Technische Universität Dresden

Articulatory Copy Synthesis
Based on the Speech Synthesizer VocalTractLab

M. Sc.

Yingming Gao

der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität Dresden

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender: Prof. Dr.-Ing. h. c. Frank Fitzek	Tag der Einreichung: 19.11.2021
Gutachter: Prof. Dr.-Ing. Peter. Birkholz	Tag der Verteidigung: 08.06.2022
Gutachter: Prof. Dr. Bernd Möbius	
Gutachter: Prof. Dr. Yves Laprie	

Declaration of Authorship

I, M.S. Yingming Gao, declare that this thesis titled, “Articulatory Copy Synthesis Based on the Speech Synthesizer VocalTractLab” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Yingming Gao

Date: 08.06.2022

TECHNISCHE UNIVERSITÄT DRESDEN

Abstract

Faculty of Electrical and Computer Engineering

Institute of Acoustics and Speech Communication

Dr.-Ing.

Articulatory Copy Synthesis Based on the Speech Synthesizer VocalTractLab

by M.S. Yingming Gao

Articulatory copy synthesis (ACS), a subarea of speech inversion, refers to the reproduction of natural utterances and involves both the physiological articulatory processes and their corresponding acoustic results. This thesis proposes two novel methods for the ACS of human speech using the articulatory speech synthesizer VocalTractLab (VTL) to address or mitigate the existing problems of speech inversion, such as non-unique mapping, acoustic variation among different speakers, and the time-consuming nature of the process.

The first method involved finding appropriate VTL gestural scores for given natural utterances using a genetic algorithm. It consisted of two steps: gestural score initialization and optimization. In the first step, gestural scores were initialized using the given acoustic signals with speech recognition, grapheme-to-phoneme (G2P), and a VTL rule-based method for converting phoneme sequences to gestural scores. In the second step, the initial gestural scores were optimized by a genetic algorithm via an analysis-by-synthesis (ABS) procedure that sought to minimize the cosine distance between the acoustic features of the synthetic and natural utterances. The articulatory parameters were also regularized during the optimization process to restrict them to reasonable values.

The second method was based on long short-term memory (LSTM) and convolutional neural networks, which were responsible for capturing the temporal dependence and the spatial structure of the acoustic features, respectively. The neural network regression models were trained, which used acoustic features as inputs and produced articulatory trajectories as outputs. In addition, to cover as much of the articulatory and acoustic space as possible, the training samples were augmented by manipulating the phonation type, speaking effort, and the vocal tract length of the synthetic utterances. Furthermore, two regularization methods were proposed: one based on the smoothness loss of articulatory trajectories and another based on the acoustic loss between original and predicted acoustic features.

The best-performing genetic algorithms and convolutional LSTM systems (evaluated in terms of the difference between the estimated and reference VTL articulatory parameters) obtained average correlation coefficients of 0.985 and 0.983 for speaker-dependent utterances, respectively, and their reproduced speech achieved recognition accuracies of 86.25% and 64.69% for speaker-independent utterances of German words, respectively. When applied to German sentence utterances, as well as English and Mandarin Chinese word utterances, the neural network based ACS systems achieved recognition accuracies of 73.88%, 52.92%, and 52.41%, respectively. The results showed that both of these methods not only reproduced the articulatory processes but also reproduced the acoustic signals of reference utterances. Moreover, the regularization methods led to more physiologically plausible articulatory processes and made the estimated articulatory trajectories be more articulatorily preferred by VTL, thus reproducing more natural and intelligible speech. This study also found that the convolutional layers, when used in conjunction with batch normalization layers, automatically learned more distinctive features from log power spectrograms. Furthermore, the neural network based ACS systems trained using German data could be generalized to the utterances of other languages.

Acknowledgements

If you want to go fast, go alone. If you want to go far, go together. My PhD has been an incredible, marvelous, and extraordinary journey in my life. There is no doubt that I could not have obtained so much without the accompany and support of many people. As I have come to the end of the big step toward completing my doctoral study, I would like to express my deepest appreciation and thanks to all of them not only for a nice study experience but also for good and happy days in Germany.

First of all I would like to show my sincere gratitude to Prof. Peter Birkholz for giving me the opportunity to pursue my PhD in such a dynamic and friendly research group and excellent supervision toward the success and completion of my doctoral study. Since the day I joined this team, his continued support is the one thing I can always count on. He showed me patience, guided and motivated me whenever the situation arose. His speciality of articulatory synthesis and acoustic phonetics deepened my knowledge of speech science. The past few years have also been a learning journey of acquiring many scientific skills and abilities under his supervision. He sets a very good example for me to conduct scientific research.

I would also like to express my deepest gratitude to Prof. Jinson Zhang, who first took me into the field of speech science, for continued help in many aspects from the past to the present. Special thanks also go to Prof. Hongwei Ding for introducing me to pursue my doctoral degree in Dresden. Both of them are resourceful and have dedicated much of their time to improving my career development. I have also broadened my horizons and greatly honed my scientific skills from collaborative projects with them.

Other thanks go to past and present colleagues, Alexander Wilbrandt, Barbara Wran, Christian Kleiner, Christoph Wagner, Falk Gabriel, Franziska Leistner, Kathleen Große, Paul Konstantin Krug, Patrick Häsner, Peter Steiner, Pouriya Amini Digehsara, Rainer Jäckel, Rémi Blandin, Rüdiger Hoffmann, Simon Stone, Steffen Kürbis, Susanne Drechsel, Ulrich Kordon, and Xinyu Zhang, for the harmonious and productive working environment we created together and their help in many aspects of my daily life. They are very nice to get along. We spent a really good time together during which I have been learning more other than science. Special thanks are given to Simon Stone since he is approachable and makes the working atmosphere more colorful and active.

I sincerely thank Simon Stone, Paul Konstantin Krug and Rainer Jäckel and other colleagues for their knowledge and suggestions during closely working together. I own particular thanks to Steffen Kürbis for his friendliness and readiness for help me with managing computers and other devices. I would also like to thank Lukas Bulla since part of my study was inspired by his previous work. I appreciate the constructive discussion on my experiments with Ju Lin and Shuju Shi. I must also give my sincere thanks to the speakers who helped me record the speech corpus and the participants of the listening experiment.

I would like to show my greatest appreciation to the funding providers of my research. My doctoral study was mainly financially supported by China Scholarship

Council. Partial scientific expenses were sponsored by Willkomm-Stiftung, enabling me to participate international conferences. The completion of my thesis was made possible by the German Academic Exchange Service (DAAD), the Association of Friends and Sponsors, and Graduate Academy of TU Dresden. In particular, I am indebted to Prof. Rüdiger Hoffmann, Ms Corina Wolfermann and Ms Lieselotte Landgraf-Dietz for their help with the funding applications.

