Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.

http://www.ub.tuwien.ac.at

TU WIEN UB Universitätsbibliothek

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.

http://www.ub.tuwien.ac.at/eng

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Visual Control of Acoustic Speech Synthesis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Jakob Johannes Hollenstein
Matrikelnummer 0625702

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ. Prof. Dipl.-Inf. Dr. rer. nat. Jens Knoop
Mitwirkung: Dr. techn. Michael Pucher

Wien, 4.12.2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Visual Control of Acoustic Speech Synthesis

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Computational Intelligence

by

Jakob Johannes Hollenstein
Registration Number 0625702

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ. Prof. Dipl.-Inf. Dr. rer. nat. Jens Knoop
Assistance: Dr. techn. Michael Pucher

Vienna, 4.12.2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Jakob Johannes Hollenstein
Spengergasse 7/2-3, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Acknowledgements

I would like to express my gratitude to my supervisors Jens Knoop and Michael Pucher for their useful comments and remarks and their guidance through the learning process of this master thesis. Furthermore I would like to thank my colleagues Dietmar Schabus and Markus Toman for the much appreciated discussions and support. Also I would like to thank Korin Richmond, Ming Lei, and Zhen-Hua Ling for providing us with an HMM-based dependency modelling system. Last but not least I am grateful to my family for their unconditional support.

This work was supported by the Austrian Science Fund (FWF): P22890-N23. The Competence Center FTW Forschungszentrum Telekommunikation Wien GmbH is funded within the program COMET - Competence Centers for Excellent Technologies by BMVIT, BMWA, and the City of Vienna. The COMET program is managed by the FFG.

Abstract

Speech synthesis based on Hidden Markov Model (HMM) has become a well known and widely applied technique. One benefit of statistical modelling of speech over signal concatenation approaches is greater flexibility since the parameter trajectories can be modified more easily than the audio signal. However, the acoustic features used for speech synthesis are high dimensional and difficult to modify even for an expert. The idea of control is to find a more intuitive representation and a mapping from this more intuitive representation to the acoustic feature space. Then changes applied in the intuitive space should be mapped to appropriate changes in the difficult to understand acoustic feature space. This would allow modification by expert knowledge and thus lead to a more flexible synthesis system. In previous work articulatory features have been used to control acoustic synthesis, since they lend themselves to modification based on linguistic knowledge. More recently formants have been used to control speech synthesis as well.

Since articulatory data is inherently more difficult to capture than visual data, the question arises whether visual data can also be used to control and modify the acoustic parameters similar to articulatory control. To answer the question whether visual data can be used for control in a similar way to formant and articulatory data, the system used for formant control was adapted to the visual features and a series of experiments was performed to gain indication on whether the relation between visual and acoustic data is sufficiently strong. The mapping investigated for articulatory and formant based control consists of state-based piecewise linear transformations from the control to the acoustic space.

It was found that restricting the visual control space appropriately leads to sufficiently distinct visual representations and thus allows for control modelling. Improvements of the acoustic synthesis quality with respect to the uncontrolled synthesis system are unlikely due to the necessary restrictions. A less restricted and more precise mapping technique would be necessary to improve the quality of the controlled synthesis system. Subjective evaluation results indicate that phonetically meaningful control by visual only features is feasible.

Kurzfassung

Hidden Markov Model-basierte Sprachsynthese hat sich zu einer etablierten und weitverbreiteten Technik entwickelt. Dabei wird das Sprachsignal analysiert und durch hochdimensionale Parameterverläufe dargestellt. Ein Vorteil der statistisch parametrischen Modellierung des Sprachsignals gegenüber Ansätzen, die Teile des ursprünglichen Signals durch Signalverarbeitungsmethoden neu zusammensetzen, ist die größere Flexibilität, da die Parameterverläufe einer Veränderung leichter zugänglich sind als dies beim Audiosignal der Fall wäre. Allerdings sind die verwendeten akustischen Features hochdimensional und selbst für einen Experten nur sehr schwer zu verändern.

Die Idee der Steuerung ist es, eine Darstellung zu finden, die intuitiver zugänglich ist, als dies bei den akustischen Features der Fall ist. Zusätzlich soll eine Abbildung aus dieser intuitiven Repräsentation in den akustischen Featurespace gefunden werden. Damit sollen Änderungen, die im intuitiven Raum vorgenommen werden, sinnvoll in den akustischen Featurespace abgebildet werden. Dies würde in weiterer Folge auch die Modifikation durch einen Experten ermöglichen und so auch die Flexibilität der Synthese vergrößern. In früheren Arbeiten wurden artikulatorische Features verwendet, um akustische Synthese zu steuern, da diese Features auf linguistischem Wissen basierende Änderungen erlauben. Desweiteren wurden auch Formant-Features zur Steuerung von akustischer Sprachsynthese erprobt.

Da artikulatorische Daten von Natur aus schwieriger zu erfassen sind als visuelle Daten, stellt sich die Frage, ob visuelle Daten verwendet werden können um die akustische Synthese zu steuern bzw. zu verändern.

In dieser Arbeit geht es um die Frage, ob visuelle Daten in ähnlicher Weise zur Steuerung oder Modifikation der akustischen Synthese eingesetzt werden können, wie dies für Formant- und artikulatorische Features gezeigt wurde. Dazu wurde das System, das für formantenbasierte Steuerung verwendet wurde, an die visuellen Features angepasst. Eine Reihe von Experimenten zur Frage, wie stark die Abhängigkeit zwischen den akustischen und visuellen Daten ist, wurde durchgeführt.

Die Abbildung für artikulatorische- und formantenbasierte Steuerung basiert auf zustandsabhängigen, stückweise linearen Transformationen vom steuerungs- in den akustischen Featurespace. Durch einschränken des visuellen Featurespaces bzw. der visuellen Steuerung werden die zughörigen visuellen Features aussagekräftiger im Bezug auf die unterschiedlichen Phone. Dadurch wird es möglich, eine Abbildung von den visuellen auf die akustischen Features zu finden. Aufgrund der notwendigen Einschränkungen sind Verbesserungen der Qualität gegenüber dem Synthesesystem ohne Steuerungserweiterung nicht zu erwarten, solange keine ausgefeilteren Abbildungstechniken gefunden werden.

Die Resultate der subjektiven Evaluierung zeigen, dass akustisch sinnvolle Steuerung durch die rein visuellen Features möglich ist.

Contents

1 Introduction **1**
- 1.1 What is the field and its history? 1
- 1.2 What is this work about? 1
- 1.3 Related work 2
- 1.4 Applications of audio-visual speech processing in language learning 3
- 1.5 Structure of the thesis 4

2 Method **5**
- 2.1 Overview of data-based parametric speech synthesis 5
  - 2.1.1 Text to speech & steps 5
  - 2.1.2 Phonetic symbols 6
  - 2.1.3 Corpus 6
  - 2.1.4 Austrian German 6
  - 2.1.5 Vocoding in parametric speech synthesis 8
  - 2.1.6 Source-filter model 8
  - 2.1.7 Spectral speech representation 8
  - 2.1.8 LPC 9
  - 2.1.9 Rough sketch of synthesis 11
  - 2.1.10 Audio/visual speech 11
  - 2.1.11 Marker recording 11
  - 2.1.12 Scope 13
- 2.2 HMM concepts 14
  - 2.2.1 What is an HMM? 14
  - 2.2.2 HMM formally 15
  - 2.2.3 Gaussian mixtures 16
  - 2.2.4 HMM formally continued 16
- 2.3 HMMs in speech processing 17
  - 2.3.1 Relation to speech recognition 17
  - 2.3.2 HMM left-to-right model 17
  - 2.3.3 Monophone HMM 18
  - 2.3.4 Three basic problems of HMM based speech processing 18
  - 2.3.5 Coarticulation effects and fullcontext 19
- 2.4 Synthesis 19
  - 2.4.1 Clustering & trees 19
  - 2.4.2 Synthesising from trees 20
  - 2.4.3 Delta and delta-delta features 20
  - 2.4.4 Spectral, visual, and F0 features 21

2.4.5 Control, control modelling and the control functions in the tree . . . 21
2.5 Training in detail . . . 21
2.5.1 What is a model? . . . 21
2.5.2 How to get the model from the data and how to start? . . . 22
2.5.3 Model initialisation . . . 22
2.5.4 Viterbi alignment and training . . . 22
2.5.5 Forward algorithm . . . 22
2.5.6 Viterbi . . . 23
2.5.7 Model initialisation and why . . . 24
2.5.8 Backward algorithm . . . 24
2.5.9 State occupancy probability . . . 25
2.5.10 Baum-Welch . . . 25
2.5.11 GMM parameter estimation . . . 26
2.5.12 Non-embedded and embedded training . . . 26
2.5.13 Fullcontext cloning . . . 26
2.5.14 Duration models . . . 27
2.6 Synthesis in detail . . . 28
2.6.1 How do we generate speech signals from the model? . . . 28
2.6.2 PDF stream - mean/variance stream . . . 29
2.6.3 SciPy implementation . . . 32
2.6.4 Banded matrix representation . . . 32
2.6.5 How does the control work, what is different with control? . . . 33
2.6.6 Linear regression . . . 33
2.6.7 Initialisation . . . 34
2.6.8 Parameter generation . . . 34
2.6.9 Parameter generation with control math detail . . . 35

3 Experiments **39**
3.1 Possible problems with using visual features for control . . . 40
3.2 Insufficient control . . . 40
3.3 Requirements for control . . . 40
3.4 Control data clustering . . . 44
3.5 Clustering of formants . . . 44
3.6 EMA data . . . 45
3.7 Restriction of markers . . . 47
3.8 Restriction to /aː eː iː oː uː/ . . . 47
3.9 Linear regression experiments . . . 48
3.10 Comparing the 41 dimensional acoustic vector to monophone models . . . 48
3.11 Common transformations and intercepts . . . 49
3.12 Experiments with common transformations and intercepts . . . 49
3.13 Approximate trajectory generation . . . 49
3.14 Exercising visual control . . . 50
3.14.1 Pitch and visual control . . . 52
3.14.2 Selection of control space . . . 53
3.15 Tools & implementation . . . 58

4 Evaluation & Findings **59**
4.1 Experiment setup . . . 59
4.2 Results . . . 61

4.2.1 Details 63
4.3 Interpretation 64
4.4 Summary of findings 65

5 Conclusion and Future Work **67**

Bibliography **69**

A Appendix **75**
A.1 Band matrix 75
A.2 WUW is a banded matrix 75
A.3 Evaluation result confidence intervals 76

Acronyms **78**

Index **79**

CHAPTER 1

Introduction

1.1 What is the field and its history?

Speech Synthesis or Text-to-Spech (TTS) is the process of converting text to speech. This involves transforming the textual representation into a phonetic representation and generating speech sounds from this phonetic representation. Attempts to create human perceivable speech sounds date back to the 18th century [1] and development of electronic methods started as early as 1928 with the invention of the Voice Encoder (vocoder) [2, 3]. This and the following improvements in speech coding techniques and thus in transforming the speech signal into slowly varying parametric representations, is what made parametric speech synthesis possible.

After rule-based synthesis systems have shown the feasibility of speech synthesis and with the advance of computer processing power, data based methods became possible. Two prominent categories of these techniques include *unit selection* (for example described by Hunt and Black [4]) and HMM based speech synthesis [5].

In unit selection, dynamic programming is used to select recorded audio samples which are then concatenated using signal processing techniques to create new utterances. HMM based speech synthesis uses fore mentioned speech coding techniques to synthesise the corresponding waveforms from parameter trajectories. Thus *HMM based speech synthesis* is part of the *parametric synthesis* methods. Naturalness of HMM based synthesis depends largely on the quality of the speech coding techniques and thus is more difficult to achieve. However HMM based speech synthesis is also more flexible with regard to the generation of utterances that were not present in the recorded corpus.

1.2 What is this work about?

This work investigates the question whether visual speech features can be used to modify the acoustic speech synthesis within the HMM based parametric speech synthesis framework.

An advantage of parametric synthesis is the possibility to directly modify the acoustic parameters. Thereby creating new or different sounds. This can be used for more flexible speech synthesis. Since the feature space is high dimensional (often forty or more dimensions) it is unintiutive and difficult to change these parameters directly, even for an expert. A viable option is to use simpler or more intuitive parameters and a function that maps modifications from the more intuitive parameters to the acoustic parameters.

In audiovisual speech synthesis some visual representation of the speaker or more commonly the speaker's face is synthesised synchronously with the audio signal. In this work parametric visual recordings based on recording three-dimensional positional trajectories of markers glued to a speaker's face are used as the visual channel. The question whether this visual representation of a speaker's face can be used as a more intuitive space to influence the generated audio is investigated. Thus in a wider sense this work is also about investigating some aspects of the relationship between visual and acoustic modalities.

Using visual-only features is different from previous experiments where information regarding the tongue position was also available. The reason to use visual-only data is that visual data is easier to record than articulatory data. Some possibilities and limitations of using visual-only data instead of articulatory data are highlighted in this work.

The data used consists of studio quality audio recordings and synchronous marker based motion capturing of the face motion. Although this kind of visual data is also difficult to record, progress in marker-less motion capturing as well as motion capturing based on off-the-shelf consumer products hint at wider availability of visual speech data in the future. This and the inherent difficulties in recording articulatory data make investigating the audio-visual relation worthwhile.

More specifically this work investigates possibilities and difficulties of using piece wise linear functions to map from synchronous facial marker recordings to the acoustic parameters. The system used in this work is based on the system used for formant control [6]. Tools for investigation and modification of the data models as well as the parameter generation algorithm have been implemented. The underlying system used for training has been patched to add the features necessary for the experiments. An experiment with subjective evaluations has been conducted to assess the feasibility of visual control.

Possible applications include intuitive explicit modification, controlling of the synthesis based on physical constraints of the visual features and providing clues for language learners.

1.3 Related work

Fundamental to HMM based speech synthesis is the generation of parameter trajectories from the HMM states. Tokuda et al. [5] describe an algorithm to generate smooth trajectories from the observation probability density functions associated with the HMM states. This algorithm is also implemented in this work. Yoshimura et al. [7, 8] show how to model duration as well as spectral and pitch features within the HMM framework. A general overview of the state of the art of HMM based speech synthesis is given by Zen et al. [9].

Flexibility of HMM based speech synthesis can be increased by modelling different speaking styles, as for example described by Yamagishi [10]. However this increase of flexibility depends on the availability of additional data, for example data of a certain speaking style. An alternative that does not rely on additional recordings is to employ control to modify the already available data.

To modify the acoustic parameters by modifying more intuitive parameters a mapping from the intuitive to the acoustic parameters has to be established. This can be done by modelling the relationship between the feature streams, for example an acoustic and a visual stream as in our case, in terms of piece-wise linear functions. This model of multiple-linear-regressions bound to the HMM states is called *Multiple Regression Hidden Markov Model (MR-HMM)* and has been described by Fujinaga et al. [11]. The feasibility of using piece-wise linear mappings for control has been demonstrated for modification by articulatory data, that is position recordings of markers glued to the tongue, as reported by Ling et al. [12, 13]. Control of acoustic HMM based speech synthesis by modifying associated formant features has been described by Lei et al. [6]. Furthermore Ling et al. [14] describe how articulatory control was used to create the sound of a vowel (/ʌ/) that was not originally present in the training corpus, thus showing that articulatory control can effectively be used to enhance the flexibility of speech synthesis. Similar to the work of Ling et al. [12] and Lei et al. [6] this work also investigates control by using piecewise linear state based transformations from the control to the acoustic feature space.

Work done by Youssef et al. [15] explores the possibility to perform acoustic-to-articulatory inversion by acoustic recognition and articulatory resynthesis. While this investigates the dependency between acoustic and articulatory features, which may be somewhat similar to the visual-to-acoustic relationship, the use of recognition and resynthesis does not easily allow for modification or control as is desired in this work. Kjellström and Engwall [16] discuss the integration of visual features into an audiovisual-to-articulatory inversion system, thus investigating the dependency of audio-visual to articulatory features. The dependency was modelled by linear as well as non-linear techniques. In contrast, this work investigates the visual-to-audio dependency using state based piecewise linear functions.

1.4 Applications of audio-visual speech processing in language learning

Audio-visual speech synthesis and audio-visual speech processing in general could be useful for language learning.

Studies indicate that perceptual accuracy in language learning can be improved by audio-visual training stimuli [17]. The effectiveness however seems to depend on the susceptibility of the student to the visual cues, but this sensitivity is also improved by audio-visual training [18]. This indicates that presenting speech for language learning in an audio-visual way is useful.

Interpretation of animated illustrations of the vocal tract have been found to be feasible without any previous training [19]. While it was more difficult for participants to identify the correct phonemes (about $\sim 20\%$ correct, with a chance level of $\sim 7\%$), identification of articulatory features (for example rounding: rounded, neutral, spreaded) was more successful ($\sim 60\%$ with a chance level at $\sim 37\%$). Using audio-articulatory inversion to infer articulatory movements from acoustic signals to provide feedback in a similar way has been investigated by Badin et al. [20].

Since audio-visual cues can help language learners, language learning systems that employ audio-visual techniques have been developed, for example in a system described by Wang et al. [21] which features audio-visual synthesis and a system described by Jokisch et al. [22] which employs video tracking to capture lip-width and lip-height with the goal to develop methods to provide feedback using these features.

Modelling the relationship of visual changes and corresponding acoustic changes, could be used to make feedback more effective, for example by capturing visual speech information during learning sessions, and providing information how changes of the visual features would influence sound production.

An utterance synthesised by the system using a dependency model could be used to experiment with manually modifying the visual feature trajectories, for example changing the mouth opening and observing how this would change the corresponding sound.

An envisioned more advanced scenario is illustrated in Figure 1.1. In this scenario the captured audio-visual information of a language learner could be used to adapt the learning system corpus to produce an audio-visual dependency model tailored to the user. Then the system could present the user with utterances and illustrations of the trajectories of the visual features as well as a visual animation of a face. The user could then modify the trajectories, for example increase mouth opening at a certain point of the trajectories, and experience how this would change the sound. This could facilitate the understanding of differences between the required pronunciation and the users pronunciation as well as necessary changes in the way the user produces the sound.

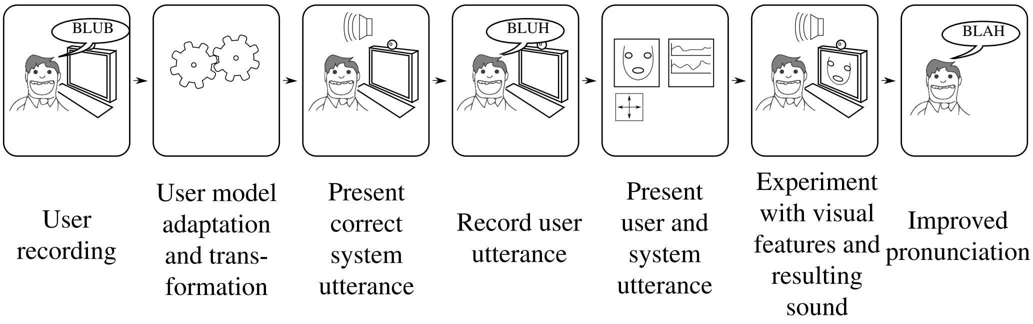
BLUB
BLUH
BLAH
User recording
User model adaptation and transformation
Present correct system utterance
Record user utterance
Present user and system utterance
Experiment with visual features and resulting sound
Improved pronunciation

Figure 1.1: *Illustration of an envisioned system employing the visual control mechanism.*

1.5 Structure of the thesis

In Chapter 2 the first steps involved in TTS and some very basic concepts are explained, followed by an explanation of the concepts required for HMM based speech synthesis. Then some details regarding the model estimation and the parameter generation with and without control are given. This is followed by Chapter 3 recapitulating the performed experiments and the conclusions drawn from these experiments, eventually leading to a system that allows for visual control in a restricted setting. In Chapter 4 the subjective evaluation experiments, performed to evaluate whether the applied control is meaningful, are explained and the findings from the experiments are summarised. This is followed by Chapter 5 concluding the thesis and outlining possible future steps. A list of frequently used acronyms is given after the Appendix.

CHAPTER 2

Method

2.1 Overview of data-based parametric speech synthesis

2.1.1 Text to speech & steps

TTS is the process of converting written language to speech. Since the mapping from written symbols to produced sounds is non-trivial the process is broken down into several steps. The first step is to analyse the input text, to detect structural elements, for example sentences, replace symbols and other constructs that merely abbreviate other words or phrases or represent other phrases or words by their pronounced counterparts (*text normalisation*), as for example needs to be done for numbers - replacing for example *12* by *twelve*. The next step is to replace words by groups of corresponding phonetic symbols which more closely relate to the produced sounds. This step may also involve resolving of ambiguities because the same written symbols or words may stand for different entities that are pronounced differently (*homographs*). Since phonetic symbols on their own do not carry enough information to produce natural speech they are augmented with additional linguistic features, such as stress indicators, word- and syllable boundaries.

From this phonetic/linguistic representation sound is generated. This can be done by employing rules derived from expert knowledge and synthesizers modelled after the human vocal tract as in rule-based synthesis or by exploiting the information of audio segments from recorded and labelled speech as in *data driven speech synthesis*.

The benefit of data driven synthesis over synthesis systems based on expert knowledge is the smaller effort required to build such a system as well as the ease of adapting it to the voice or style of a certain speaker. Data driven synthesis has essentially become possible due to the increase in available computing power. Depending on the synthesis technique used, these speech databases (called speech corpora) are of considerable size, amounting to many hours of high quality recordings of a speaker and aligned phonetic transcription.

In concatenative speech synthesis, this labelled and recorded audio data is cut and the segments conjoined using signal processing techniques. Similar, in parametric speech synthesis, synthesizer parameters are extracted from labelled recorded speech data and these extracted parameters are recombined and used to form new utterances for resynthesis. The HMM based speech synthesis used in this work is a parametric speech synthesis method.

2.1.2 Phonetic symbols

Certain speech sounds, certain segments of the audio signal, can be assigned to the same phonetic symbol. Every time a speaker utters a certain word, the acoustic realisations will vary slightly. Still, the same word will be perceived. Even though the acoustic realisations are different, they can be attributed to the same phonetic symbol sequence. In this work these symbols will interchangeably be called *phones* or *phonetic symbols*.

Some speech sounds may be different but not convey any difference in meaning, this leads to the definition of the *phoneme*: *When two sounds can be used to differentiate words they are said to belong to different phonemes. There must be a phonemic difference if two words (such as "white" and "right" or "cat" and "bat") differ in only a single sound.* [23]

Related to this definition is the definition of the *allophone*, for example the German words "ach"[aχ] and "ich"[ɪç] contain different sounds corresponding to "ch" ([χ] vs. [ç]), still the difference does not convey any meaning and both sounds thus correspond to the same phoneme (/ç/). These different groups (for example [ç] and [χ]) of realisations constitute the phoneme and are called *allophones*. Notice that in transcription /.../ is usually used for phonemic transcription and [...] for allophonic transcription.

It is beneficial if the transcriptions used for speech synthesis follows the actual produced audio more closely, i.e. an allophonic transcription (also called a narrow transcription). However if some allophones only occur rarely, which poses a problem for the generalisation to new utterances, they can be merged with other allophones and treated as the same phone in the transcription. If they occur systematically they may still be implicitly modelled in a distinct way in the synthesis system by the context dependent nature of the speech models. In the context of speech synthesis and this work, diphthongs are treated as single phones.

Table 2.1 lists the phonetic symbols used in the training corpus. The corresponding International Phonetic Alphabet (IPA) symbols and the ASCII strings used to represent the symbol in the synthesis system are provided.

2.1.3 Corpus

In general the training corpora consist of recorded utterances of one or more speakers as well as a phonetic transcription in the form of phone labels. In this work a corpus of a single speaker is used. An additional textual transcription may also be available, and may be used in conjunction with text-analysis and linguistic-analysis to automatically generate the phonetic transcription. To reduce the necessary size of a corpus, the sentences in the corpus are selected to provide a good coverage of naturally occurring phones as well as phone combinations, and contexts of the respective language. The transcription usually attempts to capture the produced sounds rather than the corresponding phonemes in order to achieve higher accuracy. If some phones occur only rarely in the corpus it is possible to merge, i.e. relabel them as similar but more frequently occurring phones - thus going into the direction of a phonemic transcription. This step can be performed semi-automatically.

2.1.4 Austrian German

A single Austrian-German speaker was recorded for the corpus used in this work. The corpus recording is described by Schabus et al. [24]. Differences from a speech-synthesis point of view between German and Austrian-German have been described by Kranzler et al. [25]. The corpus is based on the well-known *Kiel Corpus of Read Speech* [26] and includes 100 "Berlin", 100 "Marburg", 16 "Buttergeschichte"and 7 "Nordwind und Sonne" sentences.

System	IPA	Examples
		vowels - monophthongs
`E`	ɛ	Pech [ˈpɛç] , Test [ˈtɛst]
`I`	ɪ	Witz [ˈvɪts]
`O`	ɔ	Holz [ˈhɔlts], Topf [ˈtɔpf]
`P6`	ɒ	Ader [ˈʔaːdɒ] , Ufer [ˈʔuːfɒ]
`P9`	œ	Köln [ˈkœln] , öffne [ˈʔœfnə]
`U`	ʊ	Luft [ˈlʊft] , Pult [ˈpʊlt]
`Y`	ʏ	Tüv [ˈtʏf] , Tüll [ˈtʏl]
`a`	a	wann [ˈvan] , Wand [ˈvant]
`ah`	aː	Ader [ˈʔaːdɒ] , Adel [ˈʔaːdəl]
`eh`	eː	ewig [ˈʔeːvɪç] , Rehe [ˈreːə]
`ih`	iː	Sieb [ˈziːp] , Team [ˈtiːm]
`oh`	oː	hole [ˈhoːlə] , Sole [ˈzoːlə]
`schwa`	ə	gehe [ˈɡeːə] , Hase [ˈhaːzə]
`uh`	uː	Hupe [ˈhuːpə] , Ruhe [ˈruːə]
`yh`	yː	Asyl [ʔaˈzyːl] , Type [ˈtyːpə]

System	IPA	Examples
		vowels - diphthongs
`E6`	ɛɒ	Herz [ˈhɛɒts] , Nerv [ˈnɛɒf]
`Eh6`	ɛːɒ	Herd [ˈhɛːɒt] , Teer [ˈtɛːɒ]
`O6`	ɔɒ	Dorf [ˈdɔɒf] , vorn [ˈfɔɒn]
`OY`	ɔʏ	treu [ˈtrɔʏ] , Zeug [ˈtsɔʏk]
`P2h`	øː	Fön [ˈføːn] , Öle [ˈʔøːlə]
`P2h6`	øːɒ	Öhr [ˈʔøːɒ] , Göre [ˈɡøːɒrə]
`U6`	ʊɒ	Burg [ˈbʊɒk] , Turm [ˈtʊɒm]
`a6`	aɒ	Arme [ˈʔaɒmə] , warb [ˈvaɒp]
`aI`	aɪ	Leim [ˈlaɪm] , Brei [ˈbraɪ]
`aU`	aʊ	blau [ˈblaʊ] , Maus [ˈmaʊs]
`ah6`	aːɒ	Jahr [ˈjaːɒ] , zart [ˈtsaːɒt]
`ih6`	iːɒ	Tier [ˈtiːɒ] , Iris [ˈʔiːɒrɪs]
`oh6`	oːɒ	Lore [ˈloːɒrə] , Pore [ˈpoːɒrə]
`uh6`	uːɒ	Jura [ˈjuːɒraː] , pure [ˈpuːɒrə]
`yh6`	yːɒ	Tür [ˈtyːɒ] , Lyra [ˈlyːɒraː]

System	IPA	Examples
		consonants
`C`	ç	Elch [ˈʔɛlç] , Pech [ˈpɛç]
`GS`	ʔ	
`N`	ŋ	Funk [ˈfʊŋk] , lang [ˈlaŋ] ,
`S`	ʃ	spie [ˈʃpiː] , Stau [ˈʃtaʊ]
`b`	b	Eibe [ˈʔaɪbə] , Tube [ˈtuːbə]
`ch`	χ	Bach [ˈbaχ] , Loch [ˈlɔχ]
`d`	d	Ader [ˈʔaːdɒ] , doof [ˈdoːf]
`f`	f	doof [ˈdoːf] , Saft [ˈzaft]
`g`	ɡ	Yoga [ˈjoːɡaː] , vage [ˈvaːɡə]
`h`	h	Herz [ˈhɛɒts] , Holz [ˈhɔlts]
`j`	j	Koje [ˈkɔjə] , Soja [ˈzoːjaː]
`k`	k	Kohl [ˈkoːl] , sank [ˈzaŋk]
`l`	l	List [ˈlɪst] , Zahl [ˈtsaːl]
`m`	m	Norm [ˈnɔɒm] , Leim [ˈlaɪm]
`n`	n	Wand [ˈvant] , Nord [ˈnɔɒt]
`p`	p	Pech [ˈpɛç] , Pass [ˈpas]
`r`	r	grub [ˈɡruːp] , ragt [ˈraːkt]
`s`	s	Witz [ˈvɪts] , Asse [ˈʔasə]
`t`	t	Tuba [ˈtuːbaː] , West [ˈvɛst]
`v`	v	Wand [ˈvant] , Witz [ˈvɪts]
`z`	z	Sieg [ˈziːk] , Saum [ˈzaʊm]

Table 2.1: *Phone symbols present in the single-speaker Austrian-German corpus.*

2.1.5 Vocoding in parametric speech synthesis

In parametric speech synthesis, voice coding techniques are used to extract slowly varying speech parameters. These parameters can be used to perform synthesis of the speech signal. Instead of storing samples of the recordings in a database, the parameters are modelled by a statistical technique i.e. HMM [9]. Reconstruction from HMM is easier or more accurate if the parameters are smooth and slowly varying, since the HMM parameter generation (see Section 2.6.1) tends to generate very smooth trajectories.

