V1 (baseline)</td>
<td>180</td>
</tr>
<tr>
<td>A.1</td>
<td>Distances calculated between naturally occurring consecutive speech segments</td>
<td>207</td>
</tr>
<tr>
<td>A.2</td>
<td>Distances calculated between naturally occurring consecutive speech segments</td>
<td>208</td>
</tr>
<tr>
<td>A.3</td>
<td>Distances calculated between naturally occurring consecutive speech segments</td>
<td>209</td>
</tr>
<tr>
<td>B.1</td>
<td>Mean ( f_0 ) measurements of segments of /i/ in the RMS database</td>
<td>211</td>
</tr>
<tr>
<td>B.2</td>
<td>Scores for MFCC representations of /i/ in the RMS database</td>
<td>212</td>
</tr>
<tr>
<td>B.3</td>
<td>Scores for MFCC representations of /i/ in the RMS database</td>
<td>213</td>
</tr>
</tbody>
</table>
B.4  Scores for MFCC representations of /i/ in the RMS database  .......... 214
B.5  Scores for MFCC representations of /ɛ/ in the RMS database  .......... 215
B.6  Scores for MFCC representations of /ɛ/ in the RMS database  .......... 216
B.7  Mean f_0 measurements of segments of /ɛ/ in the RMS database  .......... 217
B.8  Scores for MFCC representations of /ɛ/ in the RMS database  .......... 218
B.9  Mean f_0 measurements of segments of /æ/ in the RMS database  .......... 219
B.10 Scores for MFCC representations of /æ/ in the RMS database  .......... 220
B.11 Mean f_0 measurements of segments of /a/ in the RMS database  .......... 221
B.12 Scores for MFCC representations of /a/ in the RMS database  .......... 222
B.13 Mean f_0 measurements of segments of /ɔ/ in the RMS database  .......... 223
B.14 Scores for MFCC representations of /ɔ/ in the RMS database  .......... 224
B.15 Mean f_0 measurements of segments of /o/ in the RMS database  .......... 225
B.16 Scores for MFCC representations of /o/ in the RMS database  .......... 226
B.17 Mean f_0 measurements of segments of /ʌ/ in the RMS database  .......... 227
B.18 Scores for MFCC representations of /ʌ/ in the RMS database  .......... 228
B.19 Mean f_0 measurements of segments of /u/ in the RMS database  .......... 229
B.20 Scores for MFCC representations of /u/ in the RMS database  .......... 230
B.21 Mean f_0 measurements of segments of /u/ in the RMS database  .......... 231
B.22 Scores for MFCC representations of /u/ in the RMS database  .......... 232
B.23 Mean f_0 measurements of segments of /ai/ in the RMS database  .......... 233
B.24 Scores for MFCC representations of /ai/ in the RMS database  .......... 234
B.25 Mean f_0 measurements of segments of /au/ in the RMS database  .......... 235
B.26 Scores for MFCC representations of /au/ in the RMS database  .......... 236
B.27 Mean f_0 measurements of segments of /aɪ/ in the RMS database  .......... 237
B.28 Scores for MFCC representations of /aɪ/ in the RMS database  .......... 238
B.29 Mean f_0 measurements of segments of /ʊ/ in the RMS database  .......... 240
B.30 Scores for MFCC representations of /ʊ/ in the RMS database  .......... 241
B.31 Mean f_0 measurements of segments of /v/ in the RMS database  .......... 242
B.32 Scores for MFCC representations of /v/ in the RMS database  .......... 243
B.33 Mean f_0 measurements of segments of /z/ in the RMS database  .......... 244
B.34 Scores for MFCC representations of /z/ in the RMS database .......... 245
B.35 Mean /o measurements of segments of /3/ in the RMS database .......... 246
B.36 Scores for MFCC representations of /3/ in the RMS database .......... 247
B.37 Scores for MFCC representations of /θ/ in the RMS database .......... 249
B.38 Scores for MFCC representations of /f/ in the RMS database .......... 250
B.39 Scores for MFCC representations of /s/ in the RMS database .......... 251
B.40 Scores for MFCC representations of /f/ in the RMS database .......... 252
B.41 Mean /o measurements of segments of /i/ in the BDL database .......... 254
B.42 Scores for MFCC representations of /i/ in the BDL database .......... 255
B.43 Mean /o measurements of segments of /i/ in the BDL database .......... 256
B.44 Scores for MFCC representations of /i/ in the BDL database .......... 257
B.45 Mean /o measurements of segments of /ε/ in the BDL database .......... 258
B.46 Scores for MFCC representations of /ε/ in the BDL database .......... 259
B.47 Mean /o measurements of segments of /e/ in the BDL database .......... 260
B.48 Scores for MFCC representations of /e/ in the BDL database .......... 261
B.49 Mean /o measurements of segments of /æ/ in the BDL database .......... 262
B.50 Scores for MFCC representations of /æ/ in the BDL database .......... 263
B.51 Mean /o measurements of segments of /a/ in the BDL database .......... 264
B.52 Scores for MFCC representations of /a/ in the BDL database .......... 265
B.53 Mean /o measurements of segments of /ɔ/ in the BDL database .......... 266
B.54 Scores for MFCC representations of /ɔ/ in the BDL database .......... 267
B.55 Mean /o measurements of segments of /o/ in the BDL database .......... 268
B.56 Scores for MFCC representations of /o/ in the BDL database .......... 269
B.57 Mean /o measurements of segments of /ʌ/ in the BDL database .......... 270
B.58 Scores for MFCC representations of /ʌ/ in the BDL database .......... 271
B.59 Mean /o measurements of segments of /u/ in the BDL database .......... 272
B.60 Scores for MFCC representations of /u/ in the BDL database .......... 273
B.61 Mean /o measurements of segments of /u/ in the BDL database .......... 274
B.62 Scores for MFCC representations of /u/ in the BDL database .......... 275
B.63 Mean /o measurements of segments of /ai/ in the BDL database .......... 276
B.64 Scores for MFCC representations of /ai/ in the BDL database ............... 277
B.65 Mean f0 measurements of segments of /au/ in the BDL database ............... 278
B.66 Scores for MFCC representations of /au/ in the BDL database .................. 279
B.67 Mean f0 measurements of segments of /ɔi/ in the BDL database ................. 280
B.68 Scores for MFCC representations of /ɔi/ in the BDL database .................. 281
B.69 Mean f0 measurements of segments of /ɔ/ in the BDL database ................ 283
B.70 Scores for MFCC representations of /ɔ/ in the BDL database ................... 284
B.71 Mean f0 measurements of segments of /v/ in the BDL database ................. 285
B.72 Scores for MFCC representations of /v/ in the BDL database ................... 286
B.73 Mean f0 measurements of segments of /z/ in the BDL database ................. 287
B.74 Scores for MFCC representations of /z/ in the BDL database ................... 288
B.75 Mean f0 measurements of segments of /ʒ/ in the BDL database ................. 289
B.76 Scores for MFCC representations of /ʒ/ in the BDL database ................... 290
B.77 Scores for MFCC representations of /θ/ in the BDL database ................... 291
B.78 Scores for MFCC representations of /f/ in the BDL database .................... 292
B.79 Scores for MFCC representations of /s/ in the BDL database .................... 293
B.80 Scores for MFCC representations of /ʃ/ in the BDL database ................... 294
B.81 Mean f0 measurements of segments of /i/ in the SLT database .................. 295
B.82 Scores for MFCC representations of /i/ in the SLT database .................... 296
B.83 Mean f0 measurements of segments of /ɪ/ in the SLT database .................. 297
B.84 Scores for MFCC representations of /ɪ/ in the SLT database .................... 298
B.85 Mean f0 measurements of segments of /ɛ/ in the SLT database .................. 299
B.86 Scores for MFCC representations of /ɛ/ in the SLT database .................... 300
B.87 Mean f0 measurements of segments of /e/ in the SLT database .................. 301
B.88 Scores for MFCC representations of /e/ in the SLT database .................... 302
B.89 Mean f0 measurements of segments of /æ/ in the SLT database ................. 303
B.90 Scores for MFCC representations of /æ/ in the SLT database .................... 304
B.91 Mean f0 measurements of segments of /a/ in the SLT database ................. 305
B.92 Scores for MFCC representations of /a/ in the SLT database .................... 306
B.93 Mean f0 measurements of segments of /ɔ/ in the SLT database .................. 307
B.94 Scores for MFCC representations of /ɔ/ in the SLT database ................. 310
B.95 Mean f₀ measurements of segments of /o/ in the SLT database ................. 311
B.96 Scores for MFCC representations of /o/ in the SLT database ................. 312
B.97 Mean f₀ measurements of segments of /ʌ/ in the SLT database ................. 313
B.98 Scores for MFCC representations of /ʌ/ in the SLT database ................. 314
B.99 Mean f₀ measurements of segments of /u/ in the SLT database ................. 315
B.100 Scores for MFCC representations of /u/ in the SLT database ................. 316
B.101 Mean f₀ measurements of segments of /u/ in the SLT database ................. 317
B.102 Scores for MFCC representations of /u/ in the SLT database ................. 318
B.103 Mean f₀ measurements of segments of /ai/ in the SLT database .................. 319
B.104 Scores for MFCC representations of /ai/ in the SLT database .................. 320
B.105 Mean f₀ measurements of segments of /au/ in the SLT database .................. 321
B.106 Scores for MFCC representations of /au/ in the SLT database .................. 322
B.107 Mean f₀ measurements of segments of /ɔi/ in the SLT database .................. 323
B.108 Scores for MFCC representations of /ɔi/ in the SLT database .................. 324
B.109 Mean f₀ measurements of segments of /ɔ/ in the SLT database .................. 325
B.110 Mean f₀ measurements of segments of /ɔ/ in the SLT database .................. 326
B.111 Scores for MFCC representations of /ɔ/ in the SLT database .................. 327
B.112 Scores for MFCC representations of /ɔ/ in the SLT database .................. 328
B.113 Mean f₀ measurements of segments of /v/ in the SLT database .................. 329
B.114 Scores for MFCC representations of /v/ in the SLT database .................. 330
B.115 Mean f₀ measurements of segments of /v/ in the SLT database .................. 331
B.116 Scores for MFCC representations of /v/ in the SLT database .................. 332
B.117 Mean f₀ measurements of segments of /ɔ/ in the SLT database .................. 333
B.118 Scores for MFCC representations of /ɔ/ in the SLT database .................. 334
B.119 Scores for MFCC representations of /f/ in the SLT database .................. 335
B.120 Scores for MFCC representations of /f/ in the SLT database .................. 336
B.121 Scores for MFCC representations of /s/ in the SLT database .................. 337
B.122 Scores for MFCC representations of /ʃ/ in the SLT database .................. 338
Speech synthesis is currently at the cutting edge of technological research, with far-reaching applications in the areas of education, accessibility, and gaming. Access to inexpensive memory and increased computing power have seen the unit selection speech synthesis technique in particular dominate the field of speech synthesis in the last twenty years. The success of this method is largely due to the naturalness and intelligibility of the synthetic speech output, which surpasses that of its predecessors. However, since unit selection relies on joining together segments of recorded speech, it possesses inherent flaws in the form of concatenation discontinuities, or joins, that ruin the illusion of human-like synthetic speech. This study addresses this problem by investigating ways to lessen the impact of concatenation errors on synthetic speech output.

### 1.1 Thesis Overview

The motivation of this thesis is to investigate what changes can be made to the cost function in a unit selection speech synthesis system in order to produce better synthetic speech.
There are three main propositions of this study:

1. The synthetic speech output can be improved by examining the signal processing techniques used to calculate certain components of the join cost.

2. The synthetic speech output can be improved by tailoring the spectral distance component based on the acoustic characteristics of the sound being examined.

3. An objective method of assessing the suitability of aspects of the spectral distance component would greatly benefit this line of research.

In this study, the mel-frequency cepstral coefficient (MFCC) is investigated, as the distance between vectors of MFCCs is often used to represent the spectral distance between successive segments chosen as candidates for synthetic speech output. Some mathematical properties of this important representation are presented in Chapter 3. Previous studies have shown that the calculation of MFCCs can be affected by a number of different factors. Three of these factors are investigated in this thesis:

(i) the window length used to examine the speech segments,

(ii) the time-frequency analysis performed on the signal, and

(iii) the shape of the filters in the mel filter bank.

We show that different choices of these values lead to different values used to represent the same database units. The choice of numerical representation used in the spectral distance measure affects the selection of units for synthetic output. There is no consensus as to which spectral distance measure best represents a speech signal. The contrasting results of previous studies suggested that perhaps there was no universal numerical representation that best suits all databases and speakers. By comparing the methods of parameter extraction, it was anticipated that a certain dependency of some fundamental parameters on the characteristics of the signal could be identified.

In all the research to date, the effectiveness of a metric is judged by reference to a perceptual
1.1. Thesis Overview

experiment performed by a group of test listeners. In order for these tests to have statistical significance, a large sample size is required. Unfortunately perceptual experiments of this kind are naturally extremely time consuming. This represents a significant limitation on the ability of speech researchers to conduct a large number of tests in a reasonable time frame.

In this thesis we propose a novel technique that allows us to quickly test how a range of different signal processing techniques will affect the perceived sound. The method can therefore be used for determining which of any given selection of numerical representations would best model a speech signal. The definition of a 'useful' numerical representation was ascertained as one whose values would be similar for similar-sounding signals, and very different for different sounding signals. The scoring system was based on predictions about the spectral distances that would be calculated between speech signals. In order to bypass the need for perceptual testing, the speech signals, whose numerical representations we would compare, were carefully chosen, and placed in three Categories:

**Category A:** naturally sequential segments of speech from the same signal, within the same phoneme.

**Category B:** one segment of interest compared to a generic model of a similar segment that is not of the same phoneme or signal.

**Category C:** two segments from the same phoneme and signal, which are separated by a small distance so that they are not naturally consecutive.

The scoring system was then used to rank the 243 different MFCC representations previously defined. The aims of the experiment were:

1. to ascertain whether the same numerical representations ranked in a similar fashion for all signals examined.

2. to determine whether the numerical representations ranked in a similar way for more than one speaker.
3. to investigate whether there was a relationship between certain characteristics of a signal and the values (i) – (iii) used in the calculation of the higher ranking MFCC representations for that signal.

The signals examined belonged to the following Groups:

**Group 1**: non-turbulent, periodic sounds – vowels and diphthongs.

**Group 2**: turbulent, periodic sounds – voiced fricatives.

**Group 3**: turbulent, non-periodic sounds – voiceless fricatives.

and were examined for three American English speakers of the Arctic database – RMS (male), BDL (male), and SLT (female). In order to demonstrate the benefits of a tailored cost function, an evaluation procedure was designed to test the effects of using a different numerical representation for each sound group. The cost function was tailored to each by choosing a high scoring representation for each of the three sound Groups, based on the results calculated for the RMS database in Chapter 5. The evaluation showed a preference for the modified cost function.

1.2 Background

The following sections provide a broad overview of the unit selection process, highlighting the areas for which further research could be done to improve the performance of a unit selection speech synthesis system.

1.2.1 Unit selection speech synthesis

Unit selection (Hunt and Black 1996) is a data-driven method of speech synthesis that draws from a large database of pre-recorded speech. Each sound in the database is annotated using a phonetic label. Usually the smallest unit labelled is the diphone, which describes a
sound that begins half way though one phoneme and ends half way through the adjacent phoneme, so as to capture the transition from one sound to the next. The user enters a string of text, known as the *target utterance*, as input to the system, which is then converted to a string of phonetic labels. For example the target utterance *cat* would be converted into a string of diphones /æ/-k/ /k-æ/ /æ-t/ /t-#. The task of the unit selection algorithm is to search the database for units that match the target utterance, creating a list of candidate units. In practice, there will normally be many candidates for every unit of the target utterance. Once this list has been created, the algorithm must select one candidate unit to represent each segment of the target utterance, so that they can be located in the database and their corresponding speech waveforms joined together to form synthesised speech. The unit selection process uses a *cost function* to assess which sequence of candidate units would produce the best synthesis. The cost function has two elements: the *target cost*, which is used to figure out how well candidate units represent their corresponding units in the target utterance, and the *join cost*, which measures how seamlessly the candidate units join to their potential neighbours. The higher the combined cost incurred by a sequence of segments, the less likely it is that the sequence will be chosen for synthesis.

The target cost is used to penalise candidate units for any linguistic or contextual deviation they show from the target utterance. For example, while there may be plenty of candidate units for the diphone /æ-t/, some of them may be unsuitable to represent the target diphone as they may have a different stress associated with them, or they may have had different phonetic neighbours in the database recordings. Such candidate units will be penalised accordingly by the target cost.

The join cost is used to penalise candidate units for any acoustic deviation they show from consecutive candidate units. Join cost is essentially a measurement of the perceptual quality of the concatenation of two consecutive units in a synthesised sentence. Ideally, the lower the join cost, the harder it is to perceive the existence of a join between the two segments. Factors that would make the join seem obvious would be a notable fluctuation of the fundamental frequency ($f_0$) or the amplitude of the speech signal, or a significant difference between the spectral characteristics of each segment. For example, if we wanted to con-
1.2 Challenges remaining

Despite the success of unit selection speech synthesis, many challenges remain in reducing the number of audible joins in the synthetic speech output. Some interesting topics are discussed below, which have provided the motivation for this thesis.

1.2.2.1 Spectral discontinuity

One of the major challenges for unit selection is finding an adequate measurement of the spectral discontinuity between two speech segments, an important component of the join cost. Consider again the segments /k-æ/ and /æ-t/. Calculating the spectral discontinuity between these two segments requires that the spectral characteristics of each be quantified by numerically coding the sounds as vectors of acoustic measurements. More specifically, the numeric representation is taken only on the portions of each sound that will be coming into contact, i.e. at the end of the first segment /k-æ/ and at the start of the second
1.2. Background

The speech sounds, therefore, are windowed before numerically coding the signals. The Euclidean (or similar) distance between the vectors of acoustic measurements that represent the sounds can then be calculated in order to express the perceptual difference between the sounds. The acoustic measurement used to represent the sound must therefore be one for which perceived changes in the sound are accurately mirrored by numerical changes in the acoustic representation. The MFCC remains one of the most popular speech parameterisations in both speech synthesis and speech recognition. Previous studies (Zheng, Zhang and Song 2001, Ganchev, Fakotakis and Kokkinakis 2005, Kirkpatrick, O'Brien, Scaife and Errity 2007, Kelly 2010, Kelly and Gobl 2011b, Kelly and Gobl 2011a) demonstrate that the MFCC values can be affected by many different choices made during their calculation.

A number of studies (Wouters and Macon 1998, Klabbers and Veldhuis 1998, Chen and Campbell 1999, Stylianou and Syrdal 2001, Vepa, King and Taylor 2002, Pantazis, Stylianou and Klabbers 2005, Kirkpatrick et al. 2007) have been carried out in an attempt to quantify the ability of the MFCC and other representations to accurately represent a speech signal. These studies compare different numeric representations of speech sounds by assessing listeners' capability of detecting a join. Although some of these and similar studies often conclude that one metric outperforms the others, the highest correlation achieved between such metrics and human perception of discontinuity is roughly 0.66 (Taylor 2009). Despite their popularity, MFCC-based measures of discontinuity have had mixed success in such studies. For example, Wouters and Macon (1998), concluded that the MFCC outperformed the others, while Klabbers and Veldhuis (1998) cited the MFCC as performing the worst. One reason for discrepancies in the performance of the MFCC in this case is that Wouters and Macon used a 5 ms window to examine the speech waveforms, while Klabbers and Veldhuis used a 20 ms window. Likewise Kirkpatrick et al. (2007) found that MFCCs based on wavelet analysis outperformed those calculated using the Fourier transform. Clearly it is not possible to compare the performance of different metrics when the role of such fundamental parameters such as (i) – (iii) above are not properly investigated with real speech signals. Investigating the performance of different implementations of the MFCC by varying these parameters could reveal an optimal setting for getting the best results from the
MFCC metric. Again it is worth noting the difficulties that arise for different researchers in examining this question, and the large variation in window lengths that are used in the different studies. Another major reason why these different studies cannot be directly compared is due to the different manners in which each human perception test was run. This highlights the importance of our novel insight, namely that by testing metrics on consecutive and non-consecutive speech segments we can fairly test and unambiguously compare the performance of a wide range of different metrics.

1.2.2.2 Signal characteristics

Previous studies investigating numerical representations of speech sounds have tended to focus mainly on vowels, diphthongs and other sonorant sounds, characterised by the periodicity associated with vocal fold oscillation. Studies within these sound categories (Vepa and King 2005) have shown that joins are more readily detected in diphthongs for example, than in vowels. This suggests that the inherent properties of the sound determines in part what numerical representation best represents them. Sounds such as voiceless fricatives are characterised by a turbulence resembling white noise, and display none of the periodicity of vowels and other sonorant sounds. Despite this, no studies have been carried out to investigate what parameterisation best quantifies these sounds. In the unit selection cost function, the numerical representation of speech sounds is central to the join sub-cost of spectral distance, but ultimately a blanket acoustic representation is used to calculate it, regardless of what type of sound is being joined. This raises a legitimate question about how adequate the acoustic representation of a vowel sound would be at representing a voiceless fricative, for example. Optimising the unit selection algorithm so that it uses a specially designed cost function for each type of speech sound has the potential to produce better quality speech synthesis output.
1.3 Research Question

The previous section provided an overview of unit selection speech synthesis and highlighted the lack of fundamental research into some of the most crucial signal processing questions facing unit selection speech synthesis. This gives rise to the following broad research question: How can the above challenges be addressed in order to improve the synthetic speech output of a unit selection speech synthesiser? More specifically:

- What steps can be taken to ensure that the spectral distance measurement used in the speech synthesiser is a good measure of whether or not two segments of speech are continuous or discontinuous?

- How can we investigate the importance of certain target and join sub-costs relative to others so that we can prioritise candidate units that exhibit these characteristics?

- Do these measurements depend in any way on the acoustic characteristics of the type of sound being measured?

- Will incorporating these changes into the cost function of a unit selection speech synthesiser significantly improve the synthetic speech output of the system?

1.3.1 Objectives

The following objectives were proposed in this thesis in an attempt to address the issues raised by the research question:

1. To demonstrate the effects that certain parameter choices have on the calculation of numerical speech signal representations, and ultimately on the synthetic output of a unit selection speech synthesiser.

2. To develop a method of rating numerical representations in terms of their usefulness in a way that is independent of perceptual testing.
3. To apply the devised scoring system to large datasets to establish whether there is any relationship between particular value choices and the signal under consideration, and whether these dependencies hold across speakers.

4. To demonstrate that unit selection speech synthesis output can be improved if we account for differences between signals.

1.4 Contributions of the Thesis

The research conducted for this thesis makes a number of contributions to the field:

- Identification of parameters such as (i) – (iii) and their effects on the calculation of MFCCs as key reasons for the lack of agreement among previous studies comparing spectral discontinuity measures.

- A novel framework for empirically measuring the efficacy of numerical speech signal representations without over-reliance on perceptual data.

- An investigation into the relationship between the characteristics of the signal and the numerical representation of that signal.

- The creation of a unit selection speech synthesiser with a cost function modified to account for these dependencies. This synthesiser has since been incorporated into the abair.ie Irish speech synthesis project (Ní Chasaide, Ní Chiaráin, Wendler, Berthelsen, Kelly, Gilmartin, Ní Dhonnchadh and Gobl 2011).

- An investigation of the impact of the choice of numerical representation on the output of a unit selection speech synthesiser.

- The design of an evaluation procedure to test the performance of the sound-group specific cost functions.
1.4 Contributions of the Thesis

1.4.1 List of Publications

The contributions of this thesis to the field of speech technology have been incorporated into the following publications:


1.5 Thesis Organisation

The thesis is organised as follows:

Chapter 2 begins with a description of text-to-speech systems, and the focuses on unit selection speech synthesis in particular, and provides a description of the cost function and its components. We then focus on the numerical speech representations used in the spectral component sub-cost, and outline some previous studies that have attempted to assess the efficacy of these measurements. This provides us with a guide in how to approach the investigations that this study aims to address.

Chapter 3 describes the calculation of the MFCC and discusses the particular choices that can be made in their calculation, and the effects that these choices are likely to have on the calculated values. These effects are then shown mathematically, using straightforward signals, and experimentally, using a speech signal as an example. The use of the MFCC as a component of the spectral distance measure of the join cost is then discussed.

Chapter 4 describes a method of assessing numerical representations of speech signals by predicting spectral distances between carefully selected signals. We demonstrate, using the example signal from Chapter 3, the calculation of the distances between signals. We define three Categories for which we can reasonably predict the scale of the distances that are likely to be calculated. We then define the scoring system in terms of the relationships between these distances. We then demonstrate how the scoring system can be employed to rank certain numerical representations of a database, and how the highest-scoring numerical representation may differ depending on the type of signal it is examining. The contribution of the scoring system is supported by conducting a synthesis-based evaluation. The assumptions made in devising the scoring system are then discussed, as are the limitations and contributions of the scoring system as laid out in this chapter. Finally, the contributions of the scoring system to unit selection speech synthesis are discussed.
Chapter 5 elaborates on the experiments carried out in Chapters 3 and 4 and examines many examples of real speech in order to investigate whether the highest rating MFCC representations are calculated using similar values of (i) – (iii) for every signal. In the study outlined in this chapter, we also investigate whether the results hold across speakers, and across signal types.

Chapter 6 describes the evaluation component of this project. The synthesiser, which was built for the purposes of evaluation, is described. Test sentences are held out from the Arctic database and used to evaluate the performance of the enhanced speech synthesiser compared to the baseline synthesiser.

Chapter 7 concludes by summarising the research outlined in the thesis and discussing the scope for future work in this area.
CHAPTER 2

Unit Selection Speech Synthesis

This chapter serves as an introduction to unit selection speech synthesis, beginning with an overview of general text-to-speech (TTS) systems in Section 2.1, and continuing to a more in-depth discussion of unit selection speech synthesis, particularly the cost function, in Section 2.2. Section 2.3 focusing on the spectral distance measure component of the cost function, and discusses studies in the literature that have been carried out to determine which combination of numerical representation and distance measure best represents a speech signal.

2.1 Text-to-Speech

Text-to-speech (TTS) synthesis is one of the fastest growing areas in both linguistics and computer science today. The advances made in the last few decades have transformed synthetic speech from a stereotypical American English monotone to synthetic voices that are relatively natural, human-sounding and intelligible. The methods used in creating synthetic speech have evolved from parametrical methods, which model the vocal tract filter using electrical circuits, to concatenative methods, which piece together large chunks of recorded
speech, through to the more advanced statistical methods that are growing in popularity to­day. TTS systems are now indispensable as an accessibility tool for the visually and vocally disabled, as a teaching resource for dyslexic students, and as a practical tool in industry and telecommunications. Internet real-world gaming applications and human-machine interaction will no doubt guarantee the future of text-to-speech systems.

The purpose of this section is to give an overview of the basic principles of text-to-speech synthesis, and the current technologies used to create synthetic speech. A basic text-to-speech synthesiser will consist of a natural language processing (NLP) or front-end module and a back-end synthesis module.

The front-end deals with the tasks involved in converting inputted text to a machine-readable phonetic representation, while the back-end module is responsible for the synthesis and sound output. Once the input text has been normalised and converted to a phonetic representation, the synthesis module will convert this utterance into an acoustic signal that forms the speech output of the synthesiser. There are many techniques available for creating speech output which are introduced in this chapter.

### 2.1.1 Front-end processing

The main task of the front-end module is to convert a sequence of ASCII characters into a corresponding sequence of machine-readable characters that, unlike the English alphabet for example, demonstrate a one-to-one mapping to the sounds they represent. Before this can be achieved, the text must be normalised, a process that converts numbers and symbols into words. The output of this process is a string of characters that contains only letters of the alphabet. The next stage is converting the characters to a phonetic representation.

Further front-end processing can include marking-up the text by labelling each segment with information about its position in the phrase, its left and right unit context, the level of stress associated with it, and other contextual information.

Although the process may sound straightforward, there are many opportunities for confusion.
For example, the sentence

He assumed a $30,000 loan at 15 1/4 % and put a down payment on a little 2 bedroom place (Waits 1983)

contains three challenges for a normalisation tool – changing the numbers “30,000”, “15 1/4” and “2” to words. For a human, these problems are easily solvable using context and prior knowledge, but for a machine to reach the same conclusions, a lot of thought and effort is required on the part of the designer.

The machine needs to figure out, for example, that the sequence 30,000 is a quantity of money – not a date, a time, or a sequence of digits. This could be inferred by the use of the comma, but the designer cannot count on the user always including such syntactical clues in the input text. The dollar sign provides another clue. However, pronouncing the sequence as “thirty thousand dollars” is incorrect – the machine also needs to take into account the context that the number is in. In this case it is being used as an adjective that describes the size of the loan. The very presence of a noun after the figure indicates that it should be pronounced “thirty thousand dollar loan”.

The next task is quite similar - recognising that 15 1/4 is a fraction and by viewing the context, eliminating all other possible pronunciations of the sequence, like “fifteen one slash four” and “fifteen quarters”, for example. The presence of the percentage sign should indicate that the sequence be pronounced “fifteen and one quarter percent”.

Changing the number 2 to “two” is the simplest normalisation challenge in this example. In Irish however, this task is more difficult and the number 2 could be pronounced in three separate ways depending on the context.

These ambiguities are usually overcome with the use of decision trees, which can determine in which the numbers or the words should be pronounced depending on the surrounding tokens and on the part of speech that the word is seen to belong to.

The output of the normalisation tool in this case would be
He assumed a thirty thousand dollar loan at fifteen and a quarter percent and put a down payment on a little two bedroom place.

The next step involves the conversion of inputted text to a phonetic representation. This is achieved in classical phonetics using the International Phonetic Alphabet (IPA) system, and a similar technique is used in TTS systems, the only difference being that the symbols are machine readable and can be found on a keyboard. This process is known as phonetisation and can be achieved by identifying words in an input string and consulting a pronunciation dictionary or lexicon, or, in the case of newly-coined words, or proper nouns for example, by using a set of grapheme-to-phoneme rules that govern how a certain combination of characters should be pronounced.

The output of the phonetisation tool in this case would be

```
hh iy ax s uw m d ax th uh r t iy th aw z ax n d d ao l er l ow n ax
t f ih f ti y n ax n d ax kw a r t er p er s eh n t ax n d p ah t ax
d aw n pe y m ax n t ao n ax l ih t l t uw b eh d r uw m p l ey s
```

When the text is presented in this way, it conveys no information about stress, timing, or even where a word begins and ends. This makes it very difficult to create passable synthesised speech, whether that involves finding an accurate match for the units in a recorded speech database, or creating the acoustic segments from scratch. Therefore, an important part of the text processing module of the synthesiser is marking up the text so it is chunked into words and phrases, and each phoneme is annotated with information about stress, position in the word, phrase, or syllable and right and left phonetic contexts.

An annotated input segment may look like this:

```
/ax/ =

left = /iy/
right = /s/
stressed = NO
word = 1
phrase = 3
```
The result is a target utterance that represents the inputted text as a string of machine-readable text, that also has associated prosodic and contextual information.

2.1.2 Back-end processing

The synthesis part of text-to-speech is converting this string of marked-up text, the target utterance, into physical sound. Many methods have been devised for achieving this end, including parametric synthesis and concatenative synthesis.

2.1.2.1 Parametric synthesis

Parametric speech synthesis employs a source-filter theory of speech production (Fant 1960) to model the voice source and the vocal tract filter to produce speech. The source filter theory of speech production, devised by Gunnar Fant (1960) at KTH Stockholm, views speech as the output of a source signal that has been passed through a linear time-invariant (LTI) filter. By parameterising the vocal tract filter, we can manipulate certain parts of it to create new sounds.

Articulatory synthesis The speech waveform can be viewed as the result of an input signal (the movements of the vocal folds) that has been sent through a filter (the vocal tract). If the dimensions of the vocal tract filter can be replicated, and the form of the input waveform can be accurately estimated, then it should be theoretically trivial to recreate perfectly human sounding speech. Articulatory synthesis attempts to do this by modelling the human articulatory system as accurately as possible by simulating the resonance effects of the vocal tract. A model of the vocal tract can be created by dividing it into sections of differing cross-sectional area, depending on the impedance that is represented by each (Fant 1960). The vocal tract model can be controlled by changing different dimensions of the model, the articulatory controls which control the dimensions of the vocal tract filter, and the excitation parameters, which control the characteristics of the input signal. The articulatory control
parameters may be lip aperture and protrusion, tongue height, tongue tip height and position, while the excitation parameters may be glottal aperture, vocal cord tension and lung pressure (Kröger 1993).

Data describing the operation of the human articulatory system is usually retrieved from x-ray images of the articulators during speech, but it tends to be deficient, as x-rays only provide a 2-d image of a 3-d vocal tract, and it is very hard to accurately measure the positions and movements of the articulators from the x-ray images. The tongue's movements are especially complicated, making it almost impossible to model accurately.

Due to the difficulty of the procedures involved in articulatory synthesis, and the relatively large computational load it entails, it has received less attention than other methods of synthesis, and has therefore not achieved the same levels of success.

**Formant synthesis** During the second half of the 20th century formant synthesis became the most widely used form of synthesis. Formant synthesis effectively attempts to model the vocal tract filter using only formant frequencies and amplitudes. The formant characteristics must be therefore artificially reconstructed for each sound. It is based on the source-filter theory of speech production (Fant 1960), which basically describes speech as the output of one or more sound sources which have passed through a linear filter. By modelling the properties of this linear filter, the vocal tract, using resonators, the result is a similar waveform to that of speech. These resonators are excited by a periodic source to simulate voiced sounds and a noise source to simulate voiceless sounds. Anti-resonators are used in conjunction with these to simulate the effects of nasality and frication.

The resonators can be connected in cascade, where the output of each formant resonator becomes the input of the next, or in parallel where the outputs of the three resonators are added together at the end. Cascade formant synthesisers have the advantage of needing only the formant frequencies and bandwidths as control information, so they are simpler to implement, and are therefore more effective for producing non-nasal voiced sounds (Lemmetty 1999).
The resonators in a parallel formant synthesiser are all excited simultaneously and the outputs from each are summed. A parallel structure permits additional gain control for each formant, which is good for the production of stops, nasals and fricatives.

A detailed mathematical description of parallel and cascade formant resonators is given in (Flanagan 1972). See also Holmes (1983) for more advanced parallel formant models.

**HMM-based synthesis** Hidden-Markov model-based (HMM) speech synthesis (Black and Toduka 2005) is a method of statistical parametric synthesis, which uses extracted parameters of recorded speech to create a model of each phoneme and during synthesis, regenerates speech parameters based on that model. HMM-based synthesis has two parts
2.1. **Text-to-Speech**

During training, an annotated speech database is examined and speech parameters are extracted for every frame until an average model of each phoneme is calculated, which describes how the speech parameters vary over a number of frames. This is expressed as a HMM. In order to synthesise input text, HMMs are retrieved for each contextualised phoneme in the target utterance, and speech parameters are extracted, which are then re-synthesised using a pulse train and an MLSA filter (Imai 1983). The output has a characteristically buzzy quality, but is reliable and intelligible. This method of speech synthesis is particularly important as it allows the creation of many voices on just a small amount of training data. Furthermore, speaker adaptation, voice disguise and manipulation, and expressive speech output are achievable using this method of synthesis to an extent that is not possible using other methods. As a result HMM-based speech synthesis is growing in popularity, even though it is not as natural sounding as concatenative synthesis methods.

2.1.2.2 **Concatenative synthesis**

Concatenative synthesis involves the stringing together of prerecorded units of speech to produce the synthetic speech output. It is based on the principle that only a finite number of speech sounds exist – by recording them all it should be possible to generate an infinite number of words. Obviously the larger the database of recorded speech, the more natural the resulting synthesis will sound. However, a larger database requires increased computational power and so a balance needs to be reached between realistic-sounding speech and the size of the database of recorded sounds. Advances in computational power and reductions in memory cost have seen the previous popularity of formant synthesis overshadowed by concatenative techniques.

**Diphone synthesis**  Concatenative synthesis using diphones is a common approach to achieving a balance between intelligibility and database size. Given a pair of phones, i.e. two adjacent phonemes or speech sounds, a diphone can be identified as starting halfway into the first phoneme and ending halfway through the second (Bohlenius 2005). For example the word *cat* /kæt/ contains two diphones; one begins at a steady state halfway through
the phoneme /k/ and ends in a steady state halfway through /æ/, and the other picks up at this point and ends half way through the phoneme /t/. In addition, there are two extra diphones at the start and end of the utterance – one links a silent phoneme and the first half of the phoneme /k/, while the other links the second half of the phoneme /t/ with a silent phone.

In any language, each phoneme can theoretically occur in conjunction with every other phoneme, including the ‘silence’ phoneme. For example, English has about 48 phonemes (including silence) depending on the dialect, and the square of the number of phonemes will give the number of possible diphones. Of course, some combinations of phonemes never occur in a language – in Hiberno-English, for example, the phonemes /ŋ/ and /χ/ only occur at the ends of words and will therefore never occur adjacent to one another. Omitting this, and other such combinations of phonemes, will contribute to a more efficient synthesiser.

Unit selection synthesis Over the last two decades, unit selection synthesis has emerged as the technique most capable of producing natural-sounding synthesised speech. The technique involves recording a database of naturally occurring sentences, from the public media for example, and extracting units for the synthesis of novel utterances (Clark, Richmond and King 2007). The size of the units can be phones, half phones, or diphones, or even a combination, and they are labelled in the database either automatically or by hand. Since a number of factors affect the pronunciation of each unit, its position in a word syllable or phrase for example, as well as the context in which it appears, it is desirable for the database to contain as many examples of each phoneme in the language in as many contexts as possible.

In order to synthesise a novel utterance, the synthesiser will first construct a target utterance based on text input which is a linguistic representation of the sentence to be synthesised, and which includes information about the phoneme sequence, accent placement, and syllable boundaries. It then conducts a search to select the most suitable sequence of units to use for each part of the target utterance. This is achieved by use of a cost function, which assesses the suitability of a unit sequence it considers using for the synthesised utterance.
2.2. Unit Selection Speech Synthesis

The target cost is assigned based on the position of the unit in a word, syllable or phrase and the units that appear on the right and left hand side. The join cost is assigned based on how well two consecutive units join together and is computed using only acoustic parameters such as spectral information (Clark et al. 2007). The synthesiser will select the unit sequence from the list of candidates found in the recorded database that has the lowest target and join cost.

The increase in the computational power of modern systems has greatly aided the advance of unit selection synthesis, as it is now possible to search through large databases of speech at runtime, giving the unit selection synthesiser more options when choosing the most suitable candidate units for synthesising an utterance. The unit selection method is discussed in greater detail in the following section.

2.2  Unit Selection Speech Synthesis

Unit selection synthesis is one of the most popular methods of speech synthesis used today. The process involves recording a large single-speaker database, and given a target utterance as input, searching for units in the database and joining these recorded waveforms together to create synthesised speech. The results of this method can be natural sounding and, since large chunks of sound files tend to be lifted from the database, the resulting speech can be indistinguishable from that of the recorded speaker. This section reviews the basic concepts behind unit selection synthesis, including the calculation of target and join costs, and the Viterbi algorithm.

Unit selection is a data-driven method of speech synthesis, that uses a large corpus of recorded speech as a database. The basic concept behind unit selection is the concatenation of speech units that are present in the database. The basic speech unit can be a phone, demi-phone, diphone, or any other speech segment. The main advantage of using diphones, however, is that the concatenation points occur in the middle of the phoneme and coarticulatory effects that occur during the transition from one phoneme to the next are cap-
2. Unit Selection Speech Synthesis

tured. Naturally, the larger (or more efficiently designed) the corpus, the better the potential quality of the synthesised speech, as there will exist more combinations of the contextualised units in the database.

The unit selection procedure can be viewed as a database search problem. The input of the system is a string of text, the target utterance, which is described in a machine-readable phonetic code, marked-up with information about stress, position in syllable or phrase, and left and right phoneme context. The sequence of candidate units that best matches this target utterance is then located in the database of recorded speech. The concatenated waveform is then returned as the audio output.

The quality of the synthesis therefore depends largely on how well the corpus is annotated. If the units are annotated using not just phoneme and stress identification, but also tagged with information about position in phrase or syllable, for example, then it is more likely that suitable units will be chosen to synthesise the target utterance specified. In a basic system, every unit in the database could be considered a candidate for the segment to be synthesised. A target cost is applied to each unit, penalising for contextual differences between it and the target segment. This method is computationally expensive however, and normally some kind of pre-selection is applied to the database of units before a revised list of candidates is chosen. The most obvious pre-selection strategy is to only pick units of the same phonetic description, which works very well as long as at least one example of each base unit exists in the corpus. This will depend on the manner in which the database units are annotated, and there exists a trade-off between the level of contextualisation of a phoneme, and the quantity of unique units in the database. The pre-selection of too many parameters can result in having no candidates for a segment.

Once a candidate sequence of phonemes has been identified, it must be determined whether the adjoining units are acoustically similar so that they can be concatenated without noticeable auditory artefacts. This is achieved by taking acoustic measurements of each unit in the database at the start and end of the recorded segment. Usually these acoustic measurements are \( f_0 \) and amplitude readings at the join points, and a numerical representation, for example, a vector 12 MFCCs from either edge of the segment. These measurements
are usually taken in advance for each unit, stored as a feature vector and used at runtime to calculate the join cost of a string of candidate segments. The difference between the segments is measured by calculating the Euclidean distance (or similar) between the vectors of parameter values. At this point in the search problem, there exists a number of candidates for each segment of the target utterance, each with a target cost associated with it. Each candidate can potentially join with any of the adjacent candidates to form a synthesised utterance. Calculating the join cost provides an objective measure of how well a sequence of units will join together at segment boundaries. A join cost is therefore calculated for every possible combination of candidate units.

Figure 2.3 shows the candidate selection process. The box at the top of the diagram represents the database and shows some of the candidate units available for selection. They are depicted as waveform segments. In the diagram, three of these waveform segments are being considered for concatenation. They are displayed in sequence towards the bottom of the picture, and the red lines connecting them to the collection of available candidates denotes the target cost, or the penalties incurred by each candidate for the differences they that exist between them and the units that comprise the target utterance. The grey lines pointing to other candidates denote segments which are not being considered at this time, possibly because the target cost is too high. Blue arrows connecting the sequence of candidate segments denote the concatenation cost, generally referred to in this thesis as the join cost. The weighted combination of target and join cost for a candidate sequence will determine whether or not it is selected as synthetic speech output.

Finally the join cost of each sequence of candidates is added to the target cost for each constituent segment and a Viterbi algorithm is employed to find the sequence with the lowest overall cost. The chosen sequence of candidates are then retrieved from the database and placed in an audio file, which is played back to the user as the output speech.