The computations of conducting experiments were performed partially on an HPC system at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden, and partially on the Linux server at the Center for Speech Acquisition and Intelligent Technology (SAIT) at Beijing Language and Culture University (BLCU). I thank them for generous allocations of compute resources.

I am extremely grateful to my family for always being there for me. Their immeasurable love, encouragement, and support during my PhD period and my life in general make me face every challenge bravely. I particularly owe my sincere gratitude to my parents who encouraged me to continue my studies and my wife who has always stood by me, waited for my PhD completion without any complaints, and dealt with everything well to prevent me from worrying about them throughout the years. I also thank my cousins for their various help to support my study. The undertaking of my research could not have been successfully achieved without their support.

My heartfelt appreciation goes to my friends, Delun Kong, Shuju Shi, Ju Lin, He Gao, Xiaoli Feng and countless others, for sharing happiness and sadness despite of the long physical distance to each other. I thank the friends from China and other countries I made in Dresden. In particular, I express my gratitude to the teammates (including Dong Lin, Yu Yu, Shaohuai Tang, Boli Peng, Cong Lian, Junlin Huang and quite a few more) of the CFC football team and the roommates getting along well with each other in our shared house for the unforgettable memories.

Contents

List of Figures

List of Tables

List of Abbreviations

AAI	Acoustic-to-articulatory inversion
ABS	Analysis-by-synthesis
ACS	Articulatory copy synthesis
ANOVA	Analysis of variance
API	Application programming interface
AR-ANN	Autoregressive artificial neural network
ASR	Automatic speech recognition
BiLSTM	Bidirectional LSTM neural networks
CALL	Computer aided language learning
CAP	Critical articulatory portions
CAPL	Computer assisted pronunciation learning
CMVN	Cepstral mean and variance normalization
CNN	Convolutional neural networks
CSA	Computational speech acquisition
DBN	Deep belief network
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DRMDN	Deep recurrent mixture density network
DRNL	Dual resonance nonlinear
DSL	Distal supervised learning
DTW	Dynamic time warping
EMSSL	Embodied self-supervised learning
FF-ANN	Feedforward artificial neural networks
FSS-MRHMM	Feature-space-switched multiple regression HMM
G2P	Grapheme-to-phoneme
GAS	Generic acoustic space
GMM	Gaussian mixture model
GPU	Graphics processing unit
HMM	Hidden Markov model
HPC	High performance computing
LSF	Line spectral frequencies
LSTM	Long short-term memory
MAP	Maximum a posteriori probability
MDN	Mixture density networks
MFCC	Mel-frequency cepstral coefficients
MLP	Multilayer perceptron
MRI	Magnetic resonance imaging
MSE	Mean squared error
NCAP	Non-critical articulatory portions

NMCC	Normalized modulation cepstral coefficient
NMC	Normalized modulation coefficients
PESE	Perceptual evaluation of speech quality
POV	Probability of voicing
PSE	particle swarm optimization
RASTA-PLP	Relative spectral-perceptual linear prediction
RBF	Radial basis function
RMSE	Root mean square error
RNN	Recurrent neural network
SAMPA	Speech assessment methods phonetic alphabet
SDI	Speaker-dependent inversion
SII	Speaker-independent inversion
SNR	Signal to noise ratio
STOI	Short term objective intelligibility
STP	Short term processing
SVR	Support vector regression
SyDOCC	Synchronized damped oscillator cepstral coefficients
TADA	Task dynamic model
TAM	Target approximation model
TMDN	Trajectory mixture density networks
TRM	Tube resonance model
VAD	Voice activity detection
VTL	VocalTractLab
VTLN	Vocal tract length normalization
VTLP	Vocal tract length perturbation
XRMB	X-ray microbeam
ZCR	Zero crossing rate

Chapter 1

Introduction

1.1 Background and Motivation

Speech is the most common mode of human communication. As shown in Figure 1.1, the process of communication involves three main events: the production of speech, the propagation of speech, and the perception of speech. These three events constitute the speech chain. Humans (speakers) move air particles using their vocal organs, which cause vibrations in the air. This results in a sequence of pressure waves that propagate through the air, a compressible medium. Regardless of whether the sound waves are recorded, transmitted via a digital device, or propagated directly through the air, these waves will enter a listener's ears and activate a network of tiny bones and cells that eventually enable them to hear the speaker. Hence, the investigation of speech can be roughly divided into three distinct but interdependent aspects: production, propagation, and perception, which correspond to the articulatory, acoustic, and auditory/perceptual domains, respectively.

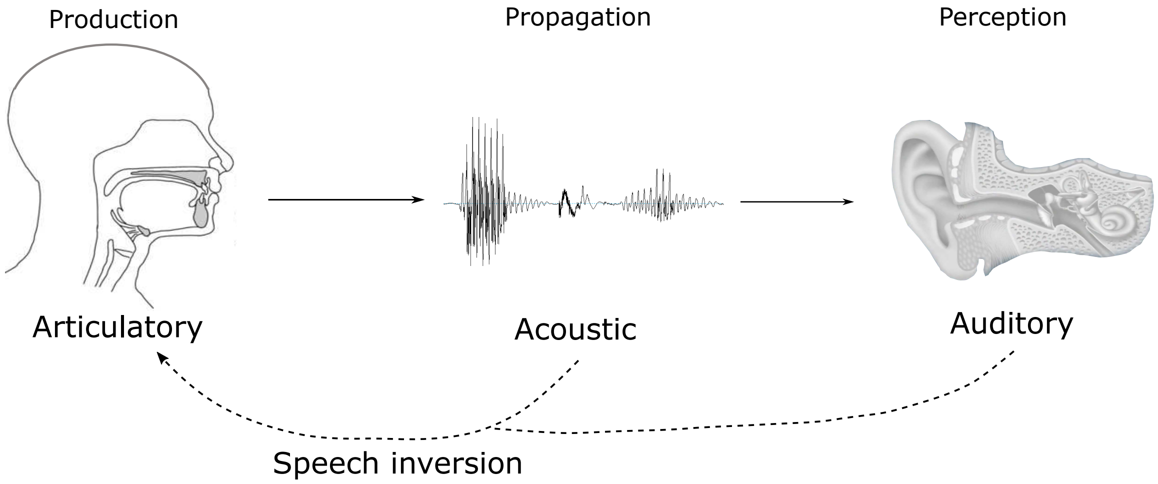


FIGURE 1.1: Diagram of speech chain and speech inversion.

Within the articulatory domain, researchers investigate how the sounds that comprise speech are produced. This usually involves a basic understanding of the anatomy of human vocal organs. For example, lungs produce the necessary energy in the form of a stream of air, the larynx serves as a modifier to the airstream and is responsible for phonation, and the vocal tract modifies and modulates the airstream through several articulators (lips, tongue, velum, etc.). From the perspective of the auditory/perceptual domain, researchers investigate the mechanisms behind the perception of human speech. This includes how the auditory system analyzes the sound waves received and how it extracts features before transmitting this information to the listener's brain. Within the acoustic domain, researchers investigate the physical properties of the speech signal by exploring the physical characteristics of human speech, such as its frequency.