2.1.6 Source-filter model

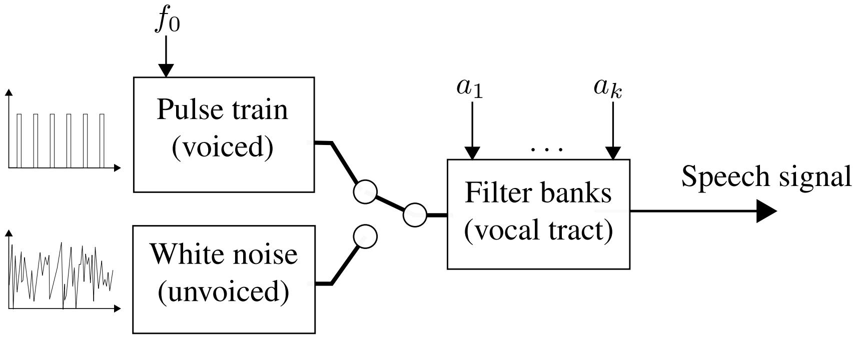

Figure 2.1: *Simple illustration of the source filter model of speech production; f_0 is the fundamental frequency of the signal generated by the vocal folds, $a_1 \dots a_k$ are the filter coefficients of the filter bank modelling the vocal tract.*

Fundamental to the voice coding is the source-filter model (illustrated in Figure 2.1) which is a simplified model of the human speech production apparatus. It models the glottis or the vocal cords as a signal source and the vocal tract as a filter bank modifying the glottis-signal. The signal produced by the vibrating vocal cords for voiced speech parts is modelled by a pulse-train and unvoiced speech parts are modelled by noise. The fundamental frequency or period of the underlying excitation signal pulse train determines the pitch and this frequency is called $f0$. The speech signal production is thus divided into a fast varying excitation signal, used to simulate the glottal pulses or the noise like sound of unvoiced parts, and slowly varying filter parameters corresponding to the shape of the vocal tract. Instead of modelling the signal values of the excitation signal at each time point, the fundamental frequency is modelled which varies much slower than the signal itself. Thus all parameters of the speech encoding are varying slowly compared to the audio signal trajectory.

2.1.7 Spectral speech representation

The continuous analog time signal is discretized with regard to time and values. This discrete signal is then stored in a loss-less way for example using the WAV file format. In this work all audio signals are assumed to be already discretized.

To extract the speech parameters the speech signal is cut into overlapping time frames. The length of these frames is called *window size* and the amount of overlapping is determined by the amount that each succeeding window is shifted forward in time with regard to the previous window. This amount is called the *frame-shift*. In this work a window size of 40ms and a frame-shift of 5ms (resulting in an overlap of 35ms) is used.

From these frames a short time spectral representation is calculated using a discrete Fourier transform. Simply cutting the signal amounts to convolution with a rectangular pulse signal which leads to smearing of the spectral components, also called *spectral leaking*. To improve the spectral leakage the signal is reshaped using a window function. See the speech signal processing literature for a more detailed description. An introduction is also provided by Pfister and Kaufmann [27].

The result from these steps are spectral representations of succeeding time points, i.e. over a sequence of overlapping time segments. This spectral representation over time is called a *spectrogram*.

2.1.8 LPC

Linear Predictive Coding (LPC) is a voice coding method based on the source filter model. A special representation of the LPC coefficients called Line Spectral Pairs (LSP) or Line Spectral Frequencies (LSF) are used as spectral features in this work. The following reproduces formulations for LSF, described by Pfister and Kaufmann [27] for the sake of completeness.

In LPC the n-th signal value $s(n)$ is predicted by a weighted sum of the previous K values $s(n-1), \ldots, s(n-K)$, the predicted signal value $\tilde{s}(n)$ is:

$$\tilde{s}(n) = -\sum_{k=1}^{K} a_k s(n-k) \tag{2.1}$$

The spectral features used in this work are based on LPC.

The error between the real signal and the predicted signal is:

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{K} a_k s(n-k) \tag{2.2}$$

The power of the error signal is given by:

$$E = \sum_n e(n)^2 = \sum_n \big[s(n) + \sum_{k=1}^{K} a_k s(n-k)\big]^2 \tag{2.3}$$

Minimising 2.3 with respect to a_i yields K equations which can be rearranged [27]:

$$\sum_{k=1}^{K} a_k \sum_n s(n-k)s(n-i) = -\sum_n s(n)s(n-i) \quad 1 \leq i \leq K \tag{2.4}$$

If we assume that the signal is multiplied with the window function the signal for a frame n, $\overline{s}_n$ is given by:

$$\overline{s}_n(m) = s(n+m)w(m), \quad 0 \leq n \leq N-1 \tag{2.5}$$

We thus also assume that the signal is 0 outside of the window. The short-time auto-correlation terms $r(i)$ are defined by:

$$r(i) = \sum_{m=0}^{N-1} \overline{s}_n(m)\overline{s}_n(m+i) \tag{2.6}$$

Using this definition of $r(i)$ and since the auto-correlation coefficients are symmetric $r(i) = r(-i)$, 2.4 can be rewritten as:

$$\sum_{k=1}^{K} r(i-k)a_k = -r(i), \quad 1 \leq i \leq K \tag{2.7}$$

By the *Wiener-Khinchin-Theorem* the auto-correlation coefficients $r'(i)$ and the power spectrum of the signal x are a Fourier pair:

$$r'(i) = \mathcal{F}^{-1}\{|\mathcal{F}\{x(j)\}|^2\} \tag{2.8}$$

because of this we can approximate the auto correlation coefficients through the short time spectrum.

By applying the z-transformation on equation 2.2 the relationship between the error signal $E(z)$ and the speech signal $S(z)$ is:

$$E(z) = S(z) + \sum_{k=1}^{K} a_k z^{-k} S(z) = \sum_{k=0}^{k} a_k z^{-k} S(z) = A(z)S(z) \tag{2.9}$$

$$S(z) = H(z)E(z) = \frac{1}{A(z)} E(z) \tag{2.10}$$

Since 2.10 reconstructs the speech signal $S(z)$ from the error signal $E(z)$, the corresponding filter $H(z)$ is called the resynthesis filter. This is the filter bank of the source-filter model. Instead of storing the error signal $E(z)$, an approximation is used, i.e. the pulse train or white noise generator of the source-filter model is used.

The LPC are difficult to interpolate and are susceptible to noise. A different representation of the LPC is based on representing the corresponding polynomial $A(z)$ by the sum of two polynomials, the palindromic polynomial $P(z)$ and the anti-palindromic polynomial $Q(z)$.

$$H(z) = \frac{1}{A(z)} = \frac{2}{P(z) + Q(z)} \tag{2.11}$$

The zeros of the polynomials $P(z)$ and $Q(z)$ are interleaved. The LSP (also called LSF) are the frequencies $0 < \omega_1 < \ldots < \omega_n < \pi$ of the zeroes of $P(z)$ and $Q(z)$. This representation is more robust with regard to noise and allows for interpolation.

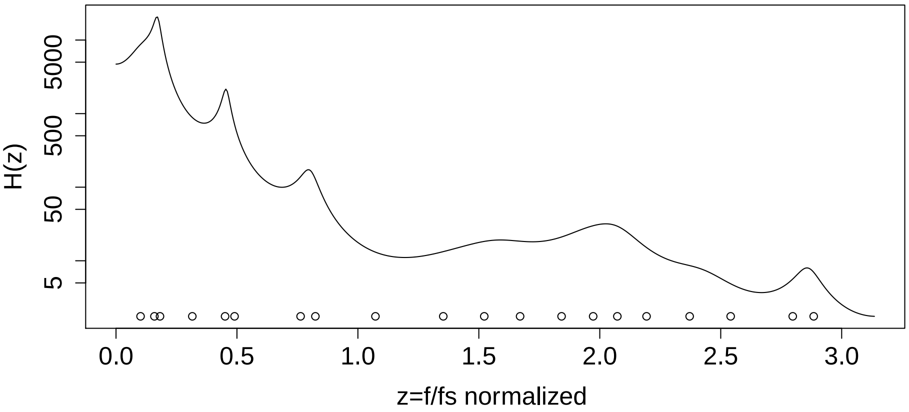

Figure 2.2: *Illustration of the spectrum as derived by the linear predictive coding and the corresponding line spectral frequency representations (circles). Notice how maxima in the gain of the transfer function $H(z)$ arise if LSFs are close to each other.*

The relationship between the polynomials and the frequencies is given by [28]:

$$P(z) = (1 + z^{-1}) \prod_{k=2,4,\dots n} (1 - 2\cos\omega_k z^{-1} + z^{-2}) \tag{2.12}$$

$$Q(z) = (1 - z^{-1}) \prod_{k=1,3,\dots n-1} (1 - 2\cos\omega_k z^{-1} + z^{-2}) \tag{2.13}$$

Efficient methods for finding the roots have also been developed, for example by Rothweiler [29, 30].

If two LSF are close to each other the transfer function $H(z)$ has a large gain. Figure 2.2 illustrates the frequency response for $H(z)$ and the corresponding LSF frequencies.

Instead of simply calculating the short time spectrum, the more sophisticated Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum (STRAIGHT) method is used [31, 32]. In STRAIGHT a pitch-adaptive window is used to reduce the effects of the excitation signal of the glottis. Additionally the spectrum is smoothed to retrieve a less periodic or less noisy spectral envelope. Calculating the LPC and respectively the LSP from the STRAIGHT envelope leads to a more stable and less noisy representation. These are the features also used by Ling et al. [12, 13]. An example of using STRAIGHT for LSPs is also given by Nguyen et al. [33] in the context of low bit-rate speech coding.

2.1.9 Rough sketch of synthesis

To perform synthesis, textual and linguistic analysis are employed to create a sequence of phonetic symbols with additional linguistic information representing the utterance. Each of the symbols in the sequence is used to determine a five state HMM corresponding to the symbol. The HMMs model the acoustic representation of the respective phonetic/linguistic symbols and are then concatenated to form a single large HMM to represent the whole utterance. This large HMM is then used to generate parameter trajectories for the vocoder to recreate the audio speech signal.

2.1.10 Audio/visual speech

In audio-visual speech synthesis an additional visual channel is synthesised. This means not only an audio signal is generated but also a representation or animation of a speaker's face. The techniques vary from three dimensional models to techniques based on two dimensional image manipulation and from human-like to cartoon like appearances of the heads. Three dimensional animation techniques to modify the appearance of the heads include marker based modification of the structure as well as key frame based interpolation techniques. An overview of different approaches to visual speech synthesis is given by Bailly et al. [34].

In this work three dimensional marker trajectory recordings are used. Visual marker trajectories are synthesised but no animation or modification of a virtual head is performed. A point cloud visualisation is used to ease the understanding of the effect of changes to the marker trajectories. A speaker with markers and the corresponding point cloud visualisation is illustrated in Figure 2.3.

2.1.11 Marker recording

The marker position trajectories are recorded synchronously with audio data from a speaker as described by Schabus et al. [24]. Principal Component Analysis (PCA) (an introduction is given by Bishop [36] or Shlens [37]) is performed on the data of X,Y,Z positions of the different facial markers in order to reduce the dimensionality. The recording system was an OptiTrack system [35] that uses several synchronised infrared cameras with infrared lighting and reflective markers glued to the speaker's face. After fixing the positions of the cameras, calibration can be done using

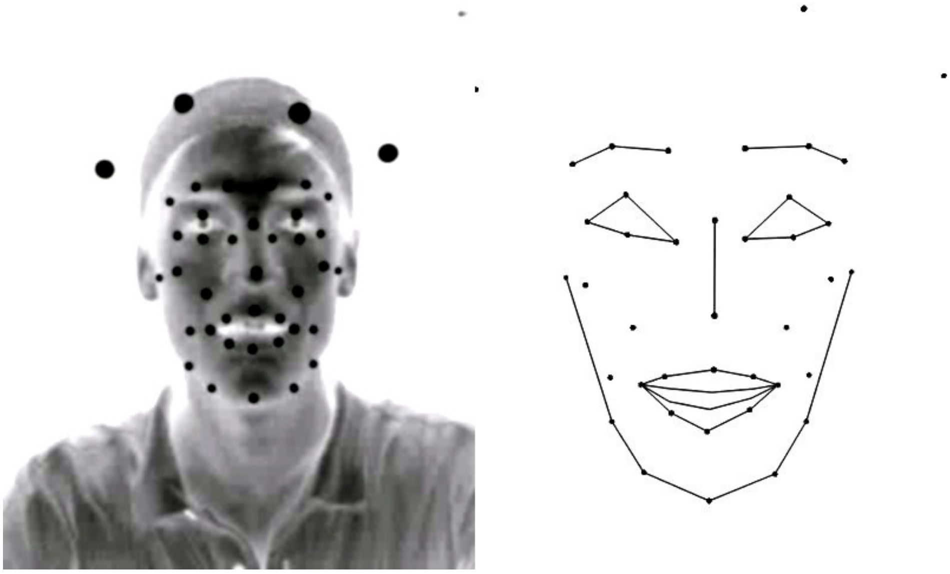

Figure 2.3: *(left) A frame from a recording session of a speaker is shown. The reflection of the markers glued to the face is clearly visible. The point cloud visualisation of the marker positions is shown on the right.*

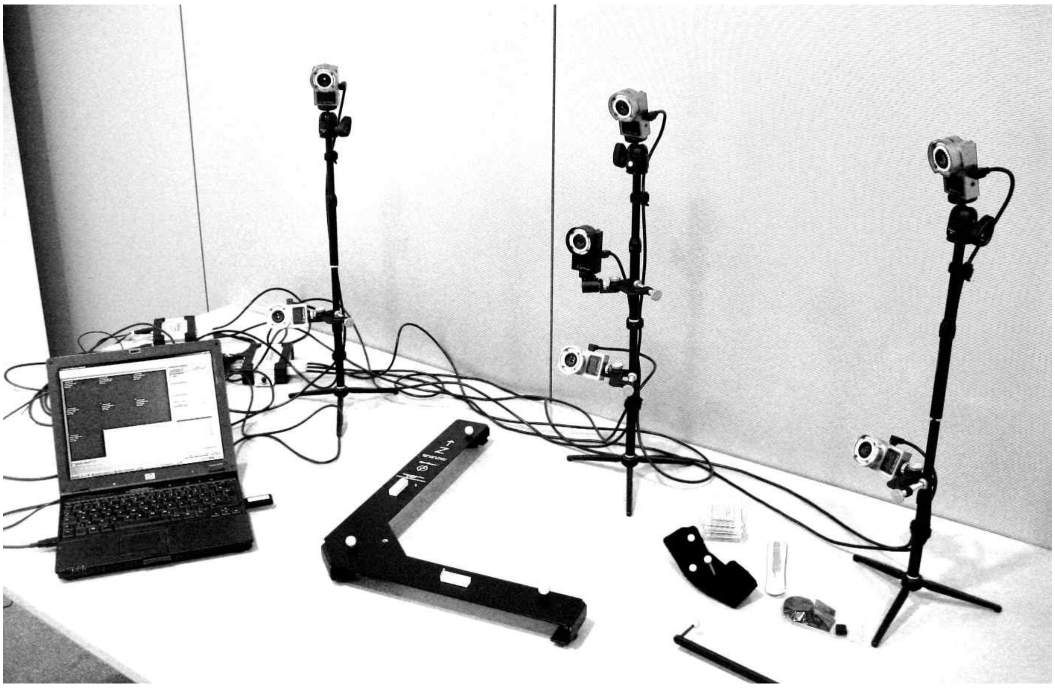

Figure 2.4: *Setup of the optical marker tracking system [35] used in the audio-visual recordings. The L-shaped calibration device is visible in the lower middle of the image.*

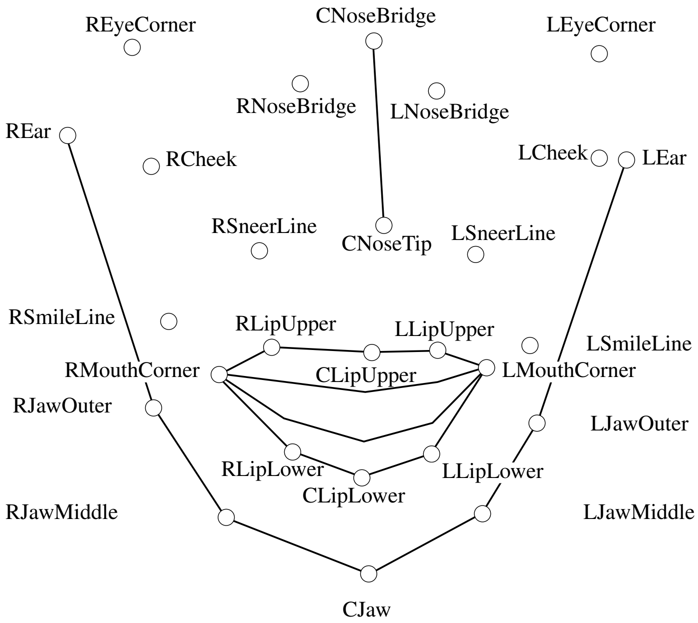

Figure 2.5: *Facial markers included in the synthesis system - eyebrow and eyelid markers removed - and their associated names*

a marker equipped device of a priori known size and orientation. Then triangulation is used to determine the positions of the marker points from the different views of the cameras. The setup of the cameras is illustrated in Figure 2.4.

Not all markers are used in our visual synthesis system since for example eye blink is not directly connected to speech production. Figure 2.5 shows the markers that are typically considered for synthesis in our system as well as their associated names. In this work no facial motion is synthesised. Only marker position trajectories are synthesised. To generate facial animations from this data retargeting is necessary: *" The problem of how to animate a talking head automatically from a sequence of parameters is called retargeting"* [38]. The retargeting problem is not dealt with in this work.

2.1.12 Scope

As described in Figure 2.6, the first steps in TTS are textual and linguistic analysis and are provided by the Festival Speech Synthesis system [39]. Conversion from the linguistic feature stream provided by the above mentioned system to voice coding parameter trajectories is done by means of HMMs. The framework used for modelling the acoustic parameters with HMMs as well as training the model and performing synthesis to generate parameter streams is the Hidden Markov Speech Synthesis System (HTS) [40]. The vocoder used to generate an audio signal from the parameter stream is derived from STRAIGHT [32] as described by Lei et al. [6] (the acoustic features are also described in Section 2.1.8).

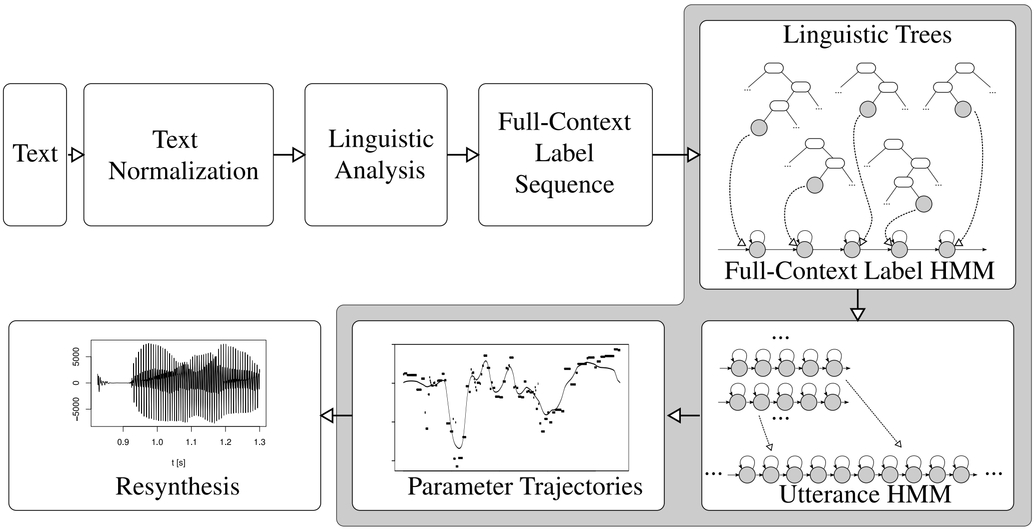

Figure 2.6: *Illustration of the steps involved in a HMM based speech synthesis system. This work focuses on the steps marked in grey.*

2.2 HMM concepts

2.2.1 What is an HMM?

An HMM is a statistical tool to model time series. In the case of speech synthesis this are the time series of acoustic parameters.

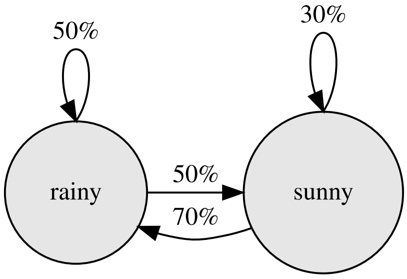

Figure 2.7: *Illustration of a Markov chain with two states, rainy and sunny, as well as their transition probabilities.*

A *Markov model* or *Markov chain* is somewhat similar to a state machine. It consists of states and transition probabilities. Given a state, the state at the next time step depends only on the current state and the transition probabilities. This is called the *Markov property*. Higher order Markov models take into account not only the current state but also - depending on the order - the n-previous states. For speech synthesis higher orders are not used. An example of a Markov chain with two states *rainy, sunny* is given in Figure 2.7, along with the transition probabilities. Given a Markov chain, possible state sequences can be enumerated and the transition probabilities can be used to assign likelihoods to these sequences, thus the most likely state sequence can be determined.

In a *hidden Markov model* the current state is not observable - hence *hidden*. However certain

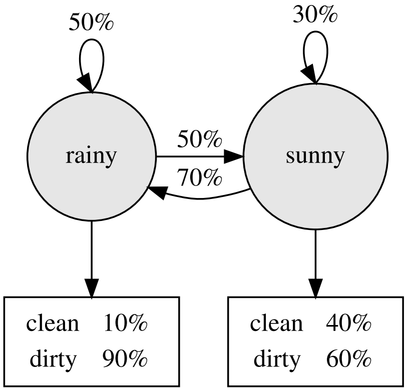

Figure 2.8: *A discrete hidden Markov model with two states, rainy and sunny, and two different observations, clean and dirty.*

properties of the world can be observed depending on the current state of the Markov model. These are called *observations*. Observations may be discrete values (for example dirty or clean shoes) or continuous values (for example different temperatures). The probability or likelihood for a certain event depends only on the state.

An example often used for illustration purposes is the case of a prisoner in a cell without a window. This prisoner is interested in the weather outside (these are the states of the model, sunny and rainy $\{r, s\}$), however the only clue is whether the shoes of the guard are dirty or clean $\{c, d\}$. This model is illustrated in Figure 2.8.

Given the prisoner knows the transition probabilities of the weather, changing from rainy to sunny $p(r, s)$ or from sunny to rainy $p(s, r)$ or staying the same $p(r, r), p(s, s)$, as well as the *output probabilities* that the guards shoes will be clean $p(c|x)\ x \in \{r, s\}$ respectively dirty $p(d|x)\ x \in \{r, s\}$ depending on the weather, the prisoner is able to calculate for all possible sequences of weather conditions the likelihood with regard to the sequence of observations of the guards shoes and is thus able to select the sequence that is most likely.

2.2.2 HMM formally

An HMM consists of *a*) N distinct states indexed by $\{1, 2, \ldots, N\}$, *b*) state transition probabilities a_{ij} from state i to state j, *c*) either a discrete set of possible observations $\mathbf{O}$, or as in our case a continuous space $\mathbf{O} \subseteq \mathbb{R}^D$ where D specifies the dimensionality of the observations and thus depends on the application, *d*) a measure of likelihood $b_j(\mathbf{o})$ that a certain observation $\mathbf{o}$ was produced by a state j.

The following properties hold for state transition probabilities a_{ij}: the transition probabilities are non-negative:

$$a_{ij} \geq 0 \quad \forall j, i$$

for each state i the transition probabilities of leaving that state sum to one:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i \neq N$$

There are no transitions leaving the end state N:

$$\forall j \;\; a_{Nj} = 0$$

For modelling the continuous parameters *Gaussian mixtures* are used.

2.2.3 Gaussian mixtures

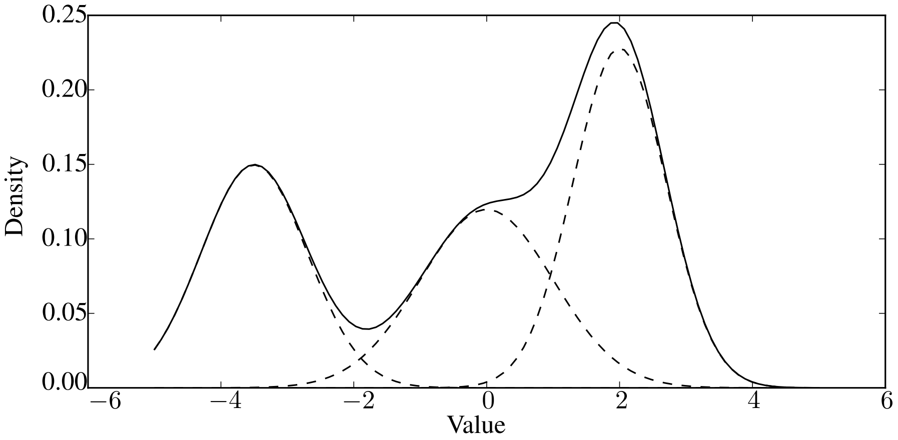

Figure 2.9: *Resulting probability density function of a three component Gaussian mixture. Gaussian mixtures can approximate arbitrarily shaped distributions.*

Gaussian Mixtures combine M different *Gaussian-* or *Normal-Distributions* $\mathcal{N}_{\mu_i,\sigma_i}$, $1 \leq i \leq M$. These are called components and have an associated weight c_k such that

$$\sum_{k=1}^{M} c_k = 1$$

By combination of the Gaussians, probability density functions of arbitrary shape can be approximated. Figure 2.9 shows an example of the corresponding combined probability density function for three Gaussians. For the combined probability density function it holds that:

$$\sum_{k=1}^{M} \int_{-\infty}^{\infty} c_k \mathcal{N}(x, \mu_k, \sigma_k) dx = 1$$

It can be shown that an HMM state with a mixture density is equivalent to a multistate single-mixture density model by splitting the state into as many states as there are mixture components and setting the transition probabilities according to the mixture weights [41].

2.2.4 HMM formally continued

Given Gaussian mixtures as defined above, the likelihood for observation $\mathbf{o}$ in state j is:

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}, \boldsymbol{\mu}_{jk}, \boldsymbol{\sigma}_{jk}), 1 \leq j \leq N \tag{2.14}$$

In the continuous case, i.e. with Gaussian mixture models, the probability for a certain observation sequence can not be assessed, because single events of a continuous probability variable do not have an assigned probability. However two events can be compared with respect to their likelihood. Again all possible state sequences can be enumerated and their likelihoods with regard to a given observation sequence can be used to select the most likely state sequence.

2.3 HMMs in speech processing

2.3.1 Relation to speech recognition

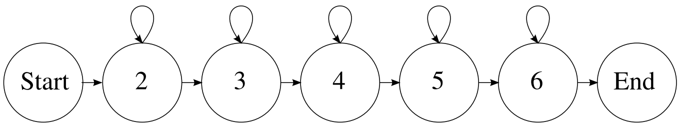

Figure 2.10: *States and possible transitions in a five emitting-state left-to-right HMM.*

HMM based speech synthesis is conceptually derived from HMM based speech recognition. In the task of speech recognition a sequence of phones and finally words should be derived from an acoustic observation sequence. Here HMM states correspond to models of segments of phonetic symbols and observations correspond to acoustic features.

In the case of speech synthesis a sequences of phones is given and the acoustic observation sequence should be derived. In both cases the observations are acoustic parameters and the HMM states are related to phone symbols. Thus for speech synthesis, given a model derived from the to-be-synthesised phone string, the optimal state sequence (i.e. durations per phone), and most likely observation vectors have to be found.

To estimate the model parameters a speaker recording and a time-aligned phonetic transcription is used. Each phonetic symbol is modelled by a five-state left-to-right HMM. These models are concatenated to represent the utterance. Each HMM state can be interpreted as modelling a certain time segment of a phonetic symbol. The observation distribution associated with each state models the associated acoustic features. The training process consists of modifying the HMM parameters such that the likelihood for the corresponding acoustic observations is maximised.

2.3.2 HMM left-to-right model

A simple left-to-right topology (illustrated in Figure 2.10) is used for the HMMs in speech synthesis. This simplifies finding the optimal state sequence. Also a more complicated HMM topology can not capture context depending variations and therefore has no additional benefit. Systematic variations of pronunciation are modelled on a higher level (for example using *fullcontext-models* and the *linguistic trees* used for clustering), not strictly within the HMMs.