In this section, we provide a mathematical description of the target and join costs, the weighting system and the Viterbi algorithm.
This general approach to unit selection was proposed in a paper by Andrew Hunt and Alan W. Black (1996), and is now referred to as the Hunt and Black algorithm. Although other variations on the unit selection problem exist, it is this approach that is adopted in Festival (The Festival Speech Synthesis System 2007) and many other synthesis systems, and will be used as the framework for explaining unit selection synthesis in this chapter. The unit selection process is dependent on a cost function, which performs the candidate selection procedure. The units in a database have been described by Hunt and Black as:

"a state transition network in which the state occupancy cost is the distance between a database unit and a target, and the transition cost is an estimate of the quality of concatenation of two consecutive units."
The term cost function generally refers to the selection of the 'best' (as per some criteria) element from a set of alternative choices. The term 'best' usually refers to the most suitable element that costs the least to select, that is, the element of the best value. In unit selection, any number of choices of candidate units may exist in a database, but the ones of the best value are the ones which satisfy two criteria:

- similarity between the candidate and target units (target cost)
- similarity between consecutive candidate units (join cost)

These criteria are further delineated into sub-components. In unit selection speech synthesis, the term cost function refers to the weighted sum of the acoustic, contextual and linguistic aspects measured by target and join sub-costs. A Viterbi algorithm is generally used to minimise the cost function by calculating the sequence of candidates with the lowest cost. Once the units have been chosen, the corresponding waveforms are lifted from the database and concatenated to make the synthetic speech output. In this section, we provide a mathematical description of the unit selection cost function as described by Hunt and Black (1996).

### 2.2.1.1 Target cost

A unit selection database should contain many examples of each unit in a variety of different contexts. A unit can be a phone, half-phone, or even a word. Assuming the base unit is the diphone, a basic unit selection database would require an example of each diphone in as many contextual settings as possible, for example, in both a stressed and an unstressed position. If the position of the diphone in the phrase was seen to be important, then each stressed and unstressed diphone example would also have to be represented at the start, middle and end of a phrase. When left and right diphone contexts are taken into account, as well as position in the word and syllable, it becomes a difficult task to design a corpus where every unit is represented in every context. The more contexts that are specified, the harder it is to find an example of every unit in every context and the more likely it is for
the unit selection engine to fail to find the exact segment that has been specified by the target utterance. The best it can achieve is the closest match present in the database and it finds this match by assigning a target cost to each candidate unit, based on its contextual similarity to the target unit, that is the unit in the utterance that is to be synthesised. The target cost is therefore a measure of how close a unit in the database is to the target unit, and is calculated by awarding a penalty to the candidate unit for every difference it displays from the target unit.

Each segment of the database is identifiable by a feature vector which contains information about the phonetic and prosodic content of the segment. The target cost is a measure of the difference between a target segment and a candidate segment and is calculated by comparing their feature vectors. For each element in the feature vector, a zero cost is incurred if the segments are the same and higher costs are incurred if they differ. The target cost is therefore the weighted sum of the differences between the feature vectors of the target and candidate segments (Hunt and Black 1996) and is given by:

$$C^t(t_i, u_i) = \sum_{p=1}^{P} w^t_p C^t_p(t_i, u_i),$$  \hspace{1cm} (2.1)

where $C^t(t_i, u_i)$ is the overall target cost for a sequence of candidates, $t_i$ and $u_i$, respectively, are the target and candidate units at position $i$ in the candidate sequence, $C^t_p(t_i, u_i)$ is the cost for each target sub-cost $p$, of a total of $P$ sub-costs, and $w^t_p$ is the weight associated with sub-cost $p$.

The sub-costs used to calculate target cost can be any feature of the speech units (as long as the measurement is available for both the target and candidate segments) and are usually some combination of the contextual features of the segment. A metric is used to numerically represent the difference between the target and candidate units for each sub-cost.
2.2. UNIT SELECTION SPEECH SYNTHESIS

2.2.1.2 Join cost

Ideally, join cost should be a measurement of the perceptual quality of the concatenation of two consecutive units in a synthesised sentence. The lower the join cost, the harder it is to perceive the existence of a join between the two segments. Factors that would make the join seem obvious would be an notable fluctuation of $f_0$ or amplitude, or a significant difference between the spectral qualities of each segment. Join cost is generally calculated by summing the weighted contribution of these acoustics measurements ($f_0$, amplitude and spectral mismatch). The weighted costs incurred by the differences found between each of the three components all contribute to the final join cost.

The difference between the $f_0$ and amplitude measurements on either side of the join are found by simply taking the measurements and comparing them, and assigning a cost that reflects the size of the difference. The significance of a spectral mismatch however, is much more difficult to ascertain, and it is this component of join cost that will be the focus of much of the thesis.

The join cost is calculated by measuring the distance between the feature vectors on either side of a concatenation point. The join cost between one unit and the next $C^j(u_{i-1}, u_i)$ is calculated by summing the sub-costs which are each multiplied by a weighting factor. This process is formalised in Equation 2.2 below:

$$C^j(u_{i-1}, u_i) = \sum_{q=1}^{Q} w_q^j C_q^j(u_{i-1}, u_i),$$

(2.2)

where $C^j(u_{i-1}, u_i)$ is the join cost for a sequence of candidates, $u_{i-1}$ and $u_i$ denote two adjacent units in a sequence, $C_q^j(u_{i-1}, u_i)$ is the cost for each join sub-cost $q$, of a total of $Q$ sub-costs, $w_q^j$ is the weight associated with sub-cost $q$.

The join cost is calculated for each possible combination of units that have been selected as candidates for each unit in the target utterance.
### Feature Weight Description

<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase</td>
<td>15</td>
<td>Position in phrase is correct (initial, medial or final)</td>
</tr>
<tr>
<td>Stress</td>
<td>10</td>
<td>Stress is correct</td>
</tr>
<tr>
<td>POS</td>
<td>6</td>
<td>Part of speech is correct (noun, verb, modifier, function word, other)</td>
</tr>
<tr>
<td>Syllable</td>
<td>5</td>
<td>Position in syllable is correct (initial, medial, final, between words)</td>
</tr>
<tr>
<td>Word</td>
<td>5</td>
<td>Position in word is correct (initial, medial, final, between words)</td>
</tr>
<tr>
<td>Left</td>
<td>4</td>
<td>Left phonetic context matches</td>
</tr>
<tr>
<td>Right</td>
<td>3</td>
<td>Right phonetic context matches</td>
</tr>
</tbody>
</table>

Table 2.1: Component cost functions for the default multisyn target cost from Clark et al. (2007)

#### 2.2.1.3 The weighting system

The target cost and the join cost are then multiplied by a weighting configuration, which prioritise some sub-costs over others by penalising mismatches in those sub-costs more heavily than others. The sub-costs and associated weights used in Festival’s multisyn unit selection algorithm are given in Table 2.1. The relative importance of each sub-cost can be controlled by increasing the weights associated with them. In the multisyn example, stress and phrase position are deemed the most important characteristics and candidate units are penalised more heavily if they do not match the target in these respects. To calculate the target cost for a unit, a penalty is awarded to the candidate unit for each difference it displays from the target. This penalty is multiplied by the weight associated with the sub-cost and the penalties for each sub-cost are added together to give the final target cost.

#### 2.2.2 Viterbi algorithm

The candidate units are now represented by a matrix or trellis, where each column represents a target segment and contains candidate units for that particular segment. In order to calculate the cost function of any combination of candidate units for each segment, the
target cost and the join cost are added together for each path in the trellis. The total cost of selecting a candidate sequence $U$ to represent a target utterance $T$ is given by $C(T, U)$ in the following equation:

$$C(T, U) = C_{(t, u_t)} + C_{(u_{t-1}, u_t)} = \sum_{p=1}^{P} w_p C_p(t, u_t) + \sum_{q=1}^{Q} w_q C_q(u_{t-1}, u_t)$$  \hspace{1cm} (2.3)$$

A dynamic programming algorithm, the Viterbi algorithm, is usually employed at this point to calculate the least expensive route through the matrix, as shown in Figure 2.4.

The problem can be visualised as in Figure 2.4 where $T$ is the number of observations, or segments in the target utterance, and $N$ is the number of states, or candidate units for each segment. The observations $T$ are organised horizontally so that the entire target utterance can be read from left to right. In this case the target utterance is Klatt, given by the sequence of diphones, or observations /k-l/ /l-æ/ /æ-t/. Each observation slot $t$ is populated with nodes that can correspond to one of $N$ states. These states are the number of candidates and the nodes are instances of that phoneme in the database. In this example, there are three instances of each diphone, but in reality there may be a different number of
candidates for each unit. Each candidate, or node, has a target cost associated with it, and also has a join cost associated with it and every node in the following observation slot. We want to select one candidate for every observation $t$ such that the sum of the target costs for each candidate selected plus the join cost between that candidate and the next is the lowest possible value for the whole utterance. This is calculated by starting at $t$ and moving horizontally to $t+1$ and calculating the cost of the transition for every possible combination in the trellis. The Viterbi algorithm then works backwards to calculate the optimum path through the trellis, i.e. the one with the lowest cost.

Mathematically, a Viterbi algorithm minimises the output as a single utterance $U$ such that:

$$
\hat{U} = \arg\min_u \sum_{p=1}^{P} w_p^{'} C_p^i(t_i,u_i) + \sum_{q=1}^{Q} w_q^{'} C_q^j(u_{i-1},u_i)
$$

(2.4)

The final sequence of candidate units is selected by the Viterbi algorithm and these units is then extracted from the recorded database and played in sequence as the synthesised speech output of the unit selection system. The segments can be directly concatenated, or some signal processing techniques can be employed across the join point to reduce any audible effects due to the discontinuity at the join, such as clicks, that may result from the concatenation.

### 2.3 Speech Parameters and Distance Measures

In this thesis, the terms *numerical representation*, *speech parameterisation* and *acoustic feature* are used to signify the acoustic representation of the speech sound. The terms *distance measure* and *metric* are used to describe the measurement of distance between the coefficients representing the acoustic signal. The term *objective spectral discontinuity measure* and similar variations are used to describe the discontinuity score calculated by using a distance measurement to find the distance between acoustic parameterisations.

In order to compare two consecutive speech units, a measure must be taken of each unit,
on either side of the join, and compared. This is effectively an arbitrary measurement of some acoustic property of the segment. Ideally, a large difference between these measurements will reflect a large perceptual difference between the sounds, so the ideal join cost speech parameter is one in which changes of its value represent equivalent changes in perception. Consider the segments /k-æ/ and /æ-t/. Calculating the spectral discontinuity between these two segments requires that the spectral characteristics of each be quantified by numerically coding the sounds as vectors of acoustic measurements. More specifically, the numeric representation is taken only on the portions of each sound that will be coming into contact, i.e. at the end of the first segment /k-æ/ and at the start of the second segment /æ-t/. The speech sounds, therefore, are windowed before numerically coding and processing the signals. The Euclidean (or similar) distance between the vectors of acoustic measurements that represent the sounds can then be calculated in order to express the perceptual difference between the sounds.

The acoustic measurement used to represent the sound must therefore be one for which perceived changes in the sound are accurately mirrored by numerical changes in the acoustic representation. The acoustic measurement, along with others such as $f_0$ and amplitude are calculated offline for the start and the end of each unit. The figures are stored in a database as a feature vector and the join cost for each pair of units is calculated at run time by measuring the distance between the feature vector at the end of the first unit with the feature vector at the start of the adjacent unit. Many studies have been conducted in an attempt to ascertain which of the many proposed numerical representations best captures the perceptually important characteristics of the speech signal.

### 2.3.1 Literature review

This section gives a review of some previous studies investigating spectral distance measures. The studies generally compare a selection of speech parameters and distance measures in order to find the representation that best predicts human perception of spectral discontinuity.
2.3.1.1 Overview of previous studies

Early studies  Research publications in this area began to appear in the 1970s. Gray and Markel (1976) conducted a theoretical and experimental study of the properties and relationships of distance measures between cepstral coefficients in speech signals. The distance measures compared were (i) the $\cosh$ distance, (ii) the cepstral distance, (iii) the likelihood ratio, and (iv) the RMS log spectral distance, which was used as a benchmark distance measure. It was shown that only the cepstral measure and the $\cosh$ measure came close to the distances predicted by their benchmark measure (iv). Nocerino, Soong, Rabiner and Klatt (1985) compared a number of spectral distance measures including the likelihood ratio and the cepstral distortion measure. For each measure they also applied Bark-scale frequency warping but they found that perceptually warped results showed no improvement. In their 1988 book Objective Measures of Speech Quality (Quackenbush, Barnwell III and Clements 1988), Quackenbush et al. compare a number of speech coding techniques. They report a correlation of 0.7 between the best automatic measures and perceptual scores.
2.3. **Speech Parameters and Distance Measures**

**Wouters and Macon**  A study by Wouters and Macon (1998) measured the subtle perceptual differences between allophonic speech segments. The database consisted of 166 words containing a vowel midsection. Subjects listened to an original utterance of a word and to a variation of the word, which was constructed by replacing the vowel midsection with another example of the vowel taken from a different phonetic context. In this study, the speech signals were sampled at 16 kHz and the parameters were extracted using a window of length 5 ms. The acoustic distances investigated were FFT-based spectra, LPC-based cepstra, line spectral frequencies, log area ratios and the Itakura distance. Each of these parameterisations were computed using the FFT-based amplitude spectrum, the PLP spectrum as described by Hermansky (1990) and a mel-warped spectrum. The distance measures used were the Mahalanobis distance, the weighted Euclidean distance, and the Itakura distance. Unlike previous studies, the best results in this study were given by perceptually warped spectra, and the mel-based distances slightly outperformed PLP-based results. Overall, mel-based cepstral distances and the symmetrised Itakura distances were found to be the most discriminating, the highest correlation with perceptual data being $p = 0.66$.

**Klabbers and Veldhuis**  Esther Klabbers and Raymond Veldhuis carried out a similar experiment on five Dutch vowels (Klabbers and Veldhuis 1998). The listening task they designed involved speech scientists making a binary decision about whether the diphone transition was continuous or discontinuous. The acoustic measurements in this case were the Euclidean distance between first and second formant ($F_1$, $F_2$) pairs, the Kullback-Leibler measure, the Euclidean distance between MFCC vectors, the likelihood ratio (LR), the mean-squared log-spectral distance (MS LSD), the loudness difference (LD), and the excitation difference (ED). In this case all measures were computed using a 40 ms window. The Kullback-Leibler distance was found to give the best measure in all vowel conditions. The authors suggested that joins occurring in consonants should be investigated in the future (Klabbers 2000, Klabbers and Veldhuis 2001).

**Chen and Campbell**  In 1999, Jing-Dong Chen and Nick Campbell (1999) compared several different acoustic transforms of the speech signal in order to assess what would be the
most reliable objective distance measure to use in the evaluation of speech synthesis systems. The distance measures investigated were LPC, LPCC, LSP, MFCC, residual MFCC (RM-MFCC), bispectrum, modified Mellin transform of the log spectrum (MMTLS), segmental MMTLS (SMMTLS), and the Wigner-Ville distribution (WVD) based cepstrum. For each measure, the distance was computed between a synthesised and a naturally spoken sentence. In order to calculate the distance between the natural sentence and the synthesised sentence they were split up into frames and the durations of the frames were made equal using dynamic time warping. The sentences were sampled at 16 kHz and the features were extracted from time frames of 20 ms along the signal. Each frame was then converted to 12 feature coefficients. The distances between feature vectors was calculated to obtain a discontinuity score. This score was then compared by correlation with perceptually-based scores obtained from a listening task. Overall the bi-spectrum performed best. The WVD performed poorly, but one possible reason given by the authors was that there was perhaps no optimal way to convert the Wigner-Ville distribution into parameter vectors. Importantly the authors did not consider the effect of sampling on the Wigner distribution, which needs to be considered very carefully (Healy, Rhodes and Sheridan 2010). The RM-MFCC measure outperformed all but the bi-spectrum. The RM-MFCC measure is the MFCC vector computed from the linear predictive residual signal.

Stylianou and Syrdal The perceptual experiment conducted by Stylianou and Syrdal (2001) also investigated detection rate of discontinuities. The test stimuli were groups of six words taken from the Modified Rhyme Test (MRT) (House, Williams, Hecker and Kryter 1963), which differ only by the first (e.g. book, took, look, cook, rook, nook) or the last consonant (e.g. teak, team, teal, teach, tear, tease). By recombining the start of every word with the end of every other word in the category, the authors were able to synthesise 36 test words per group and ask listeners to judge whether or not the test word contained a join. The distance measures compared in this study were the Euclidean distance between Log Power Spectra computed from FFT, LPC and PLP; the Euclidean distance between LSFs computed from LPC and PLP; the weighted Euclidean distance between cepstral coefficients computed from LPC and PLP; the Euclidean distance between MFCCs; the Kullback-Leibler
divergence between power spectra computed from FFT, LPC and PLP; and the the KL divergence between LSFs computed from LPC and PLP. The stimuli were all sampled at 16 kHz. The features were extracted by first taking a 40 ms window of speech at the concatenation point and then computing a 1024 point FFT. The highest rate of correlation with perceptual results was the Kullback-Leibler distance between FFT-based power spectra, followed by the well-established Euclidean distance between MFCCs. However, the authors point out that the leading distance measure was able to predict only 37% of audible concatenations, which cannot be considered high, and that further exploration of speech features should be carried out in an attempt to improve this figure (Syrdal 2001).

Donovan The study by Klabbers and Velhuis (1998) prompted further research at IBM, where it was concluded that a novel, context-dependent distance measure should outperform all those previously investigated. The new distance measure was described as a decision-tree-based context-dependent Mahalanobis distance between perceptual cepstral vectors (Donovan 2001). The database was clustered according to phonetic context using top-down decision trees and the novel distance measure was given by:

\[
D^2 = \sum_{i=1}^{n} \left[ \frac{e_i - s_i - \mu_i}{\sigma_i} \right]^2,
\]

where \( n \) is the dimensionality of the data, \( \mu_i \) is the \( i \)th element of the mean vector in leaf \( I \) of the decision tree, \( (\sigma_i^2) \) is the \( i \)th diagonal of the covariance matrix for leaf \( I \), and leaf \( I \) is the leaf reached by descending the decision tree for the context that the join is in. This distance measure was compared to the Euclidean and Mahalanobis distances between MFCCs, perceptually-based MFCCs, with and without log energy, delta and delta delta coefficients. Other measures were Euclidean distance between log FFT, the log Itakura-Saito distance, and the log Kullback-Leibler distance. In this study, all parameters were extracted from speech that was sampled at 11 kHz. Voiced speech was examined in 25 ms frames, and unvoiced speech was divided into 6 ms frames at a 3 ms frame rate. Results showed that the new distance measure was superior to other measures tested.
Vepa, King and Taylor While previous studies focussed on isolated words, Vepa and colleagues (Vepa et al. 2002, Vepa and King 2003, Vepa 2004, Vepa and King 2004, Vepa and King 2005) looked at joins occurring in polysyllabic words in natural sentences, and looked at a new parameterisation, multiple centroid analysis (MCA). Furthermore, these joins occurred in the middle of diphthongs, where their preliminary studies found that spectral discontinuity was more prominent. For the perceptual test, two natural sentences were selected for the five American English diphthongs. Participants rated the audible discontinuity on a scale of 1–5. The distance measures investigated were MFCCs, MCA, and LSF. The measures used were Euclidean, absolute, Kullback-Leibler, and Mahalanobis. They found that no distance measure outperformed any other, and the MCA parameterisation outperformed MFCCs and LSF in most cases, with the added benefit of having fewer parameters. Vepa also encouraged future investigation into phoneme class-specific joins.

Syrdal and Conkie A further study by Syrdal and Conkie (2004) used perceptual data on the detectability of mid-vowel joins to train linear regression (LR) and classification and regression trees (CART) to predict discontinuities. The study found that Euclidean cepstral distances were the superior predictor variables, and though both models achieved similar results in predicting human detection rates, the accuracy of CART model improved when it was trained using both acoustic variables and categorical phonetic predictor variables.

Bjørkan, Svendsen and Farner Bjørkan and colleagues (Bjørkan, Svendsen and Farner 2005) examine a concatenation in two Norwegian long vowels in short carrier sentences. The discontinuity measures they investigated were the symmetrical KL distance on LPC power spectra ($D_{KL}$), Euclidean distance between LPCC vectors ($D_{cep}$), mean likelihood ratio between LPC spectra ($D_{LR}$), the Euclidean distance between MFCCs ($D_{mfcc}$), and a modified pitch synchronous cross-correlation method ($D_{mpsc}$). It was found that $D_{LR}$, $D_{cep}$, and $D_{mfcc}$ performed equally well.
Kirkpatrick, O’Brien and Scaife  More recent studies (Kirkpatrick, O’Brien and Scaife 2006b, Kirkpatrick, O’Brien and Scaife 2006a) investigated feature sets derived from the phase information as well as the spectral information and found that the results were comparable as a join cost for spectral discontinuity. The features used were MFCCs computed from FFT and LPC spectra, power spectra and log power spectra computed from FFT, LPC and PLP, cepstral coefficients from PLP and LPC, LSFs computed from LPC on linear and Mel scales and from PLP coefficients. The distances used were KL, cos, Euclidean, and absolute. The drawback of using measures based on FFT is that both time and frequency resolution are constant across all frequency bands (Kirkpatrick et al. 2006b). To overcome this limitation, these measures were also derived using wavelet analysis, which allows more flexible time-frequency filtering. These measures were found to outperform all other measures. In addition, this study found that the results are sensitive to the window length used to extract the features and suggests that this may account for inconsistencies in previous studies. A further study (Kirkpatrick et al. 2007) investigated the role of spectral dynamics in predicting audible discontinuities. Results showed that measures representing spectral dynamics correlate well with human perception of discontinuity, and furthermore that the details in feature extraction impacted heavily on the correlation values, and different methods were needed for static and dynamic feature extraction.

Pantazis and Stylianou  Two new non-linear approaches for detecting discontinuities were introduced by Pantazis and colleagues (Pantazis et al. 2005, Pantazis 2006, Pantazis and Stylianou 2007). The first method was based on a harmonic model of speech and the second set was based on a technique where the speech signals were decomposed into amplitude and frequency components using the Teager energy operator. The two methods combined were shown to give good results with a detection rate of 56%.

The following section examines popular methods of attempting to correlate these measures with perceptual data.
<table>
<thead>
<tr>
<th>Study</th>
<th>Stimuli</th>
<th>Design</th>
<th>Correlation</th>
<th>Sampling Rate</th>
<th>Window Type</th>
<th>Window Length</th>
<th>Best Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gray and Markel 1976)</td>
<td>n/s</td>
<td>n/s</td>
<td>Pearson's r</td>
<td>n/s</td>
<td>n/s</td>
<td>n/s</td>
<td>RMS Log Spectral</td>
</tr>
<tr>
<td>(Nocerino, Soong, Rabiner</td>
<td>n/s</td>
<td>n/s</td>
<td>Pearson's r</td>
<td>n/s</td>
<td>n/s</td>
<td>Hamming</td>
<td>Log Likelihood ratio</td>
</tr>
<tr>
<td>and Klatt 1985)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Quackenbush et al. 1988)</td>
<td>n/s</td>
<td>n/s</td>
<td>Pearson's r</td>
<td>n/s</td>
<td>Hamming</td>
<td>n/s</td>
<td>MFCC</td>
</tr>
<tr>
<td>(Wouters and Macon 1998)</td>
<td>vowels</td>
<td>MOS</td>
<td>Pearson's r</td>
<td>16 kHz</td>
<td>Hanning</td>
<td>40 ms</td>
<td>KL</td>
</tr>
<tr>
<td>(Klabbers and Veldhuis 1998)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>16 kHz</td>
<td>n/s</td>
<td>20 ms</td>
<td>bispectrum</td>
</tr>
<tr>
<td>(Chen and Campbell 1999)</td>
<td>vowels</td>
<td>MOS</td>
<td>Pearson's r</td>
<td>16 kHz</td>
<td>n/s</td>
<td>40 ms</td>
<td>KL (FFT)</td>
</tr>
<tr>
<td>(Stylianou and Syrdal 2001)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>16 kHz</td>
<td>n/s</td>
<td>25 / 6 ms</td>
<td>perceptual cepstra</td>
</tr>
<tr>
<td>(Donovan 2001)</td>
<td>vowels</td>
<td>MOS</td>
<td>Pearson's r</td>
<td>11 kHz</td>
<td>n/s</td>
<td>25 / 6 ms</td>
<td>MCA</td>
</tr>
<tr>
<td>(Vepa et al. 2002)</td>
<td>diphthongs</td>
<td>MOS</td>
<td>Pearson's r</td>
<td>n/s</td>
<td>n/s</td>
<td>10 ms</td>
<td>Euclidean distance</td>
</tr>
<tr>
<td>(Syrdal and Conkie 2004)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>16 kHz</td>
<td>n/s</td>
<td>2 periods</td>
<td>MFCC</td>
</tr>
<tr>
<td>(Bjørkan et al. 2005)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>n/s</td>
<td>n/s</td>
<td>various</td>
<td>wavelet transform coeffs</td>
</tr>
<tr>
<td>(Kirkpatrick et al. 2006b)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>16 kHz</td>
<td>Hanning</td>
<td>various</td>
<td>AMFM/harmonic model</td>
</tr>
<tr>
<td>(Pantazis and Stylianou 2007)</td>
<td>vowels</td>
<td>Yes/No</td>
<td>ROC curve</td>
<td>16 kHz</td>
<td>Hamming</td>
<td>various</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Summary of perceptual test and feature extraction methods used in previous studies of join cost
2.3. Speech Parameters and Distance Measures

2.3.1.2 Perceptual tests

In order to investigate the perceptual salience of artificial joins, previous studies have mainly adopted one of two perceptual test approaches where listeners either decide whether or not a join is present in the data (binary decision) or rate how salient the join is on a scale of 1–5 (MOS test).

Test stimuli A number of different databases have been used to extract test stimuli in previous studies in order to gather perceptual data regarding the salience of discontinuities. A popular method employed by Stylianou and Syrdal (2001), and Kirkpatrick et al. (2007), among others has been to record a database using rhyming collections of words from the Modified Rhyme Test (MRT) (House et al. 1963), as described in Section 2.3.1. Other studies (Vepa and King 2005) used full sentences containing joins, while Klabbers and Veldhuis (2001), among others, used shorter stimuli. Recording such corpora for testing discontinuities is an obvious contribution, but one drawback is that it makes it difficult to compare the findings across speakers, as noted by Kirkpatrick (2010).

Relating perceptual results to objective measures Perceptual measures in previous studies were related to spectral distance measures using the methods described below.

- Correlation using Pearson’s $r$ Pearson’s product-moment measures the degree and direction of correlation of two data series. The correlation between $N$ measurements of pairs of values, $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, is given by:

$$r_{xy} = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N \sum x_i^2 - (\sum x_i)^2} \sqrt{N \sum y_i^2 - (\sum y_i)^2}}$$

The values of the correlation function lie between -1 and 1. A value of 1 indicates that the $x$ values are linearly proportional to the $y$ values, while a value of -1 indicates that the values are inversely proportional. A zero value means that the variables are uncorrelated.
Figure 2.6: Detection and false alarm rate for detecting discontinuities in a signal.

- **Receiver operator characteristic (ROC) curves** Receiver Operator Characteristic (ROC) curves are a method of evaluating the performance of discontinuity measures by comparing how well the measure predicts human perception of discontinuity. For all signals that contain a discontinuity, a detection rate $P_D$ signifies the number of times that both the user and the discontinuity measure spotted a discontinuity in the signal, while $P_{FA}$ is the number of times that the objective measure reported a discontinuity, where the user did not. For each discontinuity measure, $y$, two probability density functions $p(y|0)$ and $p(y|1)$ were constructed based on whether the listeners rated the signal as continuous (0) or discontinuous (1). The detection rate and false alarm measure are computed by:

$$P_D(y) = \int_y^\infty p(y|1)dy$$  \hspace{1cm} (2.7)

and

$$P_{FA}(y) = \int_y^\infty p(y|0)dy$$  \hspace{1cm} (2.8)

where $y$ defines the threshold. Normally the false alarm rate, $P_{FA}$ is set to a certain value, such as 0.05.
2.3. **Speech Parameters and Distance Measures**

![Image: Probability density functions for continuous and discontinuous test stimuli, from Pantazis (2006).](image)

Figure 2.7: Probability density functions for continuous and discontinuous test stimuli, from Pantazis (2006).

For every threshold value γ, a plot of \( P_d \) vs \( P_{fa} \) values constitutes an ROC curve. A straight line ROC curve indicates that the objective measure predicts a listener's perception of discontinuity no better than chance. The higher the curve extends towards the top left of the graph, the higher the ratio of detection to false alarm, indicating the probability density functions are moving away from each other.

### 2.3.2 Discussion

This section detailed a literature review of the studies on spectral distance measures and the attempts to correlate them with human perception in order to measure their efficacy. The main points to bear in mind about the previous studies conducted in this area are:

1. There is no one numerical representation that has been shown to best represent speech signals.

2. The window length used to extract parameters varies from study to study and therefore
it is not possible to compare the results from different tests.

3. The tests have only been performed on vowels and other voiced speech sounds, can cannot be said to hold true for all types of speech signals.

The main conclusion that can be reached from examining the literature in this active research area is that none of studies agree on which parameterisation and distance measures best predict human perception of discontinuities. This is due, in large part, to the complexity of the task at hand. Undoubtedly the human auditory and processing system is extremely complicated, and any attempt to capture its essence using a numerical metric is going to be somewhat crude. Nonetheless, the literature review shows that a large array of complex signal processing techniques have been proposed to do just this. The results of these studies, however, are inconsistent, and the manner in which the different experiments have been conducted makes it difficult, if not impossible, to cross-compare the different studies.

This lack of consensus in the different studies can be partly explained by:

1. **The variation in some of the fundamental values used to calculate parameters.** The values comprising the numerical representations themselves are entirely dependent upon several factors bearing upon their calculation. Choices of these values will result in a different set of values for the numerical representation (Kirkpatrick et al. 2006b, Kirkpatrick et al. 2007, Kirkpatrick 2010, Kelly 2010, Kelly and Gobl 2011a, Kelly and Gobl 2011b). The possibility exists that the choice of values that yields a meaningful numerical representation may be a function of the signal, governed by the signal's characteristics, such as the $f_0$, the position of the formants, or the particular idiosyncrasies of the speaker.

2. **The differences in the databases used in the study.** The segments examined in the studies were generally extracted from a database designed and recorded especially for each study in question. Although some of the databases were shared between research groups, they were generally not freely available to the general academic community.
3. The reliance on perceptual data to rate the efficacy of the measures. Each study uses a different type of perceptual test. Furthermore, the manner by which the perceptual data was compared to the numerical calculations differed per study, with some opting for a Pearson's correlation coefficient, and other using ROC curves. Another point to note is the underlying assumption that one numerical representation will suit every type of speech signal, despite the tendency of the objective discontinuity measures to work better for some sounds than for others (Vepa and King 2005, Kelly 2010, Kelly and Gobl 2011a, Kelly and Gobl 2011b). Including some kind of phonetic information increases the ability of the objective discontinuity measure to predict audible joins. None of the reviewed studies have been performed on phonetic classes other than vowels and diphthongs, despite recommendations that future work should focus more on this area (Klabbers and Veldhuis 1998).

Like the previous studies outlined in this chapter, this study proposes a procedure for determining whether a numerical speech representation is effective. This study differs from these previous studies, however, in that it attempts to address the three shortcomings listed above by devising an objective method of assessing the suitability of numerical representations of speech signals. The framework can be tested using a freely available set of databases, so that the results can be compared across studies, and the system can be applied to any collection of signals. This will allow quick and automatic assessment of the suitability of numerical representations for any type of speech signal, and will allow investigation into differences owing to speaker, database, dialect and other variables. The remainder of this chapter outlines the experimental approach adopted in this thesis and justifies a series of choices made in an attempt to showcase the findings of the method.

2.4 Conclusion

The primary aim of this thesis is to examine the acoustic representation of speech signals in a manner that takes into account the issues raised above. By approaching the problem with these issues in mind, it was hoped that a repeatable scientific method could be established
for testing the efficacy of numerical speech representations and, consequently, join costs. In this section, we provide a review of the chapter, and discuss how the issues raised in the examination of the literature have influenced the design of the studies described in this thesis.

2.4.1 Chapter review

The chapter began with an introduction to text-to-speech systems, describing the front-end and back-end processes that create synthetic speech output. We then turned our focus to unit selection speech synthesis, and discussed the cost function as defined by Hunt and Black (1996). The components of the cost function were discussed, notably the target and join cost components, the weighting system, and the Viterbi algorithm that is generally used to find a minimum cost of the output candidate sequence. The following section focussed on a particular component of the join cost – the spectral distance component. Numerous studies were described in which measurements of spectral distance were assessed in order to find one which best modelled a listener's perception of similarities between signals.

2.4.2 Discussion

The particular drawbacks of the studies conducted in this area were outlined in the discussion of Section 2.3. In this thesis, we approach the problem of evaluating the efficacy of numerical speech representations while avoiding the inherent pitfalls encountered in the previous studies.

1. The variation in some of the fundamental values used to calculate parameters. We demonstrate the importance of carefully choosing parameters to calculate numerical representations. According to Kirkpatrick (Kirkpatrick 2010):

"The choice of window length was found to be the most significant source of variation in the results. The change in performance that resulted from
changing the window length was more significant than the difference in performance between feature sets."

With this in mind, we decided to investigate one speech representation, the MFCC, and demonstrate the effects of changing important parameters like window length.

2. The differences in the databases used in the study. So as to ensure that the database used in the study was widely and freely available, the Arctic database (Kominek and Black 2003) was used during this experiment, allowing us to compare speakers and dialects as well as different signal types.

3. The reliance on perceptual data to rate the efficacy of the measures. A measure of ranking numerical signal representations is designed so that the framework can be applied to any database without the need for perceptual evaluation.

2.4.3 Concluding remarks

The examination of previous studies in this area has allowed us to lay out a method for investigating the numerical representation of speech signals. The remainder of this thesis addresses this problem and describes its contribution to unit selection speech synthesis.
CHAPTER 3

Calculating MFCC Representations

The mel-frequency cepstral coefficient (MFCC) is widely used as a numerical representation of a signal in the fields of speech synthesis and recognition. The literature has shown great variation in the ways in which the MFCC is calculated. The variation in the calculation methods, we believe, are at least partly to blame for the lack of agreement between studies investigating what the best numerical representation of speech signals might be. In this chapter, we investigate some of the parameters that affect the calculation of the MFCC, particularly:

(i) the window length used to examine the speech segments,

(ii) the time-frequency analysis performed on the signal, and

(iii) the shape of the filters in the mel filter bank.

The MFCC calculation method is described in Section 3.1, and the factors affecting calculation are discussed, with particular emphasis on (i) – (iii). In Section 3.2, we examine theoretically how choice of these values (i) – (iii) actually affect a signal, and how salient signal information can be obscured or highlighted as a direct consequence of such choices. In Section 3.3, the effects that these factors have on MFCC calculation are demonstrated.
3.1 Mel-frequency Cepstral Coefficients (MFCC)

The MFCC (Davis and Mermelstein 1980) is a signal representation that has proved to be very useful in speech synthesis and recognition. As the name suggests, the calculation of these coefficients involves representing a signal in the cepstral domain, with frequencies on the mel scale. A signal is converted to the cepstral domain by calculating the Cosine Transform (CT) of log of the square of the signal's magnitude spectrum. The frequencies are converted from the linear Hertz scale to the non-linear mel scale by multiplying the spectrum of the signal with a bank of filters that are scaled according to mel frequencies. In this section, the concept of the cepstrum and the mel scale are described, and a derivation of the MFCC calculation process is presented. Finally, the parameter choices inherent in MFCC calculation are discussed, particularly (i) – (iii), which we have chosen to examine in this study.

3.1.1 The mel scale

The mel scale is a perceptual scale of pitches on which equal distances correspond to what humans perceive as equal jumps in pitch. Due to the physiology of the human auditory system, larger and larger intervals between frequencies above about 500 Hz appear to correspond to the same pitch increment. Frequency values in Hertz \((f_{Hz})\) can be converted to mel frequencies \((f_{mel})\) using Equation 3.1 (O'Shaughnessy 1987):

\[
f_{mel} = 2595 \log_{10} \left(1 + \frac{f_{Hz}}{700}\right).
\] (3.1)
This relationship is illustrated in Figure 3.1. In order to express a frequency domain representation of a signal using mel frequencies, the signal representation must be multiplied by a series of filters that are spaced according to the mel scale. Figure 3.2 shows the logarithm of a power spectrum with frequencies in Hz. In (a), the signal undergoes filter bank analysis with evenly spaced filters, and in (b), the signal is multiplied by filters spaced according to the mel scale.

The use of the mel scale can be very valuable when dealing with concatenation costs because the quality of the join is very much dependent on the perception of the listener. As the mel scale was devised to mimic the frequency response of the human ear, it makes the MFCC a very attractive choice as a numerical representation of a speech signal.
3.1. Mel-frequency Cepstral Coefficients (MFCC)

3.1.2 The cepstral domain

Viewing a signal in different domains, such as the time domain, the frequency domain, or the cepstral domain, allows certain properties of the signal to be examined more easily. A time domain representation of a signal can be converted to an amplitude spectrum in the frequency domain by employing the Fourier Transform (FT). The spectrum of the waveform shows the relative contributions of the constituent frequencies of the signal. With some further manipulation, the signal can be converted into the cepstral domain, which allows other properties of the signal to be displayed.

Generally, the first step in converting a signal to the cepstral domain is the calculation of the spectrum of the waveform. The relative amplitudes of the constituent frequencies of the signal are thus displayed. Usually lower frequency components in speech signals are stronger than the higher ones, and so by taking the log of these amplitudes, it becomes easier to see the relative differences between the amplitudes, as shown in Figure 3.3. Note that taking the magnitude spectrum of a signal removes all phase information. This information is thrown away, although some studies (Kirkpatrick 2010) indicate that it maybe helpful in improving join cost estimates.

The calculation of the cepstrum then requires that the CT is taken of the log magnitude spectrum. This notion of taking the spectrum of the spectrum gives rise to the name ‘cepstrum’, which is derived by reversing the first four letters of the word ‘spectrum’. In order to do this, the log magnitude spectrum must be viewed as if it were a waveform in the time domain (Figure 3.3 (c)). The constituent frequencies of such a waveform could be roughly estimated.
Figure 3.3: The cepstrum of a waveform

at a glance. As a ‘time domain waveform’, it appears to have a weak high-frequency component (given by the sharp ups and downs of the harmonics), and a stronger low-frequency component that reduces the overall amplitude of the waveform over time, and gives the impression of governing the shape of the spectral envelope (Taylor 2009). The CT of the signal is therefore taken in order to see the exact contributions of the frequencies, and this is given in Figure 3.3 (d). The spike in the cepstrum at 100 ms represents the fundamental frequency of the original waveform, and the distance between the small harmonic spikes in the log magnitude spectrum. The overall time-evolving shape of the spectral envelope is represented by the spikes at the lower frequencies. We can see that a benefit of representing a signal in the cepstral domain is that the signal is displayed in such a way that an approximation of the contributions of the voice source and the vocal tract filter are displayed in separate positions along the x-axis. The ‘contribution of the source’ is represented by
the spike that occurs at the pitch of the fundamental frequency, while the 'contribution of the filter' is the frequencies that govern the shape of the spectral envelope. In reality, the separation of source and filter is a lot more complex than simply representing a signal in the cepstral domain, but doing so serves as a useable approximation for our purposes – a comprehensive study of voice source can be found in Gobl (2003).

### 3.1.3 Calculating MFCCs

A signal can be represented numerically by a vector of MFCC values, which are derived from the mel-scaled signal in the cepstral domain. The MFCC calculation process is described in this section. Consider a Hanning windowed segment of a sampled speech signal \( \{y[n]\}_{n=1}^N \). The signal is converted to the frequency domain using an \( M_s \) point DFT given by:

\[
F(k) = \sum_{n=1}^{M_s} y(n) e^{-\frac{2\pi ik}{M_s}}, \tag{3.2}
\]

where \( 0 \leq k \leq M_s - 1 \) (Bracewell 1978). The power spectrum of the signal is given by:

\[
P(k) = |F(k)|^2 \tag{3.3}
\]

The power spectrum is then multiplied by a series of filters that are spaced according to mel frequencies, which can be converted from Hertz using Equation 3.1. An example of a mel-scaled filter bank, \( \Phi(k) \), is described in Memon et al. (2009), where the frequency response of the \( n \)th filter is calculated using:

\[
\Phi_n(k) = \begin{cases} 
0 & \text{for } k < k_{bn-1} \\
\frac{k - k_{bn-1}}{k_{bn} - k_{bn-1}} & \text{for } k_{bn-1} \leq k < k_{bn} \\
\frac{k_{bn} - k}{k_{bn+1} - k_{bn}} & \text{for } k_{bn} \leq k < k_{bn+1} \\
0 & \text{for } k \geq k_{bn+1}.
\end{cases} \tag{3.4}
\]
where the boundary points of that filter, equally spaced on the mel scale are given by:

\[ k_{bk} = \left[ \frac{M}{F_s} \right] f_{mel}^{-1} \left[ f_{mel}(f_{low}) + \frac{n\{f_{mel}(f_{high}) - f_{mel}(f_{low})\}}{N + 1} \right], \]  

(3.5)

where \( f_{high} \) and \( f_{low} \) are the highest and lowest frequencies contained in the signal, \( N \) is the number of filters, and \( F_s \) is the sampling frequency. The power output of a filter in the mel scale filter bank is calculated by weighting the power value given by \( P(k) \) with the frequency response of the appropriate filter \( \Phi(k) \) such that:

\[ X_n = P(k)\Phi(k). \]  

(3.6)

The coefficients can then be extracted by calculating the Discrete Cosine Transform (DCT) of the log output of the filters, using the following equation, taken from Davis and Mermelstein (1980):

\[ c_m = \sum_{n=1}^{N} \log_{10}[X_n] \cos \left( \frac{\pi m(n-0.5)}{N} \right), m = 1, 2, \ldots, M, \]  

(3.7)

where \( M \) is the number of MFCCs, and \( n = 1, 2, \ldots, N \) is the number of mel-scaled filters.

Figure 3.5 (top) shows the waveform of the sentence "Robbery, bribery, fraud." taken from the Arctic database (RMS - male American English, 16 kHz). The speech waveform was windowed every 5 ms along the x-axis using 10 ms Hanning windows and the first 5 MFCCs.
3.1. MEL-FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

(including the 0th MFCC, which can be regarded as a collection of average energies of all frequency bands in the signal (Zheng et al. 2001)) of each windowed segment were calculated using an FFT and triangular filter bank. The MFCC values across the signal are displayed at the bottom of Figure 3.5.

"Robbery, bribery, fraud."

Figure 3.5: Speech waveform and corresponding MFCC values for the first 250 frames

3.1.4 Factors affecting MFCC calculation

It is clear from the description provided in this section that there are a number of different decisions that can be made during the MFCC calculation process that can influence the final coefficient values. Some important factors are those that comprise the pre-processing stage, such as the shape and size of the windowing function used to window the waveform.
Another factor worth examining is the process by which the spectrum of the windowed waveform is obtained. Traditionally, the FT is used to convert the signal to a frequency domain representation. However, previous research in speech processing and more recent studies in speaker recognition have suggested the benefits of employing only a partial conversion to the frequency domain — that is obtaining a time-frequency representation of the speech waveform. Another factor, which has previously been considered by Zheng (2001) focuses on the parameters related to the mel-scaled filtering process, such as the number, shape, spacing and overlap of the filters in the filter bank, as well as way in which the power spectrum is warped.

In this study, three of these factors — the width of the windowing function used in the pre-processing stage, the manner of transformation of the waveform from the time domain, and the shape of the filters used to convert the power spectrum to the mel scale — were investigated. The purpose of this investigation was to draw attention to the importance of these aspects of MFCC calculation by demonstrating how changes in these values can affect MFCC values. One of the reasons for choosing these particular factors is that one factor is examined from each of the main stages of MFCC calculation — pre-processing, transformation, and frequency conversion. The motivation for investigating these factors is explored below in greater detail.