As a result of the findings derived from these three aspects, many attendant research topics and applications have emerged, including automatic speech recognition (ASR), speech synthesis, speech coding, etc. For example, by revealing that human ears are more sensitive to lower frequency regions than higher frequency regions and that a series of critical frequency bands in the human auditory system are roughly linearly-spaced on the mel scale, mel-frequency cepstral coefficients (MFCC) were proposed and are widely used as acoustic features in tasks such as ASR and speaker recognition. As one of speech synthesis techniques, articulatory synthesis is proposed by modeling human vocal tract, vocal folds, and the articulatory processes of speech production.

Speech inversion is another research topic closely associated with the three aspects of the speech chain. As shown in Figure 1.1, instead of investigating speech from the forward direction in the speech chain (from speech production, through speech propagation, to speech perception), speech inversion attempts to gain insights from the opposite direction. Specifically, it refers to "inverse problems with respect to vocal tract shape, area function, articulatory parameters or control commands appear both in the theory of speech production and perception, and in technical applications like speech recognition, synthesis and compression" (Sorokin, 2006). Owing to the advancements in our understanding of speech production and perception, as well as in the development of technical applications, speech inversion has been extensively investigated over the last few decades and in turn facilitated many speech-related research fields. The rest of this section explains why speech inversion is worth studying, particularly in terms of its applications.

Applications of Speech Inversion

Speech inversion converts the acoustic representation of a sound to its articulatory representation. The articulatory representation is considered less variable and more robust than the acoustic representation (Mitra et al., 2010; Mitra et al., 2014). The estimated articulatory information can complement the acoustic information. This allows speech inversion to be applied to various speech-related applications.

Speech Recognition

The performance of ASR systems is inhibited due to the differences in the acoustic variations of the same phonemes as a result of speaker differences as well as prosodic and contextual influences. However, from the perspective of speech production, articulatory knowledge can mitigate the effects of such variations. In addition, the motor theory of speech perception suggests that people perceive spoken words by identifying the vocal tract gestures with which they are pronounced rather than by identifying the sound patterns that the speech generates (Liberman and Mattingly, 1985; Galantucci, Fowler, and Turvey, 2006). Articulatory information is thus considered to be beneficial to the performance of ASR systems, which has been verified in various studies (e.g., Nam et al., 2010; Mitra et al., 2013; and Mitra et al., 2014).

In addition, ASR systems usually use hand-crafted acoustic features as inputs, which may lose distinctive information for ASR due to the simplified assumptions during extraction (e.g., the decorrelation operation for extracting MFCC). However, articulatory features derived from speech inversion can provide additional information for speech recognition, which is extremely helpful when the acoustic inputs are of low quality due to noise or insufficient training data. For example, by combining articulatory and acoustic features, Frankel and King (2001) achieved better performances for both speech classification and recognition on TIMIT sentences in comparison to using either feature alone. Mitra et al. (2014) also demonstrated that articulatory representation enhanced the noise robustness of ASR systems.

Speech Synthesis and Speech Modification

Articulatory information can be incorporated into synthesis systems to modify the characteristics of synthetic speech. As an intermediate step between motor commands and acoustic features, articulatory features are physiologically meaningful and can provide a straightforward connection between the two. For example, Ling et al. (2009) proposed a method for integrating articulatory features into a hidden Markov model (HMM) based parametric speech synthesis system. The HMM states for the joint distribution of acoustic and articulatory features were first estimated using paired articulatory-acoustic training data. Acoustic features were then generated from their unified model during synthesis. A review of speech inversion with a particular focus on its application to speech synthesis can be found in Richmond, Ling, and Yamagishi (2015).

In terms of speech modification, some aspects such as duration and fundamental frequency (f_0) can be easily manipulated in the acoustic domain, while others such as regional/foreign accents as well as articulation styles can be relatively challenging. To address the latter, speaker-dependent characteristics and linguistic information should be decoupled. Felps, Geng, and Gutierrez-Osuna (2012) hypothesized that the articulatory domain provides better separation between linguistic information and speaker-dependent characteristics than the acoustic domain. In addition, they suggested that articulatory features are less speaker-dependent than acoustic features. They verified this hypothesis by tackling the problem of foreign accent conversion. They first recorded an articulatory-acoustic dataset produced by a native speaker and a non-native speaker. They then resynthesized the utterances of the non-native speaker by replacing the most accented segments with alternatives from his other utterances. These alternative segments were selected based on their articulatory or acoustic similarity to those uttered by the native speaker. Listening experiments showed that this approach of selecting segments in the articulatory domain could achieve a 20% reduction in perceived accent (Felps, Geng, and Gutierrez-Osuna, 2012).

Aryal and Gutierrez-Osuna (2013) investigated the application of speech inversion and synthesis to articulatory-based speech modification. In their inversion module, they used an articulatory-acoustic dataset to train a feed-forward radial basis function (RBF) network to estimate the six parameters of the Maeda synthesizer from MFCC features extracted via STRAIGHT (Kawahara et al., 2008). In their synthesis module, they designed a Gaussian mixture model (GMM) based forward mapping model which estimated spectral features from articulatory parameters, then used STRAIGHT to resynthesize speech with an estimated spectral envelope, aperiodicity, and pitch. The effectiveness of this proposal was validated by measuring the correlation and the root mean square error (RMSE) between the estimated and original articulatory trajectories in the articulatory domain, mel cepstral distortion between synthetic and original speech in the acoustic domain, and a listening test in the perceptual domain.

Language Learning and Speech Therapy

In computer-aided language learning (CALL), the 3D animation of vocal tracts can provide language learners with instructive information on how the articulators move when producing the sounds associated with standard speech. Badin et al. (2010a) and Badin et al. (2010b) developed a talking head that provided visual articulatory feedback for phonetic correction for second language learners. They built an OroFacial Clone (OFC) system using magnetic resonance imaging (MRI), computer tomography (CT), and video data acquired from a French speaker, henceforth referred to as the model speaker. The OFC demonstrated the articulatory process, including the movements of the internal articulators. More importantly, by recognizing new utterances from the model speaker and resynthesizing the articulatory trajectories originally measured from electromagnetic articulography (EMA) data, the OFC tool could display the new articulatory movements. Wang, Hueber, and Badin (2014) verified the usefulness of using articulatory information for second language pronunciation training. They used an articulatory talking head to teach Mandarin Chinese speakers to enunciate French vowels. The results showed that learners receiving audiovisual stimuli performed better than learners who only received auditory stimuli. In addition, articulatory features can be used to synthesize natural facial animation for language tutoring due to the close connection between articulatory features and facial features during speech production. For example, Ben-Youssef, Shimodaira, and Braude (2014) developed a talking head where the motion of the lips and head were controlled by articulatory features estimated from acoustic signals.