A left-to-right model allows only transitions to the current state or the right-hand-side state of the current state. An implication of this is that the transition probabilities form a band matrix.

$$
A = \begin{pmatrix}
0 & 1 & 0 & \dots & \dots & \dots & 0 \\
0 & a_{22} & a_{23} & \ddots & & & \vdots \\
\vdots & \ddots & a_{33} & a_{34} & \ddots & & \vdots \\
\vdots & & \ddots & a_{44} & a_{45} & \ddots & \vdots \\
\vdots & & & \ddots & a_{55} & a_{56} & 0 \\
0 & \dots & \dots & \dots & 0 & \dots & 0
\end{pmatrix}
$$

2.3.3 Monophone HMM

The simplest way to model phones is to use *monophone-HMMs*, where a single phone is represented by a single HMM. In the case of five-state models, five states represent adjacent parts of the audio-signal-trajectories. One five-state HMM models a variety of different instances of the same phone, and the model corresponds to an average of the different instances. Depending on the corresponding trajectories most of the signal may for example be modelled by a single one of the five states. This state then has a long expected duration, followed by other states with very short expected durations. More evenly temporal distribution of the states is also possible depending on the complexity and shape of the trajectories. This is somewhat similar to concatenative synthesis as these monophone models can be concatenated to represent phonetic symbol strings and thus utterances.

2.3.4 Three basic problems of HMM based speech processing

For HMM based speech recognition three general problems regarding HMMs have been identified by Rabiner [42]. These problems are important for model estimation and thus also important for speech synthesis:

1. *Given the observation sequence $O = O_1 O_2 \dots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?*
   word recognition – forward algorithm

   The forward algorithm is used to identify the probability or likelihood that a given observation sequence corresponds to a certain model. To calculate the likelihood, it is necessary to calculate the likelihoods of all possible state sequences, as well as the likelihood of each of these state sequences producing the observation sequence. This also gives us a likelihood that a certain state is responsible for a certain observation relative to the responsibility or likelihood of the other states.

2. *Find the optimal state sequence given the model. Given the observation sequence $O = O_1 O_2 \dots O_T$, and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 \dots q_t$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?*
   decoding, recognition – Viterbi algorithm

   Using the Viterbi procedure an observation-vector–state mapping can be obtained. This is also called a *forced alignment* as a predetermined phone sequence is aligned to the given training data. Forced-aligned observations can be used as training data and are used in the training procedure of this work to initialise the HMMs.

3. *How to adjust the model parameters $\lambda = (A, B, \pi)$ to maximise $P(O|\lambda)$?*
   maximum likelihood training – Baum-Welch algorithm

   The training problem consists of adjusting the model parameters λ, i.e. the transition probabilities a_{ij} as well as the observation probabilities or likelihoods $b_j(\mathbf{o})$. The observations $\mathbf{O}$ are the recorded data, and the model is constructed according to the transcription of the training data. In the Baum-Welch algorithm the likelihood of a state generating a certain observation is used to do a weighted assignment of the training vectors to the HMM states which is then used to do an estimation of the parameters of the corresponding Gaussians.

2.3.5 Coarticulation effects and fullcontext

The movements of the human articulatory organs are constrained with regard to the time it takes to change from one position to another. This physical constraint leads to an effect called coarticulation [23]. Because of the coarticulation, phones are not produced independent of their context. The preceding and succeeding phone influence the articulation. A monophone model represents several very different instances of the same phone and thus can not adequately reflect the context. Thus tri- or quin-phone models are used, taking into account the (two) preceding and (two) succeeding phones. Consider for example the word “Geschichte” /gəʃɪçtə/ (narrative,story), the monophone /ʃ/ (S) has the corresponding triphone symbol `schwa-S+I`, /ʃ/ preceded by `schwa` /ə/ and followed by `I` /ɪ/. The corresponding quinphone would be `g^schwa-S+I=C`, i.e./ʃ/ is preceded by /gə/ and succeeded by /ɪç/. Since the sound of a certain phone in a certain context may further depend on other linguistic as well as prosodic features, additional information is added to the phone. This includes but is not limited to, *stress and accent of preceding, current and succeeding syllables, numbers of phones within preceding, current, and succeeding syllables, position of current word within current phrase, numbers of syllables within preceding, current, and succeeding phrases* [9]. It is encoded as identifiers attached to the quinphone string. This aggregate is called a *fullcontext-label*. The format illustrated below has been described by Zen [43]. The quinphone described above has the corresponding fullcontext label:

```
g^schwa-S+I=C@1_3/A:0_0_2/B:0-0-3@4-2&5-2#2-1$2-2!3-0;3-1|0/
C:0+1+2/D:content_1/E:content+5@2+1&2+0#1+0/F:0_0/G:0_0/H:6=2@1=1|
L-L%/I:0=0/J:6+2-1
```

2.4 Synthesis

2.4.1 Clustering & trees

If we assume that there are 20 phones – actually there are more – and use quinphones this already results in $20^5 = 3.2 \cdot 10^6$ different possible quinphones. Since linguistic features are also taken into account the number of possible combinations is orders of magnitudes larger. Even though many of the combinations of phones and linguistic features are not meaningful, drastically reducing the total number of combinations, we will not be able to cover the full space of fullcontext labels in the recorded training corpus.

This leaves us with two problems *a*) how to synthesise fullcontext-labels that are not covered by the training corpus *b*) labels may only occur very rarely and thus the corresponding models have to be estimated from very little training data (and thus may not be statistically robust). .

To make the models statistically more robust, different fullcontext labels are merged together, thus forming clusters of acoustic observations. The clustering can be chosen based on the fullcontext labels. For example, clustering the same center-phones for different contexts if the underlying data is similar enough. We may also merge and split acoustic observations based on linguistic features, like syllables and information regarding stress. *Questions* are used that evaluate to *yes* or *no* for a given fullcontext label. This derives binary features which allow us to build a decision tree, where splitting and merging is done based on these questions. Clustering does not need to be phone- and hence HMM based, usually a tree for each of the five states of the models is built allowing for better and more appropriate clustering as coarticulation effects can be modelled more effectively. In the HMM framework the mechanism used to merge different models, or share model parameters is called *tying* [44].

2.4.2 Synthesising from trees

Decision-trees can be used to combine acoustic parameter distributions into clusters based on the linguistic information of their associated fullcontext labels. If we use decision-trees, which recursively split depending on features in the fullcontext labels, we can also traverse this tree for the features of new fullcontext labels that do not occur in the corpus, thus generating information on how new and unseen labels should be modelled.

One of the advantages of using decision tree clustering is that it allows previously unseen [tri-] phones to be synthesised. To do this, the trees must be saved ... Later if new previously unseen [tri]phones are required, for example in the pronunciation of a new vocabulary item, the existing model set... and the trees can be reloaded, and *...* *a new extended list of [tri]phones can be created...* [45].

This effectively paves the way to perform synthesis of novel utterances that would not have been possible with only the fullcontext-label based models in the training corpus. To perform synthesis from a sequence of fullcontext labels, a large conjoined HMM is built state-by-state, using the decision trees for unseen fullcontext labels. Then as explained above the most likely state sequence is determined. In a left-to-right model without the option to skip states this means determining the number of times each state is repeated. This state sequence yields a sequence of probability density functions for the observations. This sequence of probability density functions is then used to generate the most likely observation sequence which is then resynthesised into the audio speech signal.

2.4.3 Delta and delta-delta features

The Markov model does not explicitly model how parameters change over time but only models the likelihood of a certain observation given a certain state. This is not so much a problem for recognition than for generation from a Markov model. One way to generate an observation sequence from a sequence of HMM states and thus from a sequence of probability density functions, would be to generate the single most likely observation, which corresponds to the mean of each respective probability density function. However each state change will most likely be associated with a different probability density function and thus different mean values. Generating the parameter sequence based on the sequence of means would yield abruptly changing trajectories. The parameters could be interpolated between the different mean values but the model does not specify how such an interpolation should be done. To this end the feature stream is augmented with additional features. Typically delta (Δ) and delta-delta ($\Delta\Delta$) features, corresponding to discrete approximations of the first and second derivative are used. These *augmentation features* are also modelled by the probability density function stream. Given the mean, consisting of static, delta, and delta-delta features and the variance stream, the most likely observation sequence can be determined. Thus additional constraints in the form delta and delta-delta features are used to relate the otherwise independent observations over time [5].

2.4.4 Spectral, visual, and F0 features

For audio-visual synthesis based on HMMs, the probability density functions of each state, model acoustic observations and visual parameter observations. These two kinds of parameters may also be clustered independently. In the used HMM framework, these dimensions of the probability density functions that can be clustered separately are called *streams*, i.e. an acoustic stream and a visual stream.

Modelling parts of the features separately allows to recombine them which in turn may allow for smaller models due to the factorisation. Smaller models also require a smaller amount of training data. The source-filter model used for vocoding also models the excitation signal, as an approximation of the vibration of the vocal folds, and the filter coefficients, as an approximation of the shape of the vocal tract, independently. Typically separate streams for the fundamental frequency (F0) and spectral features are used [8]. In this work the visual features are modelled by an additional visual stream.

Since the source of the source-filter model generates the excitation signal for voiced- and unvoiced parts of speech, the voicedness also has to be modelled. Since in this model, there is no additional parameter for unvoiced speech, and voiced parts are modelled simply by the fundamental frequency F0, the fundamental frequency can be modelled as a continuous value in a certain range for voiced parts and as a single discrete value for unvoiced parts. The combination of discrete symbols and continuous values is modelled by a multi-space probability distribution [46].

2.4.5 Control, control modelling and the control functions in the tree

In this work ways of controlling the acoustic features via the visual features are explored. This means that a mapping from visual features to the acoustic features is associated with the HMM and that this mapping can be used to *a*) generate the acoustic features from the visual features or *b*) overlay changes from the visual features on the acoustic features. The mapping used is a linear regression from the visual features to the acoustic features. The functions mapping from the control (visual) space to the target (acoustic) space can be associated with the leaves of the linguistic trees. Several leaves can share the same transformation function. The mapping does not model dependencies between static and dynamic features, instead the relation captures dependencies between static control and static target features, as well as a dependencies between the dynamic (delta) control and the dynamic (delta) target (acoustic) features. Similar, the delta-delta control features influence the delta-delta target features.

Control is applied during synthesis. The visual parameters are changed and the mapping is used to induce changes on the acoustic features and influence the production of the acoustic parameter trajectories.

2.5 Training in detail

2.5.1 What is a model?

To perform speech synthesis using HMMs a way is needed to transform text to a phonetic transcription and a mechanism to map the transcription to the synthesizer parameter. To map the phonetic transcription to synthesizer parameter trajectories decision trees and HMMs are used. Decision trees are used to decide how unseen phone contexts should be synthesised and HMMs are employed to generate the parameter stream over time.

2.5.2 How to get the model from the data and how to start?

To train the model a corpus of recorded speech with a corresponding phonetic transcription is used. In essence audio parameter trajectory segments have to be associated with the corresponding phonetic symbols. The transcription may include a per phone alignment with respect to the audio file but should at least include an alignment on a per-sentence level, for example by cutting the audio file and the transcription into sentences. A very crude alignment can be used for initialisation: the audio signal is cut into equal length segments according to the number of phone symbols in the transcription. The model is iteratively refined in the training process which also includes refining of the alignment. This provides a way to automatically align the transcription with the audio data.

2.5.3 Model initialisation

The training begins with a very simple model. Instead of taking the full context and thus the full-context labels, into account, only the context-less *monophones* are used. First identical monophone models for all phones are created based on a prototype model. The prototype model specifies the possible transitions, i.e. the *left-to-right topology*, the transition probabilities and the number of states. The benefits of using monophone models over fullcontext models is that the monophones will occur very often and in very different contexts leading to less precise but more general models of the phones. Also the fullcontext labels will only occur very rarely in small corpora which could lead to incorrect local optimisation of the model, especially if the models are initialised using similar mean and covariance data. If corpus wide iterative refinement of the phone alignment would be done for fullcontext models that are initialised to identical data, the alignment may be less likely to capture the audio segment that actually corresponds to the phone but may as well capture a part of the trajectory before or after the phone. This is the reason for starting with a very simple model and iteratively refining it.

2.5.4 Viterbi alignment and training

As described above, if no per-phone alignment is available, equal length of the phones is assumed in the transcription. This is called a *flat-start model*. The recording of an utterance is cut into equal length segments according to the number of phones in its transcription. If alignment information is available the utterance recording can be cut into segments accordingly. Then the parameter vectors of the segments are pooled, again using equal lengths for the different *states* of the HMM, to estimate the Gaussian mixture parameters, i.e. the mean and covariance information. This initialised HMM is aligned on the corresponding segments of the training data using the *Viterbi algorithm*. This may lead to non-equal length segments for each of the states. Again the corresponding vectors are pooled to update the mixture parameters of the corresponding HMM states as well as the transition probabilities. In the HTS/Hidden Markov Model Toolkit (HTK) framework used in this work, this initialisation is performed by the command *HInit* [45]. The following sections describe the algorithms (Forward-, Backward-, Viterbi-, and Baum-Welch algorithm) in more detail. The descriptions are adapted from the descriptions given by Rabiner and Juang [41].

2.5.5 Forward algorithm

The forward algorithm calculates state occupancy likelihoods for a given observation sequence $\boldsymbol{O}$ and model parameters λ. λ signifies the transition probabilities and observation likelihoods. It is also used in the Viterbi and Baum-Welch algorithm. $\alpha_i(i)$ defines the likelihood of being in state i at time t given the observation sequence $\boldsymbol{o}_1 \dots \boldsymbol{o}_t$.

$$\alpha_t(i) = P(\boldsymbol{o}_1\boldsymbol{o}_2 \dots \boldsymbol{o}_t, q_t = i|\lambda) \tag{2.15}$$

In the first step, the algorithm begins in the starting state. Thus the likelihood to end up in state i given the first observation $\boldsymbol{o}_1$ is defined by:

$$\alpha_1(i) = a_{0i} b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq N \tag{2.16}$$

The likelihood of being in state j at time $t+1$, given the observation sequence $\boldsymbol{o}_1 \ldots \boldsymbol{o}_{t+1}$ can be calculated by using the likelihoods of possible states i at time t that could be used to reach state j given the observation likelihood $b_j(\boldsymbol{o}_{t+1})$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(\boldsymbol{o}_{t+1}), \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix} \tag{2.17}$$

The overall likelihood of the model given the observation sequence is given by the sum of the likelihoods of being in state i, given the observation sequence $\boldsymbol{o}_1 \ldots \boldsymbol{o}_T$ at the end time T over all states.

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.18}$$

2.5.6 Viterbi

The Viterbi algorithm seeks to find the state sequence $[q_1, q_2, \ldots, q_t]$ such that the likelihood of the state sequence and the observation sequence $[\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t]$ given the model parameters λ, is maximised, i.e. the most likely state sequence given the model and the observations. The most likely state at point t is given by:

$$q_t^* = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \quad 1 \leq t \leq T \tag{2.19}$$

Where $\gamma_t(i)$ denotes the state occupancy probability for state i at frame t for the complete observation sequence $\boldsymbol{o}$. The algorithm to calculate $\gamma_t(i)$ requires the forward- as well as the backward likelihoods $(\alpha_t(i), \beta_t(i))$ which will be explained below. The calculation of this state occupancy likelihood takes into account all possible state sequences, and thus all possible previous states leading to a state i. In the Viterbi-algorithm, the likelihood of a given state i at time t, is determined by the most likely sequence of preceding states instead of the overall likelihood of ending up in a state i at time t. In other words the difference between the Viterbi algorithm and the Forward algorithm is the use of maximisation instead of summation over previous states.

$\delta_t(i)$ denotes the maximum likelihood for a single state sequence that ends in state i after the first t observations. While $\delta_t(i)$ returns the likelihood for a state sequence ending in state i, $\psi_t(j)$ returns the most-likely predecessor state for a path ending in state j at time t. The Viterbi algorithm tries to find the best state sequence $\boldsymbol{q} = (q_1 q_2 \ldots q_T)$ given the observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1 \boldsymbol{o}_2 \ldots \boldsymbol{o}_T)$ under the model parameters λ:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1, q_2, \ldots q_{t-1}, q_t = i, \boldsymbol{o}_1, \boldsymbol{o}_2 \ldots \boldsymbol{o}_t | \lambda] \tag{2.20}$$

The initialisation step is defined by the probability of being in state i at time 1. Since there is no choice of predecessor state, no maximisation needs to be calculated. The predecessor state at time $t = 1$, $\psi_1(i)$, is the starting state 0.

$$\delta_1(i) = a_{0i} b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq N \tag{2.21}$$

$$\psi_1(i) = 0 \tag{2.22}$$

The maximum likelihood of being in state j at time $t+1$ given the observation o_{t+1} can be calculated from the state that is the most likely predecessor:

$$\delta_{t+1}(j) = \max_{1\leq i\leq N}[\delta_t(i)a_{ij}]b_j(\boldsymbol{o}_{t+1}), \qquad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \tag{2.23}$$

$$\psi_{t+1}(j) = \arg\max_{1\leq i\leq N}[\delta_t(i)a_{ij}] \qquad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \tag{2.24}$$

The likelihood for the complete observation sequence is given by:

$$P^* = \max_{1\leq i\leq N}[\delta_T(i)] \tag{2.25}$$

The most likely final state is given by:

$$q_T^* = \arg\max_{i\leq i\leq N}[\delta_T(i)] \tag{2.26}$$

$$\tag{2.27}$$

Then the most likely state sequence can be derived using $\psi_t(i)$ in the following way:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1 \tag{2.28}$$

When the state sequence has been calculated, the training vectors can be assigned to the HMM states and the covariance and mean parameters of the model as well as the transition probabilities can be updated accordingly.

2.5.7 Model initialisation and why

Further training is done using the Baum-Welch expectation-maximisation algorithm. The Baum-Welch algorithm consists of two steps. First likelihoods for the observation given the current model are calculated. Then, the model is updated according to the calculated likelihoods. It does not use an explicit alignment of the data vectors to the model. Instead state-occupancy probabilities are used to do a weighed assignment of the feature vectors to the HMM states that is used to re-estimate the model parameters. Thus one feature vector or frame may influence different states and not only a single state as in the Viterbi case.

The Viterbi alignment and training by pooling is simpler and may also help to estimate more distinct parameters for the different states, especially in the beginning of the training when the model may be initialised to similar or even identical parameters. The Baum-Welch algorithm tends to yield more robust estimates [47] since by its very nature more different data vectors are pooled.

To perform the Baum-Welch algorithm state occupancy probabilities have to be calculated, which can be done efficiently using the *Forward*- and *Backward*-procedures.

2.5.8 Backward algorithm

The Backward algorithm is similar to the Forward algorithm but calculates the likelihood of being in state i at time t given the *remaining* observation sequence $\boldsymbol{o}_{t+1} \ldots \boldsymbol{o}_T$, i.e.

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}\boldsymbol{o}_{t+2}\ldots\boldsymbol{o}_T|q_t = i, \lambda) \tag{2.29}$$

The algorithm is initialised by

$$\beta_T(i) = 1, \ \ 1 \leq i \leq N \tag{2.30}$$

and the recursion step is defined by

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}\beta_{t+1}(j)) \qquad \begin{array}{l} T-1 \geq t \geq 1 \\ 1 \leq j \leq N \end{array} \tag{2.31}$$

2.5.9 State occupancy probability

The state occupancy probability $\gamma_t(i)$ for state i at time t for a given observation sequence $\boldsymbol{O}$ is defined by:

$$\begin{aligned} \gamma_t(i) &= P(q_t = i|\boldsymbol{O}, \lambda) && (2.32) \\ &= \frac{P(\boldsymbol{O}, q_t = i|\lambda)}{P(\boldsymbol{O}|\lambda)} && (2.33) \\ &= \frac{P(\boldsymbol{O}, q_t = i|\lambda)}{\sum_{i=1}^{N} P(\boldsymbol{O}, q_t = i|\lambda)} && (2.34) \\ & && (2.35) \end{aligned}$$

and since $P(\boldsymbol{O}, q_t = i|\lambda)$ is equal to $\alpha_t(i)\beta_t(i)$, the state occupancy probability can be defined in terms of the forward and backward variables:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{2.36}$$

2.5.10 Baum-Welch

To re-estimate the transition probabilities the function $\xi_t(i,j)$ is defined as the probability of being in state i at time t and transitioning to state j at time $t+1$:

$$\begin{aligned} \xi_t(i,j) &= P(q_t = i, q_{t+1} = j|\boldsymbol{O}, \lambda) && (2.37) \\ &= \frac{P(q_t = i, q_{t+1} = j, \boldsymbol{O}|\lambda)}{P(\boldsymbol{O}|\lambda)} && (2.38) \\ &= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{P(\boldsymbol{O}|\lambda)} && (2.39) \\ &= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1}\beta_{t+1}(j))}{\sum_{i=1}^{N}\sum_{j+1}^{N} \alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1(j)}} && (2.40) \end{aligned}$$

This transition probability can also be used to define the state occupancy probability:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.41}$$

The transition probability ξ and the state occupancy probability γ can be interpreted in the following way: $\sum_{t=1}^{T-1} \gamma_t(i)$ is the expected number of transitions from state i given the observations $\boldsymbol{O}$ and $\sum_{t=1}^{T-1} \xi_t(i,j)$ is the expected number of transitions from state i to state j given the observations $\boldsymbol{O}$. The re-estimation formula for the transition probabilities a_{ij} is given by:

$$\begin{aligned} \hat{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} && (2.42) \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} && (2.43) \end{aligned}$$

The initial state probabilities, i.e. the transition probabilities from the initial state to state i are defined by the expected state occupancy probability in state i at time $(t = 1)$:

$$\hat{a}_{0i} = \gamma_1(i) \tag{2.44}$$

2.5.11 GMM parameter estimation

If we assume Gaussian mixture models as defined in 2.14 the likelihood that an observation at time t was generated by the i-th mixture of state j, $\gamma_t(j,k)$ can be defined as:

$$\gamma_t(j,k) = \left(\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right)\left(\frac{c_{jk}\mathcal{N}(\boldsymbol{o}_t, \boldsymbol{\mu}_{jk}\boldsymbol{\sigma}_{jk})}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(\boldsymbol{o}_t, \boldsymbol{\mu}_{jm}, \boldsymbol{\sigma}_{jm})}\right) \tag{2.45}$$

The mixture weight c_{jk} of mixture k in state j can be calculated from $\gamma_t(j,k)$ by

$$\hat{c}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M}\gamma_t(j,k)} \tag{2.46}$$

The means for the mixture can be estimated as follows

$$\hat{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)\boldsymbol{o}_t}{\sum_{t=1}^{T}\gamma_t(j,k)} \tag{2.47}$$

and the variances by

$$\hat{\boldsymbol{\sigma}}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)(\boldsymbol{o}_t - \hat{\boldsymbol{\mu}}_{jk})(\boldsymbol{o}_t - \hat{\boldsymbol{\mu}}_{jk})^\top}{\sum_{t=1}^{T}\gamma_t(j,k)} \tag{2.48}$$

which concludes the formulas necessary for re-estimation of the model.

2.5.12 Non-embedded and embedded training

After the initialisation using Viterbi, the Baum-Welch algorithm is applied on the segments cut from the recording instead of the complete utterance. In the HTK/HTS framework this is done using the *HRest* tool. Then embedded training is done. To this end, HMM models corresponding to the complete utterance are created from the phone models and Baum-Welch is used on the complete utterance model. Then the model, i.e. the component phone models, are updated accordingly. This allows for changes in the alignment beyond the initial segmentation. The re-estimation is repeated several times. The HTK/HTS framework tool for embedded re-estimation is called *HERest*.

2.5.13 Fullcontext cloning

After training monophone models using embedded training, we assume that the models are somewhat general and stable. To improve specialisation regarding coarticulation, i.e. taking the context into consideration, fullcontext models are created for the fullcontext labels in the corpus. This is done by copying the values of the monophone models corresponding to the fullcontext labels center-phone. Then embedded re-estimation is used to adapt the fullcontext phones. For example the fullcontext label

```
g^schwa-S+I=C@1_3/A:0_0_2/B:0-0-3@4-2&5-2#2-1$2-2!3-0;3-1|0/C:0+1+2/

D:content_1/E:content+5@2+1&2+0#1+0/F:0_0/G:0_0/H:6=2@1=1|L-L%/I:0=0
/
```

HCompV	computing variance floors
HInit	initialisation
HRest	re-estimation
HHEd	making monophone mmf (master macro file)
HERest	(embedded re-estimation (monophone))
HHEd	monophone to fullcontext cloning
HERest	fullcontext re-estimation
HHEd	tree based context clustering
HERest	re-estimation (clustered)
(custom)	adding dependency transformations and initialising them
HERest	re-estimation of the transform-enabled model

Table 2.2: *Sequence of used HTS Commands and Effect executed in the training script of this work*

`J:6+2-1`
is copied from the monophone model `S`. For a description of the fullcontext label see Section 2.3.5.

These cloned fullcontext models are re-estimated using several iterations of embedded Baum-Welch training. As described in Section 2.4.1 the fullcontext models are based on very few different observations and may statistically be unstable and overfit the training data. Thus clustering is used to combine different contexts and reduce the number of parameters to be estimated. The tree-based clustering is governed by a minimum-description-length criterion [48]. In the HTK/HTS framework, the tool to perform the clustering and tie the model parameters accordingly is *HHEd*. After the model has been clustered, several iterations of embedded Baum-Welch training are used to optimise the tied model. The next step performed in this work is to add dependency transformations to the model, which is done by a custom tool that also initialises the transformations. Then embedded re-estimation is used on the dependency model. Table 2.2 summarises the steps and tools involved in estimating the model for the training script of this work.

2.5.14 Duration models

By the nature of the HMM the time spent in a state, without transitioning to a different state follows a geometric distribution. Since this implies an exponentially decreasing likelihood for longer state durations, this model of the state duration does not provide an adequate representation of the temporal structure of speech [5]. To mitigate this problem the state durations are explicitly modelled. Thus the Markov property, which states that the designation of the state at the next time-step only depends on the current state and the transition probabilities, is weakened in that the probability to change to a different state is dependent on the duration of time already spent in this state. This is called a Hidden Semi-Markov Model (HSMM). Hidden Semi-Markov Models (HSMMs) have been applied to speech recognition [42] and are also used for HMM based speech synthesis [5]. Gaussians are used for the probability distributions of the duration model. This is even more important for speech synthesis than for speech recognition, since the state durations in speech recognition are influenced by the observation sequence, but have to be modelled more accurately for speech synthesis. The explicit duration modelling helps to achieve more accurate timing with regard to the generated observation sequences.

Training the HSMM models directly is different from the expectation-maximisation training discussed above. Adapted estimation formulas have been developed [49] and more recently been applied to speech synthesis [50]. However it is also possible to train a normal HMM and use alignment information of the trained model with respect to the training corpus to estimate the parameters of the state duration probability functions.

One way to do this, is to use the Viterbi algorithm to align the model with the training data (*forced alignment*) and to use the resulting state durations to estimate parameters for the probability distributions modelling the state durations in the HSMM. The problem with using only the forced-alignment information is that there are only a very small number of occurrences for each fullcontext label and thus very little information regarding the variance of the duration. However the variance of the duration is important for the modification of the speaking tempo. The forced-alignment based estimation does not make use of all the information available and a more comprehensive method is to use the state occupancy probabilities calculated in the expectation-maximisation algorithm to calculate estimates for the duration model parameters [7]. In the system used for this experiment the Viterbi-algorithm based method for estimating the duration model was used. Since no experiments regarding the artificial change of speaking tempo were made this should not pose a problem.

The estimated duration model is then clustered similar to the clustering of the spectral and F0 features to enable the modelling of unseen fullcontext-labels and to increase the statistical robustness of the model.

2.6 Synthesis in detail

2.6.1 How do we generate speech signals from the model?

From the textual representation we create a single model describing the whole utterance. That is a single string of HMM states, a left-to-right HMM covering the whole utterance. The duration of the states is described by a duration model - which turns the HMMs into HSMMs. The duration is not modelled by the states transition probabilities but rather explicitly by a model which contains a Gaussian for each state's duration. Each HMM state then is responsible for a certain duration of the parameter trajectories. The HMM states contain probability density functions which model the values the trajectory most likely assumes during the time they are responsible for modelling that part of the trajectory.

The parameter generation takes into account one stream at a time thus the following dimensions apply to each stream separately. n denotes the number of feature additions, for example three in the case of static, delta-, and delta-delta features. D denotes dimensionality of the feature vector of a single stream. T is the number of frames to be generated. τ is the window length of the filter coefficients used to generate the augmented features, i.e. delta and delta-delta features.

The HMM describes the probabilities for different observation parameters and probabilities for certain rate changes of the parameters (the delta and delta-delta features). Each state may model different parameters. Since left-to-right models are used, each state is only visited once, and thus each state is responsible for a certain part of the parameter trajectory. Because the way the parameters change is also modelled and because the state duration is also probabilistic these two, the parameter trajectories and the state duration, may influence each other and thus the results may differ between optimising the likelihood of the generated parameters and the optimal state sequence together instead of first finding the most likely state sequence $\boldsymbol{Q}$ and then finding the most likely parameter trajectory $\boldsymbol{O}$ generated by this state sequence [51], i.e. $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda)$ vs. $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda)$.