(i) Window length It can be shown theoretically, as in Section 3.2.1, or experimentally as in previous studies (Kirkpatrick et al. 2007, Kirkpatrick 2010, Kelly 2010, Kelly and Gobl 2011a, Kelly and Gobl 2011b), that the length of the window used to calculate MFCCs will have a direct effect on the values calculated. This is intuitive considering that the size of the window determines how much of the original signal is represented by the MFCC vector. More interesting is the effect that windowing has on the frequency domain representation of the signal, as the length of the window used will determine the spacing of the constituent frequency bands; too narrow a window will cause the amplitudes associated with one frequency to leak into those associated with neighbouring bands. Therefore, a classic trade-off exists between the time domain and frequency domain representations, in which the window
3.1. MEL-FREQUENCYcepstralCoefficients (MFCC)

Signal windowed by a Hanning window

Figure 3.6: Hanning windowed segment of a speech signal

length plays an important part. According to Kirkpatrick (2010), acoustic speech representations perform best when extracted using a window of one fundamental period. Reasons given for this in the literature have been speculative, and it is commonly cited that a certain window length is required to achieve adequate frequency resolution. Indeed, to our knowledge, there is no theoretical derivation of a mathematical result directly relating the effects of window length to changes in numerical perception measures, despite the fact that the window length remains a critical parameter. Furthermore, this explanation does not account for unvoiced speech.

It is important to note that it is not only the length of the window but also the type of window function used that affects the results. This is, in part, due to the fact that different window functions have different effective lengths. For instance, there is a big difference in the ef-
3. Calculating MFCC Representations

The effective length of a Blackman window and a rectangular window (having the same absolute length in number of points). We saw in Chapter 2 that there was some variation in the windowing functions used in previous studies. In order to make the experiments in this thesis comparable to future studies, we will use a Hanning window. However, in the section to follow we will use a rectangular window to demonstrate theoretically how the window length can affect the constituent values of a MFCC representation.

(ii) Time-frequency analysis (FRFT angle $\alpha$) Recent publications have demonstrated the success of time-frequency derived parameters in the field of speaker recognition (Sarikaya, Gao and Saon 2004, Wang and Wang 2005) and speech synthesis (Kirkpatrick 2010). In this chapter, we use the fast fractional Fourier transform (FRFT) to derive MFCC features using for different angles from 0 (time domain) to $\pi/2$ (frequency domain) for different window lengths for each sound category. The time-frequency plane is generally conceptualised as two orthogonal axes, the horizontal representing time and the vertical representing frequency, as in Figure 3.7 (Almeida 1994). When a signal is transformed to a domain between the time and frequency domains, the values returned depend on the angle $\alpha$, as are any values returned by any subsequent MFCC calculation. It has been suggested (Kirkpatrick 2010) that a fractional Fourier domain representation of a signal will provide a compromise in the trade off between resolution in the time and frequency domains. This kind of compromise could be very useful for numerically coding speech signals.
(iii) The mel scale filter bank  During MFCC calculation, the power spectrum is multiplied by a bank of filters arranged according to the mel frequency scale. These filters are generally triangular in shape, such as those shown in Figure 3.2 (a). However, recent research (Chakroborty and Saha 2009) has shown that MFCCs calculated using Gaussian filters perform better at speaker identification. The authors' motivation for experimenting with different filter shapes were that Gaussian shaped filters can provide much smoother transition from one sub-band to the next while preserving most of the correlation between them, and that the means and variances of these Gaussian shaped filters can be independently chosen in order to have control over the amount of overlap with neighbouring sub-bands (Chakroborty and Saha 2009). Triangular filters, in comparison, provide crisp partitions in a power spectrum, by providing non-zero weights to the portion of the spectrum covered by the filter, and zero otherwise. According to Chakroborty and Saha (2009), this phenomenon causes a loss of correlation between a sub-band output and the adjacent spectral components that are present in the other sub-bands.

In this study we calculate MFCCs using, in the mel-frequency conversion step, a bank of filters consisting of (a) traditional triangular filters, (b) rectangular filters, and (c) Gaussian filters. The effect that this parameter choice has on MFCC calculations can then be examined by comparing the resulting MFCC values. The rectangular filter bank is constructed using the standard rectangular function (Bracewell 1978) spaced according to the mel scale. The triangular filter bank is taken from Memon (2009), as outlined in Equation 3.4. The Gaussian filter bank, \( \phi^\text{Gauss}(k) \), is constructed following Chakroborty and Saha (2009), and Equation 3.4, in which:

\[
\phi^\text{Gauss}_n(k) = e^{-\frac{(k-k_0)^2}{2\sigma_n^2}},
\]

and \( \sigma_i \) is the standard deviation. The filter banks are shown in Figure 3.8.
3. Theoretical Background

In this section, the theoretical motivation for exploring factors (i) – (iii) is explained by investigating exactly how the relevant signal processing operations affect simple signals. In the previous section we discussed conceptually why these factors were interesting, but in order to demonstrate how important they actually are, it is necessary to establish mathematically what occurs when a signal is manipulated in these ways.

For the sake of transparency and in order to clearly show the effects of the operation, only straightforward sinusoids are used in these mathematical derivations. We note, however, that using Fourier synthesis we can extend the conclusions here to any physically realisable
3.2. THEORETICAL BACKGROUND

signal. Only when the effects have been determined does it make sense to experimentally demonstrate the resulting changes in the MFCC values.

3.2.1 Windowing

The Fourier theorem states that any wave can be described by a superposition of sinusoids. The FT (Bracewell 1978) of a signal $f(t)$ is given by:

$$ F(k) = \mathcal{F}\{f(t)\}(k) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi kt} dt. \quad (3.9) $$

In this section we will examine the effects of windowing a straightforward signal using a rectangular window of length $2L$. We define our rectangular windowing function as:

$$ \n(t) = \begin{cases} 0, & \text{if } |t| \geq L \\ 1, & \text{if } |t| \leq L. \end{cases} \quad (3.10) $$

Next we consider a signal $h(t)$ consisting of two cosine functions, such that:

$$ h(t) = a_1 \cos(2\pi f_1 t) + a_2 \cos(2\pi f_2 t), \quad (3.11) $$

where $a_1$ and $a_2$ are the respective amplitude contributions of the cosine functions and $f_1$ and $f_2$ are their respective frequencies.

We now use a shifted rectangular function, $\n(t - \beta)$, to window the input signal $h(t)$ such that:

$$ f(t) = \n(t - \beta) h(t). \quad (3.12) $$
We can now calculate the FT of the signal using the Fourier integral in Equation 3.9:

\[ F(k) = \mathcal{F}\{f(t)\}(k) \]
\[ = \mathcal{F}\{\triangledown(t - \beta)\}(k) \ast \mathcal{F}\{h(t)\}(k), \tag{3.13} \]

where \( \ast \) denotes the convolution operator. Using the relationship:

\[ \mathcal{F}\{\cos(2\pi ft)\}(k) = \delta(k - f) + \delta(k + f), \tag{3.14} \]

and taking the positive frequencies only, we can express \( \mathcal{F}\{h(t)\}(k) \) as:

\[ \mathcal{F}\{h(t)\}(k) = \delta(k + f_1) + \delta(k + f_2). \tag{3.15} \]

Noting that the FT of a rectangle is a sinc function:

\[ \mathcal{F}\{\triangledown(t)\}(k) = \int_{-L}^{L} e^{-i2\pi kl} dt, \]
\[ = \frac{\sin 2\pi kL}{\pi k}, \]
\[ = 2L \text{sinc}(2Lk), \tag{3.16} \]

and making use of the Fourier shift theorem:

\[ \mathcal{F}\{g(t - \beta)\}(k) = e^{-i2\pi \beta k} G(k), \tag{3.17} \]

we can express \( \mathcal{F}\{\triangledown(t - \beta)\}(k) \) as:

\[ \mathcal{F}\{\triangledown(t - \beta)\}(k) = e^{-i2\pi \beta k} (2L) \text{sinc}(2Lk). \tag{3.18} \]

Substituting expressions 3.15 and 3.18 in equation ??, we can express \( F(k) \) as:

\[ F(k) = e^{-i2\pi \beta f_1} (2L) \text{sinc}(2\pi L f_1 k) \ast [\delta(k + f_1) + \delta(k + f_2)] \]
\[ = e^{-i2\pi \beta f_1} [(2L) \text{sinc}(2\pi L f_1 k) + e^{i\varphi}(2L) \text{sinc}(2\pi L f_2 k)], \tag{3.19} \]
where \( \phi = 2\pi \beta(f_2 - f_1) \).

A cosinusoid maps to a positive and a negative frequency. Thus, the FT of the cosinusoid is two Dirac delta functions, one consisting of positive frequencies, the other of negative. The FT of a rectangular function is a sinc function. A rectangular function multiplied by cosine is then a sinc function convolved with the two delta functions. In this analysis, we concentrate solely on the positive frequencies. The sinc function interacts with the delta function such that the power associated with a specific frequency is now spread out over neighbouring frequencies. The extent of this spreading out is, strictly speaking, infinite, i.e. the total power in the sinc function is given by the integral over infinity of the intensity of the function. However for practical purposes we would like to define a region in the frequency domain where over 90% of the signal's power lies. From the following integral equation we find that this occurs between the first nulls of the sinc function such that:

\[
E(k) = \int_{-\frac{1}{2L}}^{\frac{1}{2L}} |(2L) \text{sinc}(2Lk)|^2 \, dt. \tag{3.20}
\]

We have found that approximately 90% of the power in a cosinusoid function lies within the region \(-\frac{1}{2L} < k < \frac{1}{2L}\).

Equation 3.19 therefore shows that the parameters that determine the power distribution are \( f_1, f_2, L \) and \( \beta \). Using numerical examples we will highlight some aspects of the function, particularly that:

i) a large window length will display an FT distribution where the frequency spikes are clearly separated, (see Figure 3.9, dashed line), and a smaller window length will result in a more spread out FT distribution, (see Figure 3.9, solid line).

ii) when moved across the signal, a large enough window length will produce an effectively invariant FT distribution, (see Figure 3.10, dashed line), and when moved across the signal, the smaller window length will yield a periodically varying FT distribution as the phase difference between the frequency components varies as a function of \( \phi \) (see equation 3.19 figure 3.10, solid line).
Figure 3.9: Windowed signal consisting of two cosine functions and its Fourier transform, where $\beta = 0$ ms, $f_1 = 1$ kHz and $f_2 = 2$ kHz. Solid lines represent a window of width $2L = 1$ ms, and the corresponding FT distribution. Dashed lines represent a window of width $2L = 2$ ms, and the corresponding FT distribution.

Figure 3.9 shows the magnitude FT of such a function with the values set at $f_1 = 1$ kHz, $f_2 = 2$ kHz and $\beta = 0$ ms. The dashed line in the time domain represents the length of the window of width $2L$, in this case where $2L = 2$ ms. Note that the corresponding frequency spikes, denoted also by dashed lines, are well spaced out and centred at 1 kHz and 2 kHz. The solid line represents a window length where $2L = 1$ ms and the corresponding frequency distribution in the frequency domain. In this case, a smaller rectangular window has been used, and so a wider sinc function has been convolved with the signal, resulting in a wider spread of the spikes in the frequency domain. The frequency spikes now overlap to the extent that it is no longer obvious which frequencies comprised the original signal. MFCCs are calculated from magnitude spectrums like those presented in Figure 3.9, so the values of the MFCCs would remain the same for any part of the signal for which the FT is the same.
We note that MFCCs are calculated using the square of the magnitude of the spectrum. This indicates that when the window length is large enough, see dashed line in Figure 3.9, then the magnitude of the spectrum remains invariant and hence the MFCC values do not change. This however is not the case when the window size is narrower, see solid line in Figure 3.9. Here the we expect that MFCC values will change as the rectangular window moves across the signal.

In Figure 3.10, we shift the window to the right by fixing $\beta = 0.5$ ms, introducing a phase shift between the two frequency components. Again the dashed lines denote a window length
of $2L = 2$ ms in the time domain, and the corresponding magnitude FT spectrum in the frequency domain. In this case the FT distribution is approximately the same and relatively unaltered by the phase shift, and the spikes remain centred at 1 kHz and 2 kHz. The solid line again denotes a window length of $2L = 1$ ms in the time domain, and the corresponding magnitude FT spectrum in the frequency domain. By comparing the solid line distribution in Figures 3.9 (b) and 3.10 (b), we can see significant differences between the signals. The main difference is that the single central lobe in Figure 3.9 has now been significantly changed and in fact the constituent frequencies of the signal become "resolved". We can see changes in the distribution as $\beta$ varies, so that the power associated with $f_1$ no longer leaks into $f_2$.

Since the MFCCs of a speech signal are normally calculated by first taking the magnitude Fourier transform of the windowed signal, it is clear that a different frequency distribution would lead to different MFCC values for the signal.

The purpose of this piece of analysis is to allow the reader to gain some insight into the effects of changing the window length on the Fourier plane frequency distribution. We found that for a simple input signal the frequency distribution can change quite dramatically as a function of window length and window position, which introduces a linear phase term into the frequency distribution. Although the analysis has been performed for a relatively simple signal so that insight into the process can be maintained, it clearly demonstrates the importance of this fundamental parameter in deriving a numerical representation of speech signals. Again we remind the reader that the results here apply to far more general signals. This arises because Fourier theory (synthesis) means that all physically realisable signals can be represented as an infinite sum of cosinusoidal functions of different frequencies, i.e. the signal types discussed here. From this analysis we see that the size of the function that windows the signal of interest plays a fundamentally important role in determining the MFCC value, which in turn plays an equally important role in the selection of the final synthetic speech output of a unit selection speech synthesiser.
3.2 THEORETICAL BACKGROUND

3.2.2 Fractional Fourier Transform (FRFT)

The fractional Fourier transform is a generalisation of the Fourier transform. The FRFT is dependent on the parameter $\alpha$, which determines the angle of rotation in the time-frequency domain, where $\alpha = 0$ is the identity, and $\alpha = \pi/2$ is the classical Fourier transform.

A linear operator, $R^\alpha$, corresponding to a rotation of the angle $\alpha$ which lies between 0 and $\pi/2$ would result in the representation of the signal along an axis that was at an angle $\alpha$ from the time axis. The signal representation, therefore, would lie somewhere between the time and frequency domains. Such an operator has the following properties (Almeida 1994):

1. Zero rotation: $R^0 = I$
2. Consistency with Fourier transform: $R^{\pi/2} = F$
3. Additivity of rotations: $R^\beta R^\alpha = R^{\alpha + \beta}$
4. $2\pi$ rotation: $R^{2\pi} = I$

Property (4) is a result of (2) and (3) and the fact that four successive FTs correspond to a time domain representation of the signal, such that:

$$R^{2\pi} = R^{4\pi/2} = F^4 = I$$  \hspace{1cm} (3.21)

Further properties of the FRFT can be found in (Almeida 1994).

In this section, we perform the FRFT on a straightforward signal composed of two sinu­soids in order to illustrate the effects of such an operation, and investigate its use in speech processing.

Consider again a signal $h(t)$ consisting of two cosine functions, given in Equation 3.11. We again use a shifted rectangular function, $\Pi(t - \beta)$, to window the input signal $h(t)$ such
Calculating MFCC Representations

that:

\[ f(u) = \nabla (t - \beta) h(t). \]  

(3.22)

The \( p \)-th order FRFT of a signal is given by:

\[ F^p f(u) = \int_{-\infty}^{\infty} K_p(u, u') f(u') du', \quad 0 \leq |p| \leq 2 \]  

(3.23)

where the kernel \( K_p(u, u') \) is defined as:

\[
K_p(u, u') = \begin{cases} 
\sqrt{\frac{1 - j\cot \alpha}{2\pi}} \exp \left( j \frac{u^2 + u'^2}{2} \cot \alpha - juu' \csc \alpha \right), & \alpha \neq n\pi \\
\delta(u - u') & \alpha = n\pi \\
\delta(u + u') & \alpha = 2(n+1)\pi
\end{cases}
\]

Setting \( p = 1 \) and changing the values of \( \alpha \), we can demonstrate the effects of performing the FRFT on a signal. The FRFT for a simple sinusoid of frequency 1 kHz is shown in Figure 3.11 for values of alpha ranging from 0 to 1. Note that the FRFT representation of the signal at \( \alpha = \pi/2 \) is equivalent to the FT, while the representation of the signal with \( \alpha = 0 \) is equivalent to the time domain waveform.

As mentioned in Section 3.2.1 the MFCCs of a signal can only be the same for any part of the signal for which the time-frequency representation is the same, all other variables being equal. The ability of the FRFT to represent the signal in a domain somewhere between time and frequency means that the numerical representation of the signal at this point in the MFCC calculation procedure is different for different values of \( \alpha \). This, in turn, means that the MFCCs calculated for these signals will be different for different values of \( \alpha \).

Again, the purpose of this demonstration is to illustrate the effects of performing the FRFT on a simple signal. Figure 3.11 shows very unambiguously that a vastly different 'spectrum' will result for different values of \( \alpha \). In the remainder of the chapter we will calculate MFCC for different types of speech signals to demonstrate, using real data, the extent of these effects.
3.2. THEORETICAL BACKGROUND

Figure 3.11: FRFT representations of a sinusoid for various values of $\alpha$
on more realistic speech signals.

### 3.2.3 Filter bank analysis

During the mel frequency conversion stage of calculation, the spectrum is multiplied by a bank of filters that are spaced according to the mel scale. In this section we examine the effect of the filter shape on a straightforward signal when the spectrum of the signal is multiplied by a filter bank.

The input signal in this case is a series of sinusoids of frequencies at 20 frequencies, such that the positive values of the FT of the signal are given by:

\[
\mathcal{F}\mathcal{F}\{h(t)\}(k) = \delta(k + f_1) + \delta(k + f_2) + \cdots + \delta(k + f_{20}).
\]  

(3.25)

This signal is then multiplied by each of the filters in the filter bank, such that:

\[
F_1 = \mathcal{F}\mathcal{F}\{h(t)\}(k).\Phi_1^1(k),
\]

\[
F_2 = \mathcal{F}\mathcal{F}\{h(t)\}(k).\Phi_1^2(k),
\]

\[
\vdots
\]

\[
F_{20} = \mathcal{F}\mathcal{F}\{h(t)\}(k).\Phi_1^{20}(k),
\]  

(3.26)

where \(\Phi_1^1(k), \Phi_1^2(k), \ldots, \Phi_1^{20}(k)\) are the constituent filters of the filter bank.

The filter bank output is then given by:

\[
F_{\text{out}} = F_1 + F_2 + \cdots + F_{20}.
\]  

(3.27)

Figure 3.12 (left) shows the FT of the input signal, and filter bank, for triangular, rectangular and Gaussian filters, and on the right, the output of the filter bank when it undergoes the filtering process. In this example, the frequencies of the input lie at equal intervals from 0 to 8000 Hz. We can see that the power output of the filters is a function of the shape of the
3.3 Experimental Analysis

In the previous sections, the MFCC was discussed as a numerical representation of a signal. The calculation of the MFCC was outlined, and we drew attention to some choices that are made during the calculation process. We focussed on three particular variables and provided a mathematical description of the effects that these variables have on an input signal. In this section, we demonstrate experimentally the effects of different values of these three variables using a real speech signal. In this manner, we highlight the importance of these signal processing choices, and demonstrate that different choices of (i) – (iii) result in different sets of MFCC values representing the same signal.

An example of a real speech signal was used to demonstrate the variation in MFCC values calculated using different values of (i) – (iii). The signal was extracted from the Arctic database utterance arctic.a0028: “Robbery, bribery, fraud.” (see Figure 3.5 (top)). The segment under examination was the /d/ vowel in the word “fraud”. The utterance was spoken by RMS (American English, male) and sampled at 16 kHz. In this example, 10 ms to 50 ms of a vowel sound was examined. The FRFT of the input signal was taken for angles of $\alpha$ ranging from 10° to 90° (FFT). Each calculated signal was then converted to the mel scale using a filter bank consisting of triangular, rectangular and Gaussian shaped filters.

The FRFT discussed in the previous section was the continuous form, which was used for theoretical demonstration. In order to implement the FRFT on a digital signal, it is necessary to employ the discrete version of the FRFT. The discrete version is used by means of the Matlab frft.m, the calculation of which is based on the article by Ozaktas, Arikan, Kutay, and Bozdagt (1996). The definition of the discrete fast Fourier Transform (DFT) is given by:

$$f(k) = \sum_{n=0}^{N-1} F(k,n) f(n) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n)e^{-\frac{j2\pi nk}{N}}, \; k = 0, \ldots, N - 1, \; (3.28)$$
Filters

Figure 3.12: The result of multiplying the FT of a signal consisting of sinusoids with a mel-scaled filter bank composed of (a) triangular filters, (b) rectangular filters, and (c) Gaussian filters.

Gaussian filters

Rectangular filters

Triangular filters

Pulse train multiplied by filter bank

Output of mel-scaled filters
where \( F(k,n) = \frac{W^{kn}}{\sqrt{N}} \) with \( W = e^{-j\frac{2\pi}{N}} \), and the definition of the discrete FRFT is given by:

\[
f_\alpha(k) = F^\alpha f = \sum_{n=0}^{N-1} F^\alpha(k,n)f(n), k = 0, \ldots, N - 1,
\]

where \( F^\alpha = E \Delta^\alpha E^T \), with \( F^\alpha = E \Delta E^T \) an eigenvalue decomposition of the DFT matrix (Bultheel and Martinez Sulbaran 2004).

Figures 3.13, 3.14, and 3.15 show the resulting 243 different MFCC vectors representing the signal. For each filter type, 9 separate plots were generated, each representing a different FRFT angle \( \alpha \). Each of these plots contain 9 different trends, each representing a different window length. For each window length, the vector, consisting of 12 MFCC values, is represented as a ribbon. To aid comprehension, each ribbon was given a particular colour, so that MFCCs calculated using each window length can be visually compared across all values of \( \alpha \), and across the three filter types.

### 3.3.1 Changes due to window length for fixed values of \( \alpha \) and filter

The choice of window length is a very important one. The values of the MFCCs are highly dependent on the window length used in their calculation. In Figures 3.13, 3.14, and 3.15, each ribbon denotes a set of MFCC values – its colour indicates the window length used to calculate the MFCCs, and is comparable across all plots in all three figures. The MFCC values are more sensitive to window length for some values of \( \alpha \) over others. In fact, the closer the FRFT process gets to a straight FFT (\( \alpha = 90^\circ \)), the more robust the MFCCs become when subjected to changes in window length value. That is to say, for this example, the choice of window length has less of an impact when using the FRFT with \( \alpha = 90^\circ \). In contrast, when the value of \( \alpha \) is fixed at \( 10^\circ \), each window length yields widely varying MFCC vectors. The variation of MFCC values due to window size is greater, in the case of all three filter types, for values of \( \alpha \) approaching a time domain representation.
Figure 3.13: MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of triangular filters
Figure 3.14: MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of rectangular filters.
Figure 3.15: MFCC values calculated for a windowed speech signal, using windows of different lengths, FRFT angles and a mel scaled filter bank of Gaussian filters
3.3. Experimental Analysis

3.3.2 Changes due to FRFT angle $\alpha$ for fixed window and filter values

Changes to MFCC values as a function of a change in the value of $\alpha$ are shown clearly in Figures 3.13, 3.14, and 3.15. The variation of MFCC values due to changes in $\alpha$ can be examined for a fixed window size by comparing the different coloured ribbons in each of the plots. For example, in order to examine how MFCC values change over $\alpha$ for a window length of 10 ms, we can compare the the indigo ribbon, denoting MFCC values calculated using 10 ms, on each of the plots for $\alpha = 10^\circ, 20^\circ, \ldots, 90^\circ$.

3.3.3 Changes due to filter type for fixed $\alpha$ and filter values

The variable that produced the least variance in MFCC values was the choice of filter type used in the mel scaled filter bank. By comparing Figures 3.13, 3.14, and 3.15, we can discern some slight changes introduced by the choice of filter type.

3.3.4 Discussion

We can see from Figures 3.13, 3.14, and 3.15 that changes in the values that we chose to examine result in greatly varying MFCC values for the signal under examination. Furthermore, had we tested further choices of these values, or other variables inherent in MFCC calculation, a bigger range of numerical representations would result. It is important to stress that none of these representations is more correct than the others. After all, the MFCC representation is nothing more than a signal on which a series of operations have been performed. However, some of the MFCC representations will be more useful for particular purposes than others. We require a numerical representation of a speech sound in order to assess the similarity of two speech segments that will potentially be adjacent in the synthetic speech output of a unit selection speech synthesiser. Therefore, the most valuable representation to our cause is the one for which the values are different for signals that sound different to the human ear. Furthermore, in order to bypass perceptual testing, as is the aim of this experiment, we need to be sure that the signals sound different without conducting an evaluation.
In the remainder of this chapter, we discuss how the numerical representations calculated can be used in conjunction with a distance measure to form a spectral distance measure, which can be used in the join cost of a unit selection speech synthesiser.

### 3.4 MFCCs and Join Cost

So far in this chapter, we have discussed MFCCs as a representation of a speech sound. However, what is of particular importance to this study is how such a numerical representation can be employed as measure of the similarity of two speech sounds. To recap, the join cost in a unit selection cost function is used to assess the suitability of a possible concatenation between two speech segments. This is attempted by examining the similarities between the potentially sequential segments, and penalising the sequence for any disjunction occurring between the segments that may be perceived by a human listener. Penalising for a noticeable dissimilarity between consecutive units, in theory, should result in fewer of those perceptually jarring concatenations that are an identifying feature of concatenative speech synthesis.

The join cost can be further delineated into sub-costs. Spectral distance is one such sub-cost and is used to measure the potential perceptual discontinuity between segments. Numerical representations of the speech signals are calculated and a distance measure is then used to calculate the difference between two representations. In this study, we are using particular implementations of MFCC vectors to represent our speech signals. In order to incorporate this into a spectral distance sub-cost, it is necessary to pair it with an appropriate distance measure, so that the distance between MFCC vectors, and therefore speech signals, can be calculated. Distance measures are discussed below.
3.4. MFCCs and Join Cost

3.4.1 Distance measures

In order to measure the difference between two speech sounds, or their representations, it is necessary to choose a distance measure or a metric. The concept of distance is quite intuitive when we consider the distance between two points on a line – the length of the line connecting them is the absolute distance between them. However, when we are dealing with vectors, it is more useful to visualise the vectors as coordinates on an n-dimensional plane. The MFCC vectors we have calculated, for example, can be viewed as points on a 12-dimensional plane. Calculating the distance between these points depends on the nature of the planar space that they occupy. The Euclidean metric is a natural choice for calculating distance as, not only does it correspond well with our intuitive grasp of distance, but it also represents the unique shortest distance between points on a plane.

3.4.1.1 Euclidean distance

Euclidean distance is one of the most common metrics used to calculate distance and is derived directly from Pythagoras' theorem, as illustrated in Figure 3.16.

The Euclidean distance between two points \( \mathbf{x} \) and \( \mathbf{y} \) is the length of the line segment \( \mathbf{xy} \). The general definition assumes \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) and \( \mathbf{y} = (y_1, y_2, \ldots, y_n) \) are Cartesian co-ordinates.
of two points in Euclidean n-space, and the distance between them is:

\[ \text{dist}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \]  (3.30)

In the case where \( n = 1 \), Equation 3.30 becomes

\[ \sqrt{(x - y)^2} = |x - y|, \]  (3.31)

which is the absolute distance between two points \( x \) and \( y \).

Likewise, in the case when \( n = 3 \), the equation is extended to three dimensions, as illustrated in Figure 3.17.

### 3.4.2 Calculating spectral distance

The spectral distance is calculated by finding the pairwise difference between two vectors, representing two speech segments. The numerical representations, e.g. MFCCs, of two signal segments are calculated and the distance, e.g. Euclidean distance, between these

---

Figure 3.17: Pythagoras theorem applied to distances in three-dimensional space.
vectors is calculated. The measured distance is said to be the difference between the two signals. If we have chosen an appropriate numerical representation and an appropriate distance measure, then the measured distance should reflect the perceptual difference between the signals. In unit selection speech synthesis, the two segments under scrutiny would be the trailing and leading edges of two units (e.g. diphones) from the database that may potentially be chosen as synthetic output. If the spectral distance between them is large, then it can be inferred that the perceptual difference between them is also large, and that an audible discrepancy may be detected should they be selected as an output sequence. In order to minimise the chances of this happening, a cost is assigned to them that is related in some way to the extent of the spectral distance calculated. In this manner, it is hoped that similar sounding units will incur a smaller penalty and be more likely to be chosen as the synthetic speech output.

We can see from the example given in this chapter that certain choices made during the calculation of the numerical representation will alter its resulting values. Therefore, when used in conjunction with a distance metric, different spectral distance measures will result for representations of the same unit. Therefore, it is critical to ensure the careful calculation of a signal's numerical representation in order to ensure a meaningful representation of the spectral distance measure component of the join cost of the unit selection speech synthesiser cost function. Only then can we begin to compare the calculated representations with perceptual data in a meaningful way.

3.5 Conclusion

This chapter focussed on the MFCC, as used as a numerical representation of speech signals. In this section, we summarise the findings of this chapter, discuss their implications, and conclude by highlighting how these findings can be applied to unit selection speech synthesis.
3.5.1 Chapter review

The MFCC calculation process was outlined, and attention was drawn to some of the important choices inherent in the calculation procedure. The procedure was defined in three subcategories – pre-processing, domain transformation, and frequency conversion – and one important value from each sub-category was examined: (i) window length, (ii) FRFT angle, and (iii) mel filter. The effects of changes to values (i), (ii), and (iii), were examined first from a theoretical point of view, showing mathematically, using straightforward signals as a testbed, how the signal can be affected.

Having predicted these changes from a mathematical standpoint, the effects were then demonstrated using a real speech signal. This was achieved by performing the relevant operations included in the MFCC calculation process (i.e. windowing, transformation, and filtering) using a range of values (i) – (iii), and then comparing the MFCC vector values that resulted from each calculation. The first 12 MFCC values were chosen to represent the signal and these were displayed graphically so that the changes in the MFCC values due to choices of (i) – (iii) could be easily examined.

Having established the MFCC as a speech representation, we then proceeded to discuss how this representation is used in the join cost of a unit selection cost function. The idea of spectral distance, as the calculated difference between two signals, was proposed. In order to calculate spectral distance, two signal segments are required. These segments are represented by MFCC vectors, and the spectral distance between them is ascertained with the help of a distance metric.

This study suggests the importance of carefully choosing the values of certain parameters while calculating speech representations by demonstrating the differences observed in numerical representation values as a direct result of these choices made during their calculation.
3.5.2 Discussion

There are a number of reasons why the MFCC is said to be such a true representative of a speech signal. Transforming a speech signal in the cepstral domain essentially provides a source-filter separation of the signal, making it possible to represent the spectral characteristics of the speech sound using the first few coefficients. Changes in these values represent changes in the spectral distribution of the signal. Theoretically, this makes the MFCC a good indicator of the similarity of two speech sounds, and provides a method of detecting discontinuities between two concatenated segments of speech. Furthermore, the MFCC is said to be a particularly faithful representation as the frequency scale is warped to a mel scale so that it more closely resembles the non-linear response of the ear. Many studies have been carried out in order to test the efficacy of metrics such as the MFCC, and have relied heavily on perceptual results as an indicator. However, these studies are rarely in agreement.

The calculation of MFCC values involves making decisions on the values of a number of fundamental parameters including (i) the length of the windowing function used to segment the speech signal, (ii) the time-frequency analysis performed on the signal and (iii) the type of filter used to warp the frequency axis to the mel scale. In this study we calculated the MFCC values for experimental data and examined how these values changed as the parameters (i), (ii) and (iii) were varied. The important finding of this exercise is that the MFCC values vary notably with different values of (i), (ii), and (iii). What really has been demonstrated here is that different choices of these values lead to entirely new representations of speech, which are generally grouped together under the broad heading ‘MFCCs’, without proper consideration of the values of some of these fundamental parameters, and how they affect the values of the speech representation.

3.5.3 Concluding remarks

This chapter began with some background signal processing theory, and a theoretical discussion of how the choice of window length can affect a signal's representation in the fre-
frequency domain. The mathematical derivation of the MFCC was then presented and the effects of a choice of some fundamental parameter values on the ultimate MFCC values were discussed. An experiment was then described in which the calculation of MFCCs was investigated by varying three fundamental parameters. The differences in MFCC values due to decisions made during their calculation were clearly demonstrated. The potential for these differences to impact on the naturalness and intelligibility of speech synthesis output was noted, seeing as the numerical representation is the basis for the spectral distance component of the unit selection join cost.

But just how much impact do these differences have on the resulting synthetic speech? The range of values examined for (i)–(iii) produced 243 different MFCC representations for the same database unit. However, the effects of choosing one over another have not been demonstrated. Previous studies have attempted to rank numerical representations in order of their efficacy by comparing their ability to detect discontinuities in a way that correlates with a human listener's ability to do so. However, these studies are confined to the databases on which they were tested. For example, it is impossible to compare these 243 different MFCC representations to the perceptual data used in any of the studies, as it is not freely available. Furthermore, a choice would have to be made as to which of the many datasets these representations should be compared with.

There is, therefore, a need for a more objective, repeatable, portable method of ranking numerical representations. In the next chapter, a scoring system is proposed, which ranks the numerical representations based on their ability to predict the differences between signals. The scoring system is employed to rank the 243 MFCC representations calculated for the example signal in this chapter, and the impact of the choices made in their calculation is fully demonstrated.
In the previous chapter, we examined the procedure involved in the calculation of the MFCC speech signal representation, and noted that the values of the representation could be altered by changing the values used at particular stages of the calculation procedure. Using a closed range of values for three different parameters, we calculated 243 different MFCC representations of the same database unit. Previous studies have used perceptual data to rank different numerical representations in terms of the benefits they offer to unit selection speech synthesis. Again, we stress that none of the representations are incorrect – however we are only interested in the ones that change in the same manner as a listener’s perception of the sound would change. It is quite difficult to collect reliable perceptual data with which to compare the performance of the numerical representations. For that reason, we propose that a scoring system be employed so that the numerical representations can be ranked using objective methods.

The major benefit of a scoring system is that, while perceptual data may be used to verify its veracity, it does not depend on any particular set of perceptual data in order to rank numerical representations. Whereas other studies of this nature rely on perceptual data to compare spectral distance measures (by comparing the ability of the measure to predict
discontinuities in the same manner as a listener), the scoring system depends only on a set of assumptions about predictable decisions a listener would make regarding the detection of sequential and non-sequential speech segments. The numerical representations are then ranked according to the scores they obtain. The scoring system can be used on any database, and is entirely repeatable. Scores from any study can be cross-compared without the interference of confounding factors introduced by data from unrepeatable perceptual experiments. While previous experiments have attempted to identify one numerical representation or spectral distance measure that outperforms all others, the use of a scoring system allows the investigation into the possibility that the most suitable numerical representation may be database-specific.

In this chapter, we define a scoring system by means of predicting distances between the numerical representations of carefully selected signals. We demonstrate, using the example signal from Chapter 3, the calculation of the distances between signals. We define three categories for which we can reasonably predict the scale of the distances that are likely to be calculated. We then define the scoring system in terms of the relationships between these distances. In Section 4.2, we demonstrate how the scoring system can be employed to rank certain numerical representations of a database, and how the highest-scoring numerical representation may differ depending on the type of signal it is examining. The contribution of the scoring system is supported by conducting a synthesis-based evaluation. The assumptions made in devising the scoring system are then discussed in Section 4.3, as are the limitations and contributions of the scoring system as laid out in this chapter. Finally, the contributions of the scoring system to unit selection speech synthesis are discussed.

### 4.1 Definition of the scoring system

This section describes how the use of a distance measure, (in this case the Euclidean distance between MFCC vectors, calculated using specific values of (i)-(iii)), can provide a method of ascertaining, on a preliminary level, how useful certain implementations of a numerical representation are for modelling speech sounds. This method of objectively rank-
4.1. Definition of the Scoring System

In this section, we define the scoring system. The scoring system is implemented by first calculating numerical representations of speech signals, and then calculating the distance between certain speech signals based on reasonable assumptions about how a listener would rate their perceptual continuity. These distances are defined in this section as Categories A, B, and C depending on the continuity of the signals being compared. The relationships between these distances are then used to define the scoring system.

A useful numerical representation of speech has a number of properties. The values in the vectors representing similar signals should be close, indicating that the two signals sound similar. Likewise, the values in vectors representing very different signals should be very different, indicating the dissimilarity of the sounds. Thus, a distance measured between MFCC values representing similar signals should be very small, while the distance measured between dissimilar signals should be large. These particular distance values are easy to predict, and therefore by calculating the distances using real data, we can conclude that a representation is useful if the calculated values fulfill these predictions.

For a unit selection speech synthesizer however, it is not enough that the numerical representation succeed at distinguishing one phoneme from another; it must be capable of the fine-grained resolution that allows human listeners to distinguish between two instantiations of the same phoneme. In unit selection speech synthesis, the cost function essentially compares a section of a phoneme from one complete utterance with a completely different section of another phoneme from a completely different utterance. In practice, a number of candidate sequences will be available for which the perceptual disjunction maybe very small or very big. A worthwhile numerical representation should therefore be capable of capturing the salient information encoded in a speech signal with very high resolution. As mentioned above, if two signals sound similar to the human ear, a small distance should be observed still between their representations. Traditionally, this is tested by conducting a perceptual test, like those described in the literature review of Chapter 2. However, this approach introduces all of the drawbacks inherent in such a procedure.

This presents a problem: in order to test if a numerical representation is useful for the unit selection cause, we must have a framework where we can first predict how large or small the
spectral distance will be, and then test on real data to see if our predictions have come true – if yes, then our representation is useful; if no, then it is not. However, without a perceptual test, we cannot predict the likely values of spectral distances between two same phoneme segments, some of them will sound good and therefore should have small distances, while some of them will sound bad and should have large distances.

4.1.1 Spectral distance calculation

In this section, we describe a method of circumventing that problem by using segments of a phoneme that occur within 100 ms of each other, i.e. the Category C segments described in the previous section. These segments are more likely to sound similar than segments extracted from different recordings at different times and in different contexts. Thus we predict that a numerical representation that is useful for our purposes would calculate distances between Category C segments (same phoneme, same signal, but non-consecutive) to be very close to Category A distances (same phoneme, same signal, consecutive segments).

**Category A.** Segments from the same phoneme, same signal, consecutive

– the distance between naturally consecutive segments

**Category B.** Segments from different phonemes, different signals

– the distance between different types of speech signal

**Category C.** Segments from the same phoneme, same signal but in a randomised order

– the distance between non-consecutive segments of natural speech

The segments were extracted from the Arctic database (speaker: RMS, sex: male, dialect: American English). Each phoneme examined exceeded 100 ms in duration. This was to ensure that the representative example was obscured as little as possible by the coarticulation influences of neighbouring phonemes.
4.1. Definition of the scoring system

4.1.1.1 Category A: naturally consecutive speech segments

Intuitively, the spectral distance between speech segments that occur naturally adjacent to one another should be quite small. In a unit selection speech synthesiser, it is often the case that whole words or even phrases would be lifted from the database and used as synthetic speech output. Doing so reduces the number of concatenations, and consequently, the potential for audible artefacts in the synthetic speech. It is important, therefore, to employ a join cost measure of spectral distance that is capable of recognising when speech occurs naturally, so that it can refrain from penalising consecutive segments too heavily.

For this category, one segment of the vowel /ɔ/, from the word *thaw*, in the RMS American male arctic.b0522 utterance "Nor would it thaw out his hands and feet." is compared to the following segment of the same phoneme. The distance values are shown in Figure 4.1. As expected, the distances calculated between the segments are very small for each metric calculated.

4.1.1.2 Category B: speech segments from different phonemes

Speech segments from different signals will naturally have different numerical representations. The more different one signal is from another, the greater the difference between their numerical representations, and the greater the value of the distance measure. Numerical representations of two entirely different phonemes are expected to be entirely different. In terms of join cost, should a speech synthesiser attempt to join one phoneme class to an entirely different phoneme class, it would be discouraged by the large join penalty incurred by the spectral distance sub-cost.

For this category, one segment of the vowel /ɔ/, from the word *thaw*, in the arctic.b0522 utterance "Nor would it thaw out his hands and feet.", is compared to the voiceless fricative /ʃ/, from the word *Russian* in the the arctic.b0523 utterance "The Russian music player, the Count, was her obedient slave.". The spectral characteristics of the segments are completely dissimilar, and therefore it would be expected that the distance measured between MFCC
Figure 4.1: The Euclidean distances between Category A MFCC vectors calculated using a range of values of window length, $\alpha$, and filter type.
vectors representing the segments would be large – much larger than the distance between consecutive segments of the same signal.

Figure 4.2 shows the Euclidean distance between the two segments of speech from different phonemes. It is worth noting, at this point, that while this comparison is useful for demonstrating Category B distance, it is not entirely realistic. In unit selection, this kind of comparison would mostly never occur, as all segments are concatenated at the point in the middle of the phoneme, which is assumed to be the steadiest. A more realistic situation would be if the candidate being considered for concatenation was realised in a manner that was slightly different from its potential partner, but not as different as, say the vowel /ɔ/ and fricative /ʃ/. The only situation where a phoneme change would occur across a join point would be in the case of a diphthong join, where the join point is likely to be the point of minimum stability. The problem with calculating Category B distances as the distance between two vastly different phonemes is that the numerical representation that returns a big difference may be discarding all the information coded in the signal, save for that which distinguishes one phoneme from another (for example, the periodicity of /ɔ/ or the turbulence of /ʃ/). In Chapter 5, where the scoring system is compared to a large dataset, we attempt to avoid this potential pitfall by comparing the segment of a phoneme to a general model of a phoneme from the same signal Group, instead of one that has very obvious differences.

4.1.1.3 Category C: speech segments occurring non-consecutively in the same phoneme

Segments taken from the same signal will have a lot in common, especially if taken from the same phoneme. A good speech representation would ideally remain quite stable over segments taken from the same phoneme within the same signal. As mentioned regarding Category A segments, a very low distance value would be expected between consecutive segments of the phoneme. However, a slightly larger distance might be expected between segments within that phoneme that were not naturally consecutive.

For this category, two non-consecutive segments of the vowel /ɔ/ are compared, from the
Figure 4.2: The Euclidean distances between Category B MFCC vectors calculated using a range of values of window length, $a$, and filter type.
4.1. Definition of the Scoring System

4.1.1 Definition of the scoring system

word *thaw*, in the arctic.b0522 utterance "Nor would it thaw out his hands and feet." The segments are separated by a small distance. Figure 4.3 shows the Euclidean distance between speech segments occurring non-consecutively in the same phoneme.

4.1.2 Distance predictions for Categories A, B, and C

The expected order of the categories, ranked from high to low by their spectral distance value, is given by:

- **Category B** high
- **Category C** ↓
- **Category A** low

Figures 4.4, 4.5, and 4.6 show the ranking of Categories A, B, and C, as determined by the distance measure values. In some cases, for example, when $\alpha = 90^\circ$ for all filter types, the measure has ranked the categories according to expectations: the distance measure values for Category A, where natural speech segments were compared, were very low, while those measured in Category B, where different phonemes from different signals were compared, were very high. Meanwhile, Category C distances, where segments that occurred in the same segment, but non-consecutively, were placed slightly higher than Category A and lower than Category B.