Virtual talking heads usually only provide standard pronunciations of audio examples together with prerecorded articulatory movements. Learners acquire new or correct pronunciations by imitating the pre-generated samples. However, as they cannot see the real-time movements of their own articulators, learners cannot visually compare the relative differences between their own speech and the standard articulation during speech imitation. CALL systems can thus provide customizable learning materials and provide instructive feedback based on the results of the learner's imitations. An example process is as follows: (1) The CALL system, either with or without standard pronunciations, prompts learners to produce an utterance. Meanwhile, the corresponding audio signal produced by the learners will be recorded. (2) The pronunciation errors will be diagnosed via the pronunciation error detection module. The learners' articulatory processes will be estimated using the speech inversion module. (3) The articulatory parameters of mispronounced speech will be corrected and resynthesized via the speech modification module. (4) Finally, the CALL system will provide learners with correctly resynthesized speech and a corresponding visualization of the articulatory process (e.g., a 3D animation of the tongue and lips). This approach makes it possible for language learners to hear and see both their own incorrect version of an utterance and the correct, native-like version spoken in their own voice. This direct comparison of both versions will help the learners more rapidly identify the mistakes in their pronunciation as well as receive feedback on how it could be improved.

Speech inversion can also be applied to speech therapy. For example, by providing information about tongue contact, movement, and shape via electropalatography and ultrasound imaging, articulatory visual feedback significantly improved the speech production of hearing-impaired speakers (Bernhardt et al., 2003).

Other Applications

Besides the above-mentioned applications, speech inversion can also benefit the understanding of speech production and perception by investigating the relationships between acoustic characteristics of natural utterances and corresponding articulatory and phonatory configurations. For example, speech inversion has been successfully applied to investigate the mechanisms of coarticulation and lenition by training tract-variables (TVs) estimators with paired articulatory-acoustic data (Xu, Birkholz, and Xu, 2019) or resynthesizing human speech via an articulatory synthesizer (Sivaraman et al., 2015). Moreover, it can help understand infants' early language acquisition (Kuhl, 2004; Gerazov et al., 2021). Speech inversion allows to modify specific parameters of computational speech acquisition models (the relevant concepts will be introduced in Section 2.2) and then observe and analyze the effects on speech acquisition (Howard and Messum, 2007; Philippsen, Reinhart, and Wrede, 2016).

Creating paired articulatory-acoustic datasets is another application. By using available segmentation or deriving gestural annotation from natural utterances, gestural scores (this concept will be introduced in Section 3.1) could be obtained which were further converted to articulatory trajectories via articulatory synthesizers (Nam et al., 2012; Sering et al., 2019). The derived articulatory trajectories together with acoustic features extracted from the original human utterances constitute the paired articulatory-acoustic datasets.

1.2 Discussion of Limitations of Previous Research

Although speech inversion has been extensively studied over the past few decades, it has not yet been perfected. Panchapagesan and Alwan (2011) summarized four main difficulties faced by the analysis-by-synthesis (ABS) based methods of speech inversion. When extended to include all kinds of speech inversion methods, these difficulties can be reformulated as follows:

- Insufficient data, especially in terms of the articulatory data, to train and/or evaluate inversion models. Articulatory data are usually expensive and laborious to acquire; even recording such data is invasive to subjects.
- Inherent non-uniqueness. Specifically, speech inversion has a one-to-many characteristic; for example, any given segment of a speech signal can be generated by different combinations of articulatory configurations, much less long utterances like words, phrases, and sentences. This can be explained by the theory of motor equivalence, which states that "motor equivalence enables motor tasks to be achieved in a variety of ways" (Perrier and Fuchs, 2015). The fact that the same acoustic observations can be produced by different articulatory processes is an inherent challenge faced by inversion tasks.
- The high nonlinearity of the acoustic-to-articulatory mapping. Specifically, the influence of changes in the articulatory domain on variations in the acoustic domain is non-linear. Consequently, data points that are represented in the form of a vector in acoustic space do not linearly correspond to those same points in articulatory space. Hence, simple strategies, such as codebook-based methods, usually exhibit inferior performances in speech inversion tasks.
- Physiological differences. Specifically, there are anatomical differences between the vocal tracts and vocal folds of model speakers and the speakers whose utterances are to be reproduced. Inversion methods can usually achieve good results for speaker-dependent tasks. However, these performances tend to degrade when applied to speaker-independent scenarios.
- Incomplete knowledge about speech production and perception. This influences many aspects of speech inversion such as acoustic feature selection, synthesizer design, similarity measurement, etc.
- Speech inversion methods are computationally expensive. Some tasks require forward mapping from the articulatory to acoustic domains, e.g., acoustic simulation, which usually takes more time than other modules. In addition, some methods do not have analytical solutions. Hence, articulatory movements usually have to be estimated in an iterative loop, such as via an ABS-based procedure, which is very time-consuming.
- Local optima of the cost function. Some inversion methods are strongly dependent on the initial state of the model, such as those based on self-exploration or random searches. A poor initial state will take significantly longer time to find solutions, even the final results are only locally optimal.

Due to these difficulties, most speech inversion studies have one or more of the following limitations. A detailed review of related studies will be conducted in Chapter 2.

1. The construction of mapping models relies on recorded articulatory data, which is expensive and laborious, and even invasive to speakers during data collection.
2. The inversion methods are only suitable for short-utterances like isolated vowels, "VV" sequences or simple "CV", "CVC" or "VCV" syllables.
3. The mapping is based on speaker-dependent models; in other words, both the training and testing data come from the same speaker(s).
4. ABS-based methods are time-consuming and cannot be generalized; i.e., the inversion has to be performed individually for each new utterance.
5. The performance is only evaluated in the articulatory domain; specifically, it only evaluates the correlation or RMSE between estimated and reference articulatory trajectories. Whether the estimated trajectories are suitable for synthesizing speech is not considered. Some studies that produced speech using an estimated articulatory process did not provide a systematic evaluation of the quality of the synthesized speech in terms of its intelligibility.

Given that speech inversion has various applications as summarized in Section 1.1, as well as the fact that there are still some limitations to the existing methods listed in this section, this study attempted to circumvent or mitigate these limitations.

1.3 Structure and Contributions of the Thesis

This study tackled the speech inversion problem with a particular focus on *articulatory copy synthesis* (this concept will be introduced in Section 2.3). This section outlines the content of each chapter and summarizes the main contributions of this thesis.

Thesis Structure

- Chapter 1 introduces the background of the generic research field, speech inversion, from the point of view of the speech chain. It then provides an overview of its practical and potential applications. After that, it highlights the difficulties of speech inversion and limitations of existing methods that motivated further investigation in this work.
- Chapter 2 introduces the field of speech inversion in greater detail. It is divided into three distinct but interdependent subareas: acoustic-to-articulatory inversion, computational speech acquisition, and articulatory copy synthesis. The representative literature of each of these fields is reviewed.