Since the computational cost for optimising both parameters at the same time are high, the approach of first finding the optimal state sequence and then optimising the parameter sequence has been chosen, maximising $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda)$ with respect to $\boldsymbol{O}$.

The state sequence then yields a sequence describing probability density functions. Parameter generation then means to find trajectories for the parameters that maximise the likelihood with respect to the probability density function sequence.

The standard approach [51] is described in linear algebra terms and the following is a description of the used matrices.

The likelihood for an observation x to be generated by a Gaussian j is given by:

$$b_j(\boldsymbol{o}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\sigma}_j|}} \exp(-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_j)^\top \boldsymbol{\sigma}_j^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_j)) \tag{2.49}$$

For a series of observations $\boldsymbol{O} = \begin{pmatrix}\boldsymbol{o}_1^\top & \dots & \boldsymbol{o}_T^\top\end{pmatrix}$ and a sequence of states $\boldsymbol{Q} = (q_1, \dots, q_T)$, the likelihood is denoted by:

$$P(\boldsymbol{O}|\boldsymbol{Q},\lambda) = \prod_{q_i, i\in\{1\dots T\}} b_{q_i}(\boldsymbol{o}_i) \tag{2.50}$$

$$\log P(\boldsymbol{O}|\boldsymbol{Q},\lambda) = \sum_{q_i, i\in\{1\dots T\}} \log(b_{q_i}(\boldsymbol{o}_i) \tag{2.51}$$

For a mixture j the log-likelihood for an observation $\boldsymbol{o}$ is given as:

$$\begin{aligned} log(b_j(\boldsymbol{o})) &= \boldsymbol{K} - \frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_j)^\top \boldsymbol{\sigma}_j^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_j) \\ &= \boldsymbol{K} - \frac{1}{2}(\boldsymbol{o}^\top \boldsymbol{\sigma}_j^{-1} - \boldsymbol{\mu}_j^\top \boldsymbol{\sigma}_j^{-1})(\boldsymbol{o}-\boldsymbol{\mu}_j) \\ &= \boldsymbol{K} - \frac{1}{2}(\boldsymbol{o}^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{o} - \boldsymbol{\mu}_j^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{o} - \boldsymbol{o}^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{\mu}_j) \\ &\boldsymbol{\sigma}^{-1} \text{ is symmetric and } \boldsymbol{o}, \boldsymbol{\mu}_j \text{ are vectors} \\ &= \boldsymbol{K} - \frac{1}{2}(\boldsymbol{o}^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{o} - 2\boldsymbol{\mu}_j^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{o} + \boldsymbol{\mu}_j^\top \boldsymbol{\sigma}_j^{-1}\boldsymbol{\mu}_j) \end{aligned} \tag{2.52}$$

By adequately structuring the observations $\boldsymbol{o}$, means $\boldsymbol{\mu}$ and covariances $\boldsymbol{\sigma}$ in larger matrices $\boldsymbol{O}, \boldsymbol{M}$ and $\boldsymbol{U}^{-1}$ the summation described in 2.50 can be done implicitly within the matrix multiplications.

Given for example vectors $\boldsymbol{a}_i \in \mathbb{R}^{n\times 1}$,$\boldsymbol{c}_i \in \mathbb{R}^{m\times 1}$ and a matrix $\boldsymbol{B}_i \in \mathbb{R}^{n\times m}$, the sum $\sum_i a_i^\top B_i c_i$ can also be expressed as:

$$\begin{pmatrix}\boldsymbol{a}_1^\top & \dots & \boldsymbol{a}_l^\top\end{pmatrix} \begin{pmatrix}\boldsymbol{B}_1 & & \\ & \ddots & \\ & & \boldsymbol{B}_l\end{pmatrix} \begin{pmatrix}\boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_l\end{pmatrix} \tag{2.53}$$

The structure of these matrices used to implicitly calculate the sums described in 2.51 is described below.

2.6.2 PDF stream - mean/variance stream

The total length of the utterance consists of T frames, each of which consists of D dimensions.

The observation sequence $\boldsymbol{O}$ also contains dynamic (augmented) features. These typically consist of derivations of the signal, for example the first (Δ) and second ($\Delta\Delta$) derivations. The final parameter trajectories that are used to reconstruct the speech signal do not contain these augmentations, but only the *static* (i.e. without dynamic) features. The dynamic feature sequence can be calculated from the static feature sequence. The structure of the static feature vector is given as:

$$\boldsymbol{C} = \begin{pmatrix}\boldsymbol{c}_1^\top & \dots & \boldsymbol{c}_T^\top\end{pmatrix}^\top = \begin{pmatrix}c_{1_1} & c_{1_2} & \dots & c_{1_D} & \dots & c_{T_1} & c_{T_2} & \dots & c_{T_D}\end{pmatrix}^\top \tag{2.54}$$

The vector containing the static feature vector sequence, $\boldsymbol{C}$, can be augmented with dynamic features, i.e. for example first and second derivatives. This vector containing the dynamic feature vector sequence $\boldsymbol{O}$, is structured as follows:

$$\mathbf{o}_i^\top = \begin{pmatrix} o_{i_1} & \dots & o_{i_D} & \Delta o_{i_1} & \dots & \Delta o_{i_D} & \Delta\Delta o_{i_1} & \dots & \Delta\Delta o_{i_D} \end{pmatrix}^\top \tag{2.55}$$

The vector $\boldsymbol{O}$ is a $TDn \times 1$ vector and is structured as a sequence of frames $\boldsymbol{o}_i$, which are a sequence of normal $(o_{i_1} \dots o_{i_D})$, delta $(\Delta o_{i_1} \dots \Delta o_{i_D})$ and delta-delta features $(\Delta\Delta o_{i_1} \dots \Delta\Delta o_{i_D})$.

The Δ and $\Delta\Delta$ features are calculated from the sequence of static feature vectors using the augmentation matrix $\boldsymbol{W}$ and $\boldsymbol{W} \in \mathbb{R}^{TDn \times TD}$.

$$\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C} \tag{2.56}$$

$$[TDn \times 1] = [TDn \times TD][TD \times 1] \tag{2.57}$$

The augmentation matrix $\boldsymbol{W}$ contains the augmentation coefficients necessary to calculate the augmented features. Here the coefficients for the central difference operator - the discrete derivation approximation - are used. The calculation of the augmented features is similar to feeding a signal, i.e. a segment of the original feature sequence, through a finite impulse response filter. The structure of $\boldsymbol{W}$ is described by the following partition:

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{\Omega}_0 & \dots & \boldsymbol{\Omega}_\delta & & & \\ \vdots & \ddots & & \ddots & & \\ \boldsymbol{\Omega}_{-\delta} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & \boldsymbol{\Omega}_\delta \\ & & \ddots & & \ddots & \vdots \\ & & & \boldsymbol{\Omega}_{-\delta} & \dots & \boldsymbol{\Omega}_0 \end{pmatrix} \tag{2.58}$$

where δ is related to the window length τ by $\tau = 2 \cdot \delta + 1$, and n is the number of augmentations, i.e. 3 in the case of normal, Δ and $\Delta\Delta$ features. The window length we use is 3, thus, in our case we have $\delta = 1$.

The augmentation coefficients for the j-th augmentation features (normal, Δ and $\Delta\Delta$) are denoted by $\omega_i^{(j)}$.

$$\boldsymbol{\Omega}_i = \begin{pmatrix} \boldsymbol{\Omega}_i^{(0)} \\ \boldsymbol{\Omega}_i^{(1)} \\ \vdots \\ \boldsymbol{\Omega}_i^{(n-1)} \end{pmatrix} \in \mathbb{R}^{Dn \times D} \tag{2.59}$$

$$\boldsymbol{\Omega}_i^{(j)} = \begin{pmatrix} \omega_i^{(j)} & & \\ & \ddots & \\ & & \omega_i^{(j)} \end{pmatrix} \in \mathbb{R}^{D \times D} \tag{2.60}$$

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_{-\delta}^{(0)} & \dots & \omega_i^{(0)} & \dots & \omega_\delta^{(0)} \\ \vdots & & \vdots & & \vdots \\ \omega_{-\delta}^{(n-1)} & \dots & \omega_i^{(n-1)} & \dots & \omega_\delta^{(n-1)} \end{pmatrix} \in \mathbb{R}^{n \times \tau} \tag{2.61}$$

For this work the following augmentation vectors were used:

$$\boldsymbol{\omega} = \begin{pmatrix} 0 & 1 & 0 \\ -0.5 & 0 & 0.5 \\ 0.25 & -0.5 & 0.25 \end{pmatrix} \tag{2.62}$$

To ease the understanding of the structure of $\boldsymbol{W}$ the following illustrates what $\boldsymbol{W}$ would look like for a fictional example with only three dimensions ($D = 3$) and the augmentation coefficients described in 2.62. Two instances of $\boldsymbol{\Omega}_0$ as described in 2.58 are highlighted in the illustration below 2.63.

$$\boldsymbol{W} = \begin{pmatrix}
1 & & & 0 & & & & & & \cdots \\
& 1 & & & 0 & & & & & \cdots \\
& & 1 & & & 0 & & & & \cdots \\
0 & & & 0.5 & & & & & & \cdots \\
& 0 & & & 0.5 & & & & & \cdots \\
& & 0 & & & 0.5 & & & & \cdots \\
-0.5 & & & 0.25 & & & & & & \cdots \\
& -0.5 & & & 0.25 & & & & & \cdots \\
& & -0.5 & & & 0.25 & & & & \cdots \\
& & & 1 & & & & & & \cdots \\
& & & & 1 & & & & & \cdots \\
& & & & & 1 & & & & \cdots \\
-0.5 & & & 0 & & & 0.5 & & & \cdots \\
& -0.5 & & & 0 & & & 0.5 & & \cdots \\
& & -0.5 & & & 0 & & & 0.5 & \cdots \\
0.25 & & & -0.5 & & & 0.25 & & & \cdots \\
& 0.25 & & & -0.5 & & & 0.25 & & \cdots \\
& & 0.25 & & & -0.5 & & & 0.25 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix} \tag{2.63}$$

$\boldsymbol{U}^{-1}$ is the banded matrix corresponding to the covariance matrices of the whole utterance. $\boldsymbol{U}_{q_i}^{-1} \in \mathbb{R}^{D \times D}$ is the inverse of the covariance matrix corresponding to the $i-$th frame.

$$\boldsymbol{U}^{-1} = \begin{pmatrix} \boldsymbol{\sigma}_{q_1}^{-1} & & \\ & \ddots & \\ & & \boldsymbol{\sigma}_{q_T}^{-1} \end{pmatrix} \in \mathbb{R}^{nTD \times nTD} \tag{2.64}$$

$\boldsymbol{M}$ is the sequence of the corresponding mixture means for each of the frames. $\boldsymbol{\mu}_{q_i}$ is the mean of a mixture and is structured as normal, delta and delta-delta means, similar to the structure of the observation vectors described in 2.55.

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{\mu}_{q1}^{\top} & \boldsymbol{\mu}_{q2}^{\top} & \dots & \boldsymbol{\mu}_{qT}^{\top} \end{pmatrix}^{\top} \in \mathbb{R}^{DTn \times 1} \tag{2.65}$$

From the matrices described above and the single frame likelihood $\log(b_j(\boldsymbol{o}))$ described in 2.52, the likelihood for the sequence of frames $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda)$ described in 2.50 can be rewritten as:

$$\begin{aligned}
\log P(\boldsymbol{O}|\boldsymbol{Q},\boldsymbol{\lambda}) &= -\frac{1}{2}\boldsymbol{O}^T\boldsymbol{U}^{-1}\boldsymbol{O} + \boldsymbol{O}^T\boldsymbol{U}^{-1}\boldsymbol{M} + \boldsymbol{M}^\top\boldsymbol{U}^{-1}\boldsymbol{M} + \tilde{\boldsymbol{K}} \\
\log P(\boldsymbol{WC}|\boldsymbol{Q},\boldsymbol{\lambda}) &= -\frac{1}{2}(\boldsymbol{WC})^T\boldsymbol{U}^{-1}(\boldsymbol{WC}) + (\boldsymbol{WC})^T\boldsymbol{U}^{-1}\boldsymbol{M} + \boldsymbol{M}^\top\boldsymbol{U}^{-1}\boldsymbol{M} + \tilde{\boldsymbol{K}} \quad (2.66)
\end{aligned}$$

The following rules hold for differentiation if $\boldsymbol{x}$ is a vector and $\boldsymbol{A}$ a matrix:

$$\frac{\partial \boldsymbol{x}^\top A}{\partial \boldsymbol{x}} = \boldsymbol{A} \tag{2.67}$$

$$\frac{\partial \boldsymbol{x}^\top A x}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^\top)\boldsymbol{x} \tag{2.68}$$

$$\text{if } \boldsymbol{A} \text{ is symmetric then } (\boldsymbol{A} + \boldsymbol{A}^\top)\boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x} \tag{2.69}$$

The most likely parameter sequence $\boldsymbol{C}$ can then be found by maximising the likelihood with respect to and solving for $\boldsymbol{C}$:

$$\frac{\partial \log P(\boldsymbol{WC}|\boldsymbol{Q},\lambda)}{\partial \boldsymbol{C}} = 0$$

$$\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{W}\boldsymbol{C} = \boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{M} \tag{2.70}$$

$$\boldsymbol{C} = (\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{W})^{-1}\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{M} \tag{2.71}$$

2.6.3 SciPy implementation

Solving the problem of parameter generation means to find solutions for $\boldsymbol{C}$. $\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{W}$ is a large matrix:

$$[TD \times TDn][TDn \times TDn][TDn \times TD] = [TD \times TD]$$

and the resource requirements of matrix multiplication are therefore high. However it is a banded matrix (see Section A.2 in the appendix) and thus quite sparse. Since the inverse of a sparse banded matrix is not necessarily sparse, it is beneficial that there are algorithms to solve the system of linear equations described by 2.70 without explicitly calculating the inverse of $(\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{W})$. Another point making the calculation simpler is the fact that the right hand side, $(\boldsymbol{W}^\top\boldsymbol{U}^{-1}\boldsymbol{M})$, is a column vector:

$$[TD \times TDn][TDn \times TDn][nTD \times 1] = [TD \times 1]$$

2.6.4 Banded matrix representation

The implementation of the parameter generation was done using the Python programming language and the SciPy [52] libraries which provide a convenient interface to the LAPACK suite. A solver for banded matrices is also provided. A compact representation (BLAS general banded matrix storage) is used, in SciPy this is called a “matrix diagonal ordered form”. $\boldsymbol{A}$ is a $m \times m$ matrix and has upper bandwidth u and lower bandwidth l. This means that all the elements above the upper diagonal u or below the lower diagonal l are zero. In 2.72 a 6×6 bandmatrix with upper bandwidth $u = 1$ and lower bandwidth $l = 2$ is illustrated, as well as the corresponding compact representation:

$$\begin{pmatrix} a_{00} & a_{01} & & & & \\ a_{10} & a_{11} & a_{12} & & & \\ a_{20} & a_{21} & a_{22} & a_{23} & & \\ & a_{31} & a_{32} & a_{33} & a_{34} & \\ & & a_{42} & a_{43} & a_{44} & a_{45} \\ & & & a_{53} & a_{54} & a_{55} \end{pmatrix} \Rightarrow \begin{pmatrix} * & a_{01} & a_{12} & a_{23} & a_{34} & a_{45} \\ a_{00} & a_{11} & a_{22} & a_{33} & a_{44} & a_{55} \\ a_{10} & a_{21} & a_{32} & a_{43} & a_{54} & * \\ a_{20} & a_{31} & a_{42} & a_{53} & * & * \end{pmatrix} \tag{2.72}$$

Formally the elements a_{ij} of a bandmatrix are zero:

$$\begin{gathered} \forall i, j \ \ (0 \leq i < m, 0 \leq j < m) : \\ a_{ij} = 0 \text{ if } i > j + l \text{ or } i < j - u \end{gathered}$$

$\boldsymbol{AB}$ is a compact representation of $\boldsymbol{A}$ and has size $l + u + 1 \times m$, the correspondence between the elements a_{ij} and ab_{ij} is as follows:

$$\begin{aligned} ab_{u+i-j,j} &= a_{i,j} \\ ab_{u+(j+i-u)-j,j} &= a_{j+i-u,j} \\ ab_{i,j} &= \begin{cases} a_{j+i-u,j} & \text{if } 0 \leq i + j - u \leq m \\ 0 \end{cases} \end{aligned} \tag{2.73}$$

2.6.5 How does the control work, what is different with control?

In the control case we have two streams, an acoustic and a visual stream. Thus we do not only have observations $\boldsymbol{O}$ but acoustic observations $\boldsymbol{O_X}$ as well as visual observations $\boldsymbol{O_Y}$, the static feature vectors - without the dynamic delta, and delta-delta features - are not called $\boldsymbol{C}$ but $\boldsymbol{X}$ and $\boldsymbol{Y}$ accordingly.

2.6.6 Linear regression

First a conceptual description of the generation of the features is given, then the mathematical details are described. Controlling works by modelling the dependency between the control space, in our case the visual feature space and the target space, the spectral parameters, by piece-wise linear functions. Every state of the HMM is assigned a linear function and because the trajectory is modelled by different states, the result is a piece-wise linear modelling. Piece-wise with regard to the time domain. This has been called a Multiple Regression Hidden Markov Model (MR-HMM) [11].

In the context of the articulatory control this is called a *Feature-Dependency System* and is described in detail by Ling et al. [13]. The mathematical formulations are reproduced for the sake of completeness.

The model basically assumes a linear relation on the Gaussian mixtures:

$$b_j(\boldsymbol{x}_t, \boldsymbol{y}_t) = b_j(\boldsymbol{x}_t | \boldsymbol{y}_t) b_j(\boldsymbol{y}_t) \tag{2.74}$$

$$b_j(\boldsymbol{x}_t | \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{A}_j \boldsymbol{y}_t + \boldsymbol{\mu}_{\boldsymbol{X}_j}, \boldsymbol{\sigma}_{\boldsymbol{X}_j}) \tag{2.75}$$

In essence a linear regression is used to map from the visual feature space to the spectral parameter space, however this relation is also modelled in the HMM parameter estimation and taken into account to derive the following modified expectation maximisation formulas [12, 13].

For the means $\hat{\boldsymbol{\mu}}$, variances $\hat{\boldsymbol{\sigma}}$ and transformation matrices $\hat{\boldsymbol{A}}$:
The estimation formulas for the visual/control means $\boldsymbol{\mu}\hat{_{Y_j}}$, variances $\boldsymbol{\sigma}\hat{_{Y_j}}$ is unmodified:

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{Y}_j} = \frac{\sum_{t=1}^{\top} \gamma_j(t)\boldsymbol{y}_t}{\sum_{t=1}^{\top} \gamma_j(t)} \tag{2.76}$$

$$\hat{\boldsymbol{\sigma}}_{\boldsymbol{Y}_j} = \frac{\sum_{t=1}^{\top} \gamma_j(t)(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{\boldsymbol{Y}_j})(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{\boldsymbol{Y}_j})^{\top}}{\sum_{t=1}^{\top} \gamma_j(t)} \tag{2.77}$$

The estimation formula for the transformation $\hat{\boldsymbol{A}}_j$, is related to the estimation of a linear regression and can be seen as the estimation of a linear regression with the data points weighed according to their state occupancy probability $\gamma_j(t)$

$$\hat{\boldsymbol{A}}_j = \left[\sum_{t=1}^{\top} \gamma_j(t)(\boldsymbol{x}_t - \boldsymbol{\mu}_{\boldsymbol{X}_j})\boldsymbol{y}_t^{\top}\right] \cdot \left[\sum_{t=1}^{\top} \gamma_j(t)\boldsymbol{y}_t\boldsymbol{y}_t^{\top}\right]^{-1} \tag{2.78}$$

The formula for the acoustic means $\boldsymbol{\mu}\hat{_{X_j}}$ and variances $\boldsymbol{\sigma}\hat{_{X_j}}$ is modified to incorporate the dependency:

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{X}_j} = \frac{\sum_{t=1}^{\top} \gamma_j(t)(\boldsymbol{x}_t - \hat{\boldsymbol{A}}_j\boldsymbol{y}_t)}{\sum_{t=1}^{\top} \gamma_j(t)} \tag{2.79}$$

$$\hat{\boldsymbol{\sigma}}_{\boldsymbol{X}_j} = \frac{\sum_{t=1}^{\top} \gamma_j(t)(\boldsymbol{x}_t - \hat{\boldsymbol{A}}_j\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{\boldsymbol{X}_j})(\boldsymbol{x}_t - \hat{\boldsymbol{A}}_j\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{\boldsymbol{X}_j})^{\top}}{\sum_{t=1}^{\top} \gamma_j(t)} \tag{2.80}$$

The system [6] used in this work already includes support for the dependency aware estimation formulas. By adding transformation to the model on a per-state level it is possible to restrict the use of control to some cases only. This means that the dependency expectation-maximisation formulas are only used for states that have a transformation matrix associated, in other cases the equations described in 2.47,2.48 are used.

2.6.7 Initialisation

In the training process first a model without control transformations is estimated and then the control transformations are added and the model is re-estimated. Lei et al. [6] found that this might lead to very small coefficients (close to zero) in the transformation matrix and thus very little transformation ability. To prevent this the linear regression was first estimated using forced-aligned training data and expectation maximisation was performed from this initialised model. A possible alternative to using forced-aligned training data would be use the model information, i.e. the visual and acoustic means, to estimate a rough approximation of the linear relation. The tree leaves carrying the acoustic and visual Probability Density Function (PDF) models have different weights and the information regarding this weights that is also used for tree clustering needs to be incorporated. Since there are far more training observations than mixture models in the leaves the estimation on the forced-aligned training is likely to be more robust.

2.6.8 Parameter generation

The parameter generation also needs to incorporate the appropriate changes for control. First the control space mean sequences are generated, i.e. the visual mean sequences (Figure 2.11 top). Then

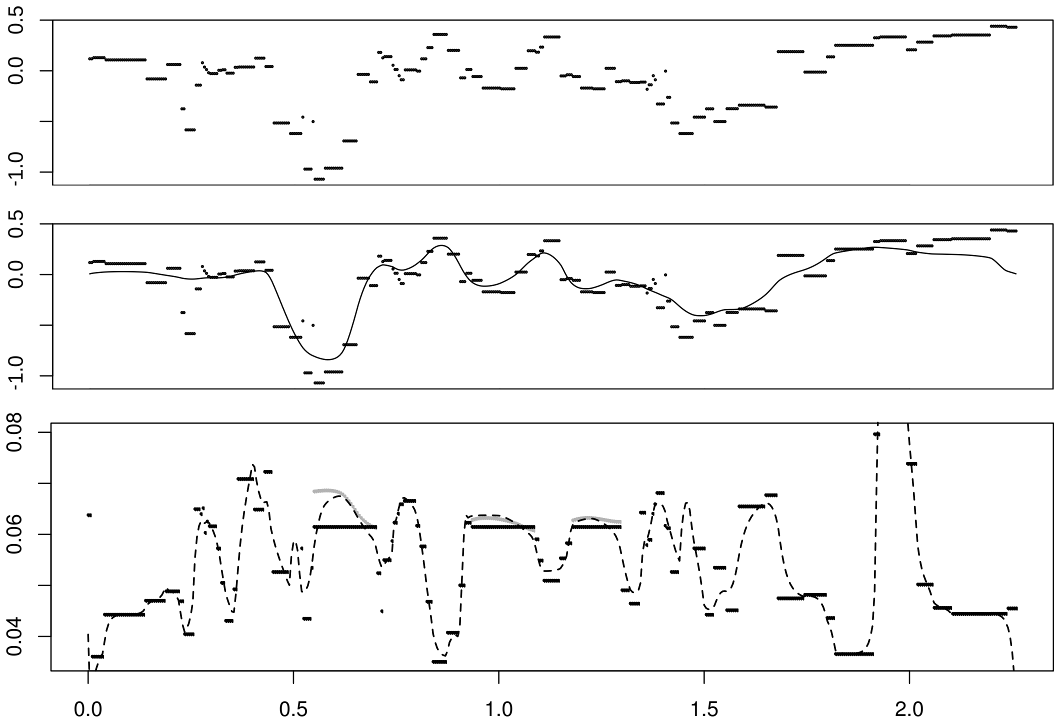

Figure 2.11: *(top) Visual mean sequence of first component, (middle) smoothed trajectory for first visual component (bottom) acoustic stream, first LSF component means in solid black horizontal line segments, controlled means solid line in grey, smooth trajectory dashed black*

the modifications are applied to the segments of the mean sequence that correspond to the phones selected for control, after which the smooth visual trajectories are estimated (Figure 2.11 middle). The modification could also be performed on the smooth visual trajectories instead of the visual means, however the transitions between the phones would have to be smoothed out explicitly. Thus the modification of the mean sequence simplifies the creation of continuous trajectories without extra effort.

Then this modified smooth trajectories are used as input for the control or transformation mapping and the resulting changes are overlaid on the acoustic mean sequences. Then this modified mean sequence is used to generate the smooth acoustic trajectories (Figure 2.11 bottom).