However, we can see from Figures 4.4, 4.5, and 4.6, that this is not always the case, and some MFCC representations calculated using certain values of (i) – (iii) rank the speech segments in a different order. This is very important to note, as a numerical representation is not very useful if it only captures the aspects of a signal that are of little perceptual importance to humans, from the point of view of unit selection. The relative sizes of the distances calculated for each Category is very important in revealing the parts of the signal that the numerical representation is modelling, and it is on these that we have based our scoring system. The value differences between Category distances are discussed in the remainder of this section.
Figure 4.3: The Euclidean distances between Category C MFCC vectors calculated using a range of values of window length, a, and filter type.
Figure 4.4: The Euclidean distances between MFCC vectors from Categories A, B, and C, calculated using a range of window lengths, \( \alpha \) values, and triangular filters.
Figure 4.5: The Euclidean distances between MFCC vectors from Categories A, B, and C calculated using a range of window lengths, \( \alpha \) values, and rectangular filters.
Figure 4.6: The Euclidean distances between MFCC vectors from Categories A, B, and C, calculated using a range of window lengths, α values, and Gaussian filters.

4.1.1. Definition of the scoring system.
4.1.2.1 Category B – Category A

If the distances measured for Category B comparisons are larger than those measure for Category A, it means that the numerical representation used is capable of distinguishing whether or not the speech segments under examination are of the same phoneme or have been lifted from entirely different phonemes. Therefore, if the opposite is true, that Category A values are higher than those from Category B, we can conclude that the numerical representation has failed to distinguish between segments of naturally occurring speech from the same phoneme, and segments lifted from different phonemes occurring in different utterances.

\[ d_{B-A} = \text{dist}(\text{MFCC}_{B, \text{left}}, \text{MFCC}_{B, \text{right}}) - \text{dist}(\text{MFCC}_{A, \text{left}}, \text{MFCC}_{A, \text{right}}) \]  (4.1)

4.1.2.2 Category B – Category C

The differences between values from Category B and those from Category C are similarly important. If Category B values (where phoneme and signal are different) are higher than Category C values (where the signals are the same and phoneme is the same, but the segments are non-consecutive), then the representation has successfully captured the parts of the signal that portray the phoneme class. This situation is very similar to the \( d_{B-A} \) measure. However, a good representation would give an even higher difference for \( d_{B-A} \) than it would for \( d_{B-C} \), indicating that the representation has picked up on the fact that the segments in A came naturally consecutively from the same signal (whereas in C, they are non-consecutive).

\[ d_{B-C} = \text{dist}(\text{MFCC}_{B, \text{left}}, \text{MFCC}_{B, \text{right}}) - \text{dist}(\text{MFCC}_{C, \text{left}}, \text{MFCC}_{C, \text{right}}) \]  (4.2)
4.1.2.3 Category C – Category A

Finally, the difference between values from Category C and those from Category A is discussed. In both cases, the segments are from the same phoneme class and from the same signal. The difference is that in Category A, the segments are consecutive, and in Category C, they are non-consecutive. A useful signal representation would show a greater distance between Category C values than Category A values, reflecting the fact that the segments in Category C are not consecutive. The difference between the Euclidean distances $d_{C-A}$ should then be positive values, but smaller than $d_{B-A}$.

$$d_{C-A} = \text{dist}(\text{MFCC}_C, \text{MFCC}_C^\text{right}) - \text{dist}(\text{MFCC}_A, \text{MFCC}_A^\text{right})$$ (4.3)

In summary, a scoring system would reward a representation that ranked the categories in the order B-C-A, and also show a larger difference between B and C values, than between C and A values.

4.1.3 Formal definition

From these observations emerges a pattern that can be used to determine whether or not a numerical speech representation has been successful at providing an accurate portrayal of the signal, in the sense that it captures information that is of use to unit selection speech synthesis:

$$d_B > d_C > d_A$$

$$= \text{dist}(\text{MFCC}_B, \text{MFCC}_B^\text{right})$$

$$> \text{dist}(\text{MFCC}_C, \text{MFCC}_C^\text{right})$$

$$> \text{dist}(\text{MFCC}_A, \text{MFCC}_A^\text{right})$$ (4.4)
Using Equations 4.1, 4.2, and 4.3, we define a representation as useful if it has each of the following properties:

1. \( d_{c-A} > 0 \)

2. \( d_{B-A} > d_{B-C} \)

A numerical representation that possesses these properties has succeeded in:

- performing a phoneme identification task – identifying that Category B distances between different-phoneme segments are greater than Category A same-phoneme consecutive segments.

- distinguishing "good joins" from "natural speech" – identifying that Category C distances between same-phoneme, same-signal, but non-consecutive speech segments are greater than than Category A same-phoneme consecutive segments.

From this we develop a score, by which we can rank numerical representations of speech signals. The score reflects the usefulness of the representation (in this study, one of the many implementations of MFCC), and is given by:

\[
\text{score} = \begin{cases} 
  d_{B-A} + d_{B-C}, & \text{for } d_{B-A} > d_{C-A} \geq 0 \\
  d_{C-A}, & \text{for } d_{C-A} < 0 
\end{cases}
\]  

(4.5)

Returning to the examples we have been dealing with in this chapter, we calculate Euclidean distances between the segment sequences examined in the Categories A, B, and C \((d_A, d_B,\) and \(d_C\) respectively), and the differences between \(d_B\) and \(d_A\), \(d_B\) and \(d_C\), and \(d_C\) and \(d_A\) \((d_{B-A}, d_{B-C},\) and \(d_{C-A}\), as given by Equations 4.1, 4.2, and 4.3 respectively).

The scores, given by equation 4.5, for each of the MFCC representations, calculated using different values for (i) – (iii) are shown in Figure 4.7. Each of the 243 MFCC representations, calculated using (i) one of 9 window lengths, (ii) one of 9 angles on the FRFT plane, and (iii) one of three mel filter types, are shown on the plots in Figure 4.7. The scores themselves are shown on the y-axis. MFCC representations that satisfy Properties 1 and 2 above have
Figure 4.7: Scores for MFCC representation of one example of the phoneme /a/.
a positive score. All others have a negative score. Positive values are ones for which $d_{B-C} > d_{C-A}$ and are indicated above the $y=0$ line (dashed), while those for which $d_{B-C} \leq d_{C-A}$ are shown in the negative $y$-axis portion of the plot. Higher scores are awarded to those MFCC representations that show greater separability between distance values calculated for Categories B, C, and A, as dictated by Equation 4.5.

We can see that for this example, those MFCC representations that were successful at ranking the speech signals obtained a higher score. The findings of this demonstration show that values of (i) – (iii) play a crucial part in MFCC calculation, and careless choices of these values can render the MFCC representation ineffectual, the consequences of which reverberate in the wider context of unit selection speech synthesis.

4.2 The Scoring System in Use

In this section, we demonstrate how the scoring system can be used in speech technology research to compare numerical speech representations. The purpose of the scoring system is to rank numerical representations of speech sounds in order of their efficacy, as defined by how useful they are to unit selection speech synthesis. In order to demonstrate the scoring system in use, we compared three different numerical speech signal representations. The database used in this experiment was the RMS speaker of the Arctic database. We calculated three numerical representations – two MFCC-based, and one based on LPC (Rabiner and Schafer 1978).

We have previously claimed that the differences in numerical representation values introduced during calculation have a direct effect on the performance of unit selection speech synthesisers. In this section, we demonstrate this by conducting a synthesis-based test. A synthesiser was built and the cost function modified so that units selected for synthetic output are chosen based solely on their spectral distance measure. In this manner, we have unrealistically amplified the contribution of the spectral distance sub-cost to the unit selection process in order to remove the confounding effects of the other costs in unit selection, and
demonstrate how poor choices of numerical representation can render the spectral distance sub-cost useless. The synthesis test is performed by using as a target utterance one of the sentences contained in the database. The hypothesis is as follows: a synthesiser (based on a corpus containing the test utterance) that uses a high-scoring numerical representation in its join cost will select more segments from the original utterance as its synthetic output than one using a low-scoring numerical representation. We show that the highest scoring representation selects the highest percentage of segments from the original utterances.

The experiment is outlined as follows:

1. Choose a set of numerical representations to compare.
2. Employ the scoring system to rate the efficacy of the representations.
3. Build a speech synthesiser that selects segments based only on spectral distance join costs.
4. Using each numerical representation in turn as part of the join cost, synthesise a set of test utterances from the database.
5. Compare synthesis results with scoring system predictions.

Details on these points are provided in the remainder of this section.

### 4.2.1 Employing the scoring system

The scoring system was employed to rank three numerical speech representations – two MFCC-based representations, and one LPC-based. These feature sets were calculated for certain signals from the Arctic database (RMS voice). The scoring system was implemented using the following steps:

**Step 1: Calculating numerical representations**

For each of the test segments, the following numerical representations were calculated:
Step 2: Calculating spectral distances for each category

For each numerical representation, the spectral distances for each category were calculated. Spectral distance is calculated as follows:

\[ d_{\text{CAT}} = \text{dist}(\text{numerical representation}_{\text{left}}, \text{numerical representation}_{\text{right}}) \]  \hspace{1cm} (4.6)

where \( \text{dist} \) is the Euclidean distance, given in Equation 3.29.

**Category A** distances were calculated by measuring the Euclidean distance between numerical representations of consecutive same-phoneme same-signal speech segments.

**Category B** distances were calculated by measuring the Euclidean distance between numerical representations of different-phoneme different-signal speech segments.

- In the previous section, an example signal was used to demonstrate the calculation of Category B distances. In this example, the phoneme under consideration was compared to one example of a phoneme from a different phonetic class. This is not entirely realistic. A numerical representation that calculates a large distance between a vowel and a voiceless fricative, say, may be doing so by discarding all information save for that relating to the most obvious differences between the two signals, in this case, the periodicity and turbulence of the respective signals. This is discussed further in Section 4.3.

- When deriving the scoring system, the example segment was compared to one other segment. Again, this is not entirely realistic. There are many examples of each segment in the database, and the comparison of the segment in question to one example of another is very limited. For this reason, the
The concept of a generic acoustic model was devised. Basically, the generic acoustic model consists of an average of the numerical representation for a particular collection of signals, obtained by calculating the numerical representations of each signal in the collection and calculating the mean, so as to capture a general snapshot of the representation of that segment in the database.

- However, instead of calculating a generic model of every phoneme, we instead calculated a generic model for every type of signal. For each phoneme in a signal Group, the averaged numerical representations of a subset of every other phoneme within the Group was used to construct a generic phoneme model for use in Category B comparisons. For example, if the phoneme in question was /ʒ/, the generic model for use as a Category B comparison would consist of averaged examples of /θ/, /z/, and /v/. This way, although the phonemes under comparison are still different, the differences between them are small enough that the numerical representation must still retain a large amount of information about each signal in order to distinguish against them.

- Although this is a crude model, it provides a more realistic situation than comparing with just one example (that may not be representative of the database or signal Group). By creating an average model, we hope to capture the aspects of the signal that are representative of the phoneme Group. This model may be refined in future studies.

**Category C** distances were calculated by measuring the Euclidean distance between numerical representations of non-consecutive same-phoneme same-signal speech segments.

**Step 3: Calculating relationships between category distance values**

The relationships between the calculated spectral distance measurements for Categories A, B, and C are needed in order to calculate the scores for the numerical
representations. These relationships, $d_{B-A}$, $d_{B-C}$, and $d_{C-A}$, are calculated using the following equations:

\begin{align*}
    d_{B-A} &= \text{dist}_B - \text{dist}_A \\
    d_{B-C} &= \text{dist}_B - \text{dist}_C \\
    d_{C-A} &= \text{dist}_C - \text{dist}_A
\end{align*}

(4.7) \hspace{1cm} (4.8) \hspace{1cm} (4.9)

**Step 4: Applying the scoring system**

Steps 2 and 3 were repeated for all three numerical representations. The scoring system was then applied to each numerical representation for each signal Group using Equation 4.5.

**4.2.1.1 Results**

The results of this experiment are shown in Figure 4.8. We can see from the plot that the MFCC-based numerical representation which was calculated with a 20 ms window length using the FFT, outperformed the other two feature sets. The LPC-based representation performed the worst.

**4.2.2 Synthesis-based evaluation**

In this thesis, we claim that the choices made during the calculation of numerical speech representations such as the MFCC have direct effects on the synthetic output of a unit selection speech synthesiser. We have shown the effects that these choices have on MFCC values calculated, but so far have not directly shown the consequences of using a numerical representation that does not ideally reflect the perceptually salient information in the signal. The research presented in this section shows that, according to our scoring system, the MFCC representation calculated using (i) 20 ms and (ii) 90° (MFCC_w20_a90) is a more useful representation than the other two that were compared. Using this information we propose
a method for demonstrating how using an unsuitable numerical representation can lead to a poor selection of units for the synthetic output of a speech synthesiser, by effectively turning the spectral distance component from a valuable indicator of potential synthesis quality to a meaningless calculation that serves only to mislead the cost function as to the quality of potential concatenations.

The cost function of a unit selection speech synthesiser is normally composed of a target cost and a join cost, each of which in turn consists of various components. A sequence of candidates for synthesis output incur penalties as dictated by the various components of the target and join costs. The spectral distance measure is just one component of the join cost. The contribution of the spectral distance measure, or indeed of any other components of the cost function, is maintained by a weighting system, which determines how much of an
impact that a penalty incurred for any sub-cost will have on the overall cost. The sequence of candidates with the lowest is chosen as the synthetic output of the system.

The ideal sequence of candidates for a given target utterance is one which can be directly lifted from the database. This minimises concatenation points and therefore the potential for audible discontinuities. The idea of the cost function is to award penalties for every aspect of a candidate sequence that differs from the target utterance, and so the cost tends to be low for a candidate sequence that already exists in its full form within the database. This includes the spectral distance component of the join cost – ideally, the segments in a database that occur consecutively should incur a smaller cost than those that are taken from different places in the database because the distance measured between their numerical representations of these segments should be smaller.

Again, the purpose of this synthesis-based test is to demonstrate that the choice of numerical representation used in the spectral distance sub-cost has the potential to directly affect the synthetic output of the system. For a given target utterance, carefully selected so that it occurs in the database in its full form, a unit selection speech synthesiser is employed to select a sequence of segments for synthetic output. In order to fully demonstrate the effects of the numerical representation, the cost function of this synthesiser is modified so that the candidate sequences incur penalties for only the spectral distance join cost component. If the signals in the database are modelled using a suitable numerical representation, a high proportion of the selected segments of a given target utterance should be lifted directly from the corresponding utterance in the database.

The synthesis test is performed using each of the three numerical representations examined in the last section: MFCC_w20.a90, MFCC_w10.a80, and LPC. By conducting this experiment we hope to demonstrate that the choice of numerical representation has a direct effect on the efficacy of the spectral distance measure, and therefore the resulting speech output; and that the scoring system that we devised for the thesis is capable of correctly indicating the which numerical representations are most useful for unit selection speech synthesis.
4.2.2.1 Synthesiser

A unit selection speech synthesiser was designed by the author, and used to evaluate the cost function modifications. It was designed using the infrastructure outlined in Hunt and Black (1996), which uses a weighted cost function consisting of target and join sub-costs, and a Viterbi search to select segments for synthetic speech output. The synthesiser was written in the Python programming language and is currently used (in its full form) in the abair.ie Irish speech synthesis project (Ni Chasaide et al. 2011). For the purposes of this experiment, the cost function has been modified so that segments are select based only on the costs incurred by the spectral distance component of the join cost. The structure of the system is briefly explained in this section.

Since the target cost is not being employed in this implementation, there was no need to use a marked-up version of the target utterance. Instead, the input to the synthesiser was a target utterance in the format of a string of base units – diphones. The target utterance is read into the candidate selection module, which then searches the database for matching segments. The names and locations of the files containing the candidate segments are returned, as are the locations of the segments within the file in the form of start and end times. The start and end times of the candidate units are then used to locate, in a corresponding coefficients file, the numerical representations of the leftmost and rightmost segments of the candidate unit. By calculating the distances between the representations at concatenation points, the spectral distance between potentially sequential units can be calculated. A Viterbi algorithm is used to trace the shortest path back through the candidate matrix and select the sequence of candidate units with the lowest overall spectral distance cost. The names and locations of the sound files that contain the selected segments, along with the start times and end times of the segments, are passed to the concatenation module, which lifts the segments from the sound files, and plays it back as audio output.
4.2.2.2 Target utterances

The most important part of this synthesis experiment is the fact that the target utterances are contained in the database being used by the synthesiser. Ten utterances, taken directly from the Arctic database, were used as target utterances. They were entered as input to the synthesis system in the form of diphones. The utterances are listed below in Table 4.1 and a screen shot of the utterances in their diphone format is given in Figure 4.9.

<table>
<thead>
<tr>
<th>idx</th>
<th>Filename</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>arctic_a0002</td>
<td>Not at this particular case, Tom, apologised Whittemore.</td>
</tr>
<tr>
<td>2</td>
<td>arctic_a0005</td>
<td>Will we ever forget it.</td>
</tr>
<tr>
<td>3</td>
<td>arctic_a0007</td>
<td>He turned sharply, and faced Gregson across the table.</td>
</tr>
<tr>
<td>4</td>
<td>arctic_a0009</td>
<td>And you always want to see it in the superlative degree.</td>
</tr>
<tr>
<td>5</td>
<td>arctic_a0019</td>
<td>I followed the line of the proposed railroad, looking for chances.</td>
</tr>
<tr>
<td>6</td>
<td>arctic_b0026</td>
<td>You have associated with some of these men.</td>
</tr>
<tr>
<td>7</td>
<td>arctic_b0244</td>
<td>You are positively soulless, he said savagely.</td>
</tr>
<tr>
<td>8</td>
<td>arctic_b0538</td>
<td>At the best, they were necessary accessories.</td>
</tr>
<tr>
<td>9</td>
<td>arctic_b0476</td>
<td>The reorganization of these countries took the form of revolution.</td>
</tr>
<tr>
<td>10</td>
<td>arctic_a0486</td>
<td>For the rest, he was a mere automaton.</td>
</tr>
</tbody>
</table>

Table 4.1: Target utterances used for synthesis test

Figure 4.9: Screen shot of target utterances in diphone form
The candidates for these target utterances were selected based on the distances between the numerical representations of the constituent segments. The candidate selection was performed three times, once for each numerical representation being compared. The results are presented in the following section.

4.2.3 Results

The results of the synthesis experiment are displayed in Figure 4.10, which shows the percentage of units selected that belonged to the corresponding utterance in the database. We can see from this plot that the choice of numerical representation chosen to model the speech segments contributes directly to the choice of candidates selected for speech synthesis output. Different segments were chosen depending on the numerical representation used. Furthermore, the numerical representation that succeeded in selecting the units from the original utterance was the one that scored highest by the scoring system.

4.2.3 Discussion

In this section, we have demonstrated how the scoring system can be used to rank any given numerical speech signal representations. We found that one particular representation outranked the other two. We then described a synthesis-based test to demonstrate that the choice of numerical representation directly affects the values chosen by the unit selection speech synthesiser. By taking care to ensure that our target utterances were also present in the database, we were able to predict that a synthesiser based on a good numerical representation would select more segments from the original utterance. We found that the scores obtained by each numerical representation with the percentage of original segments chosen. In fact, there was a significant correlation between these values, with Pearson's $\rho = 0.999$ ($p < 0.05$).

However, this high correlation value was expected. After all, the scoring system was designed so that a numerical representation for which low distance values were calculated
between consecutive segments scored highly. The synthesis test was also designed to give
a higher score (percentage of original segments) to the numerical representation that could
calculate the smallest distances between naturally consecutive same-signal segments. How­
ever, this is still relevant to speech synthesis. Often, the target input text is contained some­
where in the database, especially if the database is designed to contain many common
utterances. Therefore, a numerical representation that calculates a low score between natu­
rally consecutive speech is very useful. However, a really valuable numerical representation
also needs to be able to predict a small distance between similar sounding units from dif­
ferent places in the database. Although our scoring system was also designed to perform
this task, it was not tested using our synthesis-based evaluation described here. The real
contribution of this experiment was to demonstrate that the use of different numerical rep­
resentations does in fact lead to the selection of different segments as synthetic output.
Furthermore, since two of the numerical representations were MFCC-based, we demonstrated that changes in certain values, (i) and (ii) in this case, were just as liable to result in the choice of different segments, as choosing a completely different numerical representation.

4.3 Analysis of the Scoring System

In this section, we discuss the motivation for devising the scoring system, and justify the decisions made in its design. We then discuss the contributions and the limitations of the scoring system. The scoring system was designed to rate the efficacy of numerical representations of speech sounds. It was designed to be quick, automatic, and repeatable, in order to give a general idea of the usefulness of a numerical representation for unit selection speech synthesis.

4.3.1 Assumptions

The system is based on a number of assumptions about the behaviour of a suitable numerical representation. These assumptions are discussed below.

Assumption 1: Category A distances will be less than Category C distances. In unit selection speech synthesis, the spectral distance measure weighs in on the overall decision of the cost function to select particular units over others. This was demonstrated in Section 4.2, where a cost function consisting only of a spectral distance measure was used to select units. We found that a numerical representation that minimised the distances between units that were naturally consecutive in the database contributed to the naturalness and intelligibility of the synthetic output by selecting a higher percentage of units from original utterances within the database. A carefully designed unit selection corpus can contribute to the quality of the output sequences by including a larger constituent of commonly used expressions and therefore maximising the likelihood that a target utterance can be found in the database.
The assumption that a good numerical representation would calculate smaller Category A distances is therefore justified, as a synthesiser based on such a representation would select more original sequences from the database, resulting in fewer artificial concatenations and less opportunity for audible concatenations.

**Assumption 2: Category C distances will be close to (but greater than) Category A distances.** Most of the time in unit selection speech synthesis, the target utterance is not found in the database in its entirety. In such cases, it is necessary for the cost function to calculate costs for candidate sequences, and the spectral distance component will be used to represent the differences between the numerical representations of signals that are not naturally consecutive segments of speech. A useful numerical representation, therefore, is required to represent signals in such a way that a small distance is calculated between segments that sound similar, even if they are not naturally consecutive. Traditionally, a perceptual evaluation is carried out in order to ascertain whether differences can be discerned between certain segments, and a numerical representation is deemed useful if it predicts discontinuity in the same manner as the test subjects. However, as previously discussed, this method has certain drawbacks as the perceptual data is not always reliable, and results from different researchers is not cross-comparable. The aim of the scoring system is to remove the dependence on perceptual data, and rate numerical representations in an objective manner. In order to do this, it is necessary to make certain predictions about the perception of a listener regarding the similarity of certain speech segments. One such assumption is that Category C distances will be slightly greater than Category A distances. A larger difference between Category C distances is predicted because the segments being compared in Category C are not naturally consecutive, as they are in Category A. However, it is also predicted that the distances will be quite close to Category A distances, because they are segments taken from the same signal and phoneme, and therefore are more likely to contain similarities than those taken from completely different segments. The assumption is that numerical representations that capture the similarities between these non-consecutive segments (while still capturing the distinctions owing to the fact that the segments are not naturally consecutive), are likely to result in smaller distances being calculated between
similar sounding units in the database. Basically, we hope that a good numerical representation can encode the particular information in a signal that allows a listener to perceive the similarities inherent in same-signal segments.

**Assumption 3: Category B distances will be greater than Category A and C distances**

In order to calculate Category B distances, an acoustic model was developed of a generic 'phoneme', which combines the characteristics of a combination of phonemes from a particular phonetic class. We assume here that the numerical representation of a segment of a particular phoneme will be quite different than the numerical representation of a different phoneme, and that the distances calculated between them will be quite large. We attribute a score to the numerical representation based on its ability to distinguish between phonemes, when in reality the usefulness of a numerical representation lies in its ability to discern within-phoneme changes.

The assumption, therefore, is that the salient audible cues that allow a listener to distinguish one phoneme from another are the same ones that allow a listener to distinguish between more subtle within-phoneme distances. In reality, we do not know if that is the case. The differences between certain different vowels, say, is a distinct and significant alteration in the positions of the formants, whereas the differences between different instantiations of the same vowel would not be as obvious as finding the formants in different positions. Instead it could be something a lot more subtle, like a very slight difference in the position of the formants, or the presence of energy in other frequencies due to coarticulation effects. In short, awarding a high score to a numerical representation because of its ability to detect large differences in spectral constitution may tend to favour a representation that can perform phoneme identification, without being entirely sure that it can distinguish between phonemes that are phonetically closer. We cannot be sure that the numerical representation is successful in calculating large distances for Category B because it discards spectral detail and retains only information that captures a segment's identity.

This may still be useful to unit selection speech synthesis, however. The generic segment models that were generated in order to compare with known vowel segments were created
using a combination of other phonemes from the same category. The generic acoustic model would therefore not display any particularly exotic auditory features – a vowel segment would not display any characteristics associated with a fricative, for example, and diphthongs and vowel models were calculated separately, so that no distinguishing diphthong-like behaviour would become a recognisable part of the vowel acoustic model, and vice versa. Therefore, if the numerical representation was in fact discarding much of the salient spectral information and retaining only the information that allowed the detection of different phonemes, it would still need to retain enough information to differentiate between spectrally (relatively) similar phonemes. In this way, we hope to model the situation in unit selection wherein a candidate being considered as a companion for a particular phoneme has been realised slightly differently and therefore would result in an auditory discontinuity if the join was approved. We can assume that the spectral signature of the candidate would be slightly different from the one to which the join is being considered. If a useful numerical representation had been employed, we would expect that a distance be calculated between the numerical representations of the two signals that would discourage the choice of that candidate. The assumption, therefore, is that a numerical representation that can distinguish between a signal segment and one that displays slightly different spectral characteristics would be useful in this scenario as it would reject the candidate that has been realised differently.

To summarise, we have recognised the danger that numerical representation that return large Category B distances might in fact be discarding information relevant to making smaller, and more subtle within-phoneme changes, by discarding all information about the signals save for those that allow the identification of phonemes. We point out that in our employment of the acoustic model as the contrasting segment, we have attempted to avoid this pitfall, as we have optimised for differences between segments that are not so vastly different that very much of the information would be discarded.

**Assumption 4: Euclidean distance is a suitable metric** D'Agostino and Dardanoni (2009) define a list of properties that address the manner in which the distance between two vectors can change:
4.3. Analysis of the Scoring System

(a) by changing the value of one component of one of the two vectors, leaving everything else unchanged;

(b) by "swapping" the values of two elements in one of the two vectors, leaving everything else unchanged (D'Agostino and Dardanoni 2009).

The Euclidean distance satisfies these properties in that it changes intuitively to reflect the manner in which vectors can change. For the purposes of this study, we need a distance function that will intuitively change in this manner. The properties from D'Agostino and Dardanoni's study (2009) are all required in the measurement of spectral distance, and therefore, of all the available metrics, the Euclidean distance is a justifiably suitable choice for calculating spectral distance.

Assumption 5: $d_{B-A}$ is perceptually equivalent to $d_{C-A}$. The scoring system awards a positive score based on these two factors in equal measure for $d_{B-A} > d_{C-A} \geq 0$. However, it may be the case that one of these relationships is more important and should be weighted to reflect that relative importance. In time, the scoring system may be refined to reflect the importance of one or other of these factors, and it may be that the importance differs for different types of signals. In this study, for practical purposes, the factors were weighted equally.

4.3.2 Limitations

What the scoring system does is give an indication of how faithfully a numerical representation will represent a speech signal, in a manner that is useful for unit selection speech synthesis, based on the assumptions listed above. There are a number of the limitations to the scoring system introduced by the necessary assumptions made when devising it.

For instance, without the aid of a perceptual test, it is impossible to be certain that a numerical representation will necessarily predict differences between segments in the same way as a human would. We have made assumptions based on what we think humans would
say about certain types of segment comparisons. We have already discussed the potential limitations of the types of assumptions we have made.

4.3.3 Contributions

The major contribution of the scoring system is the concept itself, which is an objective, repeatable, and flexible method of ascertaining which numerical representation will best model the speech signals in a given database. Some of the specific contributions of the scoring system are discussed below:

1. Objective – Provides an objective manner of rating the efficacy of a numerical representation.

2. Repeatable – The fact that the scoring system is not related to any specific set of perceptual data means that the framework can be applied to any speech database.

3. Flexible – Allows for quick and automatic assessment of any database, allowing the researcher to explore the possibilities that the ideal numerical representation may be database-specific.

Furthermore, although the scoring system concept remains valid, the particular scoring system outlined in this chapter is by no means the only method of ranking the representations. This scoring system is based on assumptions, that are not without their limitations, but are, on the whole, reasonable assumptions to make about human perception of discontinuity in speech. The assumptions adopted for the design of this particular scoring system may in time be improved or refined. For example, by conducting perceptual tests evidence can be accumulated to support or oppose the claims made. If any of the assumptions is not upheld under such scrutiny, it is possible to change the assumption, and therefore the details of the scoring system, without detracting from the contributions of the scoring system concept.
4.4 Conclusion

The main contribution of this chapter was the proposal, testing and analysis of a scoring system. Of the many numerical signal representations that exist, it is highly improbable that any one of them could capture the complexity of human perception. We can, however, try to ascertain which are useful to unit selection speech synthesis, and which, when used as part of a spectral distance measure, will result in the calculation of meaningful distances between potentially consecutive database segments. In this section, we review the chapter, discuss some of the findings, and finish with some concluding remarks.

4.4.1 Chapter review

In this chapter, we proposed the concept of a scoring system as an objective measurement of the efficacy of the numerical representations of speech signals as applied to unit selection speech synthesis.

In Section 4.1, we discussed how a numerical representation in conjunction with a distance metric can be used to calculate the spectral distance between carefully selected segments. These segment comparisons were defined as belonging to one of three separate categories - A, B, and C - depending on differences inherent in the segments being compared. These categories were chosen so that the perceptual distance between the sounds within the categories could be predicted. The scoring system was then devised as the relationship between these three types of distances calculated for a database. As a demonstration, the scores were determined for 243 MFCC representations of an example phoneme.

In Section 4.2, the scoring system was employed to compare three numerical representations. The database used was the RMS Arctic database, which features a male speaker of American English. The scoring system ranked the numerical representations in order of their efficacy. In order to show how the choice of one of these representations over another would affect the synthetic output of a speech synthesiser, a synthesis test was developed. In this test, the target utterances were already contained in the speech database in their entirety,
and the cost function selected units based solely on their spectral distance measure. Therefore, for a given target utterance, a useful numerical representation would choose a higher proportion of segments from the corresponding original utterance as the synthetic output. By conducting this experiment, we demonstrated the importance of choosing a suitable numerical representation by showing the differences in synthetic output that result from one choice over another. We also noted a high correlation between the percentage of original units chosen when using a particular numerical representation with the score it received by our scoring system. We pointed out that this was expected, seeing as both tests awarded a high score for correctly identifying naturally consecutive segments, but noted also that such a tendency was very useful for unit selection speech synthesis.

In Section 4.3, we explain the assumptions made in devising the scoring system, in particular the predictions regarding the category distance values and the choice of using Euclidean distance to calculate them and discuss the limitations of the scoring system introduced by these assumptions. We then outline the contributions of the scoring system, and the benefits of the general concept of a scoring system over a direct comparison of spectral distance measurements to perceptual data.

4.4.2 Discussion

At the start of this chapter, we outline a method of rating numerical representations by using them in combination with a distance metric, to create a spectral distance measure. This spectral distance measure is then used to calculate the difference between certain pairs of segments, carefully chosen because we have some intuition as to the values of these distances, relative to each other at least. We know that the values of an accurate speech representation would change dramatically from phoneme to phoneme and even from signal to signal, but stay relatively constant within the same phoneme and signal. From this we developed a scoring system, which we can now apply to any speech representation, in order to ascertain whether it is likely to be a good numerical representation of a speech sound.
The benefit of this kind of measure is the speed with which a large number of representations can be rated. The scoring process acts as an initial screening of potential speech representations, many of which are based on loose approximations of the expected behaviour of speech mechanisms and the human ear. It is important to note, at this point, that this study has no intention of undermining the benefits of the perceptual test – in contrast, the perceptual test is the only way to comprehensively test human perception. The study does recognise, however, that the perceptual test can be expensive, time-consuming, and due to the variation in method, standard and testbeds, very difficult, if not impossible, to compare across studies. This is especially frustrating when the range of test items is large and arbitrary, like the set of possible speech signal representations. Furthermore, the results of this study can be directly compared to a similar study, say of further implementations of MFCC, or LSF, LPCC, or any other existing speech representation – the scoring system is universal and quantitative, and the results are useable and reproducible.

A scoring system, such as the one outlined here, also allows relatively simple method of exploring questions along certain lines of research. One question, which will be addressed in the remainder of this work, is whether there exists one numerical representation that is ideal for use with all types of speech sounds in a database, or if, in fact, the choice of representation should be dependent on the characteristics of the signal being examined. Another question, also addressed in this study, is whether the same numerical representations obtain similar scores for different speakers, or whether the highest scoring numerical representation is speaker specific. The scoring system allows for a quick and automatic investigation into these questions with two major benefits – it allows the researcher to explore the choices made during the calculation of numerical representations to see if any trends emerge, making it possible to learn more about speech signals; and it allows the possibility for optimising a unit selection join cost for an individual database by choosing a selection of numerical representations to use with particular sounds in order to produce better speech synthesis.
4.4.3 Concluding Remarks

Finally, we again note the benefits of the experiment we have devised, which tests the ability of various numerical representations of speech sounds, in a manner that, importantly, does not rely on human perception tests. We believe this is an important contribution for several reasons:

1. The use of a scoring system, in conjunction with a freely available database, allows experiments on new metrics to be directly compared to the results from this study.

2. New metrics can be tested for a wide range of different conditions without the need for a human perception test.

3. While we acknowledge the "gold standard" nature of a human perception test, it is a cumbersome test that is fraught with human error and insufficient numbers of test subjects. This approach allows for a quantitative comparison of the performance of different metrics.

4. By creating a framework that allows important preliminary results to be derived, we can greatly speed up research in this area, by quickly eliminating poor metrics.

5. The scoring system is repeatable on different data sets. Future work may provide supportive evidence of the integrity of the scoring system itself, but the framework can be applied to any database without perceptual studies being conducted.

In the following chapter, we address the questions that have been raised so far during this study:

1. Is there evidence that particular values of (i) – (iii) will yield a representation that best models a speech sound?

2. Do value choices hold for all types of speech sounds, or are they dependent on the characteristics of the signal?
3. How can this information be used to optimise the cost function in a unit selection speech synthesiser?

In order to address these questions, we calculate the 243 MFCC representations on a large dataset of speech signals, and use our scoring system to compare them, and we repeat this experiment for three speakers of American English.
The numerical values of MFCCs and other representations of speech signals are dependent on the values chosen for important variables used in their calculation. In Chapter 3, we demonstrated how changes in three parameters can affect the calculation of MFCC values that are used to numerically represent a speech signal. In order to assess how important these changes are, we devised a scoring system that rates the usefulness of a speech representation for unit selection speech synthesis. Using one speech signal, we showed that, according to the scoring system, some MFCC-based representations are useful for unit selection, and some are not, and furthermore, their usefulness is dependent on the choices of (i) – (iii) used in the calculation of the numerical representation.

As demonstrated using one speech signal in Chapter 4, inappropriate choices of these parameters can lead to the calculation of numerical representations that do not suitably and usefully model the speech signal. If inappropriate choices are made in the calculation of speech signal representations in a unit selection speech database, the spectral distance sub-cost of the cost function will be non-impacting at best, and, at worst, will give a green light to sequences that do not sound acceptable to the listener.

In this chapter, we extend the experiments carried out in Chapters 3 and 4 and examine
many examples of real speech in order to demonstrate first and foremost that the dependency of the MFCC values on parameters (i) – (iii) holds for a large dataset, but more importantly to investigate whether the highest rating MFCC representations are calculated using similar values of (i) – (iii) for every signal. In the study outlined in this chapter, we also investigate whether the results hold across speakers, and across signal types. In short, we wish to answer the following questions:

1. Is there something about a particular value of (i), (ii), or (iii) that yields a useful MFCC representation?

2. Does this hold true for different types of signals, or do the values of (i) – (iii) that yield useful representations change for different signal types?

3. Are the observations speaker-specific, or do they remain true regardless of the speaker?

This experiment also provides the opportunity to utilise the scoring system devised in Chapter 4. Having established the scoring system, we can now employ it in order to compare candidate feature sets for different groups of speech signals and different speakers. At this point in the thesis, 243 candidate feature sets are calculated by varying the values of (i) – (iii) during MFCC calculation, and the scoring system is employed to rank them in order of efficacy.

5.1 Test Database

Up to now, research into spectral discontinuity was performed on databases constructed and recorded by the research group in question specifically for the task at hand. While these studies were no doubt rigorous, and a marked contribution to the research area, they are unfortunately not easily repeatable. In order to compare a new metric with the perceptual results, the recordings and the perceptual data would not be freely available.

The use of the Arctic database was therefore a very important decision in this research. It allowed the experiment to be performed on three different speakers, with the option of
expanding on this research in a controlled manner by any researcher in the future. The results from these three experiments and future experiments can be cross-compared, as the databases all used the same prompt list and were recorded under similar conditions. Some information is provided below on the Arctic database.

5.1.1 The Arctic database

The CMU Arctic database is a set of carefully selected, phonetically balanced sentences, chosen from out-of-copyright publications and recorded under strict studio conditions to serve as a basis for constructing a synthetic voice. It is freely available under GNU public license and can be downloaded at http://www.festvox.org/cmu.arctic. The current release includes .wav recordings of a number of different speakers in American English (male and female), and more recently, databases of the American English dialect spoken by Canadian, Indian and Scottish speakers (all male), which were recorded with simultaneous EGG measurements. The list of sentences used in the Arctic database is taken from the online catalogue Project Gutenberg, which contains the full text of selected books released with a free software license, including a clause that allows their copyright of the work itself to be removed so that the remaining text enters the public domain. The sentences were selected from the texts using a greedy algorithm which searches through the texts and selects the sentences that contain the greatest variety of phonemes in different contexts. The search is efficient enough to create the smallest prompt list possible while still retaining sufficient phonetic coverage. The 1132 sentences used in the Arctic prompt set contain 39,153 phones (Kominek and Black 2003), including the silence phone. These phones are all instances of the 41 phonemes of American English occurring in different contexts depending on the stressed or unstressed nature of the phoneme, and the position of the phoneme in the syllable, word or phrase. The phonemes will be pronounced differently by the speaker depending on the context in which they occur.

The Arctic prompt list covers the following list of phonemes, which are represented in Table 5.1 in IPA transcription and their equivalent Festival notation. Throughout this thesis,
both label types are used. Generally, in the text, the IPA notation is favoured. However, this becomes problematic during computation, for instance, when calculating MFCC representations using Matlab, or performing synthesis. The Festival notation is used in these circumstances as they are easy to reproduce using a computer keyboard. For that reason, the Festival notation will be used in Matlab plots, and synthesis output.

<table>
<thead>
<tr>
<th>Consonants</th>
<th>IPA</th>
<th>Festival</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>k</td>
<td>k</td>
<td>k</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>ñ</td>
<td>ñ</td>
<td>ñ</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vowels</th>
<th>IPA</th>
<th>Festival</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>iy</td>
<td>i</td>
</tr>
<tr>
<td>í</td>
<td>ih</td>
<td>í</td>
</tr>
<tr>
<td>e</td>
<td>eh</td>
<td>e</td>
</tr>
<tr>
<td>ę</td>
<td>ey</td>
<td>ę</td>
</tr>
<tr>
<td>æ</td>
<td>ae</td>
<td>æ</td>
</tr>
<tr>
<td>ø</td>
<td>ao</td>
<td>ø</td>
</tr>
<tr>
<td>ą</td>
<td>aa</td>
<td>ą</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IPA</th>
<th>Festival</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>ow</td>
</tr>
<tr>
<td>ø</td>
<td>ah</td>
</tr>
<tr>
<td>u</td>
<td>uh</td>
</tr>
<tr>
<td>æ</td>
<td>uw</td>
</tr>
<tr>
<td>ę</td>
<td>ay</td>
</tr>
<tr>
<td>í</td>
<td>oy</td>
</tr>
<tr>
<td>ñ</td>
<td>aw</td>
</tr>
</tbody>
</table>

Table 5.1: American English phone set used the the Arctic prompt list in Festival and IPA notation

Figure 5.1 shows a histogram of the frequency of occurrence of the phonemes in the Arctic database, not including the silence phone, which occurs the most frequently. The most common phonemes (besides silence/pause) are /n/ and /a/, while the least common are /ɔi/ and /ʒ/.

Phonemes were extracted from three different Arctic databases:

1. RMS – American English, male
2. BDL – American English, male
3. SLT – American English, female
5.1.2 Signal groups

For the purposes of this thesis, the types of signals that constitute speech are categorised based on the presence or absence of periodicity or turbulence.

5.1.2.1 Group 1

Signals belonging to Group 1 are quasi-periodic signals that are not characterised by the presence of turbulence. In this study, we have chosen to examine certain vowels and diphthongs. In particular, the concatenation of diphthongs has been reported as particularly problematic in unit selection speech synthesis (Vepa and King 2005). In unit selection, the segments of speech are usually joined at the level of diphones, so that the concatenation point occurs at the steadiest point of the phoneme. Diphthongs are characterised by a sharp transition from one steady vowel state to another, so that the middle of the phoneme is not the steadiest state, as it would be in a monophthong.
Vowels and diphthongs are both examined in Group 1 signals in order to examine how changes in (i) – (iii) affect the MFCC representation of this type of signal. Previous studies (Kirkpatrick 2010) suggest that a window length of at least one pitch period is needed to gather enough information to sufficiently model a speech signal. In this experiment, we hope to verify these findings. Other studies (Agrawal, Chandra and Badgaiyan 2012) suggest that, for periodic signals, the FFT will better represent the signal than other angles of the FRFT. This is also investigated in this experiment.

5.1.2.2 Group 2

Signals belonging to Group 2 are quasi-periodic, but are also characterised by turbulence. In this study, we investigate whether the same values of (i) – (iii) yield the highest scoring MFCC representations. Considering that the signals still retain their periodicity, it may be that the window length that leads to the calculation of the most suitable MFCC representation is a function of the fundamental frequency of the signal, or it may be that the turbulent nature of the signal interferes with this relationship. Different angles of rotation of the FRFT may lead to the calculation of more suitable MFCC representations for Group 2 signals.

5.1.2.3 Group 3

Group 3 consists of voiceless fricatives, which are non-periodic and turbulent. Generally, the problem of discontinuity across concatenated voiceless fricatives is not addressed, as it is thought that the discontinuity should not be perceptually obvious. One study (Syrdal and Conkie 2004) looked at the broad phonetic classes of unvoiced weak and strong fricatives and found, using perceptual data, that 35.3% (weak) and 13.6% (strong) of joins were detected by listeners. It is therefore worth investigating whether a numerical representation can sufficiently model signals from this category, and whether they should be calculated using the same values of (i) – (iii) as used for signals from Groups 1 and 2.
5.1.3 Test stimuli

The test stimuli were chosen from three different categories of sound differentiated by the presence or absence of voicing (periodicity) and frication (turbulence). The units chosen for this experiment were sampled at 16 kHz and located within the Arctic database recordings using a search algorithm. The search algorithm was written in Python and used specific mark-up files (.utt), included in the Arctic database download, that contained start and end times for all the phonemes in the Arctic database. There is one .utt file for every .wav file. The search algorithm read through all the .utt files and extracted the information about the position in the corresponding .wav files of target phonemes from each Group.