- Chapter 3 introduces the fundamentals that allow to experimentally validate the proposals of this study. It first introduces the basic concepts of the articulatory synthesizer used in this study, VocalTractLab, which connects the articulatory and acoustic domains throughout the experiments in Chapters 4 and 5. This chapter also introduces the datasets, especially the ones deliberately designed for this study, as well as the acoustic representation of utterances (i.e., the acoustic features).
- Chapter 4 presents a novel approach to articulatory copy synthesis based on a genetic algorithm, which consists of two steps: the initialization and optimization of a gestural score for a given utterance. It also presents the regularization of the deviations of articulatory parameters as well as the automatic performance evaluations in the acoustic and perceptual domains.
- Chapter 5 presents a second approach to articulatory copy synthesis based on artificial neural networks. Specifically, long short-term memory (LSTM) neural networks and convolutional LSTM neural networks are employed to establish the mapping of acoustic features to the articulatory parameters of VocalTractLab. It also presents the regularization of articulatory trajectories by introducing additional losses into the loss function during the training of neural networks. In addition to the automatic performance evaluation by machines in the articulatory and perceptual domains, this chapter introduces an online perceptual experiment to evaluate the quality of reproduced speech by human listeners. Furthermore, it examines the generalization ability of the speech inversion models to utterances of other languages.
- Chapter 6 concludes this study by summarizing and discussing the findings as well as providing a description of potential future work.

Thesis Contributions

The contributions of this thesis are as follows:

1. Both the genetic algorithm and deep neural network based methods not only reproduced the articulatory processes but also reproduced the acoustic signals of reference utterances.
2. The genetic algorithm proved to be more powerful than the coordinate descent algorithm for gestural score optimization.
3. The recurrent characteristics of LSTM were capable of modeling the temporal dependence of both articulatory parameters and acoustic features. Using convolutional layers together with batch-normalization layers resulted in better acoustic representations than hand-crafted features.
4. This study confirmed the effectiveness of regularizing articulatory parameters during articulatory copy synthesis. A forward model that involved the mapping of articulatory parameters to acoustic representation was proposed, which outperformed traditional regularization methods.

5. The universality of articulatory representation regardless of language was verified by applying the speech inversion models trained on German data to English and Mandarin Chinese utterances.

6. Two datasets were deliberately created for this work. One was a phonemically-balanced acoustic corpus, the PBACU corpus, which allowed this study to use the least number of utterances to cover as many German phonemes as possible. The other was a paired articulatory-acoustic dataset, the VTL-Kiel corpus, which was created using VocalTractLab as well as the Kiel Corpus and allowed for the training of other speech inversion systems. Both of these datasets and the methods used to create them could be applied to other research tasks.

Chapter 2

Literature Review

Speech inversion is a generic concept that describes the mapping of acoustic representations to articulatory representations. This research field can be roughly divided into three research topics according to the aim, the data formats, and its theoretical basis. Acoustic-to-articulatory inversion (AAI) refers to the conversion of acoustic signals to articulatory variables that generally represent the positions of the articulators. Computational speech acquisition (CSA) refers to the development of language learning agents that acquire motor commands from ambient speech to produce human-like speech. Articulatory copy synthesis (ACS) refers to the articulatory reproduction of a target reference by estimating the actual articulatory process and subsequently resynthesizing an acoustic duplicate. The relationships between these research topics are shown in Figure 2.1(a).

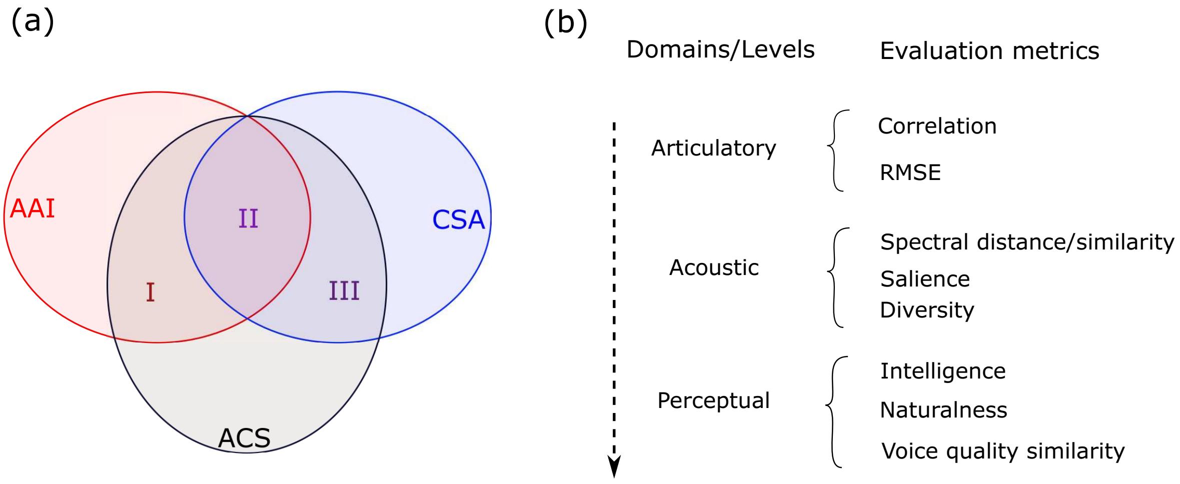


FIGURE 2.1: (a) Relation among acoustic-to-articulatory inversion (AAI), computational speech acquisition (CSA), and articulatory copy synthesis (ACS), which are represented by the red, blue, and black circles, respectively; (b) Evaluation metrics of different domains/levels.

Although the literature will be separately reviewed for each of these three research topics, there are some similarities between them. For example, each of these research topics requires acoustic and articulatory representations as well as inversion algorithms, as indicated by the overlapping area "II" in Figure 2.1(a). There are also areas shared by only two of these research topics. Area "I" is shared by AAI and ACS but not by CSA because both AAI and ACS require the acoustic signals and estimated articulatory trajectories to be matched frame-by-frame, while CSA is more focused on whether motor commands (i.e., articulatory trajectories) are mastered as well as whether the reproduced speech is linguistically meaningful; the latter does not need the speech learned by agents and ambient speech to be temporally matched. Similarly, area "III" is shared by CSA and ACS but not by AAI. This is because AAI only derives articulatory representations from acoustic signals; it does not produce new speech.