2.6.9 Parameter generation with control math detail

The explicit modelling of the dependency as described in 2.74 takes into account that the state occupancy probabilities may change due to the dependency. Thus the maximum likelihood generation of the control/visual features is also influenced by the expectation for the acoustic features and the transformation matrix. The joint log-likelihood of the visual and acoustic observation sequences is given by:

$$\begin{aligned}\log P(\boldsymbol{O_X}\boldsymbol{O_Y}|\boldsymbol{Q},\boldsymbol{\lambda}) = &-\frac{1}{2}\boldsymbol{O}_{\boldsymbol{X}}^{T}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{O_X} + \boldsymbol{O}_{\boldsymbol{X}}^{T}\boldsymbol{U}_{\boldsymbol{X}}^{-1}(\boldsymbol{A}\boldsymbol{O_Y} + \boldsymbol{M_X})\\ &+\frac{1}{2}(\boldsymbol{M}_{\boldsymbol{X}}^{\top} + \boldsymbol{A}^{\top}\boldsymbol{O}_{\boldsymbol{Y}}^{\top})\boldsymbol{U}_{\boldsymbol{X}}^{\boldsymbol{-1}}(\boldsymbol{A}\boldsymbol{O_Y} + \boldsymbol{M_X}) + \widetilde{\boldsymbol{K}}\\ &-\frac{1}{2}\boldsymbol{O}_{\boldsymbol{Y}}^{T}\boldsymbol{U}_{\boldsymbol{Y}}^{-1}\boldsymbol{O_Y} + \boldsymbol{O}_{\boldsymbol{Y}}^{T}\boldsymbol{U}_{\boldsymbol{Y}}^{-1}\boldsymbol{M_Y} + \widetilde{\widetilde{K}}\end{aligned}$$

Where $(\boldsymbol{M}_{\boldsymbol{X}}^{\top} + \boldsymbol{A}^{\top}\boldsymbol{O}_{\boldsymbol{Y}}^{\top})\boldsymbol{U}_{\boldsymbol{X}}^{\boldsymbol{-1}}(\boldsymbol{A}\boldsymbol{O_Y} + \boldsymbol{M_X})$ expands to

$$\boldsymbol{O}_{\boldsymbol{Y}}^{\top}\boldsymbol{A}^{\top}\boldsymbol{U}^{-1}\boldsymbol{A}\boldsymbol{O_Y} + 2\boldsymbol{O}_{\boldsymbol{Y}}^{\top}\boldsymbol{A}^{\top}\boldsymbol{U}^{-1}\boldsymbol{M_X} + \boldsymbol{M}_{\boldsymbol{X}}^{\top}\boldsymbol{U}^{-1}\boldsymbol{M_X} \tag{2.81}$$

The relationship of the static features $\boldsymbol{X}$ (resp. $\boldsymbol{Y}$) to the augmented, i.e. dynamic, features $\boldsymbol{O_Y}$ ($\boldsymbol{O_X}$) is established by substituting $\boldsymbol{O_Y} = \boldsymbol{W_Y}\boldsymbol{Y}$ and $\boldsymbol{O_X} = \boldsymbol{W_X}\boldsymbol{X}$, expanding to:

$$\begin{aligned}\log P(\boldsymbol{W_X}\boldsymbol{X}, \boldsymbol{W_Y}\boldsymbol{Y}|\lambda, \boldsymbol{Q}) = \ & \boldsymbol{X}^{\top}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{A}\boldsymbol{W_Y}\boldsymbol{Y}\\ &-\frac{1}{2}\boldsymbol{X}^{\top}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{W_X}\boldsymbol{X} + \boldsymbol{X}^{\top}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{M_X}\\ &-\frac{1}{2}\boldsymbol{Y}^{\top}\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U}_{\boldsymbol{Y}}^{-1} + \boldsymbol{A}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{A})\boldsymbol{W_Y}\boldsymbol{Y}\\ &+\boldsymbol{Y}^{\top}\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U}_{\boldsymbol{Y}}^{-1}\boldsymbol{M_Y} - \boldsymbol{A}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{M_X}) + K\end{aligned} \tag{2.82}$$

This leads to the following maximisation for the acoustic parameter sequence:

$$\frac{\partial \log P(\boldsymbol{W_X}\boldsymbol{X}, \boldsymbol{W_Y}\boldsymbol{Y}|\lambda, \boldsymbol{Q})}{\partial \boldsymbol{X}} = 0 \tag{2.83}$$

Solving the derivation, using the rules in 2.67-2.69, yields:

$$0 = \boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{A}\boldsymbol{W_Y}\boldsymbol{Y} - \boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{W_X}\boldsymbol{X} + \boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{M_X} \tag{2.84}$$

$$\boldsymbol{X} = (\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{W_X})^{-1}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}(\boldsymbol{M_X} + \boldsymbol{A}\boldsymbol{W_Y}\boldsymbol{Y}) \tag{2.85}$$

Maximising with respect to the visual parameter sequence $\boldsymbol{Y}$:

$$\frac{\partial \log P(\boldsymbol{W_X}\boldsymbol{X}, \boldsymbol{W_Y}\boldsymbol{Y}|\lambda, \boldsymbol{Q})}{\partial \boldsymbol{Y}} = 0 \tag{2.86}$$

Which can be solved and rewritten as:

$$\begin{aligned}0 = \ & (\boldsymbol{X}^{\top}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U_X}\boldsymbol{A}\boldsymbol{W_Y})^{\top}\\ &-\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U}_{\boldsymbol{Y}}^{-1} + \boldsymbol{A}^{\top}\boldsymbol{U_X}\boldsymbol{A})\boldsymbol{W_Y}\boldsymbol{Y}\\ &+(\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U_Y}\boldsymbol{M_Y} - \boldsymbol{A}^{\top}\boldsymbol{U_X}\boldsymbol{M_X}))^{\top}\end{aligned} \tag{2.87}$$

$$\tag{2.88}$$

This can be solved with respect to $\boldsymbol{Y}$ [13]:

$$\begin{aligned}\boldsymbol{Y} = \ & (\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U}_{\boldsymbol{Y}}^{-1} + \boldsymbol{A}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{A} - \boldsymbol{A}^{\top}\boldsymbol{Z}^{-1}\boldsymbol{A})\boldsymbol{W_Y})^{-1}\\ &\cdot\boldsymbol{W}_{\boldsymbol{Y}}^{\top}(\boldsymbol{U}_{\boldsymbol{Y}}^{-1}\boldsymbol{M_Y} + \boldsymbol{A}^{\top}\boldsymbol{Z}^{-1}\boldsymbol{M_X} - \boldsymbol{A}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{M_X})\end{aligned} \tag{2.89}$$

$$\boldsymbol{Z}^{-1} = \boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{W_X}(\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1}\boldsymbol{W_X})^{-1}\boldsymbol{W}_{\boldsymbol{X}}^{\top}\boldsymbol{U}_{\boldsymbol{X}}^{-1} \tag{2.90}$$

Since the calculation of $\boldsymbol{Z}^{-1}$ has a high computational cost, the following approximation is used:

$$\frac{\partial \log P(\boldsymbol{W_Y Y}|\lambda, q^*)}{\partial \boldsymbol{Y}} \approx \frac{\partial \log P(\boldsymbol{W_X X}, \boldsymbol{W_Y Y}|\lambda, q^*)}{\partial \boldsymbol{Y}} \tag{2.91}$$

Since this is the same as the parameter generation without dependency modelling, see 2.71, as described above this yields:

$$\boldsymbol{Y} = (\boldsymbol{W_Y^T U_Y^{-1} W_Y})^{-1} \boldsymbol{W_Y^T U_Y^{-1} M_Y} \tag{2.92}$$

CHAPTER 3

Experiments

This chapter describes the experiments and steps performed to investigate the feasibility of controlling the acoustic synthesis by modifying the associated visual features. The underlying model consists of audio-visual HMMs with linear regression dependencies for some phone-contexts. The linear-regression and the resulting changes to the HMM framework are described in Section 2.6.5. The selection of phone-contexts to be modelled was established during the series of experiments described below. The experiments were performed on a system adapted from the system used in the formant control experiments [6]. Formants are frequencies at which the spectral energy has distinctive local maxima. Figure 3.1 illustrates the first three formants F1,F2,F3 for a spectral envelope derived from LPC analysis. The first two formants can be used to differentiate between the different vowels [23].

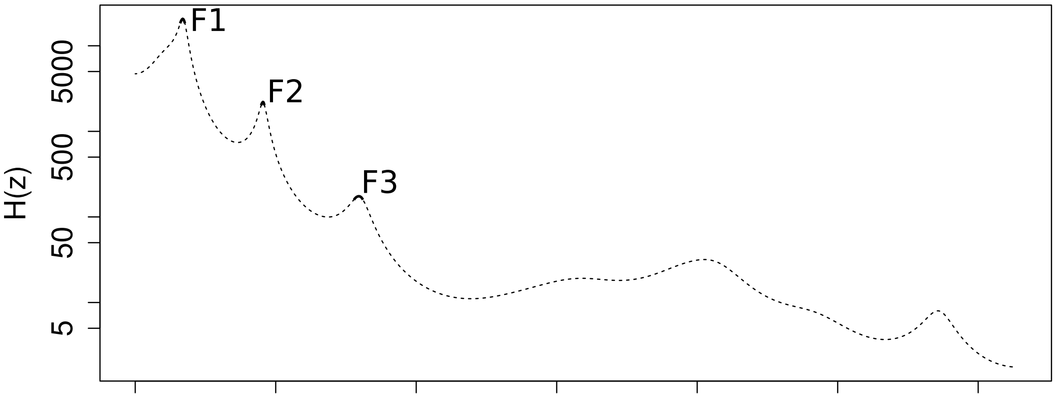

Figure 3.1: *Spectral envelope derived from linear predictive coding. The first three formants (F1 ... F3) are indicated.*

3.1 Possible problems with using visual features for control

Control by modelling the dependency of acoustic features on articulatory features has been shown to be feasible [13]. Since the articulatory features are directly involved in the sound production this relation or correspondence is more obvious than a relation of visual and acoustic features. Visual features only represent a subset of the features influencing human speech production. Thus the ability to exercise control using only the visual features is more restricted than controlling by combined visual & articulatory features. This means that the changes that can be made may be smaller or that additional information regarding, or restriction of, the target (acoustic) space is necessary.

3.2 Insufficient control

The first step to investigate the feasibility of visual control was to replace the formant stream in the formant control system with the visual stream of our corpus.

At this point the visual stream consisted of thirty components of the PCA reduction of the full 3D marker position recordings. The parameters relevant to the dependency modelling of the formant control system were also applied to the visual control system. This includes a splitting of the linguistic tree into a vowel and a non-vowel part, as well as modelling the control by tying transformation functions to the third level (counted from the tree root)of the linguistic tree. Synthesis using this system appeared to work quite well, although there was (and still is) some buzziness which was attributed to not using the *global variance* improvements [53] resulting in more over-smoothing of the acoustic parameter trajectories.

The acoustic means of the controlled phones were not tied, thus the means can be interpreted as correcting the remaining error between the control modelling and the acoustic means of the uncontrolled systems. Remember the likelihood of an observation under dependency modelling is specified by

$$b_j(\boldsymbol{x}_t|\boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{A}_j\boldsymbol{y}_t + \boldsymbol{\mu}_{X_j}, \boldsymbol{\sigma}_{X_j}) \tag{3.1}$$

If dependency modelling is enabled for a certain pair of audio-visual HMM states, the mean of the acoustic parameter distribution for the parameter generation is inferred from the visual features and the transformation, i.e. $\boldsymbol{A}_j\boldsymbol{y}_t$. Since the audio streams probability density function parameter $\boldsymbol{\mu}_{X_j}$ also describes a mean, this mean ($\boldsymbol{\mu}_{X_j}$) can be seen as modelling the residual error that remains after applying the control, especially if this parameter is untied and may be different for each of the states.

In a first experiment a phone should be modified by modifying the associated visual features. To this end a source vowel and a target vowel were selected. The visual feature means of the five-state fullcontext model of the source vowel were replaced by the visual feature means of the target vowel. The goal was to make the source phone sound more like the target phone. However the effect of replacing the visual features and the resulting acoustic changes induced by the transformation of the modified visual features did not yield a perceivable change in the quality of the sound of the source phone with regard to the target phone. Instead, only a distortion of the source phone was achieved. This was a first indication that the modelling of the acoustic features by the visual features was insufficient.

3.3 Requirements for control

To exercise meaningful control it is necessary but not sufficient that the control parameters actually induce change in the target space. No control can be exercised if the coefficients of the linear

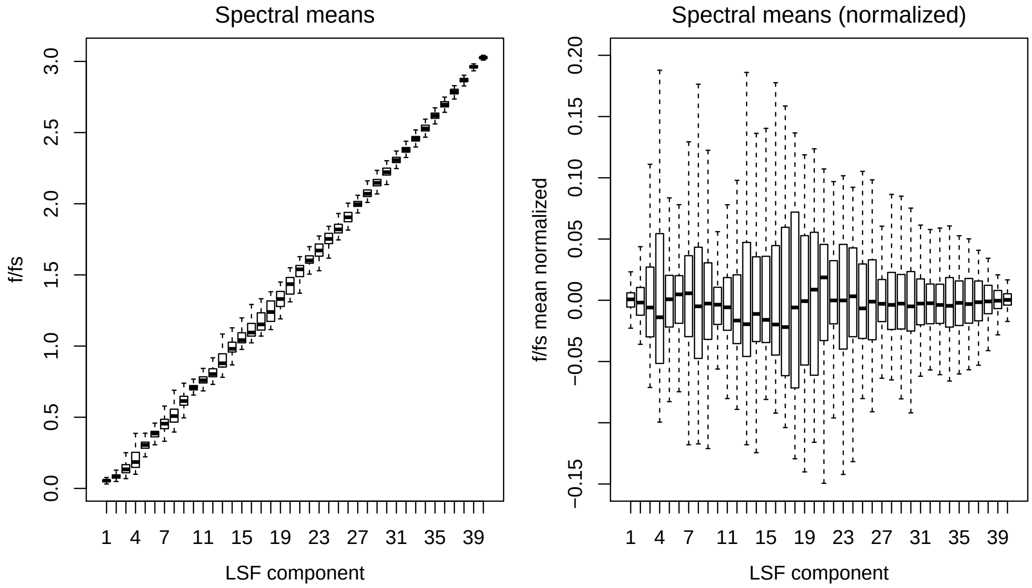

Figure 3.2: *Boxplots of the 40 LSF dimensions variations of the forced-aligned acoustic vectors corresponding to the vowels and diphthongs in the training corpus (right) and a mean-normalised illustration of the same data (left)*

relation (the A in $\mathcal{N}(\sigma, AY + \mu, X)$) are zero or very close to zero. However it is not easy to see how large these coefficients have to be, to be able to induce meaningful changes because *a*) it is a multidimensional relation and *b*) the necessary variation in the different dimensions is difficult to estimate.

Each realisation of each vowel will be slightly different each time it is uttered. Also the vowels sound differently. Thus the associated acoustic parameter representations will vary. The acoustic representations of all vowels in the recorded corpus can be pooled and the variation for each of the parameter dimensions can be calculated. If the acoustic feature vectors could be completely restored from the visual representation, the set of all visual representations could be used to calculate a corresponding set of acoustic representations. The variation within the set of calculated acoustic representations should then be comparable to the variation within the set of recorded acoustic vowel representations discussed above.

To get an idea of the variation that should be induced by the transformation, the naturally occurring variation within all naturally occurring vowels was investigated. To this end the models corresponding to vowels were pooled. That is all the fullcontext labels containing a center vowel were selected and the corresponding acoustic tree leaves determined. The acoustic mean value vectors represented by these tree leaves ($\mu_1 \ldots \mu_D$) were pooled. Each dimension was treated as a multiset. Boxplots of these multisets are shown in Figure 3.2 (left), since there is a large difference in the means, which we are not interested in, the variation is shown as mean-normalised boxplots (right).

To have a rough indication of the possible variation that can be induced by the control a random subset (40%) of the vowel leaves (corresponding to a selection of means and transformation functions) and a random subset (40%) of the means of the visual leaves was selected. The restriction to a subset was done to reduce the number of necessary comparisons. Each of the visual mean vectors

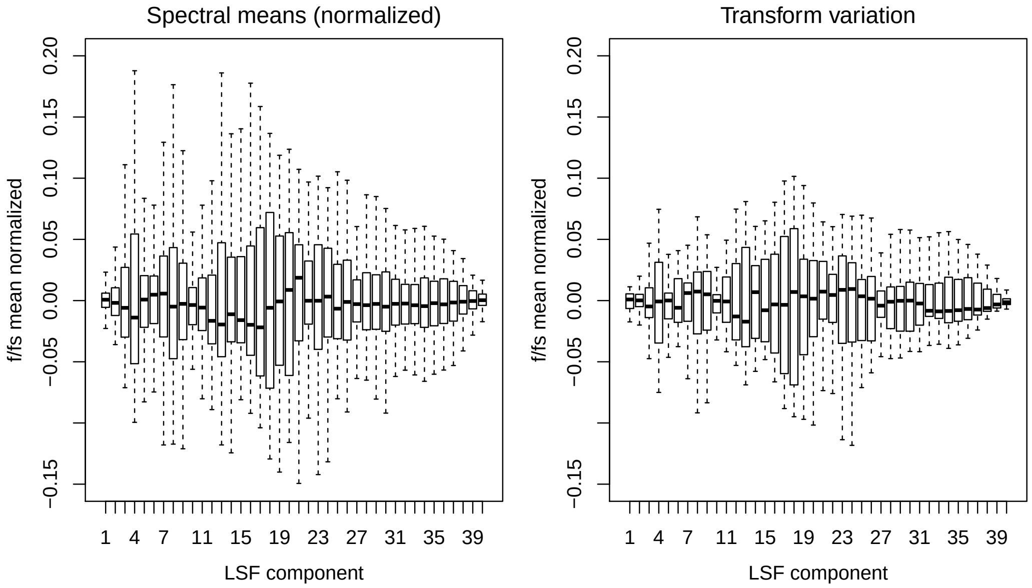

Figure 3.3: *Formant control: (left) the natural variation in the residuals (acoustic means) of the formant-controlled system is smaller than the natural variation of the acoustic means of the uncontrolled system (Figure 3.2), the variation induced by the transformations is larger than the variation of the residuals and more comparable to the variation in Figure 3.2 (right)*

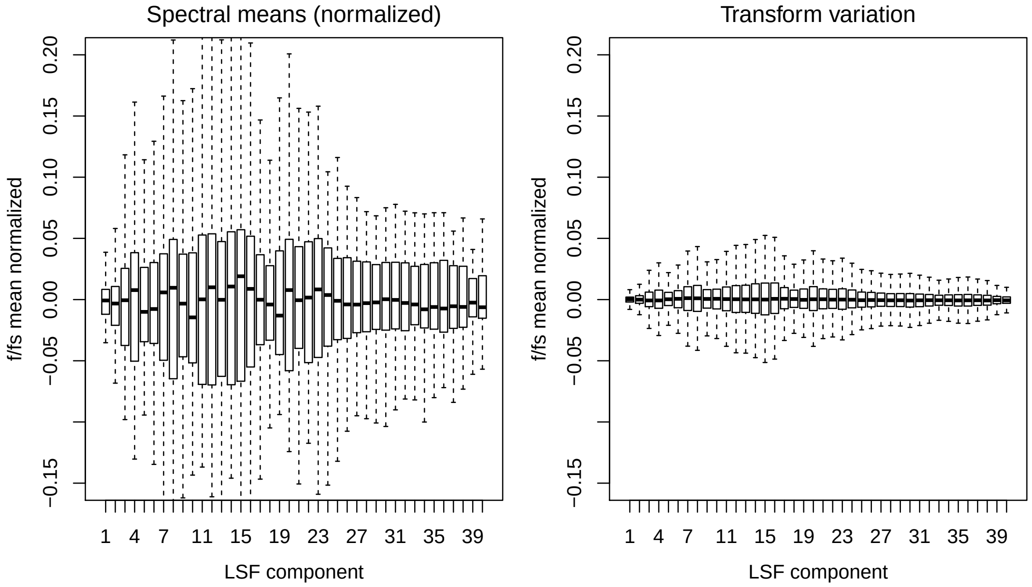

Figure 3.4: *Visual control (PCA, 30 components): The variation in the means (left) is much larger than the induced variation of the transformation functions due to the application of visual features (right) mean-acoustic-visual30pca*

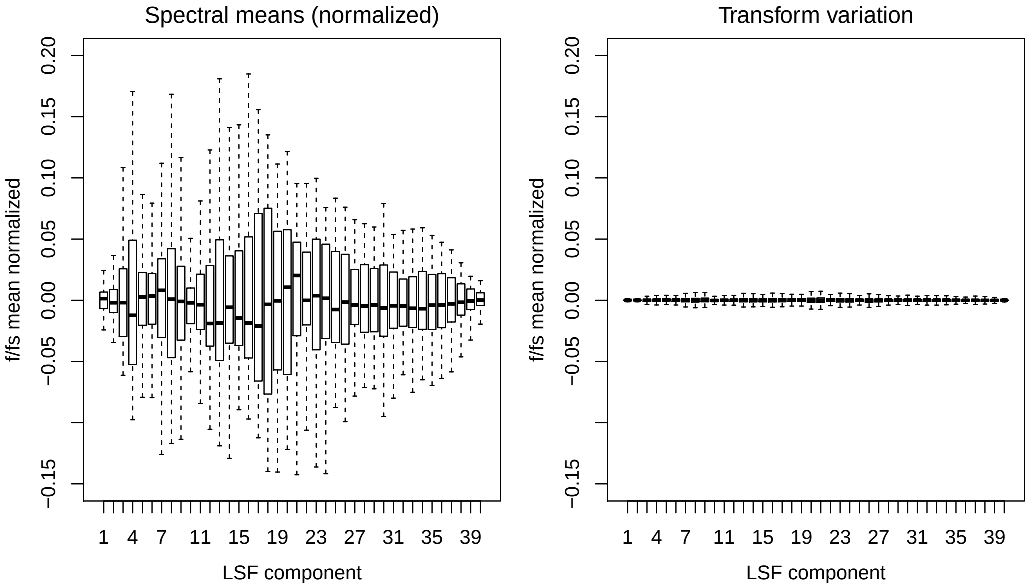

Figure 3.5: *Visual control (PCA, 5 components): the variation in the mean residuals (left) is comparable to the means in the uncontrolled case, also the variations induced by the transformations (right) are very small.*

in the subset was used as control input for each of the transformations. The resulting transformed vectors - corresponding to acoustic change vectors - were pooled and the variation calculated in accordance to the calculation of the naturally occurring variation. Because the control data represents the naturally occurring variation of the visual features and possible transformation functions, expressive control should result in acoustic features with similar variation compared to the variation within the recorded acoustic representations.

In Figure 3.3 the mean residuals (left) and the corresponding induced changes (right) are shown. The mean residuals in Figure 3.3 are much smaller than the mean variation in Figure 3.2. This hints at a strong modelling (i.e. smaller residuals) of the acoustic features by the formant features. Also the variation induced by the transformations in the formant control system is quite large compared to 1.) the variation of the residuals (acoustic means) and 2.) the variation of the means in the uncontrolled system.

In comparison Figure 3.4 shows mean variations (left) of the visual control system (using 30 PCA components) and the variation induced by control (right). The variation of the residuals is slightly different than in the formant control case which may be attributed to a different alignment caused by a different control feature stream (formant vs. visual). Also the variation induced by the transformations is much smaller than the residual variation and also smaller than the variation induced by the transformations in the formant control system.

Because this 30 dimensional visual PCA space is unintiutive and difficult to control, the dimensionality was further reduced by reducing the number of principal components from 30 to 5 components (for an investigation of the quality implications of the number of visual components see [54]). This led to a substantial decrease of the variation induced by the transformations (Figure 3.5, right). This may indicate *a*) that the principal components do capture the features that contribute most to the variation but not the most differentiating features and *b*) that the correlations between the visual features and the acoustic features were mostly random - explaining why the additional PCA components helped in modelling the acoustic features. The modelling for control apparently worked in the formant control case but not in the visual control case, raising the question of the difference in the control data between formant data and visual data.

3.4 Control data clustering

Another necessary but not sufficient condition for control is, that the control feature configurations for different target configurations differ. For example if we want to change the acoustic features of /eː/ to be more like the acoustic features of /oː/ by changing the control features from the visual appearance of /eː/ to the visual appearance of /oː/ the corresponding visual representations have to be distinct. The next step to find out why the control did not work as expected was a closer look at the control features and their clustering with respect to the different phones.

3.5 Clustering of formants

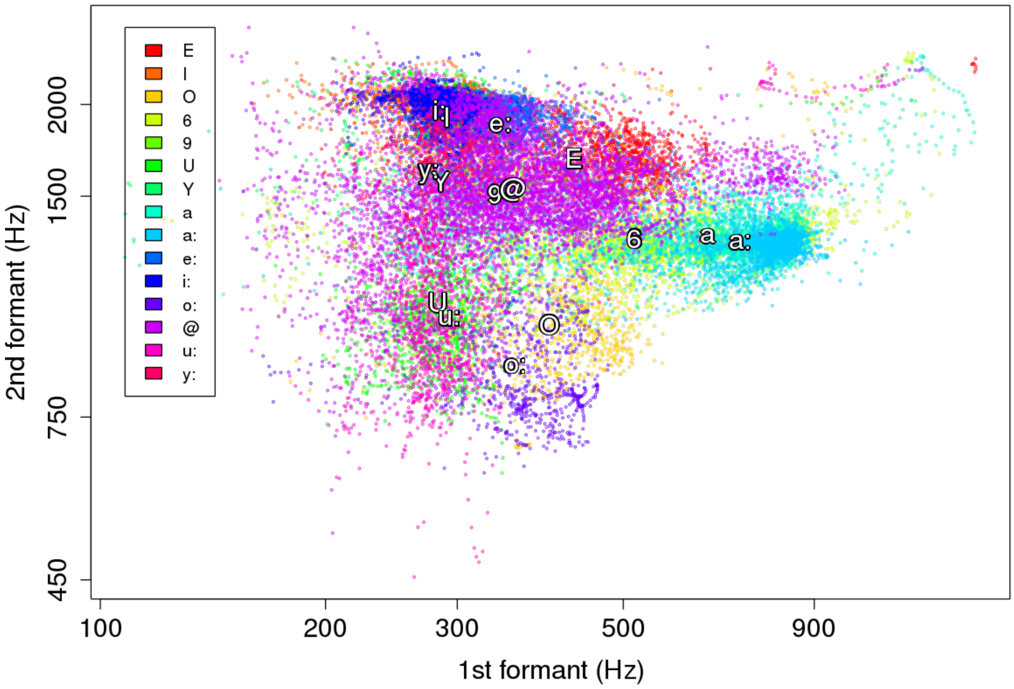

Figure 3.6: *Plot of 1st vs. 2nd formant of the phones in the training corpus. Phone labels are placed on the medians.*

To get a rough indication of the clustering of the control data, the training data corresponding to the vowels was collected from the recordings of the corpus. Figure 3.6 shows a scatter plot illustrating the 1st and 2nd formant for the data vectors of the vowels in the corpus . Different

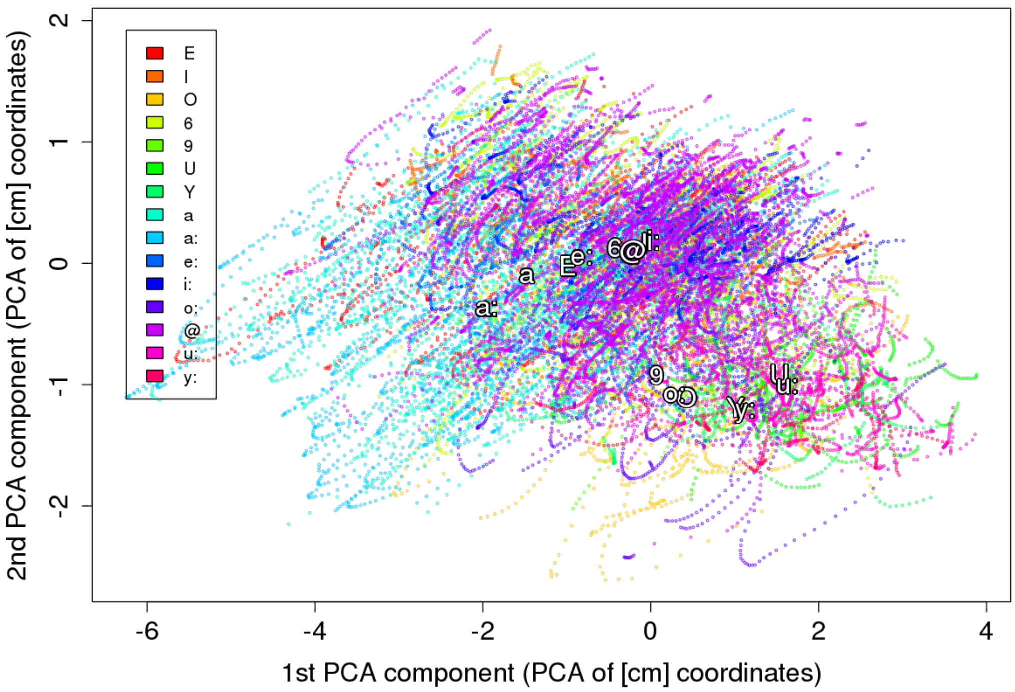

Figure 3.7: *Plot of the forced-aligned visual data (1st and 2nd PCA component) for the vowels in the training corpus. Rounded and unrounded vowels seem to cluster, other clusterings are not obvious and variation and overlap are quite large.*

colours were used for the different phones. Because of the overlap of the training vectors not all data points are visible. The plot is not meant as an illustration of the formants for certain vowels but rather as an indication of the overall clustering in the formant control space. The phone label markers are placed on the medians of the respective data sets. While the regions of the data points for different phones do overlap considerably, the medians are quite distinct and a general clustering can be perceived. Some overlap may also be attributed to coarticulation effects.

In contrast to this Figure 3.7 shows the same data for the first and second visual PCA component. There is no obvious clustering of the visual features with regard to the different phones. Also the variation of the data vectors is much larger and the overlap greater. What is interesting to note is the clustering of the unrounded vowels (a:,a, E, e:, 6, @, I, i:) and the rounded vowels (9, o:, O, Y, y:, U, u:). Roundedness or more accurately protrusion, is more adequately captured by the reduced marker set used in later experiments and the associated PCA space, see Section 3.14.

3.6 EMA data

The difference between the visual features and the formant features is quite large. The visual features do not capture the difference between different vowels nearly as precisely as the formant features do. Since control was shown to work with articulatory (EMA) data, which also consists of marker position trajectory recordings, the clustering of EMA data with respect to different phones was investigated. If the visual data can not be used as control data because of a lack of clustering with regard to different phones, and the articulatory Electromagnetic Articulography (EMA) data can be used, then there should be a difference between the two with regard to clustering.

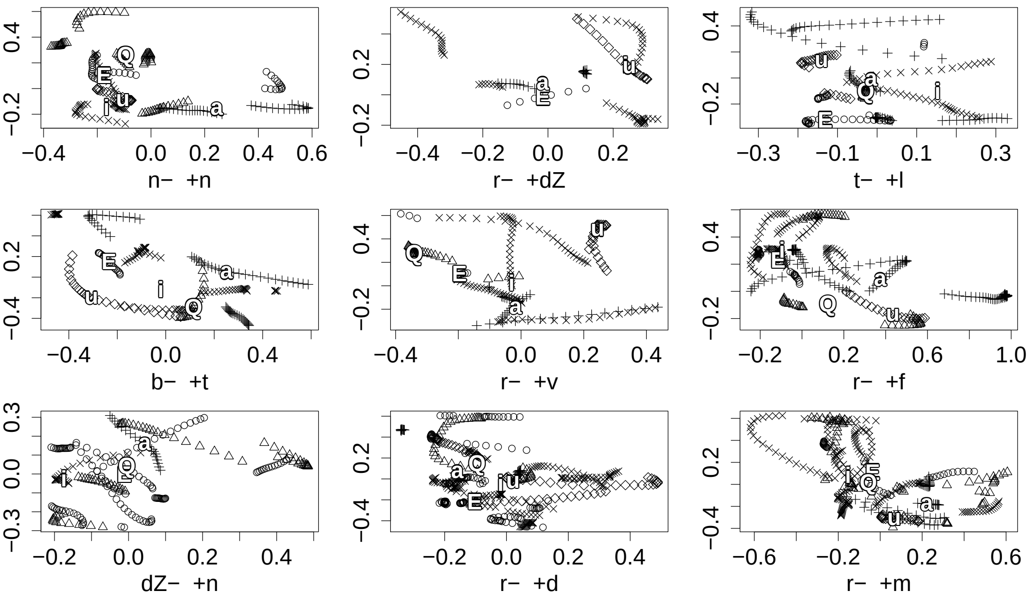

Figure 3.8: *Illustrations of the X,Y [cm,cm] dimensions of the tongue back (dorsum) marker trajectories from the mngu0 [55] corpus. Nine contexts common to the vowels i /i/ illustrated by ×, E /ɛ/ illustrated by ○, a /a/ illustrated by +, Q /ɒ/ illustrated by △, and u /u/ illustrated by ◇ (i,ɛ,a,ɒ,u) have been selected. Each label in the plot is placed on the respective median of the set of all trajectories of the given context.*

In Figure 3.8 marker trajectories of EMA data are illustrated. The data has been taken from the *mngu0* corpus [55]. The mngu0 corpus consists of multiple sources of articulatory data acquired from a single speaker, Electromagnetic Articulography (EMA) data has been used in this experiment. The subset of the corpus used is the *EMA day 1* subset, which consists of more than 1300 recorded utterances. Position trajectory recordings of markers at the upper lip, lower lip, tongue tip, tongue body and tongue back are available.