Table 5.2 below details the units examined for this experiment.

5.2 Experimental Procedure

In this section, we describe the method used whereby, for three different speakers, we apply the scoring system to many examples of speech signals belonging to different Groups.

The experimental procedure is outlined as follows:

1. The location of each phoneme of interest was identified in the Arctic database.

2. MFCC values were calculated for each segment, using different values of (i) – (iii).

3. Category A distances were calculated as the Euclidean distance between consecutive same-phoneme segments.

4. Category B distances were calculated as the Euclidean distance between a segment of the phoneme, and a segment of a generic acoustic model specifically created for use with the phoneme.

5. Category C distances were calculated as the Euclidean distance between non-consecutive same-phoneme segments.
5.2. Experimental Procedure

<table>
<thead>
<tr>
<th>Phonetic label</th>
<th>RMS</th>
<th>BDL</th>
<th>SLT</th>
<th>Periodicity</th>
<th>Turbulence</th>
</tr>
</thead>
<tbody>
<tr>
<td>/i/</td>
<td>535</td>
<td>322</td>
<td>1088</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/i/</td>
<td>195</td>
<td>135</td>
<td>1375</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/e/</td>
<td>140</td>
<td>106</td>
<td>625</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/æ/</td>
<td>350</td>
<td>317</td>
<td>541</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/a/</td>
<td>654</td>
<td>436</td>
<td>897</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/o/</td>
<td>345</td>
<td>250</td>
<td>436</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/o/</td>
<td>224</td>
<td>224</td>
<td>673</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/o/</td>
<td>288</td>
<td>251</td>
<td>444</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/æ/</td>
<td>109</td>
<td>65</td>
<td>526</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/a/</td>
<td>27</td>
<td>16</td>
<td>149</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/u/</td>
<td>186</td>
<td>135</td>
<td>435</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/u/</td>
<td>541</td>
<td>492</td>
<td>698</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/oi/</td>
<td>90</td>
<td>81</td>
<td>92</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/au/</td>
<td>234</td>
<td>187</td>
<td>237</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 5.2: Units extracted from the RMS, BDL and SLT Arctic database for testing

6. For each phoneme the means of Categories A, B, and C, were calculated.

7. The scoring system was applied using Equation 4.5.

This experiment was performed for three different signal Groups on three different Arctic databases. In the remainder of this section, the databases are discussed, and the signal Groups are defined and explained. An explanation is also provided of how the 243 different MFCC representations are calculated for each segment. Finally, details are given of how Category A, B, and C distances were calculated and how the scoring system was employed.
5.3 Results

The scores calculated using Equation 4.5 for each MFCC representation (calculated using a range of values of (i) – (iii)) are presented in this section for each database RMS, BDL, and SLT. The mean $f_0$ for each signal was calculated using the fwrapt.m script for Matlab, which tracks the $f_0$ across the signal. The mean $f_0$ values for each phoneme are presented in Appendix B.

5.3.1 RMS database

The RMS database features an American English male speaker, with an average $f_0$ of 101.18 Hz for voiced segments. The mean $f_0$ calculated for each signal examined is shown in Figure 5.2.
5.3. Results

5.3.1.1 Group 1: Periodic, non turbulent signals (vowels and diphthongs)

The average scores across the vowels and diphthongs listed in Table 5.2 were calculated for the RMS database, using the outlined procedure and Equation 4.5. These results are presented in Figure 5.3 below. We can see, from this plot, the effects that the choices (i) – (iii) had on the values calculated for the MFCC representation of the Group 1 signals from the RMS database. Window length options (i) within the range of 10 ms to 50 ms were examined, and of these, MFCC representations calculated using a window length of 10 ms or 20 ms were awarded the best score, based on their ability to identify a ranking system whereby Category A distances had values that were slightly lower than Category C distances and much lower than Category B distances. Representations calculated using a 15 ms window length were not as successful at ranking the Categories, nor were those calculated using window lengths greater than 20 ms. The mean $f_0$ for Group 1 segments was 101.14 Hz, giving a fundamental period of 9.88 ms. This means that the average window length for Group 1 is long enough to capture 1 or 2 entire pitch periods of the signal. We can therefore say that this is an adequate amount to resolve its constituent frequencies to a degree that will allow us ascertain useful information about the signal.

With the exception of $\alpha = 90^\circ$, the range of values tested for (ii), the FRFT angle, all resulted in an MFCC representation that was incapable of modelling Group 1 signals so that the distances calculated for Categories A, B, and C were in the predicted order. The FFT $\alpha = 90^\circ$ representations that ranked the distances as predicted, with Category A distances slightly lower than C and much lower than B, are shown in the positive region of the plot, above the dashed black zero line. Those in the negative portion of the plot were incapable of distinguishing between Category A and Category C – scores close to zero indicate that Category A and C distances were close, but with a lower score are less useful. The differences that resulted from using a different shaped filter to convert the frequencies to the mel scale were so very small that they are imperceptible on the score plot. This would suggest that the changes introduced by varying the value of (iii) are negligible.

The results for the constituent vowels and diphthongs examined for Group 1 are also pre-
Figure 5.3: Scores for MFCC representations of Group 1 segments in the RMS database.
5.3. Results

Figure B.2 shows the scores for the phoneme /i/. The scores are positive for (i) \( > 35 \text{ ms} \) (ii) 80\(^\circ\), and all representations calculated with (ii) 90\(^\circ\), the highest value awarded to the MFCC representation calculated using a window length of 20 ms. The mean \( f_0 \) calculated for instances of this phoneme was 101.46 Hz, meaning that at least 2 pitch periods of 9.85 ms were required to adequately resolve the constituent frequencies.

For the /i/ vowel, only the use of the FFT for (ii) resulted in adequate MFCC representations, as shown in Figure B.4. In this case, the highest scoring MFCC representation was calculated using (i) 25 ms, but all window lengths scored highly for the FFT. The average pitch period for instances of this phoneme was also 9.88 ms \( (f_0 = 101.19) \), and so at least 2 pitch periods were covered by the window size that yielded the highest scoring representation.

For the phonemes /ɛ/ and /e/, shown in Figures B.6 and B.8 respectively, again the FFT produces the only MFCC representations to achieve positive scores. The representation that scored highest for phoneme /ɛ/ was 10 ms, with a secondary peak at 20 ms. The mean pitch period was 9.93 ms \( (f_0 = 100.61) \) and the top scoring MFCC representations were calculated using a window of adequate length to examine either 1 or 2 full cycles. The MFCC representation that scored highest for /e/ was calculated using a 20 ms window. The mean pitch period was 9.88 ms \( (f_0 = 100.61) \) in this case, so two pitch periods were used to calculate this highest scoring representation. Again, no difference in score was observed by changing the value of (iii).

For the phonemes /æ/, /ao/, and /aa/, shown in Figures B.10, B.12, and B.14 respectively, again the representations calculated using the FFT rank the highest. For all three phonemes, the scores have a peak at 20 ms, but for phonemes /ao/, and /aa/, representations calculated using 10 ms were tied for the highest score. The pitch periods of the phonemes are 9.91 ms \( (f_0 = 100.86 \text{ Hz}) \), 9.95 ms \( (f_0 = 100.55 \text{ Hz}) \), and 9.92 ms \( (f_0 = 100.79 \text{ Hz}) \), so that 2 full pitch periods are covered by the 20 ms window, but only 1 pitch period for the representations calculated using a 10 ms window length.
Only the MFCC representations of /o/ that are calculated using the FFT scored positively by the scoring system. This is shown in Figure B.16. Again, value of (iii) had negligible impact on the scores. The highest score is seen for MFCCs calculated using (i) 10 ms, (ii) 90°, and (iii) any. Only one 9.88 ms pitch period \( (f_0 = 101.13 \text{ Hz}) \) is needed to give sufficient resolution of the frequencies in this case. However, there is a secondary maximum at (i) 20 ms, for which 2 pitch periods would be required.

Figure B.18 shows the scores for the phoneme /æ/. Again, only FFT-based MFCC representations achieved a positive score. The choice of (iii) had no impact on the ranking of the scores. The choice of value for (i) had an impact on the scores awarded to the MFCC representations. Those calculated with a window length of 20 ms scored the highest, with those calculated using 10 ms in close second. Higher window lengths produce representations that attained lower scores. Two 9.87 ms pitch periods \( (f_0 = 101.32 \text{ Hz}) \) are needed to give sufficient resolution of the frequencies in this case, or just 1 for the representation calculated using a (i) value of 10 ms.

Figure B.20 shows the scores for /u/. Again, the FFT is the only choice of (ii) that results in a positively scoring MFCC representation, while any choice of (iii) gives similar results. The window length that yields the highest score is 20 ms for /u/. Two 9.65 ms pitch periods \( (f_0 = 103.62 \text{ Hz}) \) are needed to give sufficient resolution of the frequencies in this case.

Figure B.22 shows the scores for /u/. The representations that yield the highest scores peak at 10 ms and decrease with increasing window size. The mean \( f_0 \) of /u/ is 100.83 Hz, meaning that only one pitch period is captured in the peak-score window length. Again, the only values of (ii) where \( \alpha = 90° \) yielded positively scoring results, and the choice of (iii) had a negligible impact on the scores.

In this Group, we also decided to include diphthongs, as problems have been noted in the past (Vepa and King 2005) regarding their numerical representation. In addition, the fact that they can be regarded for our purposes as periodic and non-turbulent, means that they satisfy the criteria for Group 1 membership. The scores for representations of the diphthong /ai/, as shown in Figure B.24, are positive only for the FFT at certain window lengths, in
keeping with Group 1 results so far. The peak window length is 10 ms for the FFT, with a dip at 15 ms and a second peak at 20 ms. Representations calculated using window lengths of above 35 ms obtained negative scores. The average pitch period for these segments was roughly \( f_0 = 100 \text{ Hz} \) 10 ms, so one period was just about covered by the 10 ms window length. Again the value of (iii) was of little or no impact to the score obtained by the MFCC representations.

Scores for MFCC representations of the diphthong /au/ are shown in Figure B.26. Again, they are positive only for FFT values of certain window lengths. The highest scoring representation was calculated using a window length of 10 ms, with those calculated with a 20 ms window scoring in second place. These window lengths are long enough to respectively cover 1 and 2 pitch periods of 10 ms \( f_0 = 100 \text{ Hz} \). Scores were negative for values of (i) greater than 25 ms. The value of (iii) had no noticeable impact on score.

The scores for diphthong /ɔi/ are similar in that only certain window lengths for the FFT yield appropriate MFCC representations for the signal, while higher, in this case window lengths greater than 20 ms, give a negative score. The average pitch period for these segments was \( f_0 = 101.83 \text{ Hz} \) was 9.82 ms, so one period was covered by the 10 ms window length. Only representations calculated using the FFT obtain a positive score. The score is largely unaffected by (iii).

The MFCC representations for diphthongs are different from those representing vowels in that there are some window lengths for which the FFT does not yield an appropriate representation. In the case of vowels, FFT representations of all window lengths obtained some kind of score, and in some cases, so did certain other values of (ii). For diphthongs, therefore, there are less combinations of (i) – (iii) that will yield a representation that will model the signal adequately for our purposes, as determined by the scoring system. These findings support the claim that the salient information contained in diphthongs is more difficult to represent numerically.
5.3.1.2 Group 2: Period, turbulent signals (voiced fricatives)

Group 2 signals are those that, while still displaying the characteristic of quasi-periodicity, also display some turbulence, or frication, brought on by the manner of articulation of the phonemes that make up the Group: voiced fricatives. In unit selection speech synthesis, the spectral distance measure of the join cost attributes a penalty to voiced fricatives based on how perceptible it deems the transition from one segment to the next. The way it calculates this distance is dependent on the numerical representation used to model the voiced fricative. While many studies have been done to assess the validity and usefulness of numerical representations for vowels and diphthongs, there are comparatively few studies that have investigated whether this suitability extends to the phoneme group of voiced fricatives.

As mentioned above, Group 2 differs from Group 1 in that the constituent sounds display a level of turbulence. We predict that this will affect the efficacy of any numerical representation that proved effective in modelling Group 1 sounds, and that the highest scoring MFCC representations for this Group will be calculated using values (i) – (iii) that differ from those used to calculate Group 1 MFCC representations.

Figure 5.5 shows the scores obtained by MFCC representations of Group 2 sounds, and the mean $f_0$ calculated for each signal examined in shown in Figure 5.4. For the signals in Group 2, positive scores were obtained by representations calculated using many different window lengths and FRFT rotation angles. These results contrast strongly with Figure 5.3, where representations calculated using the FFT were exclusively dominant. The Group 2 scores are similar to Group 1 in that the value of (iii) has only a negligible impact on the final score. The average $f_0$ of the segments in this category was 101.42 Hz, making the average pitch period 9.86 ms. The highest scoring representations were calculated using FRFT angles of 60–90°, and for window lengths of around 15–20 ms. Other values of (ii) yielded positively scoring representations for some window lengths, especially 20 ms, but negative values for most others.

Figure B.30 shows the scores for the MFCC representations of the voiced fricative /s/. FFT-based representations scored consistently highly for this phoneme, across all window
Figure 5.4: Mean f₀ of Group 2 segments in the RMS database

lengths and filter types. The highest scoring representation, however, was calculated using (i) 20 ms, (ii) 60°. FRFT values of 70° and 80° also yielded high scoring representations, especially for a window length of 20 ms. The average pitch period of these segments was 10 ms (f₀ = 100 Hz) so again, the highest representations were calculated using 2 pitch periods. Once more, the choice of (iii) did not affect the scores.

Figure B.32 shows the scores for the MFCC representations of the voiced fricative /v/. The results for /v/ displayed some similarity to Group 1 results in that the highest scoring representation was calculated using the FFT and a 25 ms window. However, other representations achieved scores in line with this, notably with α = 80°, and a 20 ms window. Unlike Group 1 signals, the representations were calculated using many different values for (ii). Many FRFT-based representations also featured in the scores with negative values. Again though, this is very different from Group 1 scores, where the FFT was the exclusive winner. The average pitch period was 10 ms (f₀ = 100 Hz), so again, one or two periods
Figure 5.5: Scores for MFCC representations of Group 2 segments in the RMS database.
5.3. RESULTS

were required to calculate the highest scoring representations.

Figures B.34 shows the scores for the MFCC representations of the voiced fricative /z/. The highest scoring representations for these segments were those that used values of (i) 10 ms and 15 ms. The \( f_0 \) for each phoneme was calculated as 101 Hz. The representations that scored highest were calculated using values of (ii) from 50° to 80°. Lower values of (ii) scored positively for the window length of 20 ms particularly. The FFT-based representations only scored positively for 10–20 ms windows.

Figure B.36 shows the scores for the MFCC representations of the voiced fricative /ʒ/. Representations calculated using angles of \( \alpha \) other than the FFT scored highly, particularly where \( \alpha = 70° \), as did those calculated using the FFT, but only for lower window lengths, with the exception of 10 ms. The mean \( f_0 \) was calculated at 100.85 Hz for /ʒ/, and the window lengths that yielded the highest scoring representations were 15 ms and 20 ms. The value of (iii) did not significantly impact the scores.

5.3.1.3 Group 3: non-periodic, turbulent signals (voiceless fricatives)

Group 3 differs from the previous Groups in that signals that constitute Group 3 have no inherent periodicity. Group 3 signals are voiceless fricatives, characterised by high-frequency turbulence. For the sounds investigated in this study, the frication is caused by using the tongue or lower lip to create an air-flow constriction during phoneme production. Figure 5.6 shows the collective scores for signals belonging to Group 3. We can see immediately that the scoring pattern resembles that of Group 1, where the constituent phonemes displayed no turbulence, more than that of Group 2, where the constituent phonemes had turbulence. This is surprising as the opposite trend was predicted. The remainder of this section shows the phonetic breakdown of the results.

The scores for representations of the phoneme /θ/ are shown in Figure B.37. These scores differ from all seen yet, in that every representation calculated gives a negative score. This indicates that each representation was incapable of distinguishing between Category A and
Figure 5.6: Scores for MFCC representations of Group 3 segments in the RMS database
Category C distances. These scores were augmented if the $d_{B-A}$ B calculated as positive, but diminished if not.

The scores for MFCC representations of the phoneme /f/ are presented in Figure B.38. The scoring pattern is like that observed for the phoneme /θ/ – none of the representations achieved a positive score, and the highest scoring negative scores were those of the FFT.

In Figure B.39, we can see that the scoring patterns of MFCC representations for the phoneme /s/ marks a return to what we saw for Group 1 segments – dominant scores for FFT-based representations. Representations calculated using a window length of 10–20 ms received the highest scores.

MFCC representation scores for the phoneme /ʃ/ are given in Figure B.40. Again, we see a dominance of the FFT-based representations, the highest calculated using window lengths of 10 ms, and the scores decreasing for increasing window length. The opposite is true for those calculated using a value of 80° for (ii), in that the scores increase with increasing window size, yielding positive score for long window lengths.

5.3.1.4 Discussion

The scores obtained for MFCC representations differed depending on the values used in their calculation. Window length and FRFT angle, (i) and (ii) respectively, caused the most variation in the scores. The extent of this variation different depending on whether the signals belonged to Groups 1, 2, or 3.

Group 1 scores were dominated by FFT-based representations. In fact, for most Group 1 signals, the FFT-based representations were the only ones that received positive scores. For these FFT-based representations, those calculated using a window length of 10 ms and 20 ms scored the highest, which corroborates with Kirkpatrick’s (2010) findings that at least one pitch period is needed to adequately resolve the constituent frequencies while still maintaining the quasi-stationarity of the signal. However, most other window lengths also
yielded positive scores. The choice of (ii) was therefore more important for Group 1 signals, as any angle for the FRFT other than 90 yielded a representation that was not as useful for unit selection speech synthesis, according to our scoring system.

Group 2 scores were high for representations calculated using other values of (ii). The choice of (i) greatly affected the scores – for any given value of (ii), the scores differed greatly for different values of (i). This is a different situation than for Group 1 signals, where the choice of (i) did not affect the scores so drastically. Generally, the lower values for (i) yielded the higher scores for Group 2 signals.

Group 3 representations scored highly for some constituent phonemes, but for others, no representation received a positive score. The phonemes for which a positive score was observed were, for the most part, modelled using FFT-based MFCC representations.

5.3.2 BDL database

The BDL Arctic database features a male speaker of American English. The mean \( f_0 \) calculated for the voiced segments of this database is 127.74 Hz, making the average pitch period 7.83 ms.

5.3.2.1 Group 1: Periodic, non turbulent signals (vowels and diphthongs)

For the Group 1 segments alone, the average pitch period is 7.83 ms, and the average \( f_0 \) is 127.74 Hz. The mean \( f_0 \) calculated for each signal examined in shown in Figure 5.7. When we look at the scores for the MFCC representations of all the Group 1 segments, shown in Figure 5.8, we can see that the value of (i) that yields the highest scoring representations is 15 ms, almost enough to allow two entire pitch periods. The scores for 10 ms and 20 ms also feature strongly, but representations calculated using higher values of (i) obtain a lower score. FFT-based representations dominate the high scores for the Group 1 segments, as they did for the RMS database. The remainder of this section shows the phonetic breakdown for the Group 1 segments of the BDL database.
5.3. Results

Mean f0 readings for examples of Group 1 signals in the BDL database

![Graph showing mean f0 readings for Group 1 signals in the BDL database.](image)

Figure 5.7: Mean f0 of Group 1 segments in the BDL database

Figure B.42 shows the scores obtained by the MFCC representations of segments of the phoneme /i/. FFT-based representations scored consistently well for every value of (i) and (iii). A low positive score was obtained for values of (ii) of 80° where the window length was 50 ms. The mean pitch period for this segment was 127 Hz.

The pattern of scoring in Figure B.44 is unlike anything seen already for Group 1 signals. All the representations scored negatively, with the exception of the low values of (ii) (10 and 20°) at 30 and 35 ms window lengths. The mean f0 of this segment was 128.35 Hz.

Figure B.46 shows the scores for MFCC representations of the vowel /ɛ/. Here we see a return to the trend of high-scoring FFT-based representations for roughly two pitch periods of the voiced segment. The mean f0 is 126.63 Hz.

The scores for phoneme /e/ are given in Figure B.48. These are very similar to those shown for /ɛ/ – the highest scoring representation was calculated using a window length of 15 ms.
Figure 5.8: Scores for MFCC representations of Group 1 signals in the BDL database
Results

$\left( f_0 = 127.31 \text{Hz}, \text{pitch period 7.85 ms} \right)$, and the FFT. All FFT-based representations scored highly, and those calculated using $60^\circ$ and $70^\circ$ obtained low positive scores for high values of (i). The choice of (iii) did not significantly affect the scores.

Figure B.50 shows the scores for MFCC representations for the phoneme /æ/. Again, FFT-based values have the highest scores, especially those calculated using low window length, particularly 15 ms. The mean $f_0$ of the segments was calculated at 127.24 Hz.

Figure B.52 shows the scores for MFCC representations of /ao/. In this case, we see rising score as the value of (i) changes from 10 ms to 20 ms, with a decrease in score thereafter. Again, this is only true for the FFT-based representations, which scored consistently highly. Representations calculated using (ii) $80^\circ$ received high scores for certain window lengths. Again, the value of (iii) had no remarkable effects on the scores. The average pitch period was $7.88 \text{ ms} \left( f_0 = 126.83 \text{ Hz} \right)$.

The scoring trend seen for Figure B.54 is similar again, with the dominant scores obtained by representations calculated using (ii) equal to $90^\circ$. The highest score was for value of (i) equal to 15 ms. However, values of (ii) equal to $80^\circ$ also received consistently positive scores, if not in the range of the FFT-based representations. The $f_0$ for /ɔ/ was 127.11 Hz, making the average pitch period 7.87 ms.

The only representations to receive a positive score for phoneme /o/ were those calculated using (ii) $90^\circ$. These are shown in Figure B.56. The highest scoring of these were calculated using a window length of 15 ms. The score decreases with increasing window size. The mean $f_0$ of the segments was 127.06 Hz, making the average pitch period 7.87 ms in length. Values of (iii) did not significantly alter the scores.

Scores for MFCC representations of the phoneme /ʌ/ are shown in Figure B.58. Again, the FFT-based representations receive the highest scores. The value of (iii) does not significantly affect the scores. The FFT-based representations that scored the highest were those calculated using 15 ms and 20 ms window lengths. The average $f_0$ for the segments is 128.77 Hz, making the average pitch period of 7.76 ms.
Figure B.60 shows the scores for MFCC representations of the phoneme /u/. The highest scoring were calculated using (ii) 90°, as for most other Group 1 signals examined so far. The value of (i) that produced the highest scoring representations was 10 ms in this case. Incidentally, the mean $f_0$ calculated for this phoneme is 130.26 Hz, higher than any other phoneme in the group, meaning that its pitch period, at 7.67 ms is shorter than any yet seen in the Group. Generally, we have been seeing a correspondence between pitch period and window length for the voiced segments.

Scores for MFCC representations for the vowel /u/ are shown in Figure B.62. For values of (i) ranging from 10–25 ms, the highest scoring representations were calculated using a value of (ii) 90°. After (i) 25 ms however, the FFT-based representations score only negatively. Representations calculated using (ii) 80° scored positively for all window lengths, and those calculated using FRFT angles of 70° received positive values using window lengths of 15 ms and greater than 25 ms. The mean $f_0$ for these segments was 128 Hz.

The scores for MFCC representations of the diphthongs /ai/, /au/, and /ɔi/ are shown in Figures B.64, B.66, and B.68 respectively. For all three phonemes, the FFT-based representations achieve the highest scores, especially those calculated with values of (i) equal to 10 ms or 15 ms. The score decreases with increasing values of (i), especially in the case of /ɔi/, where values of (i) over 20 ms yield negatively scoring MFCC representations. In the case of /ɔi/, positive scores are shown for higher values of (i) provided they were calculated using lower values of (ii), ranging from 60°–80°.

### 5.3.2.2 Group 2: Period, turbulent signals (voiced fricatives)

In this section, we examine the scores for MFCC representation of segments belonging to Group 2, those that display both periodicity and turbulence. We expect that, due to the level of turbulence, MFCC representations other than those that successfully modelled Group 1 signals, as determined by the scoring system, may be quite effective at modelling signals from Group 2. The mean $f_0$ calculated for each signal examined in shown in Figure 5.9.
5.3. Results

Mean f0 readings for examples of Group 2 signals in the BDL database

![Graph showing mean f0 readings for Group 2 signals in the BDL database](image)

Figure 5.9: Mean f0 of Group 2 segments in the BDL database

We can see from Figure 5.10 that, while representations calculated using other values of (ii), the FFT still yields the highest scoring representations. Despite the continued dominance of the FFT-based representations, the scoring pattern for Group 2 signals is significantly different than those for Group 1 signals. The most notable difference is the proliferation of scores for MFCC representations calculated using FRFT angles other than the FFT itself. In particular, those calculated using a value of (ii) ranging from 50–80° give scores ranking second only to representations calculated using the FFT. The average f0 for Group 2 segments was 127.77 Hz, giving an average pitch period of 7.8 ms.

Figure B.70 shows the scores for MFCC representations of the phoneme /ʊ/. Representations calculated using 10 ms and 15 ms window lengths performed well for the values of (ii) equal to 60°. This scoring pattern is very different from anything that was seen in Group 1. The value of (iii) did not noticeably affect the scores.

The scores in Figure B.72 are again very similar to those displayed by Group 1 signals,
Figure 5.10: Scores for MFCC representations of Group 2 signals in the BDL database

BDL: Scores for MFCC representations of Group 2

Triangular filters

Rectangular filters

Gaussian filters

window length (ms)
in that the FFT-based representations are dominant, especially for low values of (i). The scores obtained by other values of (ii), particularly 60°, and 70° at higher values of (i). The high scores of these MFCC representations make the pattern for this phoneme quite different from that of a Group 1 phoneme, despite the predominance of the FFT-based scores. The mean \( f_0 \) of these segments is 127.25 Hz.

Scores for MFCC representations of the phoneme /z/ more closely resemble those from Group 1 than any voiced fricative examined yet. The FFT-based representations score the highest for every value of (i), particularly the lower values. However, representations calculated using lower values of (ii) is noticeable, especially 80°, and a window length of 15–35ms. These are shown in Figure B.74.

Figure B.76 shows the scores for MFCC representations of the phoneme /\tilde{z}/. Again for this phoneme, the FFT-based representations score lower than those calculated using other values of (ii), especially 50°, 60°, and 70°, for a window length of 20 ms. The \( f_0 \) for this segment was 129.25 Hz and the average pitch period of 7.73 ms.

### 5.3.2.3 Group 3: Non-periodic, turbulent signals (voiceless fricatives)

The signals in Group 3 have no periodicity and are characterised by the presence of turbulence.

Figure 5.11 shows the scores for MFCC representations of Group 3 signals. Again, the FFT-based representations are dominant, the highest scoring one calculated with a 10 ms window length. However, like Group 2 signals, representations calculated using other values of (ii) obtain high scores as well, in contrast to the Group 1 scoring pattern. For values of (ii) of 60° and 80°, the highest scoring representations were calculated using a 15 ms window. The contributions of the particular phonemes examined are given in the rest of the section.

The scores for the phoneme /θ/ are given in Figure B.77. Although the representations calculated using (ii) 90° scored the highest for one particular value, the scores for those
Figure 5.11: Scores for MFCC representations of Group 3 signals in the BDL database.
calculated using other FRFT values, particularly 50° to 80° were in contention. The scores decreased generally with increasing values of (i), and the value of (iii) had little influence on the scores.

The scores for MFCC representations for /f/ are given in Figure B.78. Just like the situation for the RMS database, the representations for this phoneme did not score positively by the scoring system. Those representations calculated using (ii) 80° or 90° scored the highest.

Figure B.79 show the scores for MFCC representations of the phoneme /s/. The highest scoring representations were those calculated using an FRFT angle of 80° or 90°. The highest scoring FFT-based representation was calculated using a window length of 10 ms, and the highest scoring representation for (i) 80° was calculated using a window length of 15 ms.

Figure B.80 shows the scores for MFCC representations of the phoneme /ʃ/. Values of (ii) ranging from 50° to 90° yielded high scoring MFCC representations. For these values, the score generally decreased as the value of (i) increased. Again, the value of (iii) had little bearing on the final scores.

5.3.2.4 Discussion

The scores obtained for MFCC representations of the phonemes examined in the BDL database were dependent on the values of (i), (ii), and (iii) used in the calculation of the numerical representations. Furthermore, certain values of (i) and (ii) yielded particular representations that scored more highly than others. These values, especially those of (ii), differed depending on the signal Group under investigation.

Representations calculated using the FFT scored the highest for most Group 1 signals, to the exclusion of all other values of (ii), which only yielded negatively scoring representations. Of these, the representations calculated using a window length of 10–20 ms scored the highest. The average pitch period for the Group was about 7.8 ms, so one or two pitch periods were
needed to obtain the temporal information from the signal needed to adequately resolve the constituent frequencies. While the shorter window lengths yielded representations that scored highly, the choice of \( \alpha \) was therefore more important for Group 1 signals, as any angle for the FRFT other than 90° yielded a representation that was not as useful for unit selection speech synthesis, according to our scoring system.

Group 2 scores were high for representations calculated using the FFT and other values of \( \alpha \) at lower window lengths. When grouped together, the scores for Group 2 resemble Group 1 in that the representations calculated with \( \alpha =90° \) achieved the highest scores. However, the scoring pattern is distinct from that observed for Group 1 in that representations calculated using values of \( \alpha \) other than the FRFT scored highly, and the difference between these scores of these representations was smaller than for Group 1. That is, for Group 1 signals, the FFT was the clear winner; for Group 2 signals, the situation is not as clear-cut. Looking at the constituent phonemes, we see that for certain phonemes, the FFT-based representations dominate, but for others, those calculated using other angles of the FRFT achieve the highest scores. Again, this situation differs from Group 1, where the scores for the phonemes, with the exception of \(/i/\), were remarkably similar.

Group 3 representations scored highly for some constituent phonemes, but for others, no representation received a positive score. The phonemes for which a positive score was observed were, for the most part, modelled using FFT-based MFCC representations.

### 5.3.3 SLT database

The SLT Arctic database features a female speaker of American English. The mean \( f_0 \) for the voiced segments is 175.49 Hz, making the average pitch period 5.69 ms. The mean \( f_0 \) calculated for each signal examined in shown in Figure 5.12.
5.3. Results

Mean f0 readings for examples of Group 1 signals in the SLT database

Figure 5.12: Mean f0 of Group 1 segments in the SLT database

5.3.3.1 Group 1: Periodic, non turbulent signals (vowels and diphthongs)

The average pitch period of the Group 1 signals alone is 175.52 Hz, and the average pitch period is 5.69 ms. Again, the Group 1 signals examined are vowels and diphthongs, which have characteristic periodicity.

We can see from the scores shown in Figure 5.13, that only MFCC representations that adequately model the Group 1 sounds for the SLT database are FFT-based, with a (ii) value of 90°. All other values of (ii) achieved negative scores. Of the FFT-based representations, the highest scoring were those calculated using a (i) value of 10 ms or 15 ms. This corresponds to one to two periods for Group 1 signals.

The scores for MFCC representations for the constituent phonemes are shown in Figures B.82–B.102. Each constituent scoring pattern is representative for the entire of Group 1, showing high scores for 10 ms and that score decreasing with increasing window length.
Figure 5.13: Scores for MFCC representations of Group 1 signals in the SLT database.

SLT: Scores for MFCC representations of Group 1

Triangular filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Rectangular filters

Gaussian filters

Window length (ms)
One exception is seen in Figure B.100, where the scores for the MFCC representations of the phoneme /u/ differ from the others of Group 1 in that the score for the FFT-based representation calculated using a 10 ms window receives a negative score, whereas for every other member of the Group, it obtains one of the highest scores. Instead the highest score is awarded to the FFT-representations calculated using a 15 ms window. This is in keeping with the rest of the results. For every other plot in this Group, the FFT-based representation, calculated using either a 10 ms or a 15 ms window, (i.e. 1 or 2 pitch periods), achieves the highest score, with the score decreasing with increasing values of (i). As in all data examined so far, the value of (iii) does not bear significantly on the scores.

5.3.3.2 Group 2: Period, turbulent signals (voiced fricatives)

Group 2 scores have an average pitch period of 5.7 ms ($f_0 = 175$ Hz), and again, differ from Group 1 signals in that they also display a level of turbulence. The mean $f_0$ calculated for each signal examined is shown in Figure 5.14.

Figure 5.15 shows the scores for the MFCC representations of the Group 2 segments. The FFT-based representations for this database score the highest, and although some of those calculated using other FRFT angles also score positively, they mostly feature in the negative section of the plot. An examination of the scores obtained by the MFCC representations of the constituent phonemes is provided in the remainder of this section.

Figure B.110 shows the scores for MFCC representations of the phoneme /θ/. The FFT-based representations score consistently highest, across all window lengths, particularly 15 ms. The representation calculated using the FRFT angle of 80° and window length of 35 ms receives a positive score scores for this phoneme, but all other scores are negative.

The scores for MFCC representations of the phoneme /v/ are shown in Figure B.112. The FFT-based representations here also display a pattern similar to those found for Group 1 signals, with those calculated using lower values of (i) achieving the highest scores. Positive scores are also achieved for 80° where the window length was 20, 25 or 50 ms.
The scores for the phoneme /z/ are displayed in Figure B.114. Again the FFT-based representations score highly, but only for values of (i) in the range of 15–30 ms. Representations calculated using a value of 80° for (ii) and values of (i) ranging from 10–35 ms also receive high scores. Values of (iii) did not have a significant effect on the scores.

The scores for /3/ are shown in Figure B.116. Again the FFT-based representations score the highest, and display a pattern very like that seen for Group 1 signals. The difference with this phoneme is that representations calculated using different values of (ii) also feature strongly in the scores, particularly where (ii) is 60°. Other values of (ii) yield representations that scored negatively, with the exception of 35 ms window length for 80° and 20 ms for 70°.
Figure 5.15: Scores for MFCC representations of Group 2 signals in the SLT database
5.3.3.3 Group 3: Non-periodic, turbulent signals (voiceless fricatives)

Finally, we examine the scores obtained by the MFCC representations for Group 3 segments for the SLT database. Again, these segments are voiceless fricatives, which display no characteristic periodicity, but are instead characterised by the turbulence produced during their production.

Figure 5.16 shows the scores for the MFCC representations for all of the segments in Group 3. The highest scoring representation was produced using (i) 15 ms, and (ii) 90°. However the scores for the FFT-based representations decrease with increasing window length. Representations calculated using (ii) equal to 50° and 60° score consistently highly for all values of (i) and (iii), and for some window lengths, 70° and 80° In the remainder of this section, we examine how each constituent phoneme in the Group contributed to the overall score.

Figure B.117 shows the scores for the MFCC representations for the phoneme /θ/. In this case, FFT-based representations achieved the highest scores, especially those with low values of (i).

Figure B.118 shows the scores for the MFCC representations for the phoneme /f/. Again, the FFT-based representations score well, but only for values of (i) between 15 ms and 25 ms. Scores for representations calculated using an FRFT angle of 70° were also positive for values of (i) in the range of 10–30 ms. Values of (iii) had negligible effects on the scores.

The scores for MFCC representations of the phoneme /s/ are shown in Figure B.119. The FFT-based MFCC representations for values of (i) in the range of 15–45 ms achieve the highest scores, the score decreasing with increasing values of (i). The FRFT angle 70° also achieves a high score, but only at values of (i) equal to 25–35 ms.

The scores for MFCC representations of /ʃ/ are given in Figure B.120. In this plot, we notice that many high scoring MFCC representations were calculated using values of (ii) ranging from 50–90°. For each of these values of (ii), the corresponding value of (i) that yielded the highest scores was 15 ms.
Figure 5.16: Scores for MFCC representations of Group 3 signals in the SLT database.
5.3.3.4 Discussion

The scores for each phoneme of the Group 1 signals give similar results. They display a trend whereby the highest scoring representations are calculated using $\alpha = 90^\circ$ and a window length of 10 ms. The average pitch period is about 5.7 ms for this speaker, so again, one pitch period is needed to gather enough temporal information from the signal to adequately resolve the constituent frequencies. For Group 2 signals, the FFT generally yields the highest scores also. However, we can see that other FRFT angles yielded high scores as well, a situation not observed for Group 1 signals. Furthermore, there was phoneme-dependent variation observed within Group 2 signals. This was also true for Group 3 signals, although generally the FFT yielded the highest scoring representations here as well.

5.4 Analysis of Results

The scores obtained by 243 MFCC representations (differing from each other due to choices of (i) – (iii) values used in their calculation) were presented in this chapter for three speakers. In this section, we examine the observations across signal Groups and databases.

5.4.1 Cross-speaker comparison

Three speakers were examined in this study. Again, we note the benefits of using the Arctic database. The fact that the prompt list was the same, and the recordings conducted under similar conditions, made it possible for us to use the scoring system to directly compare across speakers. Any differences noted in the ranking of numerical representations are more likely to be speaker-related, rather than database-related. Examining three databases also allows us to explore whether the findings regarding the choices of (i)–(iii) that yield the highest scoring representations are comparable across all speakers. For all three speakers, we noted that the FFT exclusively yielded the highest scoring representations for Group 1 signals, other values of (ii) yielded high scoring representations for Group 2 signals, and
for Group 3 signals, there was too much within-Group variation to determine a trend. For Group 1 signals, the highest scoring representations were calculated using a window length that was large enough to contain one or two pitch periods of the speech data. For Group 2 signals, it was more difficult to determine, but generally, the lower window lengths yielded the highest scoring representations for all the signals examined. Group 3 signals, in some cases, also appeared to be dependent on window length, which is surprising, given the fact that they have no associated periodicity. For all speakers, and across all Groups, the choice of (iii) had no discernible effect on the ranking of the numerical representations. There were some speaker-specific distinctions also observed, especially for Groups 2 and 3. For the SLT database in particular, the FFT-based representations dominated the scores for all signal Groups: that is to say that the presence of turbulence within the signal had less effect on the scores than for other databases. For RMS in particular, the presence of frication in the signals correlated with higher scores from values of (ii) other than the FFT.

5.4.2 Global effects

5.4.2.1 Effects due to (i) window length

We generally found that the highest ranking numerical speech representations were calculated using a window length of at least one to two pitch periods. This corroborates Kirkpatrick's (2010, p. 87) findings that the "optimum windowing strategy" is one which is "pitch synchronous windowing with a window length of one pitch period." For RMS, the average pitch period was just under 10 ms, and the window lengths that yielded the highest ranking representations were 10 ms and 20 ms. BDL had a lower pitch period, of about 7.8 ms, and the value of (i) with which the highest scoring numerical representations were calculated was most often 15 ms. For SLT, a lower pitch period again was observed, corresponding to the higher $f_0$ that is characteristic of female speech. The window length yielding the highest scoring representations in this case was 10 ms, which is long enough to cover one to two 5.6 ms pitch periods. Kirkpatrick (Kirkpatrick 2010) noted that when a window length is smaller than a pitch period, the DFT can no longer adequately resolve the constituent frequencies.
However, the range that we tested in this study did not fall below one pitch period of the speaker with the lowest $f_0$. Overall, there does appear to be a correlation between the pitch period and the window length used to calculate the representations.

5.4.2.2 Effects due to (ii) FRFT angle of rotation

Overall, we see evidence that regardless of the database used, the FFT is the choice of (ii) that yields the highest scoring representation for Group 1 signals. For all three databases, we see that, once turbulence is introduced to the signal, numerical representations calculated using other values of (ii) gain scores that either surpass, or at least contend with those produced using the FFT. The effects of this are more evident for certain speakers, however, and an exact value of (ii) for Group 2 could not be identified as an optimal choice, in the same way that the FFT was for Group 1 signals. This suggests that the optimal angle of the FRFT may be specific to each speaker.

5.4.2.3 Effects due to (iii) filter type

Regardless of the database tested, the effects of changing the value of (iii) were insignificant on the final scores obtained by the representations. We noted in Chapter 3 that, although some differences in MFCC values were introduced due to the choice of this variable, the effects were not as significant as those observed from changing (i) and (ii).

5.5 Conclusion

5.5.1 Chapter review

This chapter began with a discussion about our motivation for conducting this experiment - having established the scoring system, and having highlighted the importance of careful parameter calculation, we saw it necessary to rank our different implementations of the MFCC.
for different types of signals and for different speakers. In Section 5.1, the database was described, and information was provided about the test stimuli for the three speakers. The signal Groups were then defined and discussed. In Section 5.2, the procedure for implementing the scoring system was briefly described. The results for the three signal Groups were presented in Section 5.3 for speakers RMS, BDL, and SLT. The plots for individual phonemes can be found in Appendix B. The results were discussed in Section 5.4.

5.5.2 Discussion

The purpose of this experiment was to examine the effects of particular choices made during the calculation of numerical representations of speech signals. The numerical representations were ranked according to their performance, as dictated by the scoring system. In this manner, we hoped to identify a pattern in the choices for values (i) – (iii) that yielded the highest scoring representations. Furthermore, we hoped to determine whether the same numerical representations scored highly for different speakers and signal types.

We can summarise our findings as follows:

1. The choice of (i) that yields the best numerical representation should be at least one, if not 2 pitch periods for voiced (Group 1 and Group 2) signals.

2. The choice of (ii) that yields the best representation for non-turbulent (Group 1) signals is the FFT ($\alpha = 90^\circ$).

3. The choice of (ii) for turbulent (Group 2 and 3) signals is harder to gauge, and may be phoneme-, database- or speaker-specific.

4. The choice of (iii) has no significant effect on signals from any Group.

In future work, the range of values could be explored further. For example, window lengths less than one pitch period of the database could be examined. Alternatively, the data could be examined pitch synchronously to obtain more exact results. The range of FRFT rotation angles could be explored with greater granularity. These values could also be examined for
a greater distinction of signals to see whether the highest scoring numerical representation differs depending on phonetic context, for example. In the future, we also hope to explore the suggested correlation between $f_0$ and window length, by testing a range of window functions and lengths (as mentioned previously, the effective length of the window is determined by the windowing function), and performing a correlation between the representations of each segment and the mean $f_0$ calculated for each segment.

5.5.3 Concluding remarks

The overall aim of this thesis is to optimise the cost function in a unit selection speech synthesiser. We have already seen how the MFCC values are affected by changes in (i) – (iii), and the important impact this can have on synthesis. In this chapter, we show that certain numerical representations are better at modelling speech sounds than others, according to our scoring system. We also show that the numerical representation that works best for one particular signal may not work for another. We also found that some differences depend on the speaker. The benefits of the study described in this chapter are that we know that while there may be certain universal rules for calculating a useful numerical representation, the optimal choice may be database-specific, at least for some types of signals. In addition, we have devised a framework for determining the worth of a numerical representation.