The performance of speech inversion methods can be evaluated according to different domains/levels (Figure 2.1(b)). In order from simple to complex, they are the articulatory, acoustic, and perceptual domains/levels, each of which is evaluated using different metrics. The AAI tasks are only evaluated in the articulatory domain/level since they merely estimate articulatory trajectories, and do not produce any new speech. The most common way to evaluate these tasks is by calculating the correlation coefficients and RMSE between the estimated and the reference articulatory trajectories. In contrast, CAS tasks usually aim to master speech production. Therefore, the performance of these models is mainly evaluated in the perceptual domain/level, such as by measuring the intelligibility and naturalness of speech produced by CAS systems. It can also be evaluated in the acoustic domain/level, such as by measuring the "salience" and "diversity" of the sounds produced by early-stage CAS agents; this will be explained in greater detail in Section 2.2. ACS tasks not only estimate the articulatory process of original utterances but also synthesize the acoustic duplicates to be temporally matched with the given speech. Hence, the performance of ACS tasks can be evaluated in each domain. The reproduction of reference utterances involves both the articulatory processes and the acoustic results of speech production. In the perceptual domain, the ultimate goal is to make the reproduced speech sound equivalent to the original.

In terms of the results of speech inversion, the estimated articulatory trajectories may perfectly match the reference trajectories, and the reproduced speech may be judged to be acoustically similar enough to the original recordings. Nevertheless, human listeners generally perceive the reproduced speech to be inferior to that produced by human speakers. This is due to several factors ranging from obvious errors and audible artifacts in the synthetic speech to the voice quality of different speakers (Steiner, 2010). In this context, voice quality refers to auditory characteristics that reflect elements of a speaker's voice, caused by a variety of laryngeal and supralaryngeal features. It should be noted that, to our knowledge, although it affects the perceptual judgment of speech similarity, the reproduction of voice quality has not been included in any speech inversion studies, including this work.

In the following sections, AAI, CSA, and ACS will be introduced in greater detail in Sections 2.1, 2.2, and 2.3, respectively. Section 2.4 provides some concluding remarks.

2.1 Acoustic-to-Articulatory Inversion

AAI requires paired articulatory-acoustic data in which the movements of articulators and their corresponding acoustic signals are recorded concurrently. Early recordings of articulatory data were based on X-rays. However, the dangers of exposure to X-ray radiation massively limited their use, especially with the development of other medical techniques. X-ray microbeam (XRMB) cinematography is an alternative tool in which subjects receive lower total doses of radiation. In addition, MRI scanner is a safer tool that can measure articulatory data from multiple orientations. Recording data with MRI, however, is very noisy and relatively costly. Ultrasound is another cheap and safe tool used to record articulatory data, although it provides rather limited information about articulators. Electromagnetic articulography (EMA) is the most common data collection technique in speech inversion tasks despite its downsides, such as measurement errors introduced by its tracking methods and the limited number of tracking points. Table 2.1 summarizes several articulatory-acoustic datasets widely used for AAI investigations.

TABLE 2.1: Summary table of commonly-used paired articulatory-acoustic datasets.

	Dataset	**Data format**	**Speaker**	**Utterance**
Westbury, Turner, and Dembowski, 1994	XRMB	Audio, X-ray	25 males and 32 females	Mix of vowels, syllable, words and sentences
Wrench, 1999	MOCHA-TIMIT	Audio, EMA, EPG	1 male and 1 female	460 sentences for each speaker
Richmond, Hoole, and King, 2011	mngu0	Audio, video, EMA, MRI	1 male	1300 sentences
Rudzicz, Namasivayam, and Wolff, 2012	TORGO	Audio, EMA	8 speakers with dysarthria and 7 non-dysarthric speakers	Mix of syllable, words and sentences
Narayanan et al., 2014	USC-TIMIT	Audio, EMA, MRI	5 males and 5 females	460 sentences for each speaker
Tiede et al., 2017	HPRC	Audio, EMA	4 males and 4 females	720 sentences

Although articulatory data such as EMA and MRI data are relatively expensive and laborious to acquire, they provide significantly different and complementary information compared to traditional acoustic signals. Due to the development of articulatory data acquisition and inversion algorithms, AAI has become much more feasible and has developed considerably over the last two decades. Table 2.2 summarizes the representative literature on AAI. Depending on whether there is an overlap of speakers whose data are used for the training and testing of the inverse model, AAI can be divided into two types: speaker-dependent inversion (SDI) or speaker-independent inversion (SII). The different acoustic characteristics between the training and testing speakers make SII tasks more challenging than SDI tasks.

TABLE 2.2: Summary table of literature on acoustic-to-articulatory inversion (AAI). The "SI" in the second column indicates whether the study is speaker-independent, i.e., "Y" for SII and "N" for SDI.

	SI	**Utterance**	**Dataset**	**Acoustic representation**	**Inverse algorithm**
Dusan and Deng, 2000	Y	English vowels, syllables, sentences	XRMB	Formant, MFCC	Codebook, extended Kalman Filter
Hiroya and Honda, 2004	N	Japanese sentences	their own EMA data	Mel-cepstra	HMM
Richmond, 2006; Richmond, 2007	N	English sentences	MOCHA-TIMIT	Mel-filterbank	Trajectory mixture density networks
Zhang and Renals, 2008	N	English sentences	MOCHA-TIMIT	MFCC	HMM
Mitra et al., 2010	N	English words	TADA synthesized data	MFCC, AP	TMDN, FF-ANN, AR-ANN, SVR, DSL
Uria, Renals, and Richmond, 2011; Uria et al., 2012	N	English sentences	mngu0	LSF	DBN-DNN, deep TMDN
Mitra et al., 2013; Mitra et al., 2014	Y	English sentences	TADA synthesized data	MFCC, RASTA-PLP, NMCC, SyDOCC	DNN
Afshan and Ghosh, 2015	Y	English sentences	MOCHA-TIMIT	MFCC	Generalized smoothness criterion
Zhu, Xie, and Chen, 2015	N	English sentences	mngu0	LSF, MFCC	BLSTM
Liu et al., 2015	N	English sentences	mngu0	LSF	BiLSTM, RMDN
Sivaraman et al., 2016	Y	English sentences	XRMB	MFCC	Feed-forward neural network
Mitra et al., 2017	Y	English sentences	TADA synthesized data	NMC, NMCC	DNN, CNN
Shahrebabaki et al., 2019	N	English sentences	HPRC	Filterbank, phonetic and attribute features	BiLSTM

By dividing the articulatory-acoustic function into sub-functions with a set of phonological coproduction units that were similar to diphones, Dusan and Deng (2000) constructed a codebook of articulatory and acoustic parameters. The articulatory trajectories were estimated by using an extended Kalman filtering technique from acoustic representation. HMMs were also explored as a tool for AAI tasks. For example, Hiroya and Honda (2004) proposed an HMM production model that consisted of the articulatory HMM for each phoneme and an articulatory-to-acoustic mapping for each HMM state; this allowed for a maximum a posteriori probability (MAP) estimate of articulatory parameters that could be determined for each acoustic feature.