The trajectories chosen for illustration are the X,Y coordinates of the tongue back (dorsum) marker. Again forced-aligned training data was collected for the corresponding phones. The investigated vowels present in the mngu0 corpus were i,ɛ,a,ɒ,u. Because coarticulation can be assumed to influence the position of the markers with respect to certain phones, common triphone contexts for these five vowels were selected from the corpus. The contexts - preceding and succeeding phones - were ordered by the difference between the most often occurring center phone and the least often occurring centerphone. For example the difference between the number of times the context `r-E+dZ` and the context `r-a+dZ` occurred in the corpus. The nine triphone contexts with the smallest difference were selected for visualisation. Although the data has been restricted to similar contexts and only five center phones, the clustering is not as apparent as in the formant case. Consider the placement of the phone label markers in the different context plots. It is difficult to attribute different regions of the space to different phones and it is difficult to attribute the data vectors of a single context to certain phones as well. Thus the indication at this point was that the EMA data and the visual data are not substantially different and that control using visual data may still be feasible.

3.7 Restriction of markers

Since the previous control attempts were unsuccessful and the 30 dimensional PCA vectors are difficult to interpret a subset of the markers was selected. Marker data was restricted to the markers *Lower Lip*, *Upper Lip*, and *Jaw*. From each of these markers only the 2 dimensions Y and Z corresponding to up-down and front-back were chosen resulting in 6 dimensional visual data. To visualise data from these 6 markers as a point-cloud of all facial markers a linear-regression from the 6-dimensional marker subspace to the full marker space was estimated on the training data. While this is associated with some loss of quality, no obvious artefacts were spotted in the resulting visual synthesis.

3.8 Restriction to /aː eː iː oː uː/

Since a difference in the control data for different target configurations is necessary and since the EMA data indicated that control may be possible the idea was to relax the problem by restriction to a subset of the phones. It is difficult to assess whether high-dimensional data vector representations are actually distinct from each other as was required from the visual control data vectors of the different vowels. To investigate whether the visual representations of the vowels are distinct, linear Support Vector Machine (SVM) classification was used. A linear-kernel SVM was trained on the forced-aligned visual data corresponding to the phones using the phones as classes. Then the training data was classified and the percentage of correctly classified instances interpreted as a measure for linear separability. The most simple case for the classifier is to assign all phones to the same, that is, the largest class. As a measure for separability the percentage of correctly classified instances exceeding the chance of belonging to the largest (number of training vectors) was used. Multiclass classification was done based on the *one-against-one* approach and the *LibSVM* implementation [56] was used. Since SVMs are a well established technique in machine learning and they were only used for a very small part of the experiments, no further explanation is given in this work. Information on the principles underlying SVMs are among other sources given by Bishop [36].

Phones	above chance	chance	correct
/aː, eː, iː, oː, uː/	0.430	0.319	0.749
/aː, uː/	0.295	0.682	0.977
/aː, iː, uː/	0.450	0.450	0.900
/eː, uː/	0.432	0.546	0.978
/eː, oː/	0.353	0.615	0.968
/eː, iː/	0.194	0.574	0.768
/aː, eː, uː/	0.361	0.493	0.854
/œ, yː, iː, eː, aː/	0.351	0.393	0.745

Table 3.1: *Results of the linear separability experiment performed using SVM, showing the chance level and the percentage of correctly classified instances.*

Table 3.1 shows some separability results. The first column contains the sets of phones that were compared. The second column *above chance* gives the percentage of correctly classified instances above the chance level. The third column reports the percentage of the data instances of the largest class, i.e. the chance level, since the simplest way to perform classification is to assign all instances to the most frequently occurring class. The last column shows the percentage of correctly classified instances. The phones /aː eː iː oː uː/ were selected as a trade off between separability and expressivity. Another reason for this set of phones was the assumption that the longer phone versions (indicated by ː) were visually more stable.

3.9 Linear regression experiments

The transformation functions are tied to leaves of the linguistic tree. A small modification to the HTS system also allowed the tying of the acoustic means associated with the leaves. The formant control system that was used as the basis for these experiments only provided tools for tying the leaves of subtrees based on a certain level of tree-depth of the linguistic tree, for example, tying the leaves of each subtree after the n-th split in the linguistic tree. A tool to programmatically tie specified leaves to a certain transformation function was implemented. This was used to add transformation functions only to the leaves that were used for the restricted set of vowels. Also linear-regression transformations estimated on the forced-aligned training data were used to initialise the transformations before embedded Expectation-Maximisation was used to re-estimate the dependency-model.

This allowed for modelling the dependency within an arbitrary set of phones, and the initialisation based on the forced-aligned training data allowed for quick experimentation. This lead to the question of how well the linear regression modelled the dependency and the question whether a produced acoustic feature vector sounded *right* or whether it corresponded to noise and artefacts.

3.10 Comparing the 41 dimensional acoustic vector to monophone models

Utterances were synthesised from the dependency-model, but no meaningful change was present. The process of modifying the dependency-modelling, re-estimating the model, synthesising utterances and subjectively evaluating the resulting audio is time consuming. Because the dependency modelling did not seem to work properly, a faster way of evaluating whether changes in the dependency-modelling procedure did result in any improvements was sought. The idea was to evaluate the acoustic feature vectors more directly.

The monophone models, consisting of five states each, provided a reference of acoustic feature vectors for each of the phones. This also holds for the fullcontext models or the leaves in the linguistic tree of the tied fullcontext model. However the monophone models were assumed to be statistically more stable because of the different contexts that were used in their estimation.

Thus feature vectors were compared to all states of each monophone-model using euclidean distance. Since the acoustic features have different mean values for the different dimensions, mean and variance normalisation of each dimension were used.

In the experiments performed it appeared that the best match was not always the most conclusive one. For example the best match may have been for a certain phone but several of the next best matches pointed to a different phone. Assuming that only a certain subspace of the feature space is populated by meaningful speech sounds, the average distance of the first n best matching phones can be used as an indicator of the presence of distortions and artefacts. If there are no acoustic feature vectors similar to the produced vector it is likely that the produced vector would sound artificial or distorted.

Because it is not straightforward to establish what an acoustic feature vector sounds like, the measure described above has only been used as a rough indicator whether calculated control transformations seem to do something meaningful.

3.11 Common transformations and intercepts

In order to be able to transform from one phone to another the transformation needs to be informed enough to capture required changes of the feature space. Thus the variety provided to the estimation of the transformation must be sufficiently large. A simple way to achieve this is to include the data of all phones that should be captured by the control. This way the transformation can encode training data of all relevant phones. In the original system [6] the phones share the transformation function but have phone specific y-intercepts. This leads to different resulting points in the acoustic space for the same control vector. It however also allows for an acoustic model which is more similar to the uncontrolled model and may thus retain the uncontrolled models quality better. Since the starting points are different, transforms are only meaningful relative to their starting point and may, again depending on their starting point, yield artificial sounds for the same control features. To prevent this, the model was modified and the system enhanced, to use a single acoustic mean for all phones sharing a transformation function.

3.12 Experiments with common transformations and intercepts

A centerphone /aː eː iː oː uː/ question was used to split the linguistic tree into a subtree with dependency-modelling containing only the vowels in the set /aː eː iː oː uː/ and a subtree containing all other phones. This is illustrated in Figure 3.9). The models in the dependency subtree were tied to share a common transformation and a common acoustic mean for all of their leaves. Since the acoustic means and transformations are the same for all vowels in the control subtree, the acoustic differences stem only from the differences in the visual modelling.

Synthesis was performed with this system, but the informal subjective evaluation of the perceived vowel showed the vowel to be perceived as an /eː/ sound regardless of the phone that was actually used (i.e. /aː,iː,oː,uː/). Because of this, the problem was further relaxed to include only the phones /aː uː/ in the control space. Since the training data showed complete linear separability based on the visual features control between /aː uː/ sounds should be possible. The results however were nearly the same as for the former case. Even in this most relaxed case it seemed not to work.

This result was puzzling since previous experiments using the monophone similarity suggested that linear regression from one phone to another should work at least in this relaxed case when only two phones were used to estimate the regression. In an attempt to investigate this peculiarity the visual features, the transformation and the corresponding acoustic mean of each leaf were used to calculate the control result, untie the acoustic means and replace them by the control result. The transformations were also removed. When applying the transformation in the synthesis step, the visual trajectories are used as control parameters while in the case of applying the transformations on the model the static representation of the visual mean is used. The result of applying this transformation on the model was much better than the result of applying the transformation within the system. This also held for the case of modelling the dependency for /aː,eː,iː,oː,uː/.

3.13 Approximate trajectory generation

The previous experiment showed the feasibility of control by applying the transformations to the means in the model. However in the synthesis system the control based on the trajectories did not

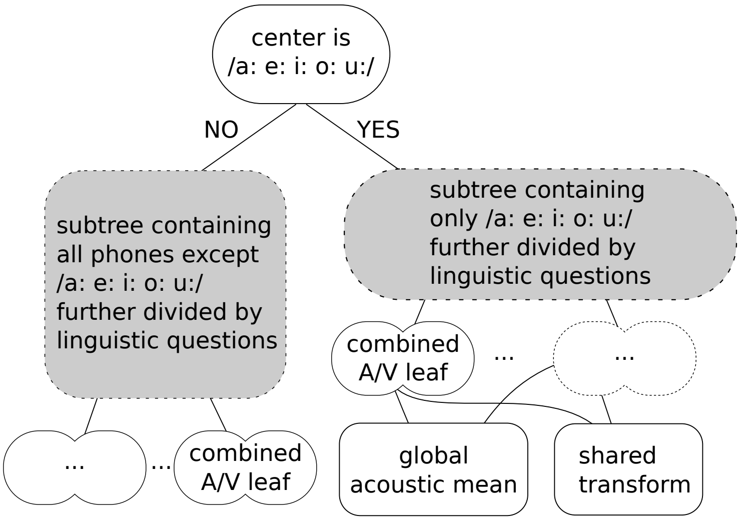
center is
/a: e: i: o: u:/
NO
YES
subtree containing all phones except /a: e: i: o: u:/ further divided by linguistic questions
subtree containing only /a: e: i: o: u:/ further divided by linguistic questions
combined A/V leaf
...
...
...
...
combined A/V leaf
global acoustic mean
shared transform

Figure 3.9: *Illustration of the /aː eː iː oː uː/ central phone question used to split the linguistic tree into a controlled (dependency) and uncontrolled subtree. In the dependency (controlled) subtree all leaves share the same transfer function and a single common acoustic mean.*

seem to work. Since it was difficult to assess what the source of the problem was - that is whether there was an implementation problem or a problem related to the application of the transformations based on the visual trajectories as opposed to the visual means - the parameter generation was implemented on a high conceptual level using linear algebra libraries. The benefit of implementing the parameter generation from scratch was the independence of possible unspecified behaviour in the original synthesis system. An approximation of the parameter generation algorithm that ignores the possible dependency of the visual features on the acoustic features was implemented as:

$$\boldsymbol{Y}_S^* = (\boldsymbol{W}_{\boldsymbol{Y}}^T \boldsymbol{U}_{\boldsymbol{Y}}^{-1} \boldsymbol{W}_{\boldsymbol{Y}})^{-1} \boldsymbol{W}_{\boldsymbol{Y}}^T \boldsymbol{U}_{\boldsymbol{Y}}^{-1} \boldsymbol{M}_{\boldsymbol{Y}} \tag{3.2}$$

instead of:

$$\begin{aligned} \boldsymbol{Y}_S^* &= (\boldsymbol{W}_{\boldsymbol{Y}}^T (\boldsymbol{U}_{\boldsymbol{Y}}^{-1} + \boldsymbol{A}^T \boldsymbol{U}_{\boldsymbol{X}}^{-1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{Z}^{-1} \boldsymbol{A}) \boldsymbol{W}_{\boldsymbol{Y}})^{-1} \\ &\cdot \boldsymbol{W}_{\boldsymbol{Y}}^T (\boldsymbol{U}_{\boldsymbol{Y}}^{-1} \boldsymbol{M}_{\boldsymbol{Y}} + \boldsymbol{A}^T \boldsymbol{Z}^{-1} \boldsymbol{M}_{\boldsymbol{X}} - \boldsymbol{A}^T \boldsymbol{U}_{\boldsymbol{X}}^{-1} \boldsymbol{M}_{\boldsymbol{X}}) \\ \boldsymbol{Z}^{-1} &= \boldsymbol{U}_{\boldsymbol{X}}^{-1} \boldsymbol{W}_{\boldsymbol{X}} (\boldsymbol{W}_{\boldsymbol{X}}^T \boldsymbol{U}_{\boldsymbol{X}}^{-1} \boldsymbol{W}_{\boldsymbol{X}})^{-1} \boldsymbol{W}_{\boldsymbol{X}}^T \boldsymbol{U}_{\boldsymbol{X}}^{-1} \end{aligned} \tag{3.3}$$

since the algorithm respecting the maximum-likelihood dependency of the visual trajectories on the acoustic models had high computational cost. The algorithm described in 3.3 was also solved using an approximation in [13] . The relevant information has been summarised in Section 2.6.1.

3.14 Exercising visual control

In order to exercise control easily the control space has to be of low dimensionality or consist of independent and meaningful dimensions. A simple way to reduce dimensionality is to use PCA. PCA was calculated on the restricted marker dimensions (Figure 3.10 shows the variance explained by the PCA components). Using PCA for dimensionality reduction also allows to visualise the associated data, Figure 3.12 shows bagplots [57] for the first, second, and third PCA component of the 6 dimensional visual data, for all vowels and diphthongs and the subset /aː eː iː oː uː/. Since the

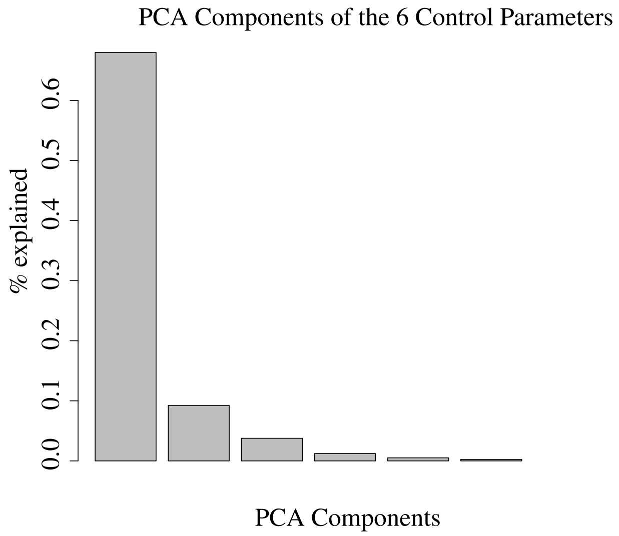

Figure 3.10: *Illustration of the percentage of variance explained by the principal components one to six*

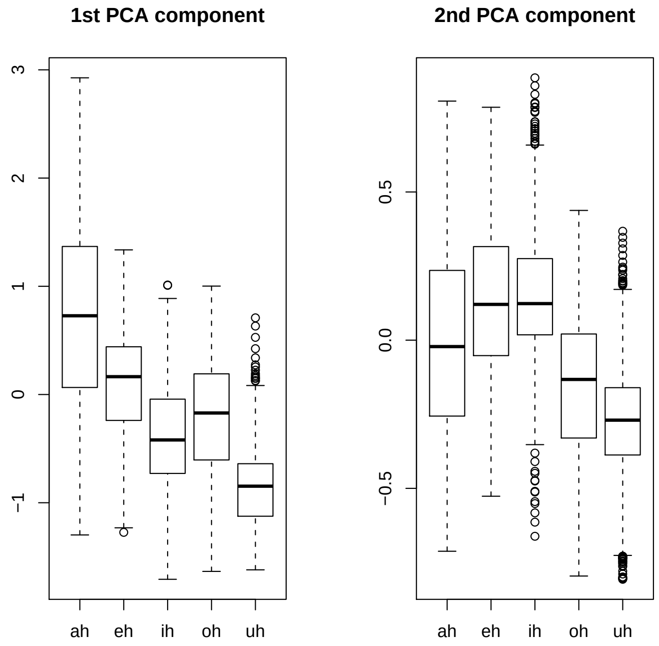

Figure 3.11: *Boxplots of the forced-aligned training vectors of the first (left) PCA component and second (right) PCA component for the vowels* aː,eː,iː,oː,uː

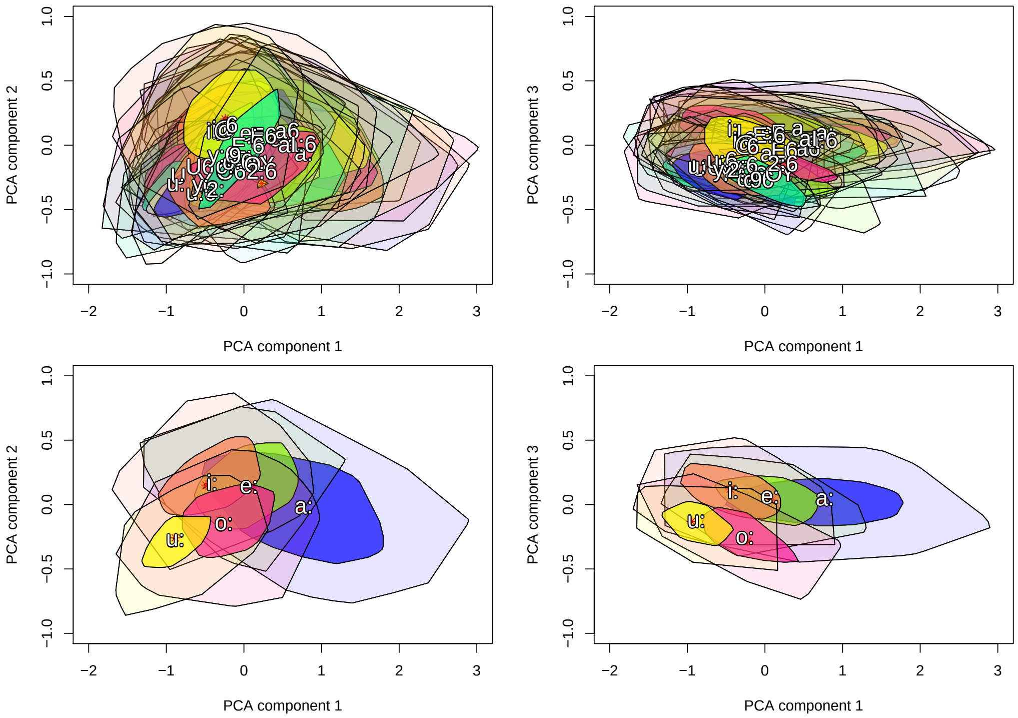

Figure 3.12: *(top) Bagplots of all vowels and diphthongs in the training corpus and of the subset /aː,eː,iː,oː,uː/ (bottom). The plot shows the first vs. the second PCA component (left) and the first vs. the third PCA component (right).*

overlap of the visual representations in the PCA1×PCA3 space is smaller than in the PCA1×PCA2 space, the former was chosen for control. Figure 3.13 illustrates the changes of the marker positions resulting from changes of the first PCA component between -1.5 and $+1.5$ and between -0.75 and $+0.75$ for the second PCA component. These can be interpreted as corresponding to mouth opening and lip rounding (or protrusion). The steps of applying control are illustrated in Figure 3.14.

3.14.1 Pitch and visual control

The *pitch* is the perceptual correlate of the fundamental frequency F0 which is the frequency of the vibration of vocal cords in voiced sounds.

Pitch variations that affect the meaning of a word are called tones [23]. A question that arises with visual control is whether visual control can also be applied to language that make use of different tones, i.e. to *tonal languages*.

If we look at our Austrian-German data of the vowels /aː eː iː oː uː/ in the control model with regard to their fundamental frequency we find that the distributions between the different vowels are very similar, see Figure 3.16, by contrast Figure 3.15 shows the variance of the spectral features for these five vowels and indicates considerable change of the LSF parameters depending on the vowel. Section 3.14 also showed that the visual representation of the selected vowels varies which is necessary (but not sufficient) to model the related acoustic changes.

If there was a significant difference of fundamental frequency for the different vowels it is possible that a simple enough control space would allow for some F0 modelling depending on

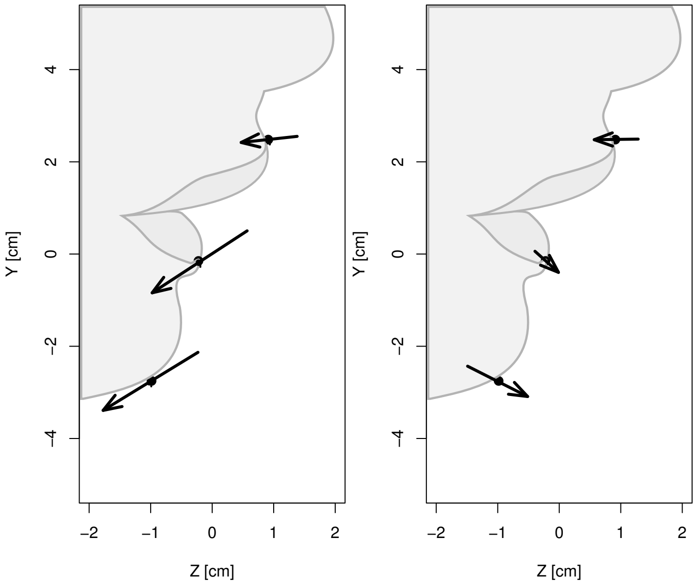

Figure 3.13: *Effects of changing the first PCA component from* -1.5 *to* $+1.5$ *(left) and changing the third PCA component from* -0.75 *to* $+0.75$ *(right).*

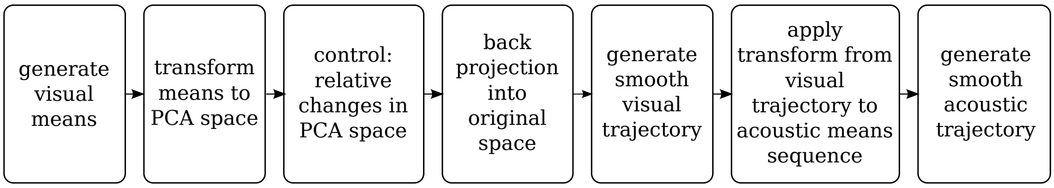
generate visual means
transform means to PCA space
control: relative changes in PCA space
back projection into original space
generate smooth visual trajectory
apply transform from visual trajectory to acoustic means sequence
generate smooth acoustic trajectory

Figure 3.14: *Illustration of the steps involved in applying visual control on the synthesised trajectories.*

the visual features – based on the relationship of the data not the physical relationship of sound production. This was also seen in the previous experiment where the vowels /oː uː/ represent *rounded back* vowels. While the roundedness and its effect on sound production can be captured by the visual features, it is unlikely that the visual features can capture this relation for the position of the tongue (i.e. front-back).

Since the source-filter model (see Section 2.1.6) models the fundamental frequency independently from the spectral features, the fundamental frequency in tonal languages can also be modelled independently. If different phones in a tonal language can be modelled by the same tones – or the tones are controlled appropriately in a different way – then there is no indication why it should not be possible to change the spectral features by visual control.

3.14.2 Selection of control space

Because of the overlapping visual representations it is necessary to limit the phones in the control space. In the previous experiment the control space was limited to the vowels /aː eː iː oː uː/, because of their more distinct visual representation (illustrated in Figure 3.12). The question arises

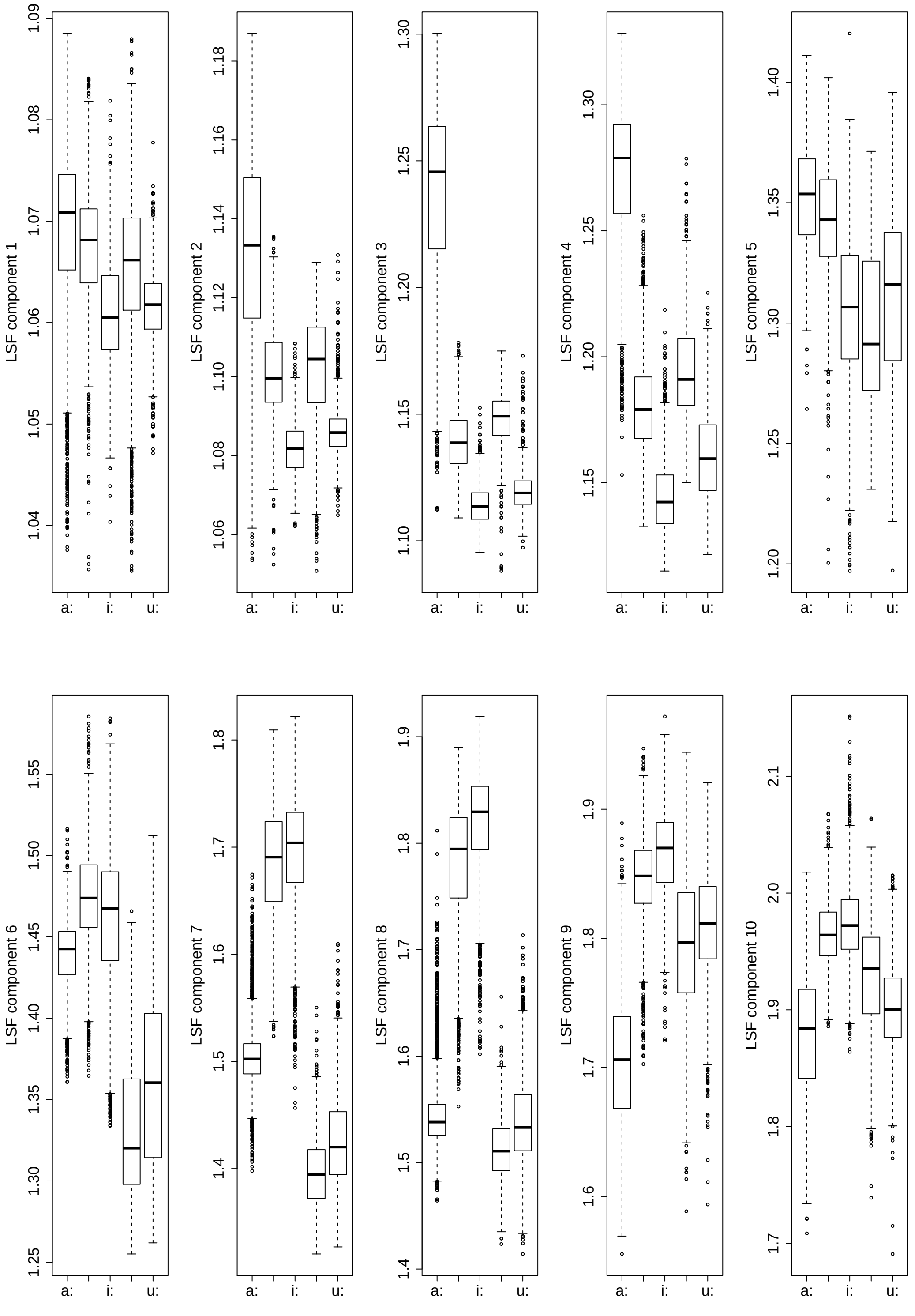

Figure 3.15: *Illustration of the variation of different LSF components for the vowels a:,e:,i:,o:,u:; the spectral LSF features (1...10) vary considerable with regard to the different vowels*

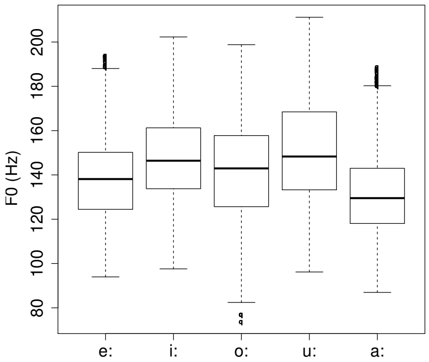

Figure 3.16: *the distribution of the F0 values for the vowel set investigated in the experiment above vary little with regard to the different vowels*

RMSE	SAMPA	IPA
0.0069	@	/ə/
0.0061	6	/ɒ/
0.0058	Y	/ʏ/
0.0031	o:	/oː/
0.0031	I	/ɪ/
0.0030	a	/a/
0.0029	a:	/aː/
0.0028	E	/ɛ/
0.0028	U	/ʊ/
0.0025	O	/ɔ/
0.0022	e:	/eː/
0.0019	u:	/uː/
0.0017	i:	/iː/
0.0015	y:	/yː/
0.0015	2:	/øː/
0.0010	9	/œ/

Table 3.2: *Linear regression over the forced-aligned training data of each of the listed phones and the corresponding RMSE. The phones are sorted by descending RMSE.*

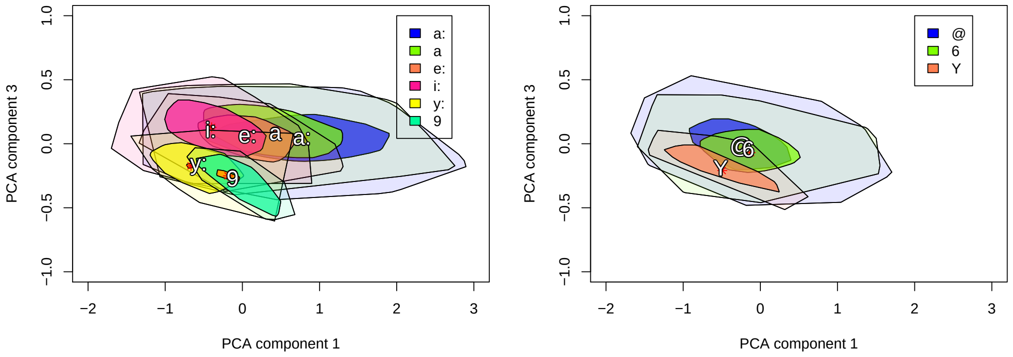

Figure 3.17: *(left) Bagplots of the visual features (PCA1×PCA3) of /a,aː,eː,iː,yː,ɘ/ and* ʏ, ə, ɒ *(right).*

RMSE single	worst	RMSE all	IPA	SAMPA	
0.0027	/yː/	0.0023	/øː,œ,eː,iː,yː/	2: 9 e: i: y:	
0.0030	/yː/	0.0026	/ɔ,eː,iː,uː,yː/	O e: i: u: y:	
0.0030	/øː/	0.0025	/ɔ,øː,œ,eː,iː,yː/	O 2: 9 e: i: y:	
0.0030	/yː/	0.0027	/ɛ,øː,œ,eː,yː/	E 2: 9 e: y:	
0.0030	/ɔ/	0.0028	/ɛ,ɔ,øː,œ,eː,yː/	E O 2: 9 e: y:	
0.0031	/ʊ/	0.0029	/ɔ,ʊ,eː,uː,yː/	O U e: u: y:	
0.0031	/a/	0.0029	/øː,œ,a,aː,eː/	2: 9 a a: e:	
0.0031	/ɔ/	0.0029	/ɔ,øː,œ,a,aː,eː,iː/	O 2: 9 a a: e: i:	
0.0031	/a/	0.0029	/a,aː,eː,iː,yː/	a a: e: i: y:	
0.0031	/a/	0.0029	/œ,a,aː,eː,iː,yː/	9 a a: e: i: y:	*
0.0031	/a/	0.0029	/øː,œ,a,aː,eː,iː,yː/	2: 9 a a: e: i: y:	
0.0031	/a/	0.0028	/øː,œ,a,aː,eː,iː/	2: 9 a a: e: i:	
0.0031	/aː/	0.0029	/ɔ,œ,a,aː,eː,iː,yː/	O 9 a a: e: i: y:	
0.0032	/a/	0.0028	/ɔ,øː,œ,a,eː,iː,yː/	O 2: 9 a e: i: y:	
0.0032	/ɪ/	0.0028	/ɪ,eː,iː,uː,yː/	I e: i: u: y:	
0.0032	/ɪ/	0.0028	/ɪ,ʊ,eː,iː,uː,yː/	I U e: i: u: y:	
0.0032	/ʊ/	0.0027	/ɔ,ʊ,eː,iː,uː,yː/	O U e: i: u: y:	
0.0032	/øː/	0.0029	/ɔ,øː,œ,a,aː,eː,iː,yː/	O 2: 9 a a: e: i: y:	
0.0032	/ɪ/	0.0028	/ɪ,ɔ,ʊ,eː,iː,uː,yː/	I O U e: i: u: y:	x
0.0037	/oː/	0.0029	/aː,eː,iː,oː,uː/	a: e: i: o: u:	*

Table 3.3: *Different sets of phones are evaluated, the forced-aligned training data are used to calculate a linear regression. The RMSE is calculated over all phones (all) and for each phone separately.The phone with the worst RMSE is listed (worst) and the corresponding RMSE (single).*

whether other combinations of phones are also possible?