This piece of research leads to the consideration: if we customise a unit selection cost function for use with a particular speaker, dialect, or even database by using a different numerical representation for different signals or even phonemes within the database, will we be able to produce better sounding synthetic output?
In this chapter, the benefits of optimising the cost function in a unit selection speech synthesiser are demonstrated using a perceptual evaluation. Previously in the thesis, we synthesised target utterances that were contained in the database in order to test how many original segments would be selected using various numerical representations as part of the spectral distance sub-cost. In this chapter, we synthesise the same target utterances, but ensure that they are not contained in the database, so that the unit selection engine must select segments from elsewhere in the database. The utterances are synthesised first using a basic cost function, and again using a cost function that has been modified to include a different numerical representation in the spectral distance sub-cost depending on the type of signal being investigated. A perception test was then conducted where listeners make a forced choice as to which synthesised sentence sounds most like the original utterance.

The chapter begins in Section 6.1 with a review of the common methods of speech synthesis evaluation. In Section 6.2, the unit selection speech synthesiser designed by the author is described, as are the cost function modifications. Section 6.3 describes the details of the experiment, and the results are presented in Section 6.4. The chapter concludes in Section 6.5 with a discussion about further possibilities of cost function personalisation.
6.1 Evaluation Methods

This section provides an overview of some of the types of evaluation that are used to assess the quality of synthetic speech.

6.1.1 Segmental level tests

A segmental level test is designed to test the intelligibility of synthetic speech at the level of the phoneme. The Modified Rhyme Test (MRT) (House et al. 1963) is an example of a segmental level test wherein the user is presented with six words that differ from one another in the initial or final phoneme. One of these words is synthesised and presented to the listener, who is asked to identify which word they are listening to. Examples of the test sets are:

- *peel, reel, feel, eel, keel, heel*

- *late, lake, lay, lace, lane, lame*

Some limitations of the MRT and other segmental tests are that the corpora are fixed and only offer test stimuli that are single consonants in stressed monosyllables appearing next to pauses (Handley and Harnel 2005). Extensions have been proposed including the use of nonsense words with open-ended responses, but they result in time-consuming evaluation procedures and results that are open to interpretation due to the fact that in English and other languages there is usually not a one-to-one correspondence between graphemes and phonemes.

6.1.2 Sentence level tests

Sentence level tests are important for generally rating the overall acceptability of synthetic speech. Intelligibility is rated on the amount of words correctly transcribed by the listener, who has been presented with a synthesised utterance. Examples of sentence level tests
are the Harvard Sentences (Egan 1948) and Haskins Sentences (Nye and Gaitenby 1973), which are often used in conjunction to separate the effects of grammatical and semantic constraints (Schmidt-Nielsen 1992).

**Harvard sentences** are phonetically balanced in that each sentence contains a wide variety of phonemes. The sentences contain five key words, and the intelligibility of the speech is rated by the user's success at identifying these words. The Harvard sentences are a fixed set and contain a wide variety of syntactic structures. Examples include:

- *A chicken leg is a rare dish*
- *The birch canoe slid on the smooth planks*

**Haskins sentences** are grammatically correct but semantically unpredictable sentences. The benefit of using these as test sentences is that the user cannot use context to predict what the individual words are and relies solely on the quality of the synthetic speech to decipher its meaning. Examples include:

- *The old corn cost the blood*
- *The great car met the milk*

Drawbacks of the Haskins and Harvard tests are that they are both fixed sets of sentences and the evaluation can only be carried out once by each listener, after which the semantic novelty of the sentences is diminished.

### 6.1.3 Comprehension tests

In comprehension tests, users listen to synthetic speech and are asked questions on the content. Such methods of evaluation are prone to inherent confounding factors, such as the listener using analytical skills and taking cues from the context of the speech.
6.1.4 Blizzard Challenge

The Blizzard Challenge is an annual event which began in 2004 to test the performance of speech synthesisers working from common datasets by evaluating the synthetic output of the systems. An evaluation procedure was devised in order to test different aspects of the system such as database labelling, pruning, join costs, and signal processing techniques (Black and Toduka 2005). The evaluation first focuses on synthesising sentences from various genres. These are discussed below, as they have been outlined in Black and Tokuda (2005).

**Novels**  The CMU Arctic dataset on which the voices were built consists of sentences from novels, and so the resulting synthesis is more suited to phrases in this style. Sentences from this category were taken from the same novels as those sentences that make up the CMU Arctic database prompt list, but were not included in the release. Examples of this type of sentence are:

- *Joe Garland lives like a good fellow.*
- *But we made no collections of eggs.*

**News**  Test sentences were taken from standard news stories.

- *The two countries agreed to resolve any conflict through diplomacy and avoid the use of force, the agency Interfax said.*

**Conversation**  Conversational sentences that more closely resemble the way that people communicate were also synthesised and evaluated.

- *Okay I would like to go to Miami, Florida.*
- *Yeah I guess it will and something downtown please.*
Phonetically confusable sentences  Words that differ from other common words by a beginning or ending phoneme were chosen and placed in carrier phrases to test the intelligibility of the synthetic speech.

- Now we will say cold again.
- Now we will say pace again.

Semantically unpredictable sentences  These were constructed using a grammatical template det adj noun verb det adj noun. The sentences were difficult to understand as they are semantically nonsensical and the user is unable to use context to distinguish the meaning.

- The unsure steaks overcame the zippy rudder.
- The dank geniuses woke the humane emptiness.

6.1.5 Discussion

The methods described above provide an insight into the measures taken to evaluate the performance of speech synthesis systems by rating the quality of their synthetic speech output. None of the designs above take into account the possibility of comparing synthesised speech to a benchmark standard. Fortunately, in the case of the evaluation required to compare two versions of synthetic output, such a benchmark standard exists in the natural utterances that make up the database. In order to evaluate the synthetic output, a test is proposed whereby sentences are withheld one at a time from the corpus from which the synthesiser selects the synthetic speech segments. The held-out sentence is then entered as a target utterance to the synthesiser. In this situation, the cost function is forced to select segments by choosing from the remaining database sentences. The perceptual experiment design is described fully in Section 6.3.
6.2 Synthesiser

A unit selection speech synthesiser was created in order to carry out the evaluation. The synthesiser was written in the Python programming language and is currently used in the *abair.ie* Irish speech synthesis project (Ni Chasaide et al. 2011). A stripped-down version was previously used in Chapter 4 of this thesis, where only the spectral distance component had any impact of the candidates selected as the synthetic output. For this evaluation, other factors are brought to bear on the choices made during candidate selection. This requires, among other things, the target utterance to be marked-up with more information. The structure of the system is briefly explained in this section.

6.2.1 Software details

The unit selection speech synthesiser was designed using the infrastructure outlined in Hunt and Black (1996), and uses a weighted cost function consisting of target and join sub-costs, and a Viterbi search to select segments for synthetic speech output. The diphone was selected as the base unit, after considering the advantages outlined in Clark et al. (2007) — that the boundaries of the diphone make for suitable concatenation points (during the steady state of the phoneme), and the choice of a fixed unit greatly simplifies the implementation of the Viterbi search. The synthesiser was designed to be modular, and uses a target utterance as input, which is converted to XML format, before being passed to the unit selection system. An example of an XML target utterance is given in Figure 6.1.

The target utterance is read into the candidate selection module, which then searches the database for matching segments. The names and locations of the files containing the candidate segments are returned, as are the locations of the segments within the file in the form of start and end times. A penalty is awarded to each of the candidate segments for every difference between it and the corresponding target segment as defined by the target sub-costs. The manner in which the target costs are applied is shown in Table 6.1.

This information is passed to the Viterbi search module. Every candidate and associated
target cost is arranged in a matrix. The target cost and the join cost are then multiplied by individual weights. The target and join sub-costs are listed in Table 6.2. The energy and $f_0$ values were calculated using Edinburgh Speech Tools (The Festival Speech Synthesis System 2007), as were the MFCCs for the baseline cost function (using a 10 ms window length). The sequence of candidate segments with the lowest cost is calculated using a Viterbi algorithm which backtracks through the matrix as it calculates the costs. The sequence with the lowest cost is chosen as the synthetic output. The names and locations of the sound files that contain the selected segments, along with the start times and end times of the segments, are passed to the next module.

The synthesiser currently uses a straightforward concatenation technique with no smoothing. The locations of the segments in the sound files are parsed and the sound segments are lifted from the database and written to a new sound file, which is played back as the synthesiser output.

Figure 6.1: Example of a target utterance in XML
Table 6.1: Component cost functions for the synthesiser's target cost

<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase</td>
<td>$w_{1}$</td>
<td>Position in phrase is correct (initial, medial or final)</td>
</tr>
<tr>
<td>Stress</td>
<td>$w_{2}$</td>
<td>Stress is correct</td>
</tr>
<tr>
<td>POS</td>
<td>$w_{3}$</td>
<td>Part of speech is correct (noun, verb, modifier, function word, other)</td>
</tr>
<tr>
<td>Syllable</td>
<td>$w_{4}$</td>
<td>Position in syllable is correct (initial, medial, final, between words)</td>
</tr>
<tr>
<td>Word</td>
<td>$w_{5}$</td>
<td>Position in word is correct (initial, medial, final, between words)</td>
</tr>
<tr>
<td>Left</td>
<td>$w_{6}$</td>
<td>Left phonetic context matches</td>
</tr>
<tr>
<td>Right</td>
<td>$w_{7}$</td>
<td>Right phonetic context matches</td>
</tr>
<tr>
<td>Orthographic word</td>
<td>$w_{8}$</td>
<td>Entire word context matches</td>
</tr>
</tbody>
</table>

6.2.2 Cost function modifications

The research described from Chapter 5 suggest that the value choices that result in the calculation of the highest-scoring MFCC representations may depend on the acoustic characteristics of the signal being examined. In some cases, this is more clear-cut, for example the tendency for higher scoring representations to be calculated using a window length dependent on the pitch period of the signal. In other cases it is less clear if a universal rule can be established. For example, it appears that the presence of turbulence in the signal increases the likelihood of a numerical representation based on some angle of the FRFT that lies between the time and frequency domains to achieve a high score. However, this tendency was not observed to the same degree across speakers. We can speculate as to why this is – perhaps the for the SLT speaker, the periodicity of the signal was a more salient aspect than the turbulence, compared to the RMS speaker.

Luckily, the scoring system framework allows us quickly and objectively test which numerical representation is likely to usefully model any group of signals. By changing the spectral distance component of the join cost we can therefore design a cost function that is tailored for different speakers and different types of signals.
For this evaluation, we have modified the cost function of the synthesiser in order to personalise it for the RMS database. The results from Chapter 5 show the following numerical representations achieve high scores for the particular signal Groups for the RMS database:

**Group 1**: MFCCs calculated using (i) 25 ms (ii) 90° (iii) triangular (MFCC.w20.a90)

**Group 2**: MFCCs calculated using (i) 10 ms (ii) 80° (iii) triangular (MFCC.w10.a80)

**Group 3**: MFCCs calculated using (i) 45 ms (ii) 90° (iii) triangular (MFCC.w20.a90)

These numerical representations were used as part of a spectral distance. The cost function was designed so that the relevant spectral distance measure was called upon for the type of units being joined together. When assessing the suitability of a potential concatenation between Group 1 units, for example, the spectral distance component will be calculated using the numerical representation MFCC.w20.a90; for Group 2 units, the numerical representation used will be MFCC.w10.a80; for Group 3 units, the spectral distance will be calculated using MFCC.w10.a80.

For this evaluation the original cost function, used by Festival is referred to as V1, while the second cost function, modified per Group as shown above, is referred to as V2.
6.3 Perceptual Experiment

The perceptual experiment outlined in this chapter explores how the experiments outlined in this thesis contribute to the optimisation of the unit selection cost function. Having reviewed the literature in the area, it is clear that in order to design an evaluation procedure that successfully reflects the user's reactions to the changes made in the system, it must be very clear what exactly it is that needs testing. The issue that needs to be addressed regarding speech synthesis is that the quality of the output is entirely dependent on the number of units in the database. Using the speech synthesis output to measure the success of the modifications effectively installs a glass ceiling on the evaluation results – no matter how much the system is improved, the synthetic output can be no better than the quality of the units in the database. As a result, the synthesised speech will not sound natural, or pleasing, or in some cases very intelligible. The danger of designing a speech synthesis evaluation that asks users to rate mediocre speech output by asking them such questions is that it will prompt the user to guess, leading to non-committal results.

However the speech synthesis output is limited by the number of units in the database, and the similarity of those units. It is important that the evaluation test designed in this chapter tests what the cost function was designed to do – pick the sequence of units that most resemble the target utterance, and that join together as seamlessly as possible. For this reason it was decided that the users should not be forced to think about the subjective and abstract labels of 'naturalness' and 'intelligibility', but instead to just pick the sentence that they prefer, after listening to the natural utterance as a reference.

Additionally, in order to ensure that the test sentences from V1 are as different as possible from those produced by V2, it was ensured that they contained as many phonemes as possible from the three sound groups.
6.3.1 Test sentences

In order to test the contributions made by the cost function modifications, ten sentences were held out from the corpus and evaluated by means of a perceptual experiment. The held-out sentences, the same ones used in Chapter 4 were entered as target utterances to the speech synthesis system. This synthesis was performed twice – the first synthetic output for each sentence was chosen using V1, and the second synthesis was created using V2.

The sentences were carefully chosen so that they contained the greatest available density of sounds from the Groups 1 – 3, so that the joins created using the MFCCs calculated with specifically chosen window lengths would be tested and ensuring that the sentences from V1 were as different as possible from the V2 utterances.

This was achieved by employing a greedy algorithm (Buchsbaum and van Santen 1996) to iteratively search through the Arctic database annotation files and select those sentences that contained the greatest number of sounds from each category. The selected sentences, and information regarding the phonetic distribution, are presented in Table 6.3. The code for the greedy algorithm was written by Harald Berthelsen.

<table>
<thead>
<tr>
<th>idx</th>
<th>Filename</th>
<th>Gr. 1</th>
<th>Gr. 2</th>
<th>Gr. 3</th>
<th>Gr 1–3</th>
<th>Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>arctic.b0244</td>
<td>13</td>
<td>3</td>
<td>4</td>
<td>20</td>
<td>31</td>
<td>64.5 %</td>
</tr>
<tr>
<td>8</td>
<td>arctic.b0538</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>18</td>
<td>28</td>
<td>64.2 %</td>
</tr>
<tr>
<td>9</td>
<td>arctic.b0476</td>
<td>18</td>
<td>9</td>
<td>3</td>
<td>30</td>
<td>47</td>
<td>63.8 %</td>
</tr>
<tr>
<td>6</td>
<td>arctic.b0026</td>
<td>11</td>
<td>5</td>
<td>3</td>
<td>19</td>
<td>28</td>
<td>67.86 %</td>
</tr>
<tr>
<td>10</td>
<td>arctic.a0486</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>26</td>
<td>57.6 %</td>
</tr>
<tr>
<td>5</td>
<td>arctic.a0019</td>
<td>16</td>
<td>5</td>
<td>3</td>
<td>24</td>
<td>47</td>
<td>51 %</td>
</tr>
<tr>
<td>3</td>
<td>arctic.a0007</td>
<td>15</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>40</td>
<td>50 %</td>
</tr>
<tr>
<td>4</td>
<td>arctic.a0009</td>
<td>13</td>
<td>1</td>
<td>5</td>
<td>19</td>
<td>38</td>
<td>50 %</td>
</tr>
<tr>
<td>1</td>
<td>arctic.a0002</td>
<td>15</td>
<td>1</td>
<td>2</td>
<td>18</td>
<td>37</td>
<td>48.6 %</td>
</tr>
<tr>
<td>2</td>
<td>arctic.a0005</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>16</td>
<td>43.7 %</td>
</tr>
</tbody>
</table>

Table 6.3: Percentage of Group 1–3 signal segments in the test stimuli
As predicted, the modifications made to the cost function resulted in different candidate segments being chosen from the database as the synthetic output. The segments chosen by V1 and V2 are given in Appendix C. The purpose of this evaluation is to determine whether the changes made to the cost function resulted in speech synthesis output that was perceptually more like the original utterance than synthesis output from the unmodified unit selection system.

6.3.2 Listening test

The listening test was set up online. The participants were presented with the 10 test sentences. They first listened to the original natural utterance and then the two synthesised versions. The instructions warned users that the utterances sounded very similar and advised that they were to pick the sentence that sounded most pleasing to them, reassuring them that there was no right or wrong answer. Users were then presented with a forced choice between the two synthesised sentences and were given the original utterance as a reference. They were able to listen to each sound file as many times as they wished. Figure 6.2 shows a screen shot from the listening test.

Figure 6.2: Screen shot of part of the online listening test.
6.4 Analysis of Results

The listening test was completed by 40 participants. The overall results are presented in Figure 6.3. The results showed a significant favour for the synthesis from the system using the modified cost function. Preference for the two types of synthesiser differ significantly according to Welsh's t-test, $t(78) = 9.1417$, $p = 5.73e-14$. The mean of the V1 synthesis was calculated at 35% while the mean of the V2 synthesis was 65%. The 95% confidence interval was 36.53% and 23.46%.

![Figure 6.3: Preference for V2 synthesis (modified cost function) compared to V1 synthesis (baseline)](image)

Utterances 6 – 10 contained more phonemes belonging to Groups 1 – 3, which each have a personalised numerical representation. These utterances therefore had a greater chance of showcasing the difference between the two synthesisers, compared to the utterances 1 – 5. The joins created in utterance 1 – 5, on the other hand, while still using the MFCCs specific
for Groups 1 – 3, did not get as many opportunities to select segments based on these new values, and therefore the synthesis is expected to be more similar to that produced by V1. As predicted, the utterances that had been especially selected based on the density of phonemes present from the three sound groups were rated higher overall. Figure 6.4 displays the preference for V1 synthesis compared to the preference for V2 synthesis for each sentence in the listening test. The results for utterances 1 – 5, while still showing a preference for V2 synthesis, reflect the similarity between the two versions. On the other hand, it is clear from the significant preference for V2 versions of utterances 6 – 10 that the synthesised versions were quite different.

Figure 6.4: Preference for utterances synthesised using V2 (modified synthesiser) compared to V1 (baseline)
6.5 Conclusion

In this section, we conclude with a review of the experiment carried out, a discussion about its implications, and some concluding remarks.

6.5.1 Chapter review

In this chapter, a perceptual evaluation was carried out in order to assess the impact of the choices of numerical representation on the synthetic speech output of a unit selection system. This differed from the experiment carried out in Chapter 4 where the target utterance was contained in the database - in this experiment, the target utterance was held out from the database so the synthesiser was forced to select output units from elsewhere in the database. The synthesis was performed twice - once using a baseline unit selection speech engine, like that used by the Festival system, and the other using one of three different numerical representations depending on the type of signals being considered for concatenation. The numerical representations were chosen as directed by the scoring system outlined in Chapter 4 for the RMS database. We predicted that a cost function that uses a modified spectral distance component would return more favourable synthetic output. The results of the evaluation confirmed this. In this chapter, we also presented a review of evaluation techniques and discussed the motivation behind the design we chosen for this experiment.

6.5.2 Discussion

The results presented here show a significant favour for the synthesised utterances produced by the synthesiser using the modified cost function. Furthermore this preference is stronger for utterances that have a greater opportunity to use the modified cost function i.e. utterances that contain more speech sounds in the three groups catered for throughout this thesis.
The results presented in this chapter very strongly demonstrate:

1. The importance of carefully chosen values for fundamental parameters, when calculating numerical speech representations, and the impact of these choices on the selection of segments by a unit selection cost function.

2. The benefit of tailoring the cost function for different types of signals contained in databases used for unit selection speech synthesis.

3. The contribution of a scoring system that can advise on the numerical representation to use for a given signal.

The results in Chapter 3 displayed the importance of values (i) – (iii) on MFCC calculation, and in Chapter 4, the impact of the choice of numerical representation on synthetic output was demonstrated. The results presented in this chapter reinforce these findings, and further support the use of the scoring system as an objective method of assessing the efficacy of an objective distance measure.

The results are particularly useful as they demonstrate the real world value of rigorous and defensible choices of fundamental values like window size. It is common to underestimate the importance of these choices, especially in a discipline like speech technology which focusses heavily on perceptual results and end-user applications. While the effects of windowing are well-known to those interested in signal processing, they are rarely taken into account when designing speech technology applications, as discussed in Chapter 2.

6.5.2.1 Weighting the Cost Function

The configuration of weights in a unit selection system controls the relative contribution of the target and concatenation sub-costs to the total cost and therefore has a direct impact on what units are chosen by the Viterbi search, and on the resulting synthesised speech. By setting the weights, the user is essentially telling the unit selection speech engine what acoustic and linguistic characteristics of the speech segments are perceptually salient, and
the extent to which their importance outweighs that of other characteristics. For example, if stress is thought to be the most important characteristic of a speech segment, setting a large weight to that sub-cost will ensure that all candidate units with a level of stress different from that of the target unit will incur such high penalties that they will most likely not be chosen in the final unit sequence. In the synthesis experiment of Chapter 4, we demonstrated the effects of the choice of numerical representation on the spectral distance component. We pointed out that if the spectral distance component is used in conjunction with other costs in the cost function, then the contribution to the overall cost incurred by a candidate sequence may be diminished. Furthermore, results from Chapter 5 show that due to the non-periodic nature of Group 3 signals, there can be no reliance on \( f_0 \) in the calculation of the representative feature set, and that setting equal weights for \( f_0 \) for these kinds of signals is counter-intuitive. A fully personalised cost function should take into accounts the weights applied to each of the contributing cost components.

The weight system is therefore an important element of any speech synthesis machine and many different methods have been used to determine the optimal weight configuration. These include hand tuning, a method used by the Festival multisyn system, for which the weights are set to equal values for the join cost, and set arbitrarily for the target cost and hand tuned to produce an acceptable level of synthesis (Clark, Richmond and King 2004). Other methods include the weight space search (Hunt and Black 1996), regression training (Hunt and Black 1995), perceptual training (Lee 2001, Wouters and Macon 2000, Toda, Kawai and Tsuzaki 2004), to name a few. Evolutionary methods have also been applied to this problem. Preliminary tests carried out during this research using a genetic algorithm, have returned mixed results, despite promising results from previous studies (Alias and Llorà 2003). Further research into cost function weighting is an exciting avenue for future research in this field.
6.5.3 Concluding remarks

The results presented in this chapter reinforce the conclusions reached in Chapter 3: that the choice of fundamental parameters used in calculating the numerical representations of speech signals have a direct effect on end-user applications for speech technology, and that the cost function in the unit selection speech synthesiser can benefit from modifications tailored for particular types of speech signals.

The results here further endorse the scoring system proposed in Chapter 4, and the demonstrated benefits of modifying the cost function based on speaker, dialect, signal type, and even phoneme, highlight a need for an objective method of assessing which numerical representations would be beneficial to use in such cases.
CHAPTER 7

Conclusion

The primary focus of this thesis was the optimisation of the cost function of a unit selection speech synthesiser. This was attempted by examining the numerical representation of speech signals, which plays a role in the spectral distance component of the join cost.

7.1 Motivation

Previous studies have attempted to ascertain which of any number of numerical representations best represents a speech signal. This was generally done by performing perceptual evaluations whereby a listener provides an opinion on the similarity of two signal segments. The conclusions reached by the many studies performed were contrasting and impossible to compare for the following reasons:

- The segments examined in the studies were generally extracted from a database designed and recorded especially for each study in question. Although some of the databases were shared between research groups, they were generally not freely available to the general academic community.
• The nature of the perceptual tests varied. In some, listeners were asked to rate the salience of a concatenation on a scale of 1–5. In others, listeners were simply asked if they could detect a concatenation point.

• The manner by which the perceptual data was compared to the numerical calculations differed per study, with some opting for a Pearson’s correlation coefficient, and other using ROC curves.

• The numerical representations tested by each study are sometimes unclear, and in certain cases scant details have been provided how each has actually been calculated. The inability to compare these studies has resulted in numerous studies with contrasting results. Furthermore, the research cannot be repeated, as the details of the numerical representation under investigation are not disclosed, the results of the perceptual testing are unclear, and the database on which it was performed is not available. This is not to deny that the studies themselves are rigorously performed. However, the contrasting results are suggestive and raise the following points:

  • Researchers have assumed that there is one numerical representation of speech that suits all speakers and databases. The previous experiments tested on one database are assuming that the results of that study suggest the existence of one numerical representation to best represent all signals. In fact, the contrasting results could, among other things, suggest that certain numerical representations are more suited to some databases or signal types.

  • The values comprising the numerical representations themselves are entirely dependent upon several factors bearing upon their calculation. Choices of these values will result in a different set of values for the numerical representation. The possibility exists that the choice of values that yields a meaningful numerical representation may be a function of the signal, governed by the signal’s characteristics, such as the $f_0$, the position of the formants, or the particular idiosyncrasies of the speaker.
Instead of conducting perceptual tests on particular sets of data, it would be useful to have a tool that would be able to predict which numerical representations would best represent a signal. The benefits of such a tool would be the lack of dependency on a particular set of perceptual data. The tool would be useable on all signals and databases, and the results that it produced would be comparable across studies.

In this thesis, we have attempted to:

1. demonstrate the effects that certain parameter choices have on the calculation of numerical speech signal representations, and ultimately on the synthetic output of a unit selection speech synthesiser.

2. develop a method of rating numerical representations in terms of their usefulness in a way that is independent of perceptual testing.

3. apply the devised scoring system to large datasets to establish whether there is any relationship between particular value choices and the signal under consideration, and whether these dependencies hold across speakers.

4. demonstrate that unit selection speech synthesis output can be improved if we account for differences between signals.

### 7.2 Summary of thesis

This study was broadly motivated by the need to improve the quality synthetic output of unit selection speech synthesis systems by lessening the characteristic concatenation errors. The cost function is the module of the unit selection synthesiser that is responsible for deciding which candidate units are chosen as the synthetic output. The experiments were carried out to update the cost function by altering the spectral distance measure used in the join cost to suit the types of sounds under examination. There were three main reasons why it was assumed that this would result in better synthetic speech output:
1. The acoustic characteristics of speech signals differ greatly depending on the phonetic category to which they belong. Vowels are characterised by their periodicity, voiceless fricatives by their turbulence, and voiced fricatives by both turbulence and periodicity. This raised the question of whether the signals should be treated in the same manner in the unit selection cost function.

2. There is no consensus as to which spectral distance measure, used in the join cost, best represents a speech signal. The contrasting results of previous studies suggested that perhaps there was no universal numerical representation that best suits all databases and speakers. By comparing the methods of parameter extraction, it was anticipated that a certain dependency of some fundamental parameters on the characteristics of the signal could be identified.

3. The development of an objective rating system would allow for the quick and reproducible assessment of different numerical representations of speech signals for any part of a database, allowing for a personalised spectral distance measure component of the join cost. In this way, the cost function of the unit selection speech synthesiser could be modified for improved synthetic speech output.

The study consisted of a number of experiments which are summarised in this section.

7.2.1 MFCC calculation

The MFCC was chosen as an example of a numerical speech representation. It was asserted that various choices made in the calculation of the MFCC would affect the values of the calculated numerical representation. The three different parameters examined were:

(i) the window length used to extract the MFCC values,

(ii) the time-frequency analysis performed on the signal,

(iii) the shape of the filters in the mel filter bank.
7.2. Summary of thesis

The effects of these three choices were discussed and we considered the implications of changing their values and the potential impact on the calculation of the MFCC representation. This was explored further by mathematically demonstrating the effects of these changes on cosinusoid signals. Finally, the effects were demonstrated experimentally for a real speech signal. By varying parameters (i) – (iii) over a closed range of values, we demonstrated that 243 different numerical representations could be calculated to represent the same database unit. The differences in the values of the MFCC representations were noted. The role of the numerical representation as part of the spectral distance component of the join cost was discussed, and we asserted that the differences in values calculated for the MFCC representation of a speech signal would impact upon the synthetic output of a unit selection speech synthesiser.

7.2.2 The scoring system

A method was then proposed for determining which of any given selection of numerical representations would best model a speech signal. The definition of a ‘useful’ numerical representation was ascertained as one whose values would be similar for similar-sounding signals, and very different for different sounding signals. The scoring system was based on predictions about the spectral distances that would be calculated between speech signals. In order to bypass the need for perceptual testing, the speech signals, whose numerical representations we would compare, were carefully chosen, and placed in three Categories:

Category A: naturally sequential segments of speech from the same signal, within the same phoneme.

Category B: one segment of interest compared to a generic model of a similar segment that is not of the same phoneme or signal.

Category C: two segments from the same phoneme and signal, which are separated by a small distance so that they are not naturally consecutive.
The predictions made about the values of these distances relative to one another became the basis of the scoring system, presented here again:

\[
\text{score} = \begin{cases} 
  d_{B-A} + d_{B-C}, & \text{for } d_{B-A} > d_{C-A} \geq 0 \\
  d_{C-A}, & \text{for } d_{C-A} < 0,
\end{cases}
\] (7.1)

where

\[
\begin{align*}
  d_{B-A} &= \text{dist}_B - \text{dist}_A & (7.2) \\
  d_{B-C} &= \text{dist}_B - \text{dist}_C & (7.3) \\
  d_{C-A} &= \text{dist}_C - \text{dist}_A & (7.4)
\end{align*}
\]

and \(\text{dist}_A, \text{dist}_B,\) and \(\text{dist}_C\) are the Euclidean distances calculated between numerical representations of Category A, B, and C segments respectively. The use of the scoring system was then demonstrated, and three different numerical representations – two MFCC-based and one LPC-based – were assessed. Having ranked the representations in order of their efficacy, we could then demonstrate the effects of the choice of numerical representation on the output of a unit selection speech synthesiser.

An experiment was carried out to demonstrate these effects. A unit selection engine was built and modified so that it selected segments based solely on the spectral distance component of the join cost. It was ensured that the target utterances used as input were contained in their entirety within the database used for synthesis. The purpose of the experiment was to demonstrate that the choice of numerical representation could affect the synthesiser’s choice of output segments. The results showed that this was indeed the case, and different segments were chosen depending on the numerical representation used in the cost function. Furthermore, it was demonstrated that the higher the score awarded to the numerical representation by the scoring system, the higher the percentage of output segments were lifted directly from the target utterance’s corresponding original utterance in the database.

In an analysis of the scoring system, we discussed the assumptions made in the creation of the scoring system, and the limitations that were the result of the predictions made. We
also discussed the contributions of the scoring system, and explained that the framework of a scoring system itself is of great value to unit selection speech synthesis as it allows fast and objective assessment of the suitability of a numerical representation to a collection of speech signals.

7.2.3 Data analysis

The scoring system was then used to rank the 243 different MFCC representations previously defined. The aims of the experiment were:

1. to ascertain whether the same numerical representations ranked in a similar fashion for all signals examined.

2. to determine whether the numerical representations ranked in a similar way for more than one speaker.

3. to investigate whether there was a relationship between certain characteristics of a signal and the values (i) – (iii) used in the calculation of the higher ranking MFCC representations for that signal.

The signals examined belonged to the following Groups:

Group 1: non-turbulent, periodic sounds – vowels and diphthongs.

Group 2: turbulent, periodic sounds – voiced fricatives.

Group 3: turbulent, non-periodic sounds – voiceless fricatives.

and were examined for three American English speakers of the Arctic database – RMS (male), BDL (male), and SLT (female). The results showed some similarities across speakers, such as high scoring representations calculated using the FFT for (ii), and no effect on the scores as a result of changing the value of (iii). Effects due to differences in the value of (i) differed across speakers, but in a way that was dependent on the $f_0$ of the signal. There were also some differences between the results for each signal type. Generally, values of
(ii) other than the FFT yielded high scoring representations for signals from Groups 2 and 3. However, the ranking of these scores compared to those attained by FFT-based representations differed across speakers. It was concluded that the best performing numerical representation may be database specific, and more than one numerical representation may need to be used in a cost function in order to improve output synthesis.

7.2.4 Perceptual test

In order to demonstrate the benefits of a tailored cost function, an evaluation procedure was designed to test the effects of using a different numerical representation for each sound group. The cost function was tailored to each by choosing a high scoring representation for each of the three sound Groups, based on the results calculated for the RMS database in Chapter 5. The chosen numerical representations for each Group were:

**Group 1**: MFCCs calculated using (i) 25 ms (ii) 90° (iii) triangular (MFCC.w20.a90)

**Group 2**: MFCCs calculated using (i) 10 ms (ii) 80° (iii) triangular (MFCC.w10.a80)

**Group 3**: MFCCs calculated using (i) 45 ms (ii) 90° (iii) triangular (MFCC.w20.a90)

Half of the ten test sentences were chosen to contain the highest density of sounds from the three sound groups under investigation. The results showed that test utterances synthesised using a modified cost function (V2) were preferred by 65% to 35% to the baseline synthesiser (V1). Furthermore the sentences that were most significantly preferred were the ones that contained the most phonemes from the three sound groups, and therefore the ones in which the joins were mostly calculated using MFCC values specific to the signal. The results strongly support the personalisation of the cost function as a method of improving the synthetic output of the unit selection speech synthesiser.
7.3 Conclusions

The crucial findings of this thesis are as follows:

- Previous studies investigating objective spectral distance measures are impossible to cross-compare due to differences in:
  - the test databases used to carry out the experiments,
  - the choices of values used to extract the parameters, and
  - the nature of the perceptual experiments used to relate the numerical representations to human perception.

- Certain choices made during the calculation of a numerical representation will directly affect the values that constitute the numerical representation. Of the values tested in this thesis, it was found that:
  - The window length used to extract the MFCCs has a significant bearing on their values.
  - The angle of rotation of the FRFT used during MFCC calculation significantly affects their values.
  - The choice of filter used to convert frequencies to the mel scale has an insignificant effect on the values of the MFCCs.

- A spectral distance measure, consisting of a numerical representation and a distance metric, was shown to return different values depending on the numerical representation used. This resulted in different segments being selected as the synthetic output of a join cost function, demonstrating the importance of the numerical representation (and the choices made during its calculation) for speech synthesis.

- The scoring system was created to rank numerical representations in order of their efficacy. The scores awarded to the numerical representations correlated with the per-
percentage of selected segments lifted directly from the target utterance's corresponding original utterance in the database, demonstrating that the scoring system does in fact rank the numerical representations in a manner that is useful to unit selection speech synthesis. The assumptions made during its derivation were explained and justified. The benefits of developing an objective, flexible and reproducible scoring system were stressed.

- According to the scoring system, the values of (i) – (iii) that result in the calculation of the highest-scoring MFCC representations depend on the acoustic characteristics of the signal being examined:
  - Group 1: the highest scoring representation was calculated using a window length that was a function of $f_0$ and the FFT.
  - Group 2: the highest scoring representation was calculated using a FRFT angle other than the FFT whose exact value differed per phoneme.
  - Group 3: the highest scoring representation was different for most phonemes.

- The highest scoring representations also differed depending on speaker.
  - RMS: numerical representations calculated using the FFT were outranked by those calculated using other values of the FRFT for Group 2 signals.
  - BDL: the non-FFT based representations scored significantly higher for Group 2 than for Group 1, but whether the FFT-based ones outranked them was dependent on the phoneme in question.
  - SLT: the FFT-based representations scored highest but were outranked by the non-FFT-based ones for all Group 1 and Group 2 signals; however, for some phonemes in Group 2, the non-FFT-based representations scored higher than they did for Group 1 signals.

- A perceptual test was carried out in order to rate the synthesis resulting from using
Festival's straightforward model of the cost function (V1) compared to the synthesis resulting from a cost function modified by using a different numerical representation for each signal Group, as determined by the scoring system (V2). The evaluation showed that:

- The test utterances synthesised using V2 were significantly preferred by the listeners to the same utterances synthesised using V1.

- In particular, the utterances which relied most on numerical representations particular to the signal Groups 1–3 were preferred more than those that contained less phonemes from the sound categories.

Overall we have demonstrated:

1. The importance of carefully chosen values for fundamental parameters when calculating numerical representations of speech signals, and the impact of these choices on the selection of segments by a unit selection cost function.

2. The benefit of tailoring the cost function for different types of signals contained in databases used for unit selection speech synthesis.

In conclusion, it has been shown that optimising the cost function based on the findings in this thesis will produce better unit selection speech synthesis.

### 7.4 Future Work

This section outlines the topics in this thesis that present promising opportunities for future research.

#### 7.4.1 Calculation of numerical representations

There are many parameter choices that affect the calculation of the MFCC:
(i) the window length used to extract the MFCC values,

(ii) the time-frequency analysis performed on the signal,

(iii) the shape of the filters in the mel filter bank,

(iv) the number of filters,

(v) the spacing of filters,

(vi) the way in which the power spectrum is warped (Zheng et al. 2001).

This thesis explored the effects of changing the values (i) – (iii). Future research could focus on examining how changes in the other parameter values may affect the ability of the MFCC to accurately represent a speech signal.

7.4.2 Time-frequency analysis

Previous findings (Kirkpatrick 2010) showed that spectral distance measures based on the wavelet transform outperformed Fourier transform-based measures as the use of wavelets was thought to provide a solution to the limitation imposed by the time-frequency trade-off. Our research found that a fractional Fourier approach performed in a similar way for turbulent signals. This presents an opportunity for further research, and we are currently investigating the use of the wavelet transform and the Wigner-Ville distribution on speech signals.

7.4.3 Software development

The synthesiser designed for testing the cost function modification techniques could benefit from further development. In particular, the synthetic speech output may be improved by catering for another group of speech sounds, non-periodic, non-turbulent, transient sounds such as stops and affricates. These sounds have a natural pause and release phase. Adjusting the cost function to ensure that these units are joined during the silence portion would
reduce the number of audible clicks that greatly diminish the perceptual quality of synthetic speech. Another enhancement that could be made to the speech synthesiser is the technique used for concatenating sound segments. As it stands, no smoothing techniques are applied during this phase, but cross-fading between segments may results in the reduction of audible discontinuities.

7.4.4 Cost function weighting

The impact of the spectral distance component of the join cost on the synthetic speech output is determined by a weighting system. We have already carried out some research in this area (Kelly 2010, Kelly, Berthelsen and Gobli 2012) outlining an experiment whereby weights are set using a genetic algorithm as an optimisation function. The results to date have been inconclusive, and the scope of the experiment has been broadened to incorporate other optimisation algorithms, such as the Nelder–Mead method (Nelder and Mead 1965) and artificial neural networks.


Kominek, J. and Black, A. W.: 2003, *CMU Arctic databases for speech synthesis*, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.


Syrdal, A. K.: 2001, Prosodic effects on listener detection of vowel concatenation, *Proceed-


APPENDIX A

Category rankings for spectral distances
Figure A.1: Distances calculated between naturally occurring consecutive speech segments.
Figure A.2: Distances calculated between naturally occurring consecutive speech segments.

Euclidean Distance

\( \alpha = 10^\circ \)

\( \alpha = 20^\circ \)

\( \alpha = 30^\circ \)

\( \alpha = 40^\circ \)

\( \alpha = 50^\circ \)

\( \alpha = 60^\circ \)

\( \alpha = 70^\circ \)

\( \alpha = 80^\circ \)

\( \alpha = 90^\circ \)

Rectangular fillers

normalised distance between MFCC vectors

window length (ms)

normalised distance between MFCC vectors

window length (ms)

normalised distance between MFCC vectors

window length (ms)

normalised distance between MFCC vectors

window length (ms)

normalised distance between MFCC vectors

window length (ms)

normalised distance between MFCC vectors

window length (ms)

Cat A
Cat B
Cat C
Cat D
Figure A.3: Distances calculated between naturally occurring consecutive speech segments.
APPENDIX B

Individual phoneme data

B.1 RMS database

B.1.1 Group 1
Mean f0 readings for examples of /iy/ in the RMS database

![Graph showing mean f0 readings for /iy/ examples in the RMS database]
Figure B.2: Scores for MFC representations of /I/ in the RMS database.
Mean f0 readings for examples of /ih/ in the RMS database
Figure B.4: Scores for MFCC representations of /i/ in the RMS database

Scores for vowel /IH/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Scores for vowel /IH/ representations

Rectangular filters

Gaussian filters
Figure B.5: Scores for MFCC representations of /ɛ/ in the RMS database
Figure B.6: Scores for MFCC representations of /EH/ in the R1MS database
Mean f0 readings for examples of /ey/ in the RMS database
Figure B.8: Scores for IV IFCC representations of /e/ in the RMS database

- Triangular filters
  - $\alpha = 10^\circ$
  - $\alpha = 20^\circ$
  - $\alpha = 30^\circ$
  - $\alpha = 40^\circ$
  - $\alpha = 50^\circ$
  - $\alpha = 60^\circ$
  - $\alpha = 70^\circ$
  - $\alpha = 80^\circ$
  - $\alpha = 90^\circ$

- Rectangular filters

- Gaussian filters

Scores for vowel /EY/ representations

Rectangular filters

Gaussian filters

0 10 15 20 25 30 35 40 45 50

window length (ms)
Mean f0 readings for examples of /æl/ in the RMS database

Figure B.9: Mean f0 measurements of segments of /æl/ in the RMS database
Figure B.10: Scores for MFCC representations of /æ/ in the RIS database.
Mean f0 readings for examples of /ao/ in the RMS database
Figure B.12: Scores for MFCC representations of /a/ in the RMS database

- Triangular filters
- Rectangular filters
- Gaussian filters
Figure B.13: Mean f0 measurements of segments of /aa/ in the RMS database

Mean f0 readings for examples of /aa/ in the RMS database

- `f0`: Mean
- `std`: Standard deviation
- `mean`: Mean

```
<table>
<thead>
<tr>
<th>no. of signals</th>
<th>f0 (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>105</td>
</tr>
<tr>
<td>50</td>
<td>110</td>
</tr>
<tr>
<td>100</td>
<td>115</td>
</tr>
<tr>
<td>150</td>
<td>120</td>
</tr>
<tr>
<td>200</td>
<td>125</td>
</tr>
<tr>
<td>250</td>
<td>130</td>
</tr>
</tbody>
</table>
```
Figure B.14: Scores for MFCC representations of \( /y/ \) in the RMS database.
Figure B.15: Mean f0 measurements of segments of /ow/ in the RMS database
Figure B.16: Scores for MFCC representations of /o/ in the RMS database.

Scores for vowel /OW/ representations

Triangular filters

Rectangular filters

Gaussian filters

Window length (ms)
Figure B.17: Mean f0 measurements of segments of /ah/ in the RMS database.

Mean f0 readings for examples of /ah/ in the RMS database.

- **f0**
- **std**
- **mean**
Figure B.18: Scores for MFCC representations of /a/ in the RMS database

- Triangular filters
- Rectangular filters
- Gaussian filters

Scores for vowel /AH/ representations

Window length (ms)
Figure B.19: Mean f0 measurements of segments of /uh/ in the RMS database

Mean f0 readings for examples of /uh/ in the RMS database

- f0
- std
- mean

no. of signals
Figure B.21: Mean f0 measurements of segments of /uw/ in the RMS database

Mean f0 readings for examples of /uw/ in the RMS database

- f0
- std
- mean

no. of signals

0 20 40 60 80 100 120 140 160 180 200

85 90 95 100 105 110 115
Triangular filters

Scores for vowel /UW/ representations

Rectangular filters

Gaussian filters

Figure B.22: Scores for MFCC representations of /u/ in the RMS database
Mean f0 readings for examples of /ay/ in the RMS database

Figure B.23: Mean f0 measurements of segments of /ai/ in the RMS database
Figure B.24: Scores for MFCC representations of /ai/ in the RMS database.
Figure B.25: Mean /aw/ measurements of segments of /aw/ in the RMS database

Mean f0 readings for examples of /aw/ in the RMS database

- f0
- std
- mean

no. of signals

f0
Figure B.26: Scores for MFCC representations of /au/ in the RMS database.