Richmond completed his PhD on the topic of the inversion mapping in which he compared mixture density networks (MDN) with feed-forward multilayer perceptron (MLP) for estimating articulatory parameters from acoustic signals, showing that MDN was more appropriate than MLP to tackle the non-uniqueness problem of speech inversion (Richmond, 2002). Extensions to this work were trajectory MDN, referred to as TMDN, by augmenting MDN with dynamic features together with an algorithm for estimating maximum likelihood trajectories (Richmond, 2006; Richmond, 2007). All these studies were conducted to estimate the 14 EMA variables of MOCHA dataset (Wrench, 1999). Richmond and his colleagues also released their own dataset ("mngu0") containing 12 EMA variables (Richmond, Hoole, and King, 2011), upon which deep belief network together with deep neural network (DBN-DNN) and deep TMDN were trained to estimate articulatory trajectories from acoustic signals (Uria, Renals, and Richmond, 2011; Uria et al., 2012). By feeding the estimated articulatory trajectories into the feature-space-switched multiple regression HMM (FSS-MRHMM) synthesizer and then measuring the quality of synthetic speech in the acoustic domain in terms of RMSE of line spectral frequencies (LSF), Ling, Richmond, and Yamagishi (2012) evaluated the performance of a range of inversion mapping methods.

By using the Haskins Laboratories' Task Dynamic model (TADA) developed by Nam et al. (2004), Nam, Mitra, and their colleagues conducted a series of studies on AAI. For example, they first generated the articulatory control parameters and corresponding synthetic speech for 420 words using TADA. Then they compared TMDN, feedforward artificial neural networks (FF-ANN), support vector regression (SVR), autoregressive artificial neural network (AR-ANN), and distal supervised learning (DSL) for training the SDI models and concluded that the speech inversion model based on a 3-hidden layer FF-ANN outperformed the models based on other algorithms (Mitra et al., 2010). They also applied the estimated articulatory information to speech recognition tasks. By generating articulatory-acoustic data of 111 929 words with TADA, they first tried training SII models with shallow neural networks in Mitra et al. (2013) and Mitra et al. (2014). Later, in Mitra et al. (2017), they tried two neural network based inversion models with DNN and convolutional neural networks (CNN) architectures respectively, and compared their performances with regard to the correlation coefficients between the ground truth and the estimated articulatory trajectories. They validated the usefulness of incorporating the articulatory information for improving the performance of speech recognition in these three studies. Besides, instead of directly concatenating the acoustic features and articulatory parameters as used in Mitra et al. (2013) and Mitra et al. (2014), they also attempted the hybrid convolutional neural network (HCNN) in which two parallel neural networks were used to model the acoustic and articulatory spaces respectively while the two branches of neural networks sharing the last network layer were jointly trained with one objective function (Mitra et al., 2017).

To tackle the problem of speaker variations in SII, Sivaraman et al. (2016) investigated vocal tract length normalization (VTLN) to reduce speaker-specific differences. They first normalized the acoustic data of each training speaker towards the acoustic space of a target test speaker using VTLN, then used the transformed data to train the SII systems. Ghosh and Narayanan (2011) conducted another study to circumvent the mismatch of acoustic characteristics between the utterances of training and test speakers. They normalized acoustic features using a generic acoustic space (GAS). They also proposed several unsupervised and supervised approaches to clustering GAS from a large pool of speakers and investigated the adaption of GAS using acoustic data of speakers in an inversion model (Afshan and Ghosh, 2015).

Zhang and Renals (2008) proposed an HMM based inversion system by jointly optimizing a combined two-stream model in which a trajectory HMM and a regular GMM-HMM modeled the articulatory and acoustic streams, respectively. The experiment results on the MOCHA-TIMIT corpus showed that the jointly trained model achieved a lower RMSE than the separately trained ones. However, the performance of this HMM-based model was still inferior to that of TMDN-based inversion models applied to the same corpus (Richmond, 2006; Richmond, 2007). Furthermore, both LSTM based models (e.g., in Zhu, Xie, and Chen (2015) and Shahrebabaki et al. (2019)) and CNN based models (e.g., in Mitra et al. (2017)) outperformed the regular feed-forward neural networks due to their modeling capability of temporal and spatial dependence, respectively. For example, by applying bidirectional LSTM neural networks (BiLSTM) to the AII task, Zhu, Xie, and Chen (2015) reduced the RMSE to 0.565 mm on the mngu0 corpus, compared to the RMSE of 0.885 mm achieved by TMDN models in Uria, Renals, and Richmond (2011) and Uria et al. (2012). Liu et al. (2015) suggested that, with LSF as acoustic features for AAI, BiLSTM performed better than deep recurrent mixture density network (DR-MDN). It is worth mentioning that Zhu, Xie, and Chen (2015) also concluded that MFCC outperformed LSF for AAI tasks despite the fact that LSF is a kind of more articulatory-originated feature.

2.2 Computational Speech Acquisition

Computational speech acquisition (CSA), also known in other studies as sensorimotor learning, (imitative) vocal learning, and vocal/speech imitation, refers to the development of the ability to imitate the articulatory process, acquire new sounds, and produce vocalizations. It is inspired by how humans learn to speak and involves three primary spaces: motor, sensory, and internal spaces. These correspond to the articulatory, acoustic, and perceptual spaces in generic speech inversion tasks, respectively. Motor commands (i.e., articulatory configurations) generate a sensory consequence (i.e., acoustic signals) which is then converted to a neural representation (i.e., processed features) in the perceptual space. An inversion framework connecting these three spaces trains the learning agent like an infant by awarding the vocal actions whose acoustic output is positively evaluated; this mimics the process by which an infant's vocalization is encouraged by a caregiver.

A growing number of computational approaches to speech acquisition have been investigated in recent years. A comparative review of the studies involving vocal imitation by sensorimotor learning models can be found in Pagliarini, Leblois, and Hinaut (2020). In this review, the authors collated the research about song learning in birds and speech acquisition in humans into a general topic. In addition to an introduction to the biological context of vocal learning, they reviewed the models in terms of the motor control device (e.g., how the models mapped muscle commands to the representation of sound); sensory system (i.e., the perceptual representation of sensory stimuli); and the learning framework, which included the architecture, learning rules, the exploration strategies, and the evaluation measures. In this study, we summarizes the representative studies and researchers that have made great or continuous contributions to CSA (Table 2.3).

TABLE 2.3: Summary table of literature on computational speech acquisition (CSA). The "SI" in the second column indicates whether the study is speaker-independent, i.e., "Y" for SII and "N" for SDI.