The visual domain allows us to control two physical aspects of speech production, i.e. the lip rounding (or protrusion) and the jaw opening. Other aspects, as for example the frequency of the vibration of the vocal cords or the tongue position can not be modelled via the visual features. The chosen vowels in the previous experiment differ by the following criteria: mouth opening increases from iː to aː as well as from uː to oː. Also aː,eː,iː are unrounded whereas oː,uː are rounded. There is however a second difference regarding tongue shape especially between the groups oː,uː and aː,eː,iː. This difference in tongue shape can not be modelled in a physically plausible way by the visual features. However, since for the selected data, mouth opening and lip protrusion loosely correlate with changes in tongue position, the linear regression is able to model the differences. This should also hold for other sets of vowels, especially for those sets that vary less with regard to tongue shape.

In Figure 3.17 (left) a set of front-vowels with distinct visual representation is illustrated. Because of the more plausible relations between the phones, i.e. all of them are front-vowels, this set should also work well as a control space. In contrast Figure 3.17 (right) shows some phones ə ʏ ɒ (@, Y, 6) that were found to have undesirable properties that made them difficult to control by their visual representation although their representations do not seem to differ much from for example the phones aː or eː. To investigate the feasibility of different control space selections, forced-aligned training data for the phones was collected. A linear-regression was then calculated from the visual to the acoustic forced-aligned data. The overall Root Mean Squared Error (RMSE) as well as the Root Mean Squared Error (RMSE) for the data corresponding to each phone was calculated. Table 3.2 shows the respective RMSE when each phone is treated separately. A linear regression from the visual features to the acoustic features is calculated for the data of just a single phone. Notice how the errors for the phones ə ɒ ʏ are about twice as large as the next largest RMSE (for oː). In the previous separability experiment the phone /ə/ was also found to severely degrade the separability with regard to the visual domain.

Table 3.3 shows RMSEs for different sets of phones. The overall RMSE (*RMSE all*) shows the error for all observations of phones in the selected set. The error is also calculated for the data vectors assigned to each phone separately. The worst performing phone is selected (*worst*) and the RMSE is reported. The sets were constructed iteratively. Starting with a vowel, the forced-aligned training data was used to calculate a linear regression and again the overall and single-phone RMSE. Then a vowel not yet part of the set was evaluated, i.e. the new set including that vowel was used to calculate the overall and single phone RMSE. The worst single-phone RMSE was chosen as the metric for this particular set. For all evaluated vowels, the vowel that increased the single-phone RMSE the least, was added to the set. For each of the vowels an initial set was created consisting only of that vowel. The sets shown in Table 3.3 have the same or a better worst-single-phone RMSE as well as the same or better overall RMSE as the set aː eː iː oː uː. Also the sets consist of at least 5 different phones. The sets indicated with * are also illustrated in the visual domain (Figure 3.17 and Figure 3.12). Some sets for example ɪ ɔ ʊ eː iː uː yː (marked with x) seem to have a good RMSE performance, but contain combinations that are not physically plausible, for example ɔ,ʊ and yː. While the RMSE as well as the monophone similarity can give an indication of interesting candidates, conclusive results can only be obtained by a subjective evaluation so far.

Since the control only influences the spectral features, no changes of voicedness can be made. It is doubtful whether this control can be used - in a meaningful way - for changing between a consonant and a vowel. It is possible that this works if the visual representations are distinct enough and both phones are voiced - since a very distinct visual representation can model interpolation between phones and should thus allow interpolation between arbitrary (voiced) phones.

3.15 Tools & implementation

The following provides a short summary of the tools required for the abovementioned experiments.

- The HTS framework was used to train the HMM models. The training scripts were re-implemented using the WAF build system [58], a python based make-like tool. This allowed for automatic rebuilding (or retraining) of the speech models, by using the facilities WAF provided to track changed files. Also setup of parallel builds was eased because the chain of commands could be specified by a python script.
- In order to perform the experiments described in this chapter, the data models generated and used by the HTS toolkit had to be analysed. Python scripts have been implemented to load, modify and store the HMM models.
- This also facilitated the implementation of the parameter generation algorithm which was implemented on a high conceptual level using python libraries to solve the linear equations [52, 59].

CHAPTER 4

Evaluation & Findings

As outlined in Section 3.12 the control space was limited to the vowels /aː eː iː oː uː/ and transforms were added for these vowels only. Then the acoustic means for all leaves in the control sub-tree were tied (see Figure 3.9). Thus only the visual features of the different leaves varied.

Utterances were synthesised using this system. The differences in the acoustic trajectories were thus derived only from the variation in the visual features of the leaves. The acoustic trajectories were generated by an approximation of the parameter generation algorithm (see Section 2.6.1) modified for control (see Section 2.6.5).

Results obtained by Schabus et al. [38] indicate that state synchronous audio-visual modelling does not decrease acoustic synthesis quality. The visual-to-acoustic control model evaluated in this experiment, required restrictions of the control space as well as restrictions with regard to the complexity of the model. In preliminary experiments conducted during system construction, this simpler model appeared to suffer from reduced acoustic synthesis quality. Because the control is restricted to a subset of vowels and quality reductions are expected, no subjective evaluation of acoustic synthesis quality has been performed, instead the experiment focuses on the evaluation of the control modelling.

In order to evaluate whether the modelling does indeed capture the audio-visual relationship in a meaningful and sufficient way, a subjective evaluation was performed. The evaluation, the experiment and the discussion of the results have also been published in [60].

4.1 Experiment setup

To evaluate the effectiveness of control German pseudowords were synthesised. Samples with and without modification of the vowels in the pseudowords were generated. For each vowel in the control set two German pseudowords were used: “bama, pata, beme, pete, bomo, poto, bimi, piti, bumu, putu” and embedded in the carrier sentence “Ich habe . . . gehört.” (I heard . . .). The control was applied by changing the associated visual feature means of the vowels in the pseudowords. These means were transformed into the PCA1×PCA3 space (described in Section 3.14), relative changes were performed, and the modified vectors were transformed back to the original visual space. No dimensionality reduction was performed. This modified sequence of visual means was then used to generate the smooth visual trajectories. The transformation functions were then used to

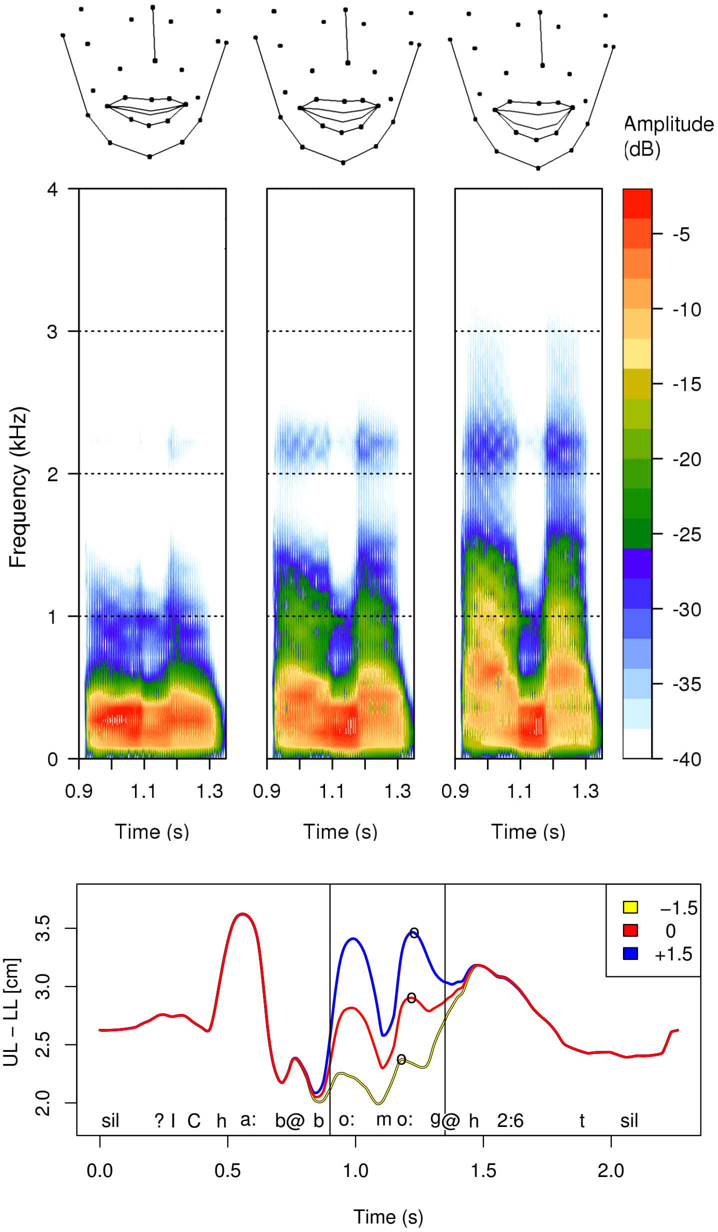

Figure 4.1: *Example outcome of modification of the first PCA component. Top: facial marker configuration at the time points indicated by small circles. Middle: spectrograms for the time segment indicated by vertical lines. Bottom: distance between the upper lip and lower lip markers over time.* [60]

overlay changes onto the acoustic mean sequences and smooth acoustic trajectories were generated from the modified acoustic mean sequences.

For each pseudoword, several versions with different modifications of the visual features were created. The first PCA component - interpreted as mouth opening - was varied by $\{-1.5, 0, +1.5\}$ and the second PCA component - interpreted as lip protrusion - was varied by $\{-0.75, 0, +0.75\}$, resulting in nine different modifications for each vowel - also including the unmodified case. These nine modifications have been applied to each of the sentences, i.e. each one of the pseudowords has been embedded in the carrier sentence once.

A subjective evaluation was performed with eleven participants, five speech experts and six non-experts, aged 18-59 years, all native Austrian-German speakers. Each participant had to listen to ninety different samples which were presented in a different randomised order.

Five vowels × two pseudowords × nine modifications lead to ninety different samples. For each sample, the participants were presented with six possible choices - the five vowels plus an additional no-match choice. The result of this evaluation is a table containing the number of times the participants perceived a certain vowel (or the no-match choice) for each of the samples. The two different pseudowords are combined, thus leading to twenty-two perception results for each base-vowel and each of the corresponding modifications.

Figure 4.1 illustrates the modification of the carrier sentence “Ich habe bomo gehört”. The bottom part illustrates the mouth opening – or the related euclidean distance of the upper lip marker and the lower lip marker over the whole duration of the utterance. Different trajectories corresponding to: a decrease, no modification, and an increase of the first PCA component are illustrated. For each variant a short time-window corresponding to the word “bomo” is selected and displayed as a spectrogram in the upper part of Figure 4.1. This time window is indicated by vertical lines in the plot illustrating the mouth opening. The time point, i.e. the frame, of widest mouth opening for the second vowel in “bomo” is marked by a small circle. The point cloud of the visual features of the frame is displayed above the spectrograms. Note that this point cloud is not synthesised from the full marker set but instead the full marker set is estimated from a linear regression using only the restricted marker set. The full marker set is not modelled in this system.

4.2 Results

Table 4.1 lists the results for each vowel and each modification - the results of the two different pseudowords are combined. The original vowel is listed in the first column, followed by the applied modification of the first (Δ_1) and third (Δ_3) PCA component. The percentages of cases a certain sound was perceived by the listeners are described in the remaining columns.

The results are also illustrated graphically in Figure 4.2. The plot consists of a group of stacked barplots for each of the original vowels (aː eː iː oː uː) and each group consists of nine stacked barplots corresponding to the nine different modifications. The orientation of the barplots layout is in accordance with the illustration of the PCA space in Figure 3.12. The middle stacked barplot corresponds to the unmodified case. The left and right columns of stacked barplots correspond to a decrease, respectively increase, of the first PCA component, which can be interpreted as mouth opening. The upper and lower rows of stacked barplots correspond to an increase, respectively a decrease, of the third PCA component. The third PCA component can be interpreted as modelling protrusion or lip rounding. See Figure 3.13 for an illustration of changes to the markers resulting from a modification of the first and third PCA components.

The following hypotheses were used to test the significance of the evaluation results. The two pseudowords corresponding to the same base vowel are combined. For each base vowel, the eight

Table 4.1: *Evaluation Results: identification percentages for each initial vowel, modified by each of nine control offset combinations [60]*

V	Δ_1	Δ_3	a	e	i	o	u	?
a	-1.5	+0.75	0	77.3	0	0	0	22.7
a	0	+0.75	4.5	72.7	0	0	0	22.7
a	+1.5	+0.75	9.1	40.9	0	0	0	50.0
a	-1.5	0	22.7	36.4	0	0	0	40.9
a	0	0	63.6	0	0	0	0	36.4
a	+1.5	0	72.7	0	0	0	0	27.3
a	-1.5	-0.75	0	0	0	68.2	0	31.8
a	0	-0.75	4.5	0	0	54.5	0	40.9
a	+1.5	-0.75	36.4	0	0	18.2	0	45.5
e	-1.5	+0.75	0	27.3	72.7	0	0	0
e	0	+0.75	0	100.0	0	0	0	0
e	+1.5	+0.75	0	100.0	0	0	0	0
e	-1.5	0	0	0	90.9	0	9.1	0
e	0	0	0	68.2	22.7	0	0	9.1
e	+1.5	0	4.5	95.5	0	0	0	0
e	-1.5	-0.75	0	0	45.5	0	50.0	4.5
e	0	-0.75	0	0	13.6	0	72.7	13.6
e	+1.5	-0.75	13.6	31.8	0	31.8	18.2	4.5
i	-1.5	+0.75	0	4.5	86.4	0	9.1	0
i	0	+0.75	0	95.5	4.5	0	0	0
i	+1.5	+0.75	0	95.5	4.5	0	0	0
i	-1.5	0	0	0	95.5	0	4.5	0
i	0	0	0	31.8	50.0	4.5	13.6	0
i	+1.5	0	0	81.8	4.5	9.1	4.5	0
i	-1.5	-0.75	0	0	54.5	0	40.9	4.5
i	0	-0.75	0	0	40.9	0	50.0	9.1
i	+1.5	-0.75	0	50.0	0	27.3	18.2	4.5
o	-1.5	+0.75	0	63.6	13.6	0	13.6	9.1
o	0	+0.75	0	86.4	0	0	0	13.6
o	+1.5	+0.75	4.5	86.4	0	0	0	9.1
o	-1.5	0	0	0	4.5	0	90.9	4.5
o	0	0	0	18.2	0	72.7	9.1	0
o	+1.5	0	72.7	9.1	0	13.6	0	4.5
o	-1.5	-0.75	0	0	0	0	100.0	0
o	0	-0.75	0	0	0	27.3	72.7	0
o	+1.5	-0.75	4.5	0	0	90.9	0	4.5
u	-1.5	+0.75	0	0	95.5	0	4.5	0
u	0	+0.75	0	22.7	68.2	0	0	9.1
u	+1.5	+0.75	0	81.8	13.6	0	0	4.5
u	-1.5	0	0	0	27.3	0	72.7	0
u	0	0	0	0	22.7	0	63.6	13.6
u	+1.5	0	0	27.3	4.5	9.1	50.0	9.1
u	-1.5	-0.75	0	0	0	0	90.9	9.1
u	0	-0.75	0	0	0	0	90.9	9.1
u	+1.5	-0.75	0	0	0	0	95.5	4.5

modification cases are compared against the respective unmodified case:

$$\begin{aligned} H0 &: \text{perception of unmodified and modified case is the same} \\ H1 &: \text{perception of unmodified and modified case differ} \end{aligned} \tag{4.1}$$

Thus for each base vowel this hypothesis is tested eight times. Significant change with respect to the unmodified case is indicated in Figure 4.2 by a black border.

Each modification case as well as the unmodified case are tested with regard to the vowel that was perceived most often. This is done for each base vowel, leading to five$\times$nine $= 45$ tests of the hypothesis:

$$\begin{aligned} H0 &: \text{no vowel was perceived significantly more often than any other vowel} \\ H1 &: \text{some vowel was perceived significantly more often than any other vowel} \end{aligned} \tag{4.2}$$

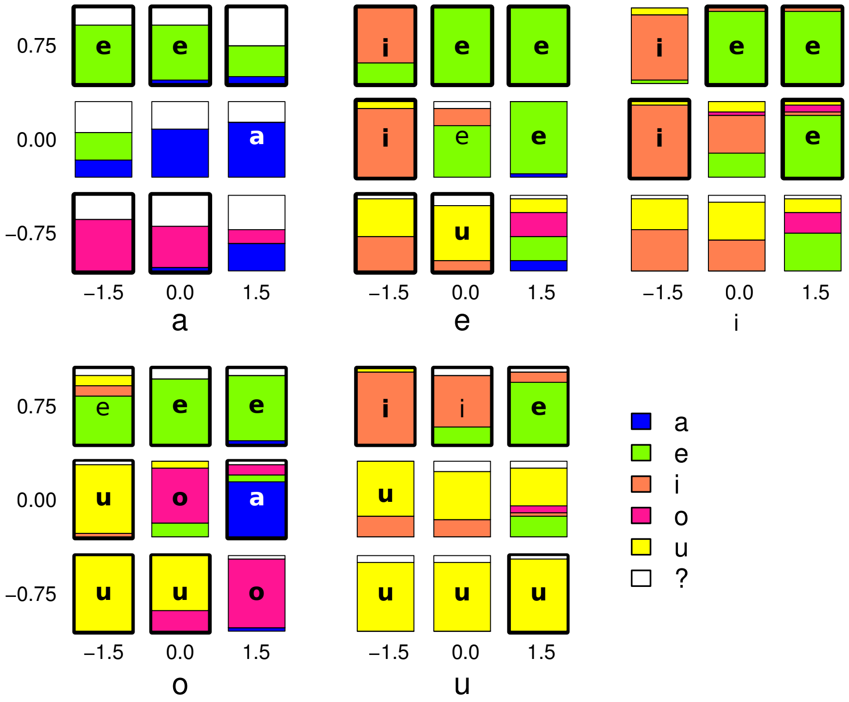

Figure 4.2: *Visualisation of the evaluation results (Table 4.1). For each initial phone, the central subplot shows the classification results for the unmodified phone. The eight surrounding subplots show the classification results for the modified phones. Colours and orientation are in line with Figure 3.12. The rejection of Hypothesis* 4.1 *(H0) (modified and unmodified case differ) is indicated in Figure 4.2 by a black border. The acceptance of the hypotheses (H1) corresponding to the vowel perceived in the majority of cases (Hypothesis* 4.2*) and the vowel perceived in more than* 50% *of the cases (Hypothesis* 4.3*), are indicated by the corresponding vowel name printed in the middle of the bar plot in normal and bold font respectively.* [60]

Rejection of $H0$ is indicated in Figure 4.2 by plotting the vowel perceived in the majority of cases in the middle of the corresponding bar plot.

The following slightly stronger requirement is also tested 45 times:

$$\begin{aligned} H0 &: \text{no vowel was perceived significantly more often than in } 50\% \text{ of the cases} \\ H1 &: \text{a vowel was perceived significantly more often than in } 50\% \text{ of the cases} \end{aligned} \quad (4.3)$$

Rejection of $H0$ is indicated in Figure 4.2 in the same way as the previous hypothesis, but using a bold font instead. In most cases when Hypothesis $H0$ 4.2 is rejected, Hypothesis $H0$ 4.3 is rejected as well.

4.2.1 Details

The evaluation results can be modelled by a multinomial distribution:

> The multinomial variate is a multidimensional generalization of the binomial. Consider a trial that can result in only one of k possible distinct outcomes, labeled $A_i, i = 1, \ldots k$. Outcome A_i occurs with probability p_i . The multinomial distribution relates

> to a set of n-independent trials of this type. The multinomial multivariate is $\boldsymbol{M} = [\boldsymbol{M}_i]$, where $\boldsymbol{M}_i$ is the variate "number of times event A_i occurs," $i = 1, \ldots, k$. The quantile is a vector $x = [x_1, \ldots, x_k]$. For the multinomial variate, x_i is the quantile of $\boldsymbol{M}_i$ and is the number of times event A_i occurs in the n trials. [61]

The abovementioned hypotheses can be tested using confidence intervals of the parameters of the multinomial distributions, i.e. the category probabilities. A Chi-squared homogeneity test can not be performed because the required minimal cell count of each choice category in each sub-evaluation (≥ 5) is not fulfilled. An R implementation by Villacorta [62] based on the method of Glaz and Sison [63] was used to calculate the confidence intervals.

Hypothesis 4.1 H0 is rejected if, for at least one category, the confidence intervals of the category probability parameters do not overlap, i.e. for at least one category the probability parameters in the modified and unmodified case differ significantly.

Hypothesis 4.2 H0 is rejected if the lower tail of the confidence interval of the most frequently perceived vowel's probability parameter is above the upper tails of the respective confidence intervals of the remaining categories' probability parameters.

Hypothesis 4.3 H0 is rejected if the lower tail of the confidence interval of the most frequently perceived vowel's category probability is strictly above 50%.

The calculated confidence intervals are given in Table A.1.

4.3 Interpretation

- Figure 4.2 shows that in most cases there is a clear majority regarding the perceived vowel.
- In general the results of perception are consistent with the expected changes with regard to the control space (see Figure 3.12 for an illustration of the control space).
- For all five vowels there is a direction of modification where the perceived vowel is more clearly identified. This direction can be interpreted as moving away from the other vowels and thus from regions with overlapping visual representations.
- The vowel perception did not reach a statistically significant majority in some cases as for example in the unmodified cases for the vowels /a i u/. While the majority was not significant the mode still corresponds to the underlying vowel, i.e. an unmodified /i/ vowel was still perceived as /i/ in the majority of cases. This less clear perception is attributed to overlapping visual features.
- The overlapping visual features do not necessarily reduce the quality of the synthesis system since *a*) the uncontrolled variant could be used for synthesis when available and *b*) better mappings may mitigate this effect.
- In a case where a vowel was perceived in the majority of cases, but the overall result was not significantly different from the unmodified case, the produced sound has become more distinct and the mode of the percentage of observed vowels has not changed. An example for this is the modification of the base vowel /u/ when changing the 1st PCA component by -1.5 and the 3rd PCA component by -0.75.

- In the case of /a/ there also is a fairly large number of "no match" votes ("?") in all nine cases. Part of this is assumed to be due to acoustic artefacts (e.g., buzzing or distortions regarding amplitude). These artefacts can be attributed to leaving the area of the PCA space in which observations are naturally occurring and thus creating artificial visual features and inducing artificial sound.

- In a case with significant change (indicated by a black border in Figure 4.2) and a vowel perceived with significant majority (indicated by a vowel name overlaid on the bar plot in Figure 4.2), the control *changed* the perceived sound and the majority indicates a *meaningful change*, see for example the modification of /a/ when changing the 3rd PCA component by 0.75.

- In Figure 4.1 the spectrograms corresponding to the vowels show a shift of the energy distribution towards higher frequencies. This shift corresponds to increased mouth opening. The evaluation results show perception of /uː/ in case of decreased and /aː/ in case of increased mouth opening. The formant frequencies are expected to increase from /uː/ towards /aː/ thus showing that the change in the spectrogram is consistent with the perception results.

4.4 Summary of findings

This section provides a short summary of the findings from the subjective evaluation as well as from the experiments described in the previous chapter.

Visual control of acoustic features can be achieved on a restricted set of vowels. If the visual data are restricted appropriately, for example by limiting the involved data to a certain set of phones, the visual features exhibit more distinct representations and meaningful mappings from the visual to the acoustic features are possible.

The visual-to-acoustic relationship appears to generalise less well than the formant-to-acoustic relationship. The formant-to-acoustic control is discussed by Lei et al. [6]. While local variations appeared to be sufficient for the formant features, this could not be reproduced for the visual features. In other words, the visual features of much different contexts had to be combined in order to retrieve transformations that allow for changes in the quality of the sound and the visual features corresponding to different vowels are far more overlapping.

EMA and visual data seem to be similar with respect to smoothing and overlapping. The EMA data trajectories and the visual marker trajectories seem to be similar with respect to smoothness and overlap regarding different vowels, though the EMA data are more expressive with respect to the acoustic features since the tongue position is also captured.

Distinct enough visual representations model uncaptured dependencies. If the visual representation is distinct enough, relationships that are not captured by the visual features directly can be modelled, for example the vowels /aː eː iː/ and /oː uː/ differ in tongue position which is obviously not captured by the visual features.

The mapping is susceptible to favour more often occurring vowels. Since the mapping does not weigh the training vectors according to their acoustic representation, the result of a mapping will be biased towards the most frequently occurring phone (largest number of training vectors) if the visual representations overlap.

Control using visual features is possible, but overlap decreases quality. The experiments showed that visual features can indeed be used to change the perceived vowels if the control space is limited accordingly. Evaluations showed that vowel perception was less clear in regions with overlapping visual representations, but that the majority of participants still perceived the respective vowels correctly.

Intuitive control features were derived from PCA of the restricted set of visual markers.