Scores for diphthong /AW/ representations

Triangular filters

Rectangular filters

Gaussian filters

$\alpha = 10^\circ$

$\alpha = 20^\circ$

$\alpha = 30^\circ$

$\alpha = 40^\circ$

$\alpha = 50^\circ$

$\alpha = 60^\circ$

$\alpha = 70^\circ$

$\alpha = 80^\circ$

$\alpha = 90^\circ$

window length (ms)

window length (ms)

window length (ms)
Mean f0 readings for examples of /oy/ in the RMS database

![Graph showing mean f0 readings for /oy/ in the RMS database. The x-axis represents the number of signals, and the y-axis represents f0 values ranging from 90 to 120. The graph includes lines for mean, std, and f0.]
Figure B.28: Scores for MFCC representations of /OY/ in the RMS database
B.1.2  Group 2
Figure B.29: Mean \( f_0 \) measurements of segments of /\( \delta \)/ in the RMS database
Figure B.30: Scores for voiced fricative /DH/ representations of /θ/ in the RMS database.
Figure B.31: Mean $f_0$ measurements of segments of /v/ in the RMS database
Figure B.32: Scores for voiced fricative /v/ in the RMS database

Scores for voiced fricative /v/ representations

Triangular filters

Rectangular filters

Gaussian filters
Mean f0 readings for examples of /zz/ in the RMS database

Figure B.33: Mean f0 measurements of segments of /z/ in the RMS database

- f0
- std
- mean

Mean f0 readings for examples of /zz/ in the RMS database

<table>
<thead>
<tr>
<th>No. of signals</th>
<th>f0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>85</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>200</td>
<td>95</td>
</tr>
<tr>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>400</td>
<td>105</td>
</tr>
<tr>
<td>500</td>
<td>110</td>
</tr>
<tr>
<td>600</td>
<td>115</td>
</tr>
</tbody>
</table>

B. INDIVIDUAL PHONEME DATA
Figure B.34: Scores for MFCC representations of /z/ in the RMS database.
Mean f0 readings for examples of /zh/ in the RMS database
Figure B.36: Scores for MFCC representations of /ZH/ in the RMS database.
B.1.3 Group 3
Scores for voiceless fricative /θ/ representations

Triangular filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Rectangular filters

Gaussian filters

Figure B.37: Scores for MFCC representations of /θ/ in the RMS database
Figure B.38: Scores for MFCC representations of /f/ in the RMS database.
Figure B.39: Scores for voiceless fricative /s/ in the RMS database

Scores for voiceless fricative /s/ representations

- Triangular filters
- Rectangular filters
- Gaussian filters

- \( \alpha = 10^\circ \)
- \( \alpha = 20^\circ \)
- \( \alpha = 30^\circ \)
- \( \alpha = 40^\circ \)
- \( \alpha = 50^\circ \)
- \( \alpha = 60^\circ \)
- \( \alpha = 70^\circ \)
- \( \alpha = 80^\circ \)
- \( \alpha = 90^\circ \)

Window length (ms)
Figure B.40: Scores for MFCC representations of /ʃ/ in the RMS database

Scores for voiceless fricative /ʃ/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Score

Score

Score

Window length (ms)

Window length (ms)

Window length (ms)
B.2  BDL database

B.2.1  Group 1
Figure B.41: Mean /o/ measurements of segments of /iy/ in the BDL database

Mean f0 readings for examples of /iy/ in the BDL database
Figure B.42: Scores for MFCC representations of /i/ in the BDL database

<table>
<thead>
<tr>
<th></th>
<th>Triangular filters</th>
<th>Scores for vowel /IY/ representations</th>
<th>Rectangular filters</th>
<th>Gaussian filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter Type</td>
<td>α = 10°</td>
<td>α = 20°</td>
<td>α = 30°</td>
<td>α = 40°</td>
</tr>
<tr>
<td>Window Length</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>Score</td>
<td>-1.0</td>
<td>-0.5</td>
<td>0.0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The graphs show the variation of scores for different filter types and window lengths.
Mean f0 readings for examples of /ih/ in the BDL database

Figure B.43: Mean f0 measurements of segments of /i/ in the BDL database
Figure B.44: Scores for MFC representations of /i/ in the BDL database
Figure B.45: Mean $f_0$ measurements of segments of /ɛ/ in the BDL database
Figure B.46: Scores for MFC representations of /e/ in the BDL database.
Figure B.47: Mean /o/ measurements of segments of /e/ in the BDL database

Mean f0 readings for examples of /ey/ in the BDL database
Figure B.4b: Scores for MFCC representations of /e/ in the BDL database.
Figure B.49: Mean f0 measurements of segments of /æl/ in the BDL database

Mean f0 readings for examples of /æl/
in the BDL database

- f0
- std
- mean

no. of signals

0 50 100 150 200 250 300 350 400 450

110 115 120 125 130 135 140 145 150
Figure B.50: Scores for MFCC representations of /æ/ in the BDL database.

- Triangular filters
- Rectangular filters
- Gaussian filters
Figure B.51: Mean f0 measurements of segments of /aol/ in the BDL database

Mean f0 readings for examples of /aol/ in the BDL database

- f0
- std
- mean

no. of signals

0 50 100 150 200 250 300

105 110 115 120 125 130 135 140 145 150
Figure B.52: Scores for MFCC representations of /a/ in the BDL database.

**Triangular filters**
- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

**Rectangular filters**

**Gaussian filters**

Scores for vowel /AO/ representations
Figure B.53: Mean /a/ measurements of segments of /al/ in the BDL database

Mean f0 readings for examples of /aa/ in the BDL database

- f0
- std
- mean

no. of signals

0 50 100 150 200 250

110 115 120 125 130 135 140 145

266 B. INDIVIDUAL PHONEME DATA
Figure B.55: Mean $f_0$ measurements of segments of /o/ in the BDL database
Figure B.56: Scores for MFCC representations of /o/ in the BDL database.

Scores for vowel /OW/ representations

Triangular filters

Rectangular filters

Gaussian filters
Figure B.57: Mean $f_0$ measurements of segments of /ʌ/ in the BDL database
Figure B.58: Scores for MFCC representations of /A/ in the BDL database.
Figure B.59: Mean $f_0$ measurements of segments of /u/ in the BDL database
Figure B.60: Scores for MFCC representations of /i/ in the BDL database.

**Triangular filters**

- \( \alpha = 10^\circ \)
- \( \alpha = 20^\circ \)
- \( \alpha = 30^\circ \)
- \( \alpha = 40^\circ \)
- \( \alpha = 50^\circ \)
- \( \alpha = 60^\circ \)
- \( \alpha = 70^\circ \)
- \( \alpha = 80^\circ \)
- \( \alpha = 90^\circ \)

**Rectangular filters**

- \( \alpha = 10^\circ \)
- \( \alpha = 20^\circ \)
- \( \alpha = 30^\circ \)
- \( \alpha = 40^\circ \)
- \( \alpha = 50^\circ \)
- \( \alpha = 60^\circ \)
- \( \alpha = 70^\circ \)
- \( \alpha = 80^\circ \)
- \( \alpha = 90^\circ \)

**Gaussian filters**

- \( \alpha = 10^\circ \)
- \( \alpha = 20^\circ \)
- \( \alpha = 30^\circ \)
- \( \alpha = 40^\circ \)
- \( \alpha = 50^\circ \)
- \( \alpha = 60^\circ \)
- \( \alpha = 70^\circ \)
- \( \alpha = 80^\circ \)
- \( \alpha = 90^\circ \)
Mean f0 readings for examples of /uw/ in the BDL database

Figure B.61: Mean f0 measurements of segments of /u/ in the BDL database
Scores for vowel /UW/ representations

Triangular filters
Rectangular filters
Gaussian filters

Figure B.62: Scores for MFCC representations of /u/ in the BDL database
Figure B.63: Mean f0 measurements of segments of /ai/ in the BDL database

Mean f0 readings for examples of /ay/ in the BDL database
Figure B.64: Scores for MFCC representations of /ai/ in the BDL database

Scores for diphthong /Ay/ representations

Triangular filters

Rectangular filters

Gaussian filters
Figure B.65: Mean /o/ measurements of segments of /aw/ in the BDL database

Mean f0 readings for examples of /aw/ in the BDL database

![Graph showing mean f0 readings](image-url)
Figure B.66: Scores for MFCC representations of /au/ in the BDL database.
Mean f0 readings for examples of /oy/ in the BDL database
Figure B.68: Scores for MFCC representations of /ɔ/ in the BDL database
B.2.2 Group 2
Figure B.69: Mean f0 measurements of segments of /dh/ in the BDL database

Mean f0 readings for examples of /dh/ in the BDL database

- f0
- std
- mean

150 -
145 -
140 -
135 -
130 -
125 -
120 -
115 -
110 -

0 10 20 30 40 50 60 70

no. of signals
Figure B.70: Scores for voiced fricative /DH/ representations of /ɣ/ in the BDL database.

Scores for voiced fricative /DH/ representations

Triangular filters

Rectangular filters

Gaussian filters

Window length (ms)
Figure B.71: Mean f0 measurements of segments of /v/ in the BDL database

Mean f0 readings for examples of /v/ in the BDL database
Scores for voiced fricative /\textit{v}/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Figure B.72: Scores for MFCC representations of /\textit{v}/ in the BDL database
Mean f0 readings for examples of /zz/ in the BDL database.

- f0
- std
- mean

0 20 40 60 80 100 120 140 160 180
no. of signals

110 115 120 125 130 135 140 145
f0

std
mean

B.2. BDL DATABASE

Figure B.73: Mean f0 measurements of segments of /z/ in the BDL database.
Figure B.74: Scores for MFCC representations of /Z/ in the BDL database

Scores for voiced fricative /ZZ/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Score vs. window length (ms)
Mean f0 readings for examples of /zh/ in the BDL database

Figure B.75: Mean f0 measurements of segments of /zh/ in the BDL database
Figure B.76: Scores for MFCC representations of /ZH/ in the BDL database.
B.2.3 Group 3
Figure B.77: Scores for MFCC representations of /θ/ in the BDL database

Scores for voiceless fricative /θ/ representations

Triangular filters
- α = 10°
- α = 20°
- α = 30°
- α = 40°
- α = 50°
- α = 60°
- α = 70°
- α = 80°
- α = 90°

Rectangular filters

Gaussian filters

Individual phone data
Figure B.78: Scores for voiceless fricative /F/ representations in the BDL database.
Figure B.79: Scores for voiceless fricative /s/ representations in the BDL database.
Figure B.80: Scores for voiceless fricative /SH/ representations in the BDL database.
B.3 SLT database

B.3.1 Group 1
Mean f0 readings for examples of /iy/
in the SLT database

Figure B.81: Mean f0 measurements of segments of /iy/ in the SLT database
Figure B.82: Scores for MFCC representations of /i/ in the SLT database.

Triangular filters

Scores for vowel /I/ representations

Rectangular filters

Gaussian filters
Mean f0 readings for examples of /ih/ in the SLT database
Figure B.84: Scores for MFCC representations of /i/ in the SLT database.

Scores for vowel /IH/ representations

Triangular filters

Rectangular filters

Gaussian filters
Mean f0 readings for examples of /eh/ in the SLT database

Figure B.85: Mean f0 measurements of segments of /e/ in the SLT database
Figure B.86: Scores for MFCC representations of /e/ in the SLT database

Triangular filters

Scores for vowel /EH/ representations

Rectangular filters

Gaussian filters

Individual phoneme data
Mean f0 readings for examples of /ey/ in the SLT database

Figure B.87: Mean f0 measurements of segments of /e/ in the SLT database
Figure B.88: Scores for MFCC representations of /e/ in the SLT database.

Scores for vowel /EY/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

window length (ms)
Mean f0 readings for examples of /æː/ in the SLT database

Figure B.89: Mean f0 measurements of segments of /æː/ in the SLT database
Figure B.90: Scores for MFCC representations of /æ/ in the SLT database.
Figure B.91: Mean f0 measurements of /ao/ in the SLT database.
Figure B.92: Scores for MFCC representations of /a/ in the SLT database

Triangular filters

Scores for vowel /AO/ representations

Rectangular filters

Gaussian filters

Scores for vowel /AO/ representations

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

B. INDIVIDUAL PHONEME DATA
Figure B.93: Mean f0 readings for examples of /aa/ in the SLT database
Figure B.94: Scores for MFCC representations of /A/ in the SLT database
Figure B.95: Mean /o/ measurements of segments of /o/ in the SLT database.
Figure B.96: Scores for MFCC representations of /o/ in the SLT database.

Triangular filters

Scores for vowel /OW/ representations

Rectangular filters

Gaussian filters
Mean f0 readings for examples of /ah/ in the SLT database
Figure B.98: Scores for MFCC representations of /A/ in the SLT database.

Scores for vowel /AH/ representations

Triangular filters

Rectangular filters

Gaussian filters

window length (ms)
Mean f0 readings for examples of /uh/ in the SLT database
Figure B.100: Scores for MFCC representations of /UH/ in the SLT database.
Mean $f_0$ readings for examples of /uw/ in the SLT database.

![Graph showing mean $f_0$ measurements for /uw/ in the SLT database.](image)
Figure B.102: Scores for MFCC representations of /u/ in the SLT database.
Figure B.103: Mean f0 measurements of /ay/ in the SLT database.
Figure B.1.04: Scores for MFCC representations of \(/\text{ai}/\) in the SLT database

Scores for diphthong \(/\text{AI}/\) representations

Triangular filters

Rectangular filters

Gaussian filters
Mean f0 readings for examples of /aw/ in the SLT database
Figure B.106: Scores for MFCC representations of /au/ in the SLT database.
Mean f0 readings for examples of /oy/ in the SLT database

Figure B.107: Mean f0 measurements of segments of /oy/ in the SLT database
Figure B.108: Scores for MFCC representations of /i/ in the SLT database.
B.3.2 Group 2
Figure B.109: Mean f0 measurements of /dh/ in the SLT database
Figure B.10: Scores for MFCC representations of /3/ in the SLT database.

- Triangular filters
- Rectangular filters
- Gaussian filters

Scores for voiced fricative /DH/ representations

- Window length (ms)
Figure B.111: Mean $f_0$ measurements of segments of /v/ in the SLT database
Figure B.112: Scores for MFCC representations of /v/ in the SLT database.
Figure B.113: Mean f0 measurements of segments of /zz/ in the SLT database

Mean f0 readings for examples of /zz/ in the SLT database

- f0
- std
- mean

no. of signals

0 50 100 150 200 250 300 350 400

150 160 170 180 190 200 210

Individual phoneme data
Scores for voiced fricative /ZZ/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

Figure B.114: Scores for MFCC representations of /z/ in the SLT database

B.3. SLT DATABASE
Mean f0 readings for examples of /zh/
in the SLT database

Figure B.115: Mean f0 measurements of segments of /zh/ in the SLT database
Figure B.116: Scores for MFCC representations of /ʃ/ in the SLT database

Scores for voiced fricative /ZH/ representations

- Triangular filters
- Rectangular filters
- Gaussian filters

\[ \alpha = 10^\circ \]
\[ \alpha = 20^\circ \]
\[ \alpha = 30^\circ \]
\[ \alpha = 40^\circ \]
\[ \alpha = 50^\circ \]
\[ \alpha = 60^\circ \]
\[ \alpha = 70^\circ \]
\[ \alpha = 80^\circ \]
\[ \alpha = 90^\circ \]
B.3.3 Group 3
Figure B.117: Scores for voiceless fricative /θ/ in the SLT database.

Scores for voiceless fricative /θ/ representations

- Triangular filters
- Rectangular filters
- Gaussian filters

*α* = 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°, 90°
Figure B.118: Scores for voiceless fricative /ff/ representations.

Scores for voiceless fricative /ff/ representations:
- Triangular filters
- Rectangular filters
- Gaussian filters

Different values of alpha (α) are shown for each type of filter:
- α = 10°
- α = 20°
- α = 30°
- α = 40°
- α = 50°
- α = 60°
- α = 70°
- α = 80°
- α = 90°

The x-axis represents the window length (ms), and the y-axis represents the score.
Figure B.119: Scores for voiceless fricative /s/ in the SLT database

Scores for voiceless fricative /s/ representations

Triangular filters

Rectangular filters

Gaussian filters

- $\alpha = 10^\circ$
- $\alpha = 20^\circ$
- $\alpha = 30^\circ$
- $\alpha = 40^\circ$
- $\alpha = 50^\circ$
- $\alpha = 60^\circ$
- $\alpha = 70^\circ$
- $\alpha = 80^\circ$
- $\alpha = 90^\circ$

score

window length (ms)
Figure B.120: Scores for voiceless fricative /SH/ representations

Scores for MFCC representations of /J/ in the SLT database

Triangular filters

Rectangular filters

Gaussian filters
C.1 Baseline cost function

"You have associated with some of those men."

<table>
<thead>
<tr>
<th></th>
<th>Population</th>
<th>Coef 1</th>
<th>Coef 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>y-uw</td>
<td>arctic_b0408.coef</td>
<td>0.116</td>
<td>0.277</td>
</tr>
<tr>
<td>uw-hh</td>
<td>arctic_b0408.coef</td>
<td>0.277</td>
<td>0.342</td>
</tr>
<tr>
<td>hh-ae</td>
<td>arctic_b0408.coef</td>
<td>0.342</td>
<td>0.422</td>
</tr>
<tr>
<td>ae-v</td>
<td>arctic_b0408.coef</td>
<td>0.422</td>
<td>0.537</td>
</tr>
<tr>
<td>v-ax</td>
<td>arctic_b0439.coef</td>
<td>3.572</td>
<td>3.632</td>
</tr>
<tr>
<td>ax-s</td>
<td>arctic_b0193.coef</td>
<td>2.107</td>
<td>2.207</td>
</tr>
<tr>
<td>s-ow</td>
<td>arctic_b0459.coef</td>
<td>2.237</td>
<td>2.332</td>
</tr>
<tr>
<td>ow-s</td>
<td>arctic_b0459.coef</td>
<td>2.332</td>
<td>2.432</td>
</tr>
<tr>
<td>s-iy</td>
<td>arctic_b0459.coef</td>
<td>2.432</td>
<td>2.542</td>
</tr>
<tr>
<td>iy-ey</td>
<td>arctic_b0450.coef</td>
<td>2.407</td>
<td>2.477</td>
</tr>
<tr>
<td>ey-t</td>
<td>arctic_b0450.coef</td>
<td>2.477</td>
<td>2.537</td>
</tr>
<tr>
<td>t-ax</td>
<td>arctic_a0347.coef</td>
<td>1.447</td>
<td>1.522</td>
</tr>
<tr>
<td>ax-d</td>
<td>arctic_a0347.coef</td>
<td>1.522</td>
<td>1.647</td>
</tr>
</tbody>
</table>
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>d-w</th>
<th>arctic_a0347.coef</th>
<th>1.647 1.807</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-ih</td>
<td>arctic_b0310.coef</td>
<td>2.022 2.107</td>
</tr>
<tr>
<td>ih-dh</td>
<td>arctic_b0310.coef</td>
<td>2.107 2.147</td>
</tr>
<tr>
<td>dh-s</td>
<td>arctic_b0310.coef</td>
<td>2.147 2.237</td>
</tr>
<tr>
<td>s-ah</td>
<td>arctic_b0310.coef</td>
<td>2.237 2.327</td>
</tr>
<tr>
<td>ah-m</td>
<td>arctic_a0542.coef</td>
<td>1.292 1.362</td>
</tr>
<tr>
<td>m-ah</td>
<td>arctic_a0542.coef</td>
<td>1.362 1.432</td>
</tr>
<tr>
<td>ah-v</td>
<td>arctic_b0476.coef</td>
<td>1.582 1.697</td>
</tr>
<tr>
<td>v-dh</td>
<td>arctic_b0476.coef</td>
<td>1.697 1.772</td>
</tr>
<tr>
<td>dh-iy</td>
<td>arctic_b0476.coef</td>
<td>1.772 1.842</td>
</tr>
<tr>
<td>iy-z</td>
<td>arctic_b0476.coef</td>
<td>1.842 1.932</td>
</tr>
<tr>
<td>z-m</td>
<td>arctic_b0131.coef</td>
<td>0.912 1.007</td>
</tr>
<tr>
<td>m-eh</td>
<td>arctic_b0131.coef</td>
<td>1.007 1.097</td>
</tr>
<tr>
<td>eh-n</td>
<td>arctic_b0131.coef</td>
<td>1.097 1.247</td>
</tr>
</tbody>
</table>

"You are positively soulless, he said savagely."

<table>
<thead>
<tr>
<th>y-uw</th>
<th>arctic_b0105.coef</th>
<th>0.922 1.042</th>
</tr>
</thead>
<tbody>
<tr>
<td>uw-aa</td>
<td>arctic_b0105.coef</td>
<td>1.042 1.127</td>
</tr>
<tr>
<td>aa-r</td>
<td>arctic_b0105.coef</td>
<td>1.127 1.197</td>
</tr>
<tr>
<td>r-p</td>
<td>arctic_b0074.coef</td>
<td>2.992 3.097</td>
</tr>
<tr>
<td>p-aa</td>
<td>arctic_b0180.coef</td>
<td>1.117 1.227</td>
</tr>
<tr>
<td>aa-z</td>
<td>arctic_a0296.coef</td>
<td>0.902 1.007</td>
</tr>
<tr>
<td>z-ax</td>
<td>arctic_a0296.coef</td>
<td>1.007 1.122</td>
</tr>
<tr>
<td>ax-t</td>
<td>arctic_a0587.coef</td>
<td>1.067 1.132</td>
</tr>
<tr>
<td>t-ih</td>
<td>arctic_a0587.coef</td>
<td>1.132 1.197</td>
</tr>
<tr>
<td>ih-v</td>
<td>arctic_a0587.coef</td>
<td>1.197 1.267</td>
</tr>
<tr>
<td>v-l</td>
<td>arctic_a0587.coef</td>
<td>1.267 1.352</td>
</tr>
<tr>
<td>l-iy</td>
<td>arctic_a0587.coef</td>
<td>1.352 1.412</td>
</tr>
<tr>
<td>iy-s</td>
<td>arctic_a0331.coef</td>
<td>2.337 2.437</td>
</tr>
</tbody>
</table>
**C.1. Baseline Cost Function**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>s-ow</td>
<td>arctic_a0244.coef</td>
<td>0.812</td>
<td>1.042</td>
<td></td>
</tr>
<tr>
<td>ow-l</td>
<td>arctic_a0244.coef</td>
<td>1.042</td>
<td>1.232</td>
<td></td>
</tr>
<tr>
<td>l-l</td>
<td>arctic_a0539.coef</td>
<td>2.772</td>
<td>2.862</td>
<td></td>
</tr>
<tr>
<td>l-ax</td>
<td>arctic_b0532.coef</td>
<td>3.517</td>
<td>3.547</td>
<td></td>
</tr>
<tr>
<td>ax-s</td>
<td>arctic_b0532.coef</td>
<td>3.547</td>
<td>3.607</td>
<td></td>
</tr>
<tr>
<td>s-hh</td>
<td>arctic_a0531.coef</td>
<td>2.142</td>
<td>2.257</td>
<td></td>
</tr>
<tr>
<td>hh-iy</td>
<td>arctic_a0288.coef</td>
<td>1.602</td>
<td>1.717</td>
<td></td>
</tr>
<tr>
<td>iy-s</td>
<td>arctic_a0288.coef</td>
<td>1.717</td>
<td>1.842</td>
<td></td>
</tr>
<tr>
<td>s-eh</td>
<td>arctic_a0288.coef</td>
<td>1.842</td>
<td>1.987</td>
<td></td>
</tr>
<tr>
<td>eh-d</td>
<td>arctic_a0288.coef</td>
<td>1.987</td>
<td>2.102</td>
<td></td>
</tr>
<tr>
<td>d-s</td>
<td>arctic_b0063.coef</td>
<td>2.217</td>
<td>2.302</td>
<td></td>
</tr>
<tr>
<td>s-ae</td>
<td>arctic_b0063.coef</td>
<td>2.302</td>
<td>2.417</td>
<td></td>
</tr>
<tr>
<td>ae-v</td>
<td>arctic_a0234.coef</td>
<td>0.922</td>
<td>1.037</td>
<td></td>
</tr>
<tr>
<td>v-ih</td>
<td>arctic_a0252.coef</td>
<td>1.757</td>
<td>1.822</td>
<td></td>
</tr>
<tr>
<td>ih-jh</td>
<td>arctic_a0497.coef</td>
<td>1.602</td>
<td>1.722</td>
<td></td>
</tr>
<tr>
<td>jh-l</td>
<td>arctic_a0209.coef</td>
<td>1.097</td>
<td>1.212</td>
<td></td>
</tr>
<tr>
<td>l-iy</td>
<td>arctic_b0299.coef</td>
<td>2.397</td>
<td>2.482</td>
<td></td>
</tr>
</tbody>
</table>

"At the best, they are necessary accessories."

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ae-t</td>
<td>arctic_a0036.coef</td>
<td>1.002</td>
<td>1.087</td>
<td></td>
</tr>
<tr>
<td>t-dh</td>
<td>arctic_a0036.coef</td>
<td>1.087</td>
<td>1.137</td>
<td></td>
</tr>
<tr>
<td>dh-ax</td>
<td>arctic_a0036.coef</td>
<td>1.137</td>
<td>1.172</td>
<td></td>
</tr>
<tr>
<td>ax-b</td>
<td>arctic_a0180.coef</td>
<td>2.877</td>
<td>2.962</td>
<td></td>
</tr>
<tr>
<td>b-eh</td>
<td>arctic_a0180.coef</td>
<td>2.962</td>
<td>3.107</td>
<td></td>
</tr>
<tr>
<td>eh-s</td>
<td>arctic_b0034.coef</td>
<td>0.587</td>
<td>0.707</td>
<td></td>
</tr>
<tr>
<td>s-t</td>
<td>arctic_b0034.coef</td>
<td>0.707</td>
<td>0.822</td>
<td></td>
</tr>
<tr>
<td>t-dh</td>
<td>arctic_b0034.coef</td>
<td>0.822</td>
<td>0.947</td>
<td></td>
</tr>
<tr>
<td>dh-ey</td>
<td>arctic_a0293.coef</td>
<td>0.131</td>
<td>0.302</td>
<td></td>
</tr>
<tr>
<td>ey-w</td>
<td>arctic_a0293.coef</td>
<td>0.302</td>
<td>0.402</td>
<td></td>
</tr>
</tbody>
</table>
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>Pair</th>
<th>File</th>
<th>Coeffs</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-er</td>
<td>arctic_a0293.coef</td>
<td>0.402 0.497</td>
</tr>
<tr>
<td>er-n</td>
<td>arctic_b0276.coef</td>
<td>0.407 0.487</td>
</tr>
<tr>
<td>n-eh</td>
<td>arctic_a0465.coef</td>
<td>0.367 0.442</td>
</tr>
<tr>
<td>eh-s</td>
<td>arctic_a0429.coef</td>
<td>4.942 5.052</td>
</tr>
<tr>
<td>s-ax</td>
<td>arctic_a0429.coef</td>
<td>5.052 5.137</td>
</tr>
<tr>
<td>ax-s</td>
<td>arctic_a0332.coef</td>
<td>0.672 0.762</td>
</tr>
<tr>
<td>s-eh</td>
<td>arctic_b0243.coef</td>
<td>2.717 2.832</td>
</tr>
<tr>
<td>eh-r</td>
<td>arctic_b0253.coef</td>
<td>0.397 0.472</td>
</tr>
<tr>
<td>r-iy</td>
<td>arctic_b0253.coef</td>
<td>0.472 0.562</td>
</tr>
<tr>
<td>iy-ae</td>
<td>arctic_a0497.coef</td>
<td>1.057 1.142</td>
</tr>
<tr>
<td>ae-k</td>
<td>arctic_b0463.coef</td>
<td>1.637 1.727</td>
</tr>
<tr>
<td>k-s</td>
<td>arctic_b0463.coef</td>
<td>1.727 1.832</td>
</tr>
<tr>
<td>s-eh</td>
<td>arctic_b0463.coef</td>
<td>1.832 1.922</td>
</tr>
<tr>
<td>eh-s</td>
<td>arctic_a0540.coef</td>
<td>2.022 2.107</td>
</tr>
<tr>
<td>s-er</td>
<td>arctic_a0540.coef</td>
<td>2.107 2.232</td>
</tr>
<tr>
<td>er-iy</td>
<td>arctic_a0308.coef</td>
<td>1.982 2.087</td>
</tr>
<tr>
<td>iy-z</td>
<td>arctic_a0308.coef</td>
<td>2.087 2.202</td>
</tr>
<tr>
<td>dh-ax</td>
<td>arctic_a0218.coef</td>
<td>0.136 0.302</td>
</tr>
<tr>
<td>ax-r</td>
<td>arctic_a0218.coef</td>
<td>0.302 0.412</td>
</tr>
<tr>
<td>r-iy</td>
<td>arctic_a0522.coef</td>
<td>2.577 2.667</td>
</tr>
<tr>
<td>iy-ao</td>
<td>arctic_a0522.coef</td>
<td>2.667 2.732</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_a0522.coef</td>
<td>2.732 2.812</td>
</tr>
<tr>
<td>r-g</td>
<td>arctic_a0522.coef</td>
<td>2.812 2.882</td>
</tr>
<tr>
<td>g-ax</td>
<td>arctic_a0522.coef</td>
<td>2.882 2.927</td>
</tr>
<tr>
<td>ax-n</td>
<td>arctic_a0522.coef</td>
<td>2.927 2.972</td>
</tr>
<tr>
<td>n-ax</td>
<td>arctic_a0522.coef</td>
<td>2.972 3.027</td>
</tr>
<tr>
<td>ax-z</td>
<td>arctic_a0522.coef</td>
<td>3.027 3.102</td>
</tr>
</tbody>
</table>

"The reorganization of these countries took the form of revolution. "

<table>
<thead>
<tr>
<th>Pair</th>
<th>File</th>
<th>Coeffs</th>
</tr>
</thead>
<tbody>
<tr>
<td>dh-ax</td>
<td>arctic_a0218.coef</td>
<td>0.136 0.302</td>
</tr>
<tr>
<td>ax-r</td>
<td>arctic_a0218.coef</td>
<td>0.302 0.412</td>
</tr>
<tr>
<td>r-iy</td>
<td>arctic_a0522.coef</td>
<td>2.577 2.667</td>
</tr>
<tr>
<td>iy-ao</td>
<td>arctic_a0522.coef</td>
<td>2.667 2.732</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_a0522.coef</td>
<td>2.732 2.812</td>
</tr>
<tr>
<td>r-g</td>
<td>arctic_a0522.coef</td>
<td>2.812 2.882</td>
</tr>
<tr>
<td>g-ax</td>
<td>arctic_a0522.coef</td>
<td>2.882 2.927</td>
</tr>
<tr>
<td>ax-n</td>
<td>arctic_a0522.coef</td>
<td>2.927 2.972</td>
</tr>
<tr>
<td>n-ax</td>
<td>arctic_a0522.coef</td>
<td>2.972 3.027</td>
</tr>
<tr>
<td>ax-z</td>
<td>arctic_a0522.coef</td>
<td>3.027 3.102</td>
</tr>
<tr>
<td>Transition</td>
<td>Source Code</td>
<td>z-ey</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.102 3.207</td>
</tr>
</tbody>
</table>
C. Synthesis of Test Utterances

v-ax arctic_b0224.coef 1.667 1.717
ax-l arctic_b0224.coef 1.717 1.797
l-uw arctic_b0224.coef 1.797 1.882
uw-sh arctic_b0224.coef 1.882 1.982
sh-ax arctic_b0224.coef 1.982 2.077
ax-n arctic_b0224.coef 2.077 2.137

“For the rest, he was a mere automaton.”

f-ao arctic_a0176.coef 0.121 0.267
ao-r arctic_a0176.coef 0.267 0.322
r-dh arctic_a0176.coef 0.322 0.372
dh-ax arctic_a0176.coef 0.372 0.412
ax-r arctic_b0350.coef 2.862 2.957
r-eh arctic_b0350.coef 2.957 3.092
eh-s arctic_b0332.coef 1.257 1.387
s-t arctic_b0332.coef 1.387 1.527
t-hh arctic_b0332.coef 1.527 1.642
hh-iy arctic_b0332.coef 1.642 1.737
iy-w arctic_a0317.coef 0.297 0.427
w-aa arctic_a0317.coef 0.427 0.512
aa-z arctic_b0108.coef 0.432 0.497
z-ax arctic_b0108.coef 0.497 0.557
ax-m arctic_b0108.coef 0.557 0.632
m-ih arctic_b0108.coef 0.632 0.717
ih-r arctic_b0108.coef 0.717 0.802
r-ao arctic_a0404.coef 0.292 0.417
ao-t arctic_b0537.coef 1.687 1.832
t-aa arctic_b0537.coef 1.832 1.952
aa-m arctic_b0537.coef 1.952 2.047
C.1. BASELINE COST FUNCTION

m-ax  arctic_b0537.coef  2.047  2.122
ax-t  arctic_b0537.coef  2.122  2.212
t-aa  arctic_b0537.coef  2.212  2.377
aa-n  arctic_b0537.coef  2.377  2.552

"Not at this particular case, Tom, apologised Whittemore."

n-aa  arctic_a0457.coef  1.447  1.577
aa-t  arctic_a0457.coef  1.577  1.692
t-ae  arctic_a0457.coef  1.692  1.802
ae-t  arctic_b0082.coef  2.452  2.597
t-dh  arctic_b0082.coef  2.597  2.682
dh-ih  arctic_b0082.coef  2.682  2.732
ih-s  arctic_b0082.coef  2.732  2.817
s-p  arctic_a0543.coef  3.297  3.377
p-er  arctic_b0246.coef  1.807  1.947
er-t  arctic_b0030.coef  3.287  3.372
t-ih  arctic_b0030.coef  3.372  3.467
ih-k  arctic_a0463.coef  3.397  3.457
k-y  arctic_a0463.coef  3.457  3.522
y-ax  arctic_a0463.coef  3.522  3.557
ax-l  arctic_a0463.coef  3.557  3.637
l-er  arctic_b0482.coef  1.147  1.252
er-k  arctic_a0008.coef  0.862  0.957
k-ey  arctic_a0008.coef  0.957  1.072
ey-s  arctic_a0214.coef  1.797  1.902
s-t  arctic_b0264.coef  0.612  0.712
t-aa  arctic_a0496.coef  0.547  0.667
aa-m  arctic_a0496.coef  0.667  0.772
m-ax  arctic_a0507.coef  2.952  3.017
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>Pair</th>
<th>File</th>
<th>Coef 1</th>
<th>Coef 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ax-p</td>
<td>arctic_b0066.coef</td>
<td>2.037</td>
<td>2.137</td>
</tr>
<tr>
<td>p-aa</td>
<td>arctic_b0066.coef</td>
<td>2.137</td>
<td>2.252</td>
</tr>
<tr>
<td>aa-l</td>
<td>arctic_b0066.coef</td>
<td>2.252</td>
<td>2.337</td>
</tr>
<tr>
<td>l-ax</td>
<td>arctic_b0066.coef</td>
<td>2.337</td>
<td>2.407</td>
</tr>
<tr>
<td>ax-jh</td>
<td>arctic_b0066.coef</td>
<td>2.407</td>
<td>2.487</td>
</tr>
<tr>
<td>jh-ay</td>
<td>arctic_b0066.coef</td>
<td>2.487</td>
<td>2.662</td>
</tr>
<tr>
<td>ay-z</td>
<td>arctic_b0287.coef</td>
<td>1.822</td>
<td>2.007</td>
</tr>
<tr>
<td>z-d</td>
<td>arctic_b0287.coef</td>
<td>2.007</td>
<td>2.082</td>
</tr>
<tr>
<td>d-w</td>
<td>arctic_b0287.coef</td>
<td>2.082</td>
<td>2.157</td>
</tr>
<tr>
<td>w-ih</td>
<td>arctic_b0020.coef</td>
<td>2.597</td>
<td>2.667</td>
</tr>
<tr>
<td>ih-t</td>
<td>arctic_b0020.coef</td>
<td>2.667</td>
<td>2.722</td>
</tr>
<tr>
<td>t-m</td>
<td>arctic_b0020.coef</td>
<td>2.722</td>
<td>2.807</td>
</tr>
<tr>
<td>m-ao</td>
<td>arctic_b0020.coef</td>
<td>2.807</td>
<td>2.927</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_b0020.coef</td>
<td>2.927</td>
<td>3.032</td>
</tr>
</tbody>
</table>

"Will we ever forget it"

<table>
<thead>
<tr>
<th>Pair</th>
<th>File</th>
<th>Coef 1</th>
<th>Coef 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-ih</td>
<td>arctic_a0411.coef</td>
<td>0.317</td>
<td>0.382</td>
</tr>
<tr>
<td>ih-l</td>
<td>arctic_a0411.coef</td>
<td>0.382</td>
<td>0.417</td>
</tr>
<tr>
<td>l-w</td>
<td>arctic_b0393.coef</td>
<td>0.792</td>
<td>0.922</td>
</tr>
<tr>
<td>w-iy</td>
<td>arctic_b0381.coef</td>
<td>1.132</td>
<td>1.247</td>
</tr>
<tr>
<td>iy-eh</td>
<td>arctic_b0381.coef</td>
<td>1.247</td>
<td>1.352</td>
</tr>
<tr>
<td>eh-v</td>
<td>arctic_b0381.coef</td>
<td>1.352</td>
<td>1.427</td>
</tr>
<tr>
<td>v-er</td>
<td>arctic_b0381.coef</td>
<td>1.427</td>
<td>1.507</td>
</tr>
<tr>
<td>er-f</td>
<td>arctic_b0381.coef</td>
<td>1.507</td>
<td>1.617</td>
</tr>
<tr>
<td>f-er</td>
<td>arctic_a0460.coef</td>
<td>0.627</td>
<td>0.717</td>
</tr>
<tr>
<td>er-g</td>
<td>arctic_a0460.coef</td>
<td>0.717</td>
<td>0.792</td>
</tr>
<tr>
<td>g-eh</td>
<td>arctic_a0591.coef</td>
<td>1.427</td>
<td>1.542</td>
</tr>
<tr>
<td>eh-t</td>
<td>arctic_b0338.coef</td>
<td>1.977</td>
<td>2.092</td>
</tr>
<tr>
<td>t-ih</td>
<td>arctic_b0338.coef</td>
<td>2.092</td>
<td>2.207</td>
</tr>
<tr>
<td>ih-t</td>
<td>arctic_b0338.coef</td>
<td>2.207</td>
<td>2.277</td>
</tr>
</tbody>
</table>
And you always want to see it in the superlative degree.

ae-n    arctic_b0181.coef  0.156  0.342
n-d     arctic_b0181.coef  0.342  0.397
d-y     arctic_b0181.coef  0.397  0.467
y-uw    arctic_b0181.coef  0.467  0.527
uw-ao   arctic_a0334.coef  0.762  0.902
ao-l    arctic_a0246.coef  0.787  0.902
l-w     arctic_a0246.coef  0.902  0.997
w-ey    arctic_a0246.coef  0.997  1.087
ey-z    arctic_a0246.coef  1.087  1.222
z-w     arctic_b0306.coef  1.912  2.052
w-aa    arctic_b0353.coef  0.377  0.492
aa-n    arctic_b0353.coef  0.492  0.567
n-t     arctic_b0353.coef  0.567  0.632
t-t     arctic_b0353.coef  0.632  0.717
t-ax    arctic_a0408.coef  3.692  3.772
ax-s    arctic_a0408.coef  3.772  3.867
s-iy    arctic_a0408.coef  3.867  3.982
iy-ih   arctic_a0253.coef  0.322  0.487
ih-t    arctic_a0253.coef  0.487  0.557
t-ih    arctic_a0380.coef  0.257  0.347
ih-n    arctic_a0380.coef  0.347  0.447
n-dh    arctic_a0380.coef  0.447  0.497
dh-ax   arctic_a0427.coef  0.877  0.937
ax-s    arctic_a0427.coef  0.937  1.022
s-uh    arctic_b0387.coef  1.632  1.717
uh-p    arctic_b0387.coef  1.717  1.797
p-er    arctic_b0387.coef  1.797  1.912
er-l    arctic_b0387.coef  1.912  2.012
l-ax    arctic_b0387.coef  2.012  2.077
C. SYNTHESIS OF TEST UTTERANCES

<table>
<thead>
<tr>
<th>Segment</th>
<th>File</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>ax-t</td>
<td>arctic_b0387.coef</td>
<td>2.077 2.152</td>
</tr>
<tr>
<td>t-ih</td>
<td>arctic_b0468.coef</td>
<td>2.182 2.257</td>
</tr>
<tr>
<td>ih-v</td>
<td>arctic_b0468.coef</td>
<td>2.257 2.322</td>
</tr>
<tr>
<td>v-d</td>
<td>arctic_b0468.coef</td>
<td>2.322 2.427</td>
</tr>
<tr>
<td>d-ih</td>
<td>arctic_b0468.coef</td>
<td>2.427 2.537</td>
</tr>
<tr>
<td>ih-g</td>
<td>arctic_b0157.coef</td>
<td>2.687 2.767</td>
</tr>
<tr>
<td>g-r</td>
<td>arctic_b0464.coef</td>
<td>1.067 1.182</td>
</tr>
<tr>
<td>r-iy</td>
<td>arctic_b0464.coef</td>
<td>1.182 1.322</td>
</tr>
</tbody>
</table>

"He turned sharply and faced Gregson across the table."