Simple control features and trajectory smoothing hinder lip closure. In some examples (e.g., the one in Figure 4.1) the closure between the two modified vowels was not synthesised correctly in the visual domain as the smooth trajectory generation prevented the lip movement from reaching the closure point. This could be prevented, e.g., by decreasing the variances and thus forcing the trajectories closer to the specified means, or by modifying the dynamic features. This also indicates that PCA transformation for control alone does not capture all possible modifications of the underlying feature space adequately and a different parameterization for exercising control may be necessary.

Larger changes in control features than natural variation are required for control. The experiments also showed that the offsets applied in PCA space needed to be larger than the variation of the visual features implied (see Figure 3.12) in order to properly induce changes.

CHAPTER 5

Conclusion and Future Work

HMM based speech synthesis provides a flexible and intelligible speech synthesis system while also maintaining speaker specific characteristics. While HMM based speech synthesis is flexible, the acoustic features are complex and high-dimensional and thus difficult to modify even for an expert. The idea of control is to find a more intuitive representation and a mapping from this more intuitive representation to the acoustic feature space. Then changes applied in the intuitive space should be mapped to appropriate changes in the difficult to understand acoustic feature space, thus enabling an expert to modify the acoustic parameter trajectories in a meaningful way and further increasing the flexibility of the synthesis system. In previous work the feasibility of using articulatory data to control acoustic speech synthesis was shown. Since visual features are inherently easier to record, this work has investigated the possibility to control acoustic synthesis by visual-only features. The visual-only features consist of three-dimensional marker trajectory recordings of markers glued to the speaker's face.

Results obtained in this work indicate the feasibility of controlling acoustic speech by visual-only features in a phonetically meaningful way. The control by visual features was exercised based on PCA components which were interpreted as mouth opening and lip protrusion. The main difference compared to control using articulatory data is the lack of information regarding the tongue position. This led to less distinct features for different phones. Restricting the control to a small set of phones (/aː eː iː oː uː/) led to sufficiently distinct visual-only features, thus allowing the modelling of the dependency. Because of the distinct enough visual representation even the acoustic difference between /a: e: i:/ and /oː uː/ was modelled, which includes changes to the tongue position that are not captured by the visual features. The modelling of this difference is attributed to the corresponding difference in lip protrusion. The transformation of each of the vowels in the restricted control set to at least one other vowel by changing only the visual features was demonstrated as indicated by the results of the subjective evaluation. The evaluation results indicate significant differences to the unmodified case for many of the samples. Also for many samples perception of a certain vowel reached a significant majority, thus indicating that the result of the control modification is meaningful. Due to the restrictions required for dependency modelling, no improvements of acoustic synthesis quality with respect to the uncontrolled system are expected.

In future work a less restricted and more precise mapping could improve the quality of the controlled synthesis. The control model used in this work is rather simple. A single transform was used for the control space and only a single restricted control space was used. It would be interesting to evaluate more sophisticated control models for example taking the context into account to model the visual-to-acoustic relationship while considering the effects of co-articulation. This could be done by tying different transform functions to different states and contexts.

The control space has been selected and restricted in an informed way. A more unsupervised partitioning or restriction of the control space would be necessary for larger applications. This could be done, for example by performing acoustic-to-articulatory inversion and clustering of the spaces with different visual but otherwise similar articulatory configurations. A synchronous audio–visual–articulatory corpus would be helpful and could also be used to verify the applicability of clustering methods that ultimately rely only on the acoustic features.

The visual features used in the above experiments were minimalistic. Asymmetry was not captured since only the center markers have been used. Using all of the visual markers and additional features could help recover audio-visual relations not captured by the features used in this experiment. Additional features could for example also be lip shape features derived from the recorded images which could capture aspects that are lost in the marker based tracking.

The visual perception of the modified audio-visual speech synthesis has not been evaluated. It would be interesting to determine whether the changes in visual space and the induced acoustic changes are perceived to be consistent.

The experiments have been performed on a single speaker corpus. A broader experiment should compare the differences in visual representation with regard to different speakers.

Only LSF based acoustic features have been investigated in this experiment. Since other acoustic features besides LSF are also used for speech synthesis it would be interesting to see if this control system can also be used for other speech coding methods (see for example the comparison of different vocoder types by Airaksinen [64]).

Also LSF features and the corresponding LPC features can be used to model the vocal tract as a loss-less tube. This works well for vowels and could potentially also be used to cluster and partition the control space. However, simple LSF coding can not accurately capture sounds involving turbulent airflow as well as sounds requiring acoustic models with branched tubes (for example nasals). It would thus also be interesting to look at other coding mechanisms as well as the applicability of visual control to sounds other than vowels.

Bibliography

[1] S. Lemmetty, "Review of speech synthesis technology," *Helsinki University of Technology*, 1999.

[2] B. Juang and T. Chen, "The past, present, and future of speech processing," *Signal Processing Magazine, IEEE*, vol. 15, no. 3, pp. 24–48, 1998.

[3] R. Hoffmann, "On the development of early vocoders," in *Proc. HISTELCON'10*, 2010, pp. 1–6.

[4] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP'96*, vol. 1, 1996, pp. 373–376 vol. 1.

[5] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," in *Proc. ICASSP'95*, vol. 1, 1995, pp. 660–663 vol.1.

[6] M. Lei, J. Yamagishi, K. Richmond, Z.-H. Ling, S. King, and L.-R. Dai, "Formant-controlled hmm-based speech synthesis," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[7] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis." in *ICSLP*, vol. 98, 1998, pp. 29–31.

[8] ——, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis." in *EUROSPEECH*. ISCA, 1999.

[9] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[10] J. Yamagishi, "Average-voice-based speech synthesis," *Tokyo Institute of Technology*, 2006.

[11] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama, "Multiple-regression hidden markov model," in *Proc. ICASSP'01*, vol. 1. IEEE, 2001, pp. 513–516.

[12] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Articulatory control of HMM-based parametric speech synthesis driven by phonetic knowledge," in *Proc. Interspeech*, Brisbane, Australia, Sep. 2008, pp. 573–576.

[13] ——, "Integrating articulatory features into hmm-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[14] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Vowel creation by articulatory control in hmm-based parametric speech synthesis." in *INTERSPEECH*, 2012.

[15] A. B. Youssef, P. Badin, G. Bailly, P. Heracleous *et al.*, “Acoustic-to-articulatory inversion using speech recognition and trajectory formation based on phoneme hidden markov models,” *Interspeech 2009*, pp. 2255–2258, 2009.

[16] H. Kjellström and O. Engwall, “Audiovisual-to-articulatory inversion,” *Speech Communication*, vol. 51, no. 3, pp. 195 – 209, 2009.

[17] D. M. Hardison, “Acquisition of second-language speech: Effects of visual cues, context, and talker variability,” *Applied Psycholinguistics*, vol. 24, no. 04, pp. 495–522, 2003.

[18] V. Hazan, A. Sennema, M. Iba, and A. Faulkner, “Effect of audiovisual perceptual training on the perception and production of consonants by japanese learners of english,” *Speech communication*, vol. 47, no. 3, pp. 360–378, 2005.

[19] B. Kröger, V. Graf-Borttscheller, and A. Lowit, “Two and three-dimensional visual articulatory models for pronunciation training and for treatment of speech disorders,” in *Interspeech, 9th Annual Conference of the International Speech Communication Association*, 2008.

[20] P. Badin, A. Ben Youssef, G. Bailly, F. Elisei, and T. Hueber, “Visual articulatory feedback for phonetic correction in second language learning,” *Actes de SLATE*, pp. P1–10, 2010.

[21] L. Wang, Y. Qian, M. R. Scott, G. Chen, and F. K. Soong, “Computer-assisted audiovisual language learning,” *Computer*, vol. 45, no. 6, pp. 38–47, 2012.

[22] O. Jokisch, U. Koloska, D. Hirschfeld, and R. Hoffmann, “Pronunciation learning and foreign accent reduction by an audiovisual feedback system,” in *Affective Computing and Intelligent Interaction*. Springer, 2005, pp. 419–425.

[23] P. Ladefoged and K. Johnson, *A course in phonetics*. Wadsworth Publishing Company, 2010.

[24] D. Schabus, M. Pucher, and G. Hofer, “Building a Synchronous Corpus of Acoustic and 3D Facial Marker Data for Adaptive Audio-visual Speech Synthesis,” *Proc. LREC, Istanbul, Turkey*, pp. 3313–3316, 2012.

[25] C. Kranzler, F. Pernkopf, R. Muhr, M. Pucher, and F. Neubarth, “Text-to-speech engine with austrian german corpus,” in *International Conference on Speech and Computer, SPECOM*, Jun 2009.

[26] C. Brinckmann, “The Kiel corpus of read speech as a resource for speech synthesis,” Ph.D. dissertation, Citeseer, 2004.

[27] B. Pfister and T. Kaufmann, *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer DE, 2008.

[28] F. Zheng, Z. Song, L. Li, W. Yu, F. Zheng, and W. Wu, “The distance measure for line spectrum pairs applied to speech recognition.” in *ICSLP*, 1998.

[29] J. Rothweiler, “A rootfinding algorithm for line spectral frequencies,” in *Proc. ICASSP’99*, vol. 2. IEEE, 1999, pp. 661–664.

[30] ——, “On polynomial reduction in the computation of lsp frequencies,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 592–594, 1999.

[31] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, “Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,” *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[32] H. Kawahara, “STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds,” *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006. [Online]. Available: http://dx.doi.org/10.1250/ast.27.349

[33] P. C. Nguyen, O. Takao, and M. Akagi, “Modified restricted temporal decomposition and its application to low rate speech coding,” *IEICE TRANSACTIONS on Information and Systems*, vol. 86, no. 3, pp. 397–405, 2003.

[34] G. Bailly, M. Bérar, F. Elisei, and M. Odisio, “Audiovisual speech synthesis,” *International Journal of Speech Technology*, vol. 6, no. 4, pp. 331–346, 2003.

[35] Naturalpoint, 2013. [Online]. Available: http://www.naturalpoint.com/optitrack/

[36] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, Oct. 2007.

[37] J. Shlens, “A tutorial on Principal Component Analysis,” *Systems Neurobiology Laboratory, University of California at San Diego*, 2005.

[38] D. Schabus, M. Pucher, and G. Hofer, “Joint audiovisual Hidden Semi-Markov Model-based speech synthesis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, pp. 1–1, 2013.

[39] A. Black, P. Taylor, R. Caley, R. Clark, K. Richmond, S. King, V. Strom, and H. Zen, “The Festival speech synthesis system, version 1.4. 2,” *Unpublished document available via http://www. cstr. ed. ac. uk/projects/festival. html*, 2001.

[40] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, “The HMM-based speech synthesis system (HTS) version 2.0,” in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, 2007, pp. 294–299.

[41] L. Rabiner and B.-H. Juang, “Fundamentals of speech recognition,” 1993.

[42] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[43] H. Zen, “An example of context-dependent label format for HMM-based speech synthesis in english,” *The HTS CMUARCTIC demo*, 2006.

[44] S. Young, “The general use of tying in phoneme-based HMM speech recognisers,” in *Proc. ICASSP’92*, vol. 1, 1992, pp. 569–572 vol.1.

[45] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[46] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, “Hidden Markov Models based on Multi-Space probability Distribution for pitch pattern modeling,” in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 229–232.

[47] L. J. Rodríguez and I. Torres, “Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition,” in *Pattern Recognition and Image Analysis*. Springer, 2003, pp. 847–857.

[48] K. Shinoda and T. Watanabe, “Acoustic modeling based on the MDL criterion for speech recognition,” in *Proc. Eurospeech*, vol. 1, 1997, pp. 99–102.

[49] S. Levinson, “Continuously variable duration Hidden Markov Models for speech analysis,” in *Proc. ICASSP’86*, vol. 11. IEEE, 1986, pp. 1241–1244.

[50] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Hidden Semi-Markov Model based speech synthesis.” in *INTERSPEECH*, 2004.

[51] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in *Proc. ICASSP’00*, vol. 3. IEEE, 2000, pp. 1315–1318.

[52] E. Jones, T. Oliphant, and P. Peterson, “Scipy: Open source scientific tools for python,” *http://www. scipy. org/*, 2001.

[53] T. Tomoki and K. Tokuda, “A speech parameter generation algorithm considering global variance for hmm-based speech synthesis,” *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.

[54] D. Schabus, M. Pucher, and G. Hofer, “Objective and subjective feature evaluation for speaker-adaptive visual speech synthesis,” in *International Conference on Auditory-Visual Speech Processing (AVSP)*, Annecy, France, Sept 2013, pp. 37–42.

[55] K. Richmond, P. Hoole, and S. King, “Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus.” in *INTERSPEECH*, 2011, pp. 1505–1508.

[56] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[57] P. J. Rousseeuw, I. Ruts, and J. W. Tukey, “The bagplot: a bivariate boxplot,” *The American Statistician*, vol. 53, no. 4, pp. 382–387, 1999.

[58] “waf - the meta build system.” [Online]. Available: http://code.google.com/p/waf/

[59] T. E. Oliphant, “Python for scientific computing,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[60] J. Hollenstein, M. Pucher, and D. Schabus, “Visual Control of Hidden-Semi-Markov-Model based Acoustic Speech Synthesis,” in *Proceedings of the 12th International Conference on Auditory-Visual Speech Processing (AVSP), Annecy, France*. Inria, Sep. 2013, pp. 31–36.

[61] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley and Sons, Inc., 2011.

[62] P. J. Villacorta, *MultinomialCI: Simultaneous confidence intervals for multinomial proportions according to the method by Sison and Glaz*, 2012, r package version 1.0. [Online]. Available: http://CRAN.R-project.org/package=MultinomialCI

[63] J. Glaz and C. P. Sison, “Simultaneous confidence intervals for multinomial proportions,” *Journal of Statistical Planning and Inference*, vol. 82, no. 1, pp. 251–262, 1999.

[64] M. Airaksinen, “Analysis/synthesis comparison of vocoders utilized in statistical parametric speech synthesis,” 2012.

APPENDIX A

Appendix

A.1 Band matrix

For a band-matrix $\boldsymbol{A}$ with upper bandwidth u and lower bandwidth l the following holds for all non-zero elements:

$$a_{ij}: \;\; j-i \leq u, \; i-j \leq l \tag{A.1}$$

For the product $\boldsymbol{AB}$ of two band matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$ with bandwidth u_A, l_A respectively u_B, l_B, also $\boldsymbol{AB}$ is a band-matrix with:

$$j-i \leq u_A + u_B \text{ and } i-j \leq l_A + l_B \tag{A.2}$$

The matrix $\boldsymbol{AB}$ consists of elements $(ab_{ij}) = \sum_{k=1}^{m} a_{ik} b_{kj}$, which are potentially non-zero if: $\exists k$: a_{ik} non-zero $\wedge$ b_{kj} non-zero thus for all potentially non-zero elements ab_{ij} of $\boldsymbol{AB}$ holds:

$$\begin{gathered} \exists k, k \in \{1 \dots m\}: \\ k-i \leq u_A \;\wedge\; i-k \leq l_A \;\wedge \\ j-k \leq u_B \;\wedge\; k-j \leq l_B \end{gathered}$$

Which can be reconciled into the following form

$$\begin{gathered} j-i \leq u_A + u_B \\ i-j \leq l_A + l_B \end{gathered}$$

which by reinterpreting as the general property of bandmatrices (i.e. A.1) results in the summation of upper and lower bandwidths for the product of bandmatrices, i.e.: $u_{AB} = u_A + u_B$ and $l_{AB} = l_A + l_B$.

A.2 WUW is a banded matrix

Since $\boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{W}$ can be partitioned into $T \times T$ block matrices, each of which is in $\mathbb{R}^{Dn \times D}$

$$
\boldsymbol{W} = \begin{pmatrix}
\boldsymbol{\Omega}_0 & \dots & \boldsymbol{\Omega}_\delta & & & \\
\vdots & \ddots & & \ddots & & \\
\boldsymbol{\Omega}_{-\delta} & & \ddots & & \ddots & \\
& \ddots & & \ddots & & \boldsymbol{\Omega}_\delta \\
& & \ddots & & \ddots & \vdots \\
& & & \boldsymbol{\Omega}_{-\delta} & \dots & \boldsymbol{\Omega}_0
\end{pmatrix} \tag{A.3}
$$

and since $\boldsymbol{U}^{-1}$ is also partitioned into $T \times T$ blocks,

$$
\boldsymbol{U}^{-1} = \begin{pmatrix}
\boldsymbol{\sigma}_{q_1}^{-1} & & \\
& \ddots & \\
& & \boldsymbol{\sigma}_{q_T}^{-1}
\end{pmatrix} \tag{A.4}
$$

the product of $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ can also be partitioned into $T \times T$ blocks. The blocks of $\boldsymbol{W}$ have a shape of $\mathbb{R}^{Dn \times D}$ while the blocks of $\boldsymbol{W}^\top$ are $\mathbb{R}^{D \times Dn}$ and the blocks of $\boldsymbol{U}^{-1}$ are $\mathbb{R}^{Dn \times Dn}$. Thus the product of $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ is in $\mathbb{R}^{TD \times TnD}$ and its blocks are in $\mathbb{R}^{D \times nD}$. If $\boldsymbol{A}$ is a band-matrix with upper, lower diagonal u, l then $\boldsymbol{A}^\top$ has upper, lower diagonal l, u. Because the blocks of the abovementioned matrices are aligned properly, blocks in the result of $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ can be calculated by matrix multiplication from the blocks of $\boldsymbol{W}^\top$ and $\boldsymbol{U}^{-1}$, i.e. since the product of two band-matrices is a band-matrix, and since the partitioned matrices are banded, the resulting partitioned matrix is also banded.

Since $\boldsymbol{U}^{-1}$ is a diagonal block matrix, the upper and lower bandwidths are 0. Thus the upper and lower bandwidth of the partitioned matrix $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ are δ_- and δ_+.

Multiplying $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ by $\boldsymbol{W}$, results in a matrix which is in $\mathbb{R}^{TD \times TD}$ and can be partitioned into $T \times T$ blocks of size $\mathbb{R}^{D \times D}$.

Since the partitioned matrix of $\boldsymbol{W}^\top \boldsymbol{U}^{-1}$ is a banded matrix (with upper,lower diagonals δ_-, δ_+) and $\boldsymbol{W}$ also is a banded matrix (with upper,lower diagonals δ_+, δ_-) the product $\boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{W}$ also is a banded partitioned matrix with upper and lower diagonals: $\delta_+ + \delta_-$. Since the blocks are in $\mathbb{R}^{D \times D}$, the matrix without partitioning is a banded matrix with $(\delta_+ + \delta_-) \cdot D$ upper and lower bandwidth.

A.3 Evaluation result confidence intervals

Table A.1 lists the subjective evaluation results as well as their 0.01 significance confidence intervals. The significance of Hypothesis 4.1 is indicated in column ($\neq$). The significantly most often perceived vowel (Hypothesis 4.2) and the vowel perceived significantly more than 50% of cases (Hypothesis 4.3), is indicated by the respective vowel name in column ($>$) in normal and bold font respectively.

Table A.1: *Evaluation results: identification percentages and* 0.01 *confidence intervals for each initial vowel modified by each of nine control offset combinations, significant change to unmodified case (*) and majority vowel (>)*

V	Δ_1	Δ_2	a_-	a	a_+	e_-	e	e_+	i_-	i	i_+	o_-	o	o_+	u_-	u	u_+	$?_-$	$?$	$?_+$	$\neq$	$>$
a	-1.5	-0.75	0	0	0.26	0	0	0.26	0	0	0.26	0.50	0.68	0.94	0	0	0.26	0.14	0.32	0.58	*	
a	-1.5	0	0	0.23	0.52	0.14	0.36	0.66	0	0	0.30	0	0	0.30	0	0	0.30	0.18	0.41	0.70		
a	-1.5	0.75	0	0	0.23	0.64	0.77	1.00	0	0	0.23	0	0	0.23	0	0	0.23	0.09	0.23	0.45	*	**e**
a	0	-0.75	0	0.05	0.32	0	0	0.28	0	0	0.28	0.32	0.55	0.82	0	0	0.28	0.18	0.41	0.68	*	
a	0	0	0.41	0.64	0.87	0	0	0.23	0	0	0.23	0	0	0.23	0	0	0.23	0.14	0.36	0.60		
a	0	0.75	0	0.05	0.27	0.55	0.73	0.95	0	0	0.22	0	0	0.22	0	0	0.22	0.05	0.23	0.45	*	**e**
a	1.5	-0.75	0.14	0.36	0.65	0	0	0.29	0	0	0.29	0	0.18	0.47	0	0	0.29	0.23	0.45	0.75		
a	1.5	0	0.55	0.73	0.96	0	0	0.23	0	0	0.23	0	0	0.23	0	0	0.23	0.09	0.27	0.50		**a**
a	1.5	0.75	0	0.09	0.38	0.18	0.41	0.69	0	0	0.28	0	0	0.28	0	0	0.28	0.27	0.50	0.78	*	
e	-1.5	-0.75	0	0	0.28	0	0	0.28	0.23	0.45	0.74	0	0	0.28	0.27	0.50	0.78	0	0.05	0.33	*	
e	-1.5	0	0	0	0.13	0	0	0.13	0.82	0.91	1.00	0	0	0.13	0	0.09	0.22	0	0	0.13	*	**i**
e	-1.5	0.75	0	0	0.23	0.09	0.27	0.50	0.55	0.73	0.96	0	0	0.23	0	0	0.23	0	0	0.23	*	**i**
e	0	-0.75	0	0	0.21	0	0	0.21	0	0.14	0.35	0	0	0.21	0.55	0.73	0.94	0	0.14	0.35	*	**u**
e	0	0	0	0	0.25	0.50	0.68	0.93	0.05	0.23	0.47	0	0	0.25	0	0	0.25	0	0.09	0.34		e
e	0	0.75	0	0	0.08	1.00	1.00	1.00	0	0	0.08	0	0	0.08	0	0	0.08	0	0	0.08	*	**e**
e	1.5	-0.75	0	0.14	0.41	0.09	0.32	0.60	0	0	0.28	0.09	0.32	0.60	0	0.18	0.46	0	0.05	0.32		
e	1.5	0	0	0.05	0.15	0.91	0.95	1.00	0	0	0.10	0	0	0.10	0	0	0.10	0	0	0.10		**e**
e	1.5	0.75	0	0	0.08	1.00	1.00	1.00	0	0	0.08	0	0	0.08	0	0	0.08	0	0	0.08	*	**e**
i	-1.5	-0.75	0	0	0.28	0	0	0.28	0.32	0.55	0.82	0	0	0.28	0.18	0.41	0.68	0	0.05	0.32		
i	-1.5	0	0	0	0.10	0	0	0.10	0.91	0.95	1.00	0	0	0.10	0	0.05	0.15	0	0	0.10	*	**i**
i	-1.5	0.75	0	0	0.18	0	0.05	0.22	0.77	0.86	1.00	0	0	0.18	0	0.09	0.27	0	0	0.18		**i**
i	0	-0.75	0	0	0.28	0	0	0.28	0.18	0.41	0.69	0	0	0.28	0.27	0.50	0.78	0	0.09	0.38		
i	0	0	0	0	0.28	0.09	0.32	0.60	0.27	0.50	0.78	0	0.05	0.32	0	0.14	0.41	0	0	0.28		
i	0	0.75	0	0	0.10	0.91	0.95	1.00	0	0.05	0.15	0	0	0.10	0	0	0.10	0	0	0.10	*	**e**
i	1.5	-0.75	0	0	0.28	0.27	0.50	0.78	0	0	0.28	0.05	0.27	0.55	0	0.18	0.46	0	0.05	0.32		
i	1.5	0	0	0	0.18	0.68	0.82	1.00	0	0.05	0.23	0	0.09	0.27	0	0.05	0.23	0	0	0.18	*	**e**
i	1.5	0.75	0	0	0.10	0.91	0.95	1.00	0	0.05	0.15	0	0	0.10	0	0	0.10	0	0	0.10	*	**e**
o	-1.5	-0.75	0	0	0.08	0	0	0.08	0	0	0.08	0	0	0.08	1.00	1.00	1.00	0	0	0.08	*	**u**
o	-1.5	0	0	0	0.13	0	0	0.13	0	0.05	0.17	0	0	0.13	0.82	0.91	1.00	0	0.05	0.17	*	**u**
o	-1.5	0.75	0	0	0.26	0.45	0.64	0.89	0	0.14	0.39	0	0	0.26	0	0.14	0.39	0	0.09	0.35	*	e
o	0	-0.75	0	0	0.23	0	0	0.23	0	0	0.23	0.09	0.27	0.50	0.55	0.73	0.96	0	0	0.23	*	**u**
o	0	0	0	0	0.21	0	0.18	0.40	0	0	0.21	0.55	0.73	0.94	0	0.09	0.30	0	0	0.21		**o**
o	0	0.75	0	0	0.18	0.77	0.86	1.00	0	0	0.18	0	0	0.18	0	0	0.18	0.05	0.14	0.32	*	**e**
o	1.5	-0.75	0	0.05	0.17	0	0	0.13	0	0	0.13	0.82	0.91	1.00	0	0	0.13	0	0.05	0.17		**o**
o	1.5	0	0.55	0.73	0.93	0	0.09	0.30	0	0	0.21	0	0.14	0.34	0	0	0.21	0	0.05	0.25	*	**a**
o	1.5	0.75	0	0.05	0.22	0.77	0.86	1.00	0	0	0.18	0	0	0.18	0	0	0.18	0	0.09	0.27	*	**e**
u	-1.5	-0.75	0	0	0.13	0	0	0.13	0	0	0.13	0	0	0.13	0.82	0.91	1.00	0	0.09	0.22		**u**
u	-1.5	0	0	0	0.23	0	0	0.23	0.09	0.27	0.50	0	0	0.23	0.55	0.73	0.96	0	0	0.23		**u**
u	-1.5	0.75	0	0	0.10	0	0	0.10	0.91	0.95	1.00	0	0	0.10	0	0.05	0.15	0	0	0.10	*	**i**
u	0	-0.75	0	0	0.13	0	0	0.13	0	0	0.13	0	0	0.13	0.82	0.91	1.00	0	0.09	0.22		**u**
u	0	0	0	0	0.27	0	0	0.27	0.05	0.23	0.49	0	0	0.27	0.45	0.64	0.90	0	0.14	0.40		
u	0	0.75	0	0	0.25	0.05	0.23	0.47	0.50	0.68	0.93	0	0	0.25	0	0	0.25	0	0.09	0.34	*	i
u	1.5	-0.75	0	0	0.10	0	0	0.10	0	0	0.10	0	0	0.10	0.91	0.95	1.00	0	0.05	0.15	*	**u**
u	1.5	0	0	0	0.27	0.05	0.27	0.54	0	0.05	0.31	0	0.09	0.36	0.27	0.50	0.77	0	0.09	0.36		
u	1.5	0.75	0	0	0.19	0.68	0.82	1.00	0	0.14	0.32	0	0	0.19	0	0	0.19	0	0.05	0.23	*	**e**

Acronyms

EMA Electromagnetic Articulography.

HMM Hidden Markov Model.

HSMM Hidden Semi-Markov Model.

HTK Hidden Markov Model Toolkit.

HTS Hidden Markov Speech Synthesis System.

IPA International Phonetic Alphabet.

LPC Linear Predictive Coding.

LSF Line Spectral Frequencies.

LSP Line Spectral Pairs.

MR-HMM Multiple Regression Hidden Markov Model.

PCA Principal Component Analysis.

PDF Probability Density Function.

RMSE Root Mean Squared Error.

STRAIGHT Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum.

SVM Support Vector Machine.

TTS Text-to-Spech.

vocoder Voice Encoder.

Index

Markov chain, 14

acoustic stream, 21
alignment, 22
allophone, 6
audiovisual speech synthesis, 2
augmentation features, 20
augmentation matrix, 30
augmented features, 28, 29
auto-correlation, 9

Backward, 24
Baum-Welch algorithm, 19, 24
buzziness, 40

clustering, 19
coarticulation, 19
concatenative speech synthesis, 5

data driven speech synthesis, 5
decision tree clustering, 20
delta-delta features, 20, 28, 33
duration, 28
dynamic features, 29

embedded re-estimation, 26

filter coefficients, 21
five-state model, 18
flat-start model, 22
forced aligned, 44
forced alignment, 18, 28
forced-aligned, 46
formant control, 39
forward algorithm, 18
Forward-, 24
frame-shift, 8
fullcontext label, 19
fullcontext-models, 17

fundamental frequency, 21

homographs, 5

Kiel Corpus of Read Speech, 6

leaves, 21
left-to-right model, 28
left-to-right topology, 22
linear regression, 21
linguistic trees, 17

mapping, 21
Markov model, 14
Markov property, 14
monophones, 22
multi-space probability distribution, 21

observation sequence, 20, 22, 29
observations, 15
output probabilities, 15

parametric synthesis, 1
phoneme, 6
phonetic symbol, 6
piece-wise linear, 2, 33
pitch, 52

spectral features, 21
spectral leaking, 9
spectrogram, 9
state occupancy likelihoods, 22
states, 22
streams, 21

text normalisation, 5
tie, 27
tied, 27, 49
tonal languages, 52
transition probabilities, 15

tree-based clustering, 27
triphone symbol, 19
tying, 20

unit selection, 1

visual stream, 21
Viterbi algorithm, 18, 22, 28

window size, 8