<table>
<thead>
<tr>
<th>Segment</th>
<th>File</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>hh-iy</td>
<td>arctic_b0010.coef</td>
<td>0.076 0.192</td>
</tr>
<tr>
<td>iy-t</td>
<td>arctic_b0010.coef</td>
<td>0.192 0.292</td>
</tr>
<tr>
<td>t-er</td>
<td>arctic_b0010.coef</td>
<td>0.292 0.432</td>
</tr>
<tr>
<td>er-n</td>
<td>arctic_b0010.coef</td>
<td>0.432 0.532</td>
</tr>
<tr>
<td>n-d</td>
<td>arctic_b0010.coef</td>
<td>0.532 0.582</td>
</tr>
<tr>
<td>d-sh</td>
<td>arctic_b0498.coef</td>
<td>3.677 3.822</td>
</tr>
<tr>
<td>sh-aa</td>
<td>arctic_a0472.coef</td>
<td>2.352 2.467</td>
</tr>
<tr>
<td>aa-r</td>
<td>arctic_a0472.coef</td>
<td>2.467 2.527</td>
</tr>
<tr>
<td>r-p</td>
<td>arctic_a0472.coef</td>
<td>2.527 2.592</td>
</tr>
<tr>
<td>p-l</td>
<td>arctic_a0472.coef</td>
<td>2.592 2.667</td>
</tr>
<tr>
<td>l-iy</td>
<td>arctic_a0472.coef</td>
<td>2.667 2.722</td>
</tr>
<tr>
<td>iy-ae</td>
<td>arctic_a0136.coef</td>
<td>3.352 3.487</td>
</tr>
<tr>
<td>ae-n</td>
<td>arctic_a0136.coef</td>
<td>3.487 3.617</td>
</tr>
<tr>
<td>n-d</td>
<td>arctic_a0136.coef</td>
<td>3.617 3.692</td>
</tr>
<tr>
<td>d-f</td>
<td>arctic_a0375.coef</td>
<td>4.042 4.117</td>
</tr>
<tr>
<td>f-ey</td>
<td>arctic_a0046.coef</td>
<td>0.582 0.697</td>
</tr>
<tr>
<td>ey-s</td>
<td>arctic_a0046.coef</td>
<td>0.697 0.802</td>
</tr>
<tr>
<td>s-t</td>
<td>arctic_a0046.coef</td>
<td>0.802 0.887</td>
</tr>
<tr>
<td>t-g</td>
<td>arctic_b0517.coef</td>
<td>1.272 1.382</td>
</tr>
<tr>
<td>g-r</td>
<td>arctic_b0031.coef</td>
<td>0.106 0.252</td>
</tr>
</tbody>
</table>
r-eh  arctic_b0031.coef  0.252 0.312  
gh-g  arctic_b0031.coef  0.312 0.372  
g-s  arctic_b0031.coef  0.372 0.477  
s-ax  arctic_b0031.coef  0.477 0.562  
ax-n  arctic_b0031.coef  0.562 0.642  
n-ax  arctic_b0389.coef  3.042 3.227  
ax-k  arctic_b0292.coef  4.137 4.257  
k-r  arctic_b0292.coef  4.257 4.387  
r-ao  arctic_b0292.coef  4.387 4.472  
ao-s  arctic_b0292.coef  4.472 4.582  
s-dh  arctic_b0420.coef  2.282 2.452  
dh-ax  arctic_b0420.coef  2.452 2.527  
ax-t  arctic_b0019.coef  2.607 2.697  
t-ey  arctic_b0019.coef  2.697 2.832  
ey-b  arctic_b0019.coef  2.832 2.947  
b-ax  arctic_b0019.coef  2.947 3.017  
ax-1  arctic_b0019.coef  3.017 3.092  

“I followed the line of the proposed railroad, looking for chances.”

ay-f  arctic_a0476.coef  1.092 1.227  
f-aa  arctic_b0133.coef  0.407 0.517  
aa-l  arctic_b0133.coef  0.517 0.612  
l-ow  arctic_b0133.coef  0.612 0.707  
ow-d  arctic_b0133.coef  0.707 0.812  
d-dh  arctic_a0253.coef  2.172 2.247  
dh-ax  arctic_a0253.coef  2.247 2.297  
ax-1  arctic_a0498.coef  0.292 0.387  
l-ay  arctic_a0498.coef  0.387 0.562  
ay-n  arctic_b0498.coef  1.577 1.772  
<table>
<thead>
<tr>
<th>Word Pair</th>
<th>File/Type</th>
<th>Coefficient 1</th>
<th>Coefficient 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-ah</td>
<td>arctic_b0498.coef</td>
<td>1.772</td>
<td>1.882</td>
</tr>
<tr>
<td>ah-v</td>
<td>arctic_b0498.coef</td>
<td>1.882</td>
<td>1.957</td>
</tr>
<tr>
<td>v-dh</td>
<td>arctic_b0498.coef</td>
<td>1.957</td>
<td>2.027</td>
</tr>
<tr>
<td>dh-ax</td>
<td>arctic_b0498.coef</td>
<td>2.027</td>
<td>2.077</td>
</tr>
<tr>
<td>ax-p</td>
<td>arctic_a0293.coef</td>
<td>1.272</td>
<td>1.412</td>
</tr>
<tr>
<td>p-r</td>
<td>arctic_a0293.coef</td>
<td>1.412</td>
<td>1.532</td>
</tr>
<tr>
<td>r-ax</td>
<td>arctic_a0503.coef</td>
<td>3.857</td>
<td>3.907</td>
</tr>
<tr>
<td>ax-p</td>
<td>arctic_a0503.coef</td>
<td>3.907</td>
<td>3.982</td>
</tr>
<tr>
<td>p-ow</td>
<td>arctic_b0515.coef</td>
<td>1.887</td>
<td>2.017</td>
</tr>
<tr>
<td>ow-z</td>
<td>arctic_b0515.coef</td>
<td>2.017</td>
<td>2.142</td>
</tr>
<tr>
<td>z-d</td>
<td>arctic_b0515.coef</td>
<td>2.142</td>
<td>2.222</td>
</tr>
<tr>
<td>d-r</td>
<td>arctic_a0121.coef</td>
<td>1.832</td>
<td>1.917</td>
</tr>
<tr>
<td>r-ey</td>
<td>arctic_b0269.coef</td>
<td>1.292</td>
<td>1.402</td>
</tr>
<tr>
<td>ey-l</td>
<td>arctic_b0269.coef</td>
<td>1.402</td>
<td>1.497</td>
</tr>
<tr>
<td>l-r</td>
<td>arctic_b0269.coef</td>
<td>1.497</td>
<td>1.612</td>
</tr>
<tr>
<td>r-ow</td>
<td>arctic_b0269.coef</td>
<td>1.612</td>
<td>1.737</td>
</tr>
<tr>
<td>ow-d</td>
<td>arctic_b0269.coef</td>
<td>1.737</td>
<td>1.862</td>
</tr>
<tr>
<td>d-l</td>
<td>arctic_a0539.coef</td>
<td>0.512</td>
<td>0.622</td>
</tr>
<tr>
<td>l-uh</td>
<td>arctic_b0399.coef</td>
<td>0.452</td>
<td>0.522</td>
</tr>
<tr>
<td>uh-k</td>
<td>arctic_b0399.coef</td>
<td>0.522</td>
<td>0.587</td>
</tr>
<tr>
<td>k-ih</td>
<td>arctic_b0399.coef</td>
<td>0.587</td>
<td>0.647</td>
</tr>
<tr>
<td>ih-ng</td>
<td>arctic_b0399.coef</td>
<td>0.647</td>
<td>0.727</td>
</tr>
<tr>
<td>ng-f</td>
<td>arctic_a0176.coef</td>
<td>2.742</td>
<td>2.882</td>
</tr>
<tr>
<td>f-ao</td>
<td>arctic_a0176.coef</td>
<td>2.882</td>
<td>3.012</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_a0176.coef</td>
<td>3.012</td>
<td>3.132</td>
</tr>
<tr>
<td>r-ch</td>
<td>arctic_a0418.coef</td>
<td>0.502</td>
<td>0.592</td>
</tr>
<tr>
<td>ch-ae</td>
<td>arctic_a0038.coef</td>
<td>1.312</td>
<td>1.447</td>
</tr>
<tr>
<td>ae-n</td>
<td>arctic_a0038.coef</td>
<td>1.447</td>
<td>1.537</td>
</tr>
<tr>
<td>n-s</td>
<td>arctic_a0038.coef</td>
<td>1.537</td>
<td>1.617</td>
</tr>
<tr>
<td>s-ax</td>
<td>arctic_a0038.coef</td>
<td>1.617</td>
<td>1.707</td>
</tr>
</tbody>
</table>
### C.2 Modified cost function

"You have associated with some of those men."

<table>
<thead>
<tr>
<th>Phoneme Pairs</th>
<th>File/Type</th>
<th>Coefficient 1</th>
<th>Coefficient 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ax-z</td>
<td>arctic_a0038.coef</td>
<td>1.707</td>
<td>1.852</td>
</tr>
<tr>
<td>y-uw</td>
<td>vowels/arctic_b0408.coef</td>
<td>0.116</td>
<td>0.277</td>
</tr>
<tr>
<td>uw-hh</td>
<td>arctic_b0408.coef</td>
<td>0.277</td>
<td>0.342</td>
</tr>
<tr>
<td>hh-ae</td>
<td>vowels/arctic_b0408.coef</td>
<td>0.342</td>
<td>0.422</td>
</tr>
<tr>
<td>ae-v</td>
<td>voiced/arctic_b0408.coef</td>
<td>0.422</td>
<td>0.537</td>
</tr>
<tr>
<td>v-ax</td>
<td>vowels/arctic_a0278.coef</td>
<td>1.082</td>
<td>1.157</td>
</tr>
<tr>
<td>ax-s</td>
<td>voiceless/arctic_a0278.coef</td>
<td>1.157</td>
<td>1.247</td>
</tr>
<tr>
<td>s-ow</td>
<td>vowels/arctic_b0459.coef</td>
<td>2.237</td>
<td>2.332</td>
</tr>
<tr>
<td>ow-s</td>
<td>voiceless/arctic_b0459.coef</td>
<td>2.332</td>
<td>2.432</td>
</tr>
<tr>
<td>s-iy</td>
<td>vowels/arctic_b0459.coef</td>
<td>2.432</td>
<td>2.542</td>
</tr>
<tr>
<td>iy-ey</td>
<td>vowels/arctic_a0445.coef</td>
<td>0.602</td>
<td>0.702</td>
</tr>
<tr>
<td>ey-t</td>
<td>arctic_a0445.coef</td>
<td>0.702</td>
<td>0.772</td>
</tr>
<tr>
<td>t-ax</td>
<td>vowels/arctic_a0347.coef</td>
<td>1.447</td>
<td>1.522</td>
</tr>
<tr>
<td>ax-d</td>
<td>arctic_a0347.coef</td>
<td>1.522</td>
<td>1.647</td>
</tr>
<tr>
<td>d-w</td>
<td>arctic_a0347.coef</td>
<td>1.647</td>
<td>1.807</td>
</tr>
<tr>
<td>w-ih</td>
<td>vowels/arctic_b0310.coef</td>
<td>2.022</td>
<td>2.107</td>
</tr>
<tr>
<td>ih-dh</td>
<td>voiced/arctic_b0310.coef</td>
<td>2.107</td>
<td>2.147</td>
</tr>
<tr>
<td>dh-s</td>
<td>voiceless/arctic_b0310.coef</td>
<td>2.147</td>
<td>2.237</td>
</tr>
<tr>
<td>s-ah</td>
<td>vowels/arctic_a0542.coef</td>
<td>1.182</td>
<td>1.292</td>
</tr>
<tr>
<td>ah-m</td>
<td>arctic_a0542.coef</td>
<td>1.292</td>
<td>1.362</td>
</tr>
<tr>
<td>m-ah</td>
<td>vowels/arctic_a0542.coef</td>
<td>1.362</td>
<td>1.432</td>
</tr>
<tr>
<td>ah-v</td>
<td>voiced/arctic_b0476.coef</td>
<td>1.582</td>
<td>1.697</td>
</tr>
<tr>
<td>v-dh</td>
<td>voiced/arctic_b0476.coef</td>
<td>1.697</td>
<td>1.772</td>
</tr>
<tr>
<td>dh-iy</td>
<td>vowels/arctic_b0476.coef</td>
<td>1.772</td>
<td>1.842</td>
</tr>
<tr>
<td>iy-z</td>
<td>voiced/arctic_b0476.coef</td>
<td>1.842</td>
<td>1.932</td>
</tr>
</tbody>
</table>
z-m  arctic_b0131.coef  0.912 1.007
m-eh vowels/arctic_b0131.coef  1.007 1.097
eh-n arctic_b0131.coef  1.097 1.247

"You are positively soulless, he said savagely."

y-uw vowels/arctic_b0105.coef  0.922 1.042
uw-aa vowels/arctic_b0105.coef  1.042 1.127
aa-r arctic_b0105.coef  1.127 1.197
r-p arctic_b0074.coef  2.992 3.097
p-aa vowels/arctic_a0077.coef  0.992 1.082
aa-z voiced/arctic_a0147.coef  3.247 3.312
z-ax vowels/arctic_a0487.coef  1.342 1.422
ax-t arctic_a0587.coef  1.067 1.132
t-ih vowels/arctic_a0587.coef  1.132 1.197
ih-v voiced/arctic_a0587.coef  1.197 1.267
v-l arctic_a0587.coef  1.267 1.352
l-iy vowels/arctic_a0587.coef  1.352 1.412
iy-s voiceless/arctic_a0094.coef  2.137 2.247
s-ow vowels/arctic_b0453.coef  1.902 2.017
ow-l arctic_b0453.coef  2.017 2.102
l-1 arctic_a0539.coef  2.772 2.862
l-ax vowels/arctic_a0289.coef  3.192 3.257
ax-s voiceless/arctic_a0289.coef  3.257 3.317
s-hh arctic_b0492.coef  0.402 0.492
hh-iy vowels/arctic_a0288.coef  1.602 1.717
iy-s voiceless/arctic_a0288.coef  1.717 1.842
s-eh vowels/arctic_a0288.coef  1.842 1.987
eh-d arctic_a0288.coef  1.987 2.102
d-s voiceless/arctic_b0063.coef  2.217 2.302
### C.2. Modified Cost Function

<table>
<thead>
<tr>
<th>Transition</th>
<th>Context</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-ae</td>
<td>vowels/arctic_b0063.coef</td>
<td>2.302 2.417</td>
</tr>
<tr>
<td>ae-v</td>
<td>voiced/arctic_b0329.coef</td>
<td>1.027 1.117</td>
</tr>
<tr>
<td>v-ih</td>
<td>vowels/arctic_a0506.coef</td>
<td>1.472 1.532</td>
</tr>
<tr>
<td>ih-jh</td>
<td>arctic_a0497.coef</td>
<td>1.602 1.722</td>
</tr>
<tr>
<td>jh-l</td>
<td>arctic_a0209.coef</td>
<td>1.097 1.212</td>
</tr>
<tr>
<td>l-iy</td>
<td>vowels/arctic_b0299.coef</td>
<td>2.397 2.482</td>
</tr>
</tbody>
</table>

"At the best, they are necessary accessories."

<table>
<thead>
<tr>
<th>Transition</th>
<th>Context</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>ae-t</td>
<td>arctic_b0254.coef</td>
<td>0.116 0.292</td>
</tr>
<tr>
<td>t-dh</td>
<td>voiced/arctic_b0254.coef</td>
<td>0.292 0.372</td>
</tr>
<tr>
<td>dh-ax</td>
<td>vowels/arctic_b0254.coef</td>
<td>0.372 0.422</td>
</tr>
<tr>
<td>ax-b</td>
<td>arctic_a0410.coef</td>
<td>1.437 1.507</td>
</tr>
<tr>
<td>b-eh</td>
<td>vowels/arctic_b0180.coef</td>
<td>0.797 0.892</td>
</tr>
<tr>
<td>eh-s</td>
<td>voiceless/arctic_b0034.coef</td>
<td>0.587 0.707</td>
</tr>
<tr>
<td>s-t</td>
<td>arctic_b0034.coef</td>
<td>0.707 0.822</td>
</tr>
<tr>
<td>t-dh</td>
<td>voiced/arctic_b0034.coef</td>
<td>0.822 0.947</td>
</tr>
<tr>
<td>dh-ey</td>
<td>vowels/arctic_a0031.coef</td>
<td>0.111 0.257</td>
</tr>
<tr>
<td>ey-w</td>
<td>arctic_a0031.coef</td>
<td>0.257 0.337</td>
</tr>
<tr>
<td>w-er</td>
<td>arctic_a0031.coef</td>
<td>0.337 0.412</td>
</tr>
<tr>
<td>er-n</td>
<td>arctic_b0276.coef</td>
<td>0.407 0.487</td>
</tr>
<tr>
<td>n-eh</td>
<td>vowels/arctic_a0233.coef</td>
<td>0.307 0.392</td>
</tr>
<tr>
<td>eh-s</td>
<td>voiceless/arctic_a0121.coef</td>
<td>1.997 2.067</td>
</tr>
<tr>
<td>s-ax</td>
<td>vowels/arctic_a0153.coef</td>
<td>2.302 2.412</td>
</tr>
<tr>
<td>ax-s</td>
<td>voiceless/arctic_a0153.coef</td>
<td>2.412 2.482</td>
</tr>
<tr>
<td>s-eh</td>
<td>vowels/arctic_b0076.coef</td>
<td>1.887 1.982</td>
</tr>
<tr>
<td>eh-r</td>
<td>arctic_b0470.coef</td>
<td>0.407 0.477</td>
</tr>
<tr>
<td>r-iy</td>
<td>vowels/arctic_b0470.coef</td>
<td>0.477 0.552</td>
</tr>
<tr>
<td>iy-ae</td>
<td>vowels/arctic_a0350.coef</td>
<td>2.847 2.937</td>
</tr>
<tr>
<td>ae-k</td>
<td>arctic_b0463.coef</td>
<td>1.637 1.727</td>
</tr>
</tbody>
</table>
C. SYNTHESIS OF TEST UTTERANCES

k-s voiceless/arctic_b0463.coef  1.727 1.832
s-eh vowels/arctic_a0472.coef  4.187 4.272
eh-s voiceless/arctic_a0472.coef  4.272 4.367
s-er arctic_b0206.coef  4.457 4.607
er-iy vowels/arctic_a0378.coef  1.502 1.632
iy-z voiced/arctic_a0378.coef  1.632 1.782

"The reorganization of these countries took the form of revolution."

dh-ax vowels/arctic_a0218.coef  0.136 0.302
ax-r arctic_a0218.coef  0.302 0.412
r-iy vowels/arctic_a0522.coef  2.577 2.667
iy-ao vowels/arctic_a0522.coef  2.667 2.732
ao-r arctic_a0522.coef  2.732 2.812
r-g arctic_a0522.coef  2.812 2.882
g-ax vowels/arctic_a0522.coef  2.882 2.927
ax-n arctic_a0522.coef  2.927 2.972
n-ax vowels/arctic_a0522.coef  2.972 3.027
ax-z voiced/arctic_a0522.coef  3.027 3.102
z-ey vowels/arctic_a0522.coef  3.102 3.207
ey-sh voiceless/arctic_a0522.coef  3.207 3.337
sh-ax vowels/arctic_b0479.coef  0.702 0.802
ax-n arctic_b0479.coef  0.802 0.907
n-ah vowels/arctic_b0479.coef  0.907 1.057
ah-v voiced/arctic_b0479.coef  1.057 1.177
v-dh voiced/arctic_b0026.coef  2.082 2.162
dh-iy vowels/arctic_b0026.coef  2.162 2.262
iy-z voiced/arctic_b0026.coef  2.262 2.402
z-k arctic_a0128.coef  2.067 2.157
k-ah vowels/arctic_a0522.coef  0.637 0.727
### C.2. Modified Cost Function

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Dataset</th>
<th>Coef 1</th>
<th>Coef 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ah-n</td>
<td>arctic_a0522.coef</td>
<td>0.727</td>
<td>0.782</td>
</tr>
<tr>
<td>n-t</td>
<td>arctic_a0522.coef</td>
<td>0.782</td>
<td>0.852</td>
</tr>
<tr>
<td>t-r</td>
<td>arctic_a0522.coef</td>
<td>0.852</td>
<td>0.927</td>
</tr>
<tr>
<td>r-iy</td>
<td>vowels/arctic_a0522.coef</td>
<td>0.927</td>
<td>1.017</td>
</tr>
<tr>
<td>iy-z</td>
<td>voiced/arctic_a0522.coef</td>
<td>1.017</td>
<td>1.157</td>
</tr>
<tr>
<td>z-t</td>
<td>arctic_b0159.coef</td>
<td>0.352</td>
<td>0.472</td>
</tr>
<tr>
<td>t-uh</td>
<td>vowels/arctic_b0140.coef</td>
<td>2.042</td>
<td>2.132</td>
</tr>
<tr>
<td>uh-k</td>
<td>arctic_b0133.coef</td>
<td>1.767</td>
<td>1.867</td>
</tr>
<tr>
<td>k-dh</td>
<td>voiced/arctic_b0133.coef</td>
<td>1.867</td>
<td>2.002</td>
</tr>
<tr>
<td>dh-ax</td>
<td>vowels/arctic_a0096.coef</td>
<td>3.287</td>
<td>3.317</td>
</tr>
<tr>
<td>ax-f</td>
<td>voiceless/arctic_a0096.coef</td>
<td>3.317</td>
<td>3.412</td>
</tr>
<tr>
<td>f-ao</td>
<td>vowels/arctic_a0360.coef</td>
<td>2.882</td>
<td>3.012</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_a0360.coef</td>
<td>3.012</td>
<td>3.102</td>
</tr>
<tr>
<td>r-m</td>
<td>arctic_a0360.coef</td>
<td>3.102</td>
<td>3.207</td>
</tr>
<tr>
<td>m-ah</td>
<td>vowels/arctic_a0360.coef</td>
<td>3.207</td>
<td>3.322</td>
</tr>
<tr>
<td>ah-v</td>
<td>voiced/arctic_a0360.coef</td>
<td>3.322</td>
<td>3.412</td>
</tr>
<tr>
<td>v-r</td>
<td>arctic_a0464.coef</td>
<td>3.357</td>
<td>3.477</td>
</tr>
<tr>
<td>r-eh</td>
<td>vowels/arctic_b0224.coef</td>
<td>1.542</td>
<td>1.617</td>
</tr>
<tr>
<td>eh-v</td>
<td>voiced/arctic_b0224.coef</td>
<td>1.617</td>
<td>1.667</td>
</tr>
<tr>
<td>v-ax</td>
<td>vowels/arctic_b0224.coef</td>
<td>1.667</td>
<td>1.717</td>
</tr>
<tr>
<td>ax-l</td>
<td>arctic_b0224.coef</td>
<td>1.717</td>
<td>1.797</td>
</tr>
<tr>
<td>l-uw</td>
<td>vowels/arctic_b0224.coef</td>
<td>1.797</td>
<td>1.882</td>
</tr>
<tr>
<td>uw-sh</td>
<td>voiceless/arctic_b0224.coef</td>
<td>1.882</td>
<td>1.982</td>
</tr>
<tr>
<td>sh-ax</td>
<td>vowels/arctic_b0224.coef</td>
<td>1.982</td>
<td>2.077</td>
</tr>
<tr>
<td>ax-n</td>
<td>arctic_b0224.coef</td>
<td>2.077</td>
<td>2.137</td>
</tr>
</tbody>
</table>

“For the rest, he was a mere automaton.”

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Dataset</th>
<th>Coef 1</th>
<th>Coef 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-ao</td>
<td>vowels/arctic_a0176.coef</td>
<td>0.121</td>
<td>0.267</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_a0176.coef</td>
<td>0.267</td>
<td>0.322</td>
</tr>
</tbody>
</table>
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>Component</th>
<th>File Reference</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-dh voiced/arctic_a0176.coef</td>
<td>0.322 0.372</td>
<td></td>
</tr>
<tr>
<td>dh-ax vowels/arctic_b0350.coef</td>
<td>2.807 2.862</td>
<td></td>
</tr>
<tr>
<td>ax-r arctic_b0350.coef</td>
<td>2.862 2.957</td>
<td></td>
</tr>
<tr>
<td>r-eh vowels/arctic_b0350.coef</td>
<td>2.957 3.092</td>
<td></td>
</tr>
<tr>
<td>eh-s voiceless/arctic_b0350.coef</td>
<td>3.092 3.242</td>
<td></td>
</tr>
<tr>
<td>s-t arctic_b0210.coef</td>
<td>1.222 1.347</td>
<td></td>
</tr>
<tr>
<td>t-hh arctic_b0210.coef</td>
<td>1.347 1.457</td>
<td></td>
</tr>
<tr>
<td>hh-iy vowels/arctic_b0210.coef</td>
<td>1.457 1.557</td>
<td></td>
</tr>
<tr>
<td>iy-w arctic_b0489.coef</td>
<td>0.242 0.327</td>
<td></td>
</tr>
<tr>
<td>w-aa vowels/arctic_b0489.coef</td>
<td>0.327 0.402</td>
<td></td>
</tr>
<tr>
<td>aa-z voiced/arctic_b0489.coef</td>
<td>0.402 0.467</td>
<td></td>
</tr>
<tr>
<td>z-ax vowels/arctic_b0108.coef</td>
<td>0.497 0.557</td>
<td></td>
</tr>
<tr>
<td>ax-m arctic_b0108.coef</td>
<td>0.557 0.632</td>
<td></td>
</tr>
<tr>
<td>m-ih vowels/arctic_b0108.coef</td>
<td>0.632 0.717</td>
<td></td>
</tr>
<tr>
<td>ih-r arctic_b0108.coef</td>
<td>0.717 0.802</td>
<td></td>
</tr>
<tr>
<td>r-ao vowels/arctic_b0197.coef</td>
<td>1.107 1.252</td>
<td></td>
</tr>
<tr>
<td>ao-t arctic_b0537.coef</td>
<td>1.687 1.832</td>
<td></td>
</tr>
<tr>
<td>t-aa vowels/arctic_b0537.coef</td>
<td>1.832 1.952</td>
<td></td>
</tr>
<tr>
<td>aa-m arctic_b0537.coef</td>
<td>1.952 2.047</td>
<td></td>
</tr>
<tr>
<td>m-ax vowels/arctic_b0537.coef</td>
<td>2.047 2.122</td>
<td></td>
</tr>
<tr>
<td>ax-t arctic_b0537.coef</td>
<td>2.122 2.212</td>
<td></td>
</tr>
<tr>
<td>t-aa vowels/arctic_b0537.coef</td>
<td>2.212 2.377</td>
<td></td>
</tr>
<tr>
<td>aa-n arctic_b0537.coef</td>
<td>2.377 2.552</td>
<td></td>
</tr>
</tbody>
</table>

“Not at this particular case, Tom, apologised Whittemore.”
<table>
<thead>
<tr>
<th>Pair</th>
<th>Directory</th>
<th>Cost Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>t-dh</td>
<td>voiced/arctic_b0082.coef</td>
<td>2.597 2.682</td>
</tr>
<tr>
<td>dh-ih</td>
<td>vowels/arctic_b0082.coef</td>
<td>2.682 2.732</td>
</tr>
<tr>
<td>ih-s</td>
<td>voiceless/arctic_b0082.coef</td>
<td>2.732 2.817</td>
</tr>
<tr>
<td>s-p</td>
<td>arctic_b0490.coef</td>
<td>1.357 1.447</td>
</tr>
<tr>
<td>p-er</td>
<td>arctic_b0490.coef</td>
<td>1.447 1.557</td>
</tr>
<tr>
<td>er-t</td>
<td>arctic_b0215.coef</td>
<td>1.102 1.192</td>
</tr>
<tr>
<td>t-ih</td>
<td>vowels/arctic_a0463.coef</td>
<td>3.312 3.397</td>
</tr>
<tr>
<td>ih-k</td>
<td>arctic_a0463.coef</td>
<td>3.397 3.457</td>
</tr>
<tr>
<td>k-y</td>
<td>arctic_a0463.coef</td>
<td>3.457 3.522</td>
</tr>
<tr>
<td>y-ax</td>
<td>vowels/arctic_a0463.coef</td>
<td>3.522 3.557</td>
</tr>
<tr>
<td>ax-l</td>
<td>arctic_b0427.coef</td>
<td>3.047 3.097</td>
</tr>
<tr>
<td>l-er</td>
<td>arctic_b0427.coef</td>
<td>3.097 3.252</td>
</tr>
<tr>
<td>er-k</td>
<td>arctic_a0197.coef</td>
<td>2.027 2.157</td>
</tr>
<tr>
<td>k-ey</td>
<td>vowels/arctic_b0075.coef</td>
<td>0.607 0.757</td>
</tr>
<tr>
<td>ey-s</td>
<td>voiceless/arctic_b0075.coef</td>
<td>0.757 0.942</td>
</tr>
<tr>
<td>s-t</td>
<td>arctic_a0095.coef</td>
<td>0.997 1.137</td>
</tr>
<tr>
<td>t-AA</td>
<td>vowels/arctic_a0480.coef</td>
<td>0.086 0.237</td>
</tr>
<tr>
<td>AA-M</td>
<td>arctic_a0480.coef</td>
<td>0.237 0.347</td>
</tr>
<tr>
<td>M-AX</td>
<td>vowels/arctic_a0486.coef</td>
<td>1.957 2.017</td>
</tr>
<tr>
<td>AX-P</td>
<td>arctic_b0066.coef</td>
<td>2.037 2.137</td>
</tr>
<tr>
<td>P-AA</td>
<td>vowels/arctic_b0066.coef</td>
<td>2.137 2.252</td>
</tr>
<tr>
<td>AA-L</td>
<td>arctic_b0066.coef</td>
<td>2.252 2.337</td>
</tr>
<tr>
<td>L-AX</td>
<td>vowels/arctic_b0066.coef</td>
<td>2.337 2.407</td>
</tr>
<tr>
<td>AX-JH</td>
<td>arctic_b0066.coef</td>
<td>2.407 2.487</td>
</tr>
<tr>
<td>JH-AY</td>
<td>vowels/arctic_b0066.coef</td>
<td>2.487 2.662</td>
</tr>
<tr>
<td>AY-Z</td>
<td>voiced/arctic_b0066.coef</td>
<td>2.662 2.877</td>
</tr>
<tr>
<td>Z-D</td>
<td>arctic_b0287.coef</td>
<td>2.007 2.082</td>
</tr>
<tr>
<td>D-W</td>
<td>arctic_b0287.coef</td>
<td>2.082 2.157</td>
</tr>
<tr>
<td>W-IH</td>
<td>vowels/arctic_b0020.coef</td>
<td>2.597 2.667</td>
</tr>
<tr>
<td>IH-T</td>
<td>arctic_b0020.coef</td>
<td>2.667 2.722</td>
</tr>
</tbody>
</table>
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Source</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>t-m</td>
<td>arctic_b0020.coef</td>
<td>2.722 2.807</td>
</tr>
<tr>
<td>m-ao</td>
<td>vowels/arctic_b0020.coef</td>
<td>2.807 2.927</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_b0020.coef</td>
<td>2.927 3.032</td>
</tr>
</tbody>
</table>

**“Will we ever forget it”**

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Source</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-ih</td>
<td>vowels/arctic_a0153.coef</td>
<td>3.337 3.407</td>
</tr>
<tr>
<td>ih-l</td>
<td>arctic_b0393.coef</td>
<td>0.692 0.792</td>
</tr>
<tr>
<td>l-w</td>
<td>arctic_b0393.coef</td>
<td>0.792 0.922</td>
</tr>
<tr>
<td>w-iy</td>
<td>vowels/arctic_b0381.coef</td>
<td>1.132 1.247</td>
</tr>
<tr>
<td>iy-eh</td>
<td>vowels/arctic_b0381.coef</td>
<td>1.247 1.352</td>
</tr>
<tr>
<td>eh-v</td>
<td>voiced/arctic_b0381.coef</td>
<td>1.352 1.427</td>
</tr>
<tr>
<td>v-er</td>
<td>arctic_b0381.coef</td>
<td>1.427 1.507</td>
</tr>
<tr>
<td>er-f</td>
<td>voiceless/arctic_b0381.coef</td>
<td>1.507 1.617</td>
</tr>
<tr>
<td>f-er</td>
<td>arctic_a0143.coef</td>
<td>0.952 1.037</td>
</tr>
<tr>
<td>er-g</td>
<td>arctic_a0143.coef</td>
<td>1.037 1.117</td>
</tr>
<tr>
<td>g-eh</td>
<td>vowels/arctic_a0461.coef</td>
<td>1.932 2.047</td>
</tr>
<tr>
<td>eh-t</td>
<td>arctic_a0525.coef</td>
<td>2.987 3.082</td>
</tr>
<tr>
<td>t-ih</td>
<td>vowels/arctic_a0525.coef</td>
<td>3.082 3.207</td>
</tr>
<tr>
<td>ih-t</td>
<td>arctic_a0525.coef</td>
<td>3.207 3.307</td>
</tr>
</tbody>
</table>

**“And you always want to see it in the superlative degree.”**

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Source</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>ae-n</td>
<td>arctic_b0181.coef</td>
<td>0.156 0.342</td>
</tr>
<tr>
<td>n-d</td>
<td>arctic_b0181.coef</td>
<td>0.342 0.397</td>
</tr>
<tr>
<td>d-y</td>
<td>arctic_b0181.coef</td>
<td>0.397 0.467</td>
</tr>
<tr>
<td>y-uw</td>
<td>vowels/arctic_b0181.coef</td>
<td>0.467 0.527</td>
</tr>
<tr>
<td>uw-ao</td>
<td>vowels/arctic_b0503.coef</td>
<td>2.777 2.947</td>
</tr>
<tr>
<td>ao-l</td>
<td>arctic_a0246.coef</td>
<td>0.787 0.902</td>
</tr>
<tr>
<td>l-w</td>
<td>arctic_a0246.coef</td>
<td>0.902 0.997</td>
</tr>
<tr>
<td>w-ey</td>
<td>vowels/arctic_a0246.coef</td>
<td>0.997 1.087</td>
</tr>
<tr>
<td>ey-z</td>
<td>voiced/arctic_a0246.coef</td>
<td>1.087 1.222</td>
</tr>
</tbody>
</table>
"He turned sharply and faced Gregson across the table."
### C. Synthesis of Test Utterances

<table>
<thead>
<tr>
<th>Vowel/Phoneme</th>
<th>File/Group</th>
<th>Coefficient 1</th>
<th>Coefficient 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>hh-iy</td>
<td>vowels/arctic_b0010.coef</td>
<td>0.076</td>
<td>0.192</td>
</tr>
<tr>
<td>iy-t</td>
<td>arctic_b0010.coef</td>
<td>0.192</td>
<td>0.292</td>
</tr>
<tr>
<td>t-er</td>
<td>arctic_b0010.coef</td>
<td>0.292</td>
<td>0.432</td>
</tr>
<tr>
<td>er-n</td>
<td>arctic_b0010.coef</td>
<td>0.432</td>
<td>0.532</td>
</tr>
<tr>
<td>n-d</td>
<td>arctic_b0298.coef</td>
<td>0.637</td>
<td>0.727</td>
</tr>
<tr>
<td>d-sh</td>
<td>voiceless/arctic_b0298.coef</td>
<td>0.727</td>
<td>0.872</td>
</tr>
<tr>
<td>sh-aa</td>
<td>vowels/arctic_a0472.coef</td>
<td>2.352</td>
<td>2.467</td>
</tr>
<tr>
<td>aa-r</td>
<td>arctic_a0472.coef</td>
<td>2.467</td>
<td>2.527</td>
</tr>
<tr>
<td>r-p</td>
<td>arctic_a0472.coef</td>
<td>2.527</td>
<td>2.592</td>
</tr>
<tr>
<td>p-l</td>
<td>arctic_a0472.coef</td>
<td>2.592</td>
<td>2.667</td>
</tr>
<tr>
<td>l-iy</td>
<td>vowels/arctic_a0472.coef</td>
<td>2.667</td>
<td>2.722</td>
</tr>
<tr>
<td>iy-ae</td>
<td>vowels/arctic_a0387.coef</td>
<td>3.332</td>
<td>3.607</td>
</tr>
<tr>
<td>ae-n</td>
<td>arctic_a0387.coef</td>
<td>3.607</td>
<td>3.807</td>
</tr>
<tr>
<td>n-d</td>
<td>arctic_b0440.coef</td>
<td>2.942</td>
<td>3.032</td>
</tr>
<tr>
<td>d-f</td>
<td>voiceless/arctic_b0440.coef</td>
<td>3.032</td>
<td>3.142</td>
</tr>
<tr>
<td>f-ey</td>
<td>vowels/arctic_a0046.coef</td>
<td>0.582</td>
<td>0.697</td>
</tr>
<tr>
<td>ey-s</td>
<td>voiceless/arctic_a0046.coef</td>
<td>0.697</td>
<td>0.802</td>
</tr>
<tr>
<td>s-t</td>
<td>arctic_a0513.coef</td>
<td>0.467</td>
<td>0.542</td>
</tr>
<tr>
<td>t-g</td>
<td>arctic_a0513.coef</td>
<td>0.542</td>
<td>0.622</td>
</tr>
<tr>
<td>g-r</td>
<td>arctic_a0012.coef</td>
<td>0.071</td>
<td>0.177</td>
</tr>
<tr>
<td>r-eh</td>
<td>vowels/arctic_a0012.coef</td>
<td>0.177</td>
<td>0.232</td>
</tr>
<tr>
<td>eh-g</td>
<td>arctic_a0012.coef</td>
<td>0.232</td>
<td>0.292</td>
</tr>
<tr>
<td>g-s</td>
<td>voiceless/arctic_a0012.coef</td>
<td>0.292</td>
<td>0.372</td>
</tr>
<tr>
<td>s-ax</td>
<td>vowels/arctic_a0012.coef</td>
<td>0.372</td>
<td>0.432</td>
</tr>
<tr>
<td>ax-n</td>
<td>arctic_a0012.coef</td>
<td>0.432</td>
<td>0.492</td>
</tr>
<tr>
<td>n-ax</td>
<td>vowels/arctic_b0422.coef</td>
<td>1.842</td>
<td>1.967</td>
</tr>
<tr>
<td>ax-k</td>
<td>arctic_b0292.coef</td>
<td>4.137</td>
<td>4.257</td>
</tr>
<tr>
<td>k-r</td>
<td>arctic_b0292.coef</td>
<td>4.257</td>
<td>4.387</td>
</tr>
<tr>
<td>r-ao</td>
<td>vowels/arctic_b0292.coef</td>
<td>4.387</td>
<td>4.472</td>
</tr>
<tr>
<td>ao-s</td>
<td>voiceless/arctic_b0292.coef</td>
<td>4.472</td>
<td>4.582</td>
</tr>
</tbody>
</table>
### C.2. Modified Cost Function

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Model</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-dh</td>
<td>voiced/arctic_b0292.coef</td>
<td>4.582 4.682</td>
</tr>
<tr>
<td>dh-ax</td>
<td>vowels/arctic_b0019.coef</td>
<td>2.577 2.607</td>
</tr>
<tr>
<td>ax-t</td>
<td>arctic_b0019.coef</td>
<td>2.607 2.697</td>
</tr>
<tr>
<td>t-ey</td>
<td>vowels/arctic_b0019.coef</td>
<td>2.697 2.832</td>
</tr>
<tr>
<td>ey-b</td>
<td>arctic_b0019.coef</td>
<td>2.832 2.947</td>
</tr>
<tr>
<td>b-ax</td>
<td>vowels/arctic_b0019.coef</td>
<td>2.947 3.017</td>
</tr>
<tr>
<td>ax-l</td>
<td>arctic_b0019.coef</td>
<td>3.017 3.092</td>
</tr>
</tbody>
</table>

"I followed the line of the proposed railroad, looking for chances."

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Model</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>ay-f</td>
<td>voiceless/arctic_b0366.coef</td>
<td>0.136 0.337</td>
</tr>
<tr>
<td>f-AA</td>
<td>vowels/arctic_b0133.coef</td>
<td>0.407 0.517</td>
</tr>
<tr>
<td>aa-l</td>
<td>arctic_b0133.coef</td>
<td>0.517 0.612</td>
</tr>
<tr>
<td>l-ow</td>
<td>vowels/arctic_b0133.coef</td>
<td>0.612 0.707</td>
</tr>
<tr>
<td>ow-d</td>
<td>arctic_b0133.coef</td>
<td>0.707 0.812</td>
</tr>
<tr>
<td>d-dh</td>
<td>voiced/arctic_a0253.coef</td>
<td>2.172 2.247</td>
</tr>
<tr>
<td>dh-ax</td>
<td>vowels/arctic_a0498.coef</td>
<td>0.136 0.292</td>
</tr>
<tr>
<td>ax-l</td>
<td>arctic_a0498.coef</td>
<td>0.292 0.387</td>
</tr>
<tr>
<td>l-ay</td>
<td>vowels/arctic_a0498.coef</td>
<td>0.387 0.562</td>
</tr>
<tr>
<td>ay-n</td>
<td>arctic_b0498.coef</td>
<td>1.577 1.772</td>
</tr>
<tr>
<td>n-ah</td>
<td>vowels/arctic_b0498.coef</td>
<td>1.772 1.882</td>
</tr>
<tr>
<td>ah-v</td>
<td>voiced/arctic_b0498.coef</td>
<td>1.882 1.957</td>
</tr>
<tr>
<td>v-dh</td>
<td>voiced/arctic_b0498.coef</td>
<td>1.957 2.027</td>
</tr>
<tr>
<td>dh-ax</td>
<td>vowels/arctic_a0073.coef</td>
<td>0.121 0.262</td>
</tr>
<tr>
<td>ax-p</td>
<td>arctic_a0073.coef</td>
<td>0.262 0.337</td>
</tr>
<tr>
<td>p-r</td>
<td>arctic_a0073.coef</td>
<td>0.337 0.417</td>
</tr>
<tr>
<td>r-ax</td>
<td>vowels/arctic_a0073.coef</td>
<td>0.417 0.457</td>
</tr>
<tr>
<td>ax-p</td>
<td>arctic_b0047.coef</td>
<td>0.282 0.372</td>
</tr>
<tr>
<td>p-ow</td>
<td>vowels/arctic_b0047.coef</td>
<td>0.372 0.522</td>
</tr>
<tr>
<td>ow-z</td>
<td>voiced/arctic_b0047.coef</td>
<td>0.522 0.672</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>z-d</td>
<td>arctic_a0065.coef</td>
<td>0.747 0.827</td>
</tr>
<tr>
<td>d-r</td>
<td>arctic_b0532.coef</td>
<td>2.662 2.742</td>
</tr>
<tr>
<td>r-ey</td>
<td>vowels/arctic_b0269.coef</td>
<td>1.292 1.402</td>
</tr>
<tr>
<td>ey-l</td>
<td>arctic_b0269.coef</td>
<td>1.402 1.497</td>
</tr>
<tr>
<td>l-r</td>
<td>arctic_b0269.coef</td>
<td>1.497 1.612</td>
</tr>
<tr>
<td>r-ow</td>
<td>vowels/arctic_b0269.coef</td>
<td>1.612 1.737</td>
</tr>
<tr>
<td>ow-d</td>
<td>arctic_b0269.coef</td>
<td>1.737 1.862</td>
</tr>
<tr>
<td>d-l</td>
<td>arctic_a0539.coef</td>
<td>0.512 0.622</td>
</tr>
<tr>
<td>l-uh</td>
<td>vowels/arctic_b0399.coef</td>
<td>0.452 0.522</td>
</tr>
<tr>
<td>uh-k</td>
<td>arctic_b0399.coef</td>
<td>0.522 0.587</td>
</tr>
<tr>
<td>k-ih</td>
<td>vowels/arctic_b0399.coef</td>
<td>0.587 0.647</td>
</tr>
<tr>
<td>ih-ng</td>
<td>arctic_b0415.coef</td>
<td>0.867 0.937</td>
</tr>
<tr>
<td>ng-f</td>
<td>voiceless/arctic_b0415.coef</td>
<td>0.937 1.042</td>
</tr>
<tr>
<td>f-ao</td>
<td>vowels/arctic_b0415.coef</td>
<td>1.042 1.107</td>
</tr>
<tr>
<td>ao-r</td>
<td>arctic_b0415.coef</td>
<td>1.107 1.182</td>
</tr>
<tr>
<td>r-ch</td>
<td>arctic_a0418.coef</td>
<td>0.502 0.592</td>
</tr>
<tr>
<td>ch-ae</td>
<td>vowels/arctic_a0038.coef</td>
<td>1.312 1.447</td>
</tr>
<tr>
<td>ae-n</td>
<td>arctic_a0038.coef</td>
<td>1.447 1.537</td>
</tr>
<tr>
<td>n-s</td>
<td>voiceless/arctic_a0038.coef</td>
<td>1.537 1.617</td>
</tr>
<tr>
<td>s-ax</td>
<td>vowels/arctic_a0038.coef</td>
<td>1.617 1.707</td>
</tr>
<tr>
<td>ax-z</td>
<td>voiced/arctic_a0038.coef</td>
<td>1.707 1.852</td>
</tr>
</tbody>
</table>