	SI	**Utterance**	**Synthesizer (No. of involved parameters)**	**Acoustic representation**	**Inversion algorithm**	**Evaluation**
Howard and Huckvale, 2005	Y	5 English vowels and /b/ and /g/ and their combinations	Maeda synthesizer (9)	JSRU vocoder features	Multilayer perceptron	Subjective comparison of spectrogram and informal listening test
Howard and Messum, 2007	N	8 English vowels	Maeda synthesizer (9)	Salience	Self-exploration and reinforcement	Informal listening test
Howard and Messum, 2011	N	English Vs, CVs, VCs and CVVs	Maeda synthesizer (10)	Salience and diversity	Self-exploration and reinforcement	Informal listening test
Philippsen, Reinhart, and Wrede, 2014	N	German syllables	VocalTractLab (22)	MFCC	ESN	MSE in acoustic domain, informal listening test
Philippsen, Reinhart, and Wrede, 2015	N	German /a/, /i/, /u/, /e/	DIVA version of Maeda synthesizer(10)	Cochleograms	Radial basis function network	Average reproduction rate of cluster centers
Murakami et al., 2015	N	German /a/, /i/, /u/	VocalTractLab (16)	DRNL filterbank features	ESN, reinforcement learning	Formant similarity
Philippsen, Reinhart, and Wrede, 2016	N	German /a/, /i/, /u/, /e/, /o/	VocalTractLab (17)	MFCC	Radial basis function network	Competence of acoustic reproduction
Howard and Birkholz, 2018	N	14 German vowels	VocalTractLab (14)	Filter-bank features	Multilayer perceptron	Subjective comparison of spectrogram and informal listening test
Howard and Birkholz, 2019	N	8 German vowels	VocalTractLab (14)	Salience, diversity and effort	Self-exploration and reinforcement	Subjective comparison of spectrogram and informal listening test
Shitov, 2020	N	English vowel-to-vowel	VocalTractLab (23)	MFCC	Soft-DTW-triplet, reinforcement learning	Soft-DTW distance
Philippsen, 2021	N	German /a/, /i/, /u/, /e/, /o/, /ba/, /ma/	VocalTractLab (21)	MFCC	Radial basis function network	Euclidean distance in acoustical goal space and articulatory space

Howard and his colleagues used computational models to conduct a series of studies on speech acquisition. They divided speech acquisition into two phases: the babbling phase and the reinforcement phase. The babbling phase has two purposes. The first purpose was to train the agent/infant (i.e., the articulatory synthesizer) to discover potentially useful articulatory configurations; this was done in an unsupervised manner. For example, the motor patterns (i.e., the articulatory configurations) were first generated through vocal self-exploration. The corresponding speech was subsequently synthesized via acoustic simulation. The quality of the synthetic sounds was evaluated using acoustic metrics, such as salience and diversity (Howard and Messum, 2007; Howard and Messum, 2011; Howard and Birkholz, 2019). Salience is a sensor metric that measured the quality of discovered sound. It is calculated as the weighted sum of the acoustic power, touch contact, and spectral balance. In contrast, diversity is a motor metric that measured how diverse the new motor pattern and its acoustic and tactile sensory consequences were compared to those of previously discovered motor patterns; it is calculated using the weighted sum of motor diversity, tactile diversity, and sensory diversity (Howard and Messum, 2011). Motor patterns with high scores or low cost were used to develop the babbling ability of the agent/infant. The other purpose of the babbling phase is to create training samples for the reinforcement phase. For example, the predefined articulatory configurations were fed into the synthesizer to produce synthetic speech. The articulatory configurations and acoustic features extracted from the corresponding synthetic speech constituted the articulatory-acoustic samples. In the reinforcement phase, the agent/infant reinforced its ability to produce speech-like sounds by interacting with caregivers. Alternatively, a neural network (i.e., the inverse model) could be trained by using acoustic features as its inputs and articulatory configurations as its targets (Howard and Huckvale, 2005; Howard and Birkholz, 2018). Following these two phases, articulatory configurations could be obtained by imitating the given speech or feeding the acoustic features into the trained inverse model.

Philippsen and her colleague have also been making continuous contributions to CSA. Philippsen, Reinhart, and Wrede (2014) implemented an articulatory-acoustic model where the forward and inverse mappings were learned by an echo state network (ESN). The model was initially trained on a small set of training samples before being improved by the imitation-based refinement, which allowed for syllable production. They also investigated goal-directed babbling methods, inspired by infants' speech learning, for vowel acquisition with an active selection of targets (Philippsen, Reinhart, and Wrede, 2015) or adaptive exploration noise (Philippsen, Reinhart, and Wrede, 2016). Their recent approach to CSA consisted of two phases. The first phase established a low-dimensional goal space from ambient speech. The second phase involved an iterative loop of goal-directed babbling in which an inverse model that mapped the acoustic goal space to the articulatory space was learned via exploration and adaptation steps. The final model successfully learned how to produce five German vowels as well as the /ba/ and /ma/ syllables (Philippsen, 2021).

Shitov, Pirogova, and Lech (2018) assessed four types of acoustic features (formant, MFCC, MFCC-PCA, and the hidden states of neural networks with MFCC as input) in modeling speech acquisition, and concluded that the features that were automatically learned by convolutional neural networks using LSTM layers outperformed all other features. In his PhD thesis, Shitov (2020) also designed neural network based CSA models for static vowel and vowel-to-vowel imitation. He proposed the Soft-DTW-triplet model for learning distinctive features for word discrimination tasks, the training data for which were composed of reference, positive, and negative samples. Soft dynamic time warping (Soft-DTW) distance was incorporated into the loss function to train the triplet neural network models. The trained model was then used as a feature extractor in a speech imitation task where reinforcement learning was employed to learn the policies that controlled the vocal tracts of VocalTractLab as well as the dynamics of speech production. However, this study involved only five English vowels (including isolated vowels and their vowel-vowel combinations). Furthermore, these proposals were validated using speaker-dependent data; in other words, both the training and testing utterances were created using the same synthesizer.

Unlike AAI, which uses articulatory-acoustic data recorded in parallel, CAS uses an articulatory synthesizer as the learning agent. Another difference is that CSA is normally used to investigate the acquisition of short utterances (see the "Utterance" column of Table 2.3). Different acoustic features inspired by biological research are also used in CSA, such as the cochleograms used in Philippsen, Reinhart, and Wrede (2015) and the dual resonance nonlinear (DRNL) filterbank features used in Murakami et al. (2015). In addition, all studies, with the exception of the study by Howard and Huckvale (2005), developed speaker-dependent (SD) CSA systems. The ambient speech used as the learning reference is often created in advance using the same synthesizer. In the context of speech acquisition, this means that the infant and the caregiver were acoustically the same. Howard and Huckvale (2005) suggested that the main limitations in the performance of resynthesis systems were not due to convexity problems or the lack of goal-directed learning, but were more likely to be due to normalization issues between the CSA system and human speakers.

2.3 Articulatory Copy Synthesis

ACS differs from AAI and CSA because it emphasizes the exact reproduction of reference utterances, and involves both the physiological articulation process and its corresponding acoustic results. In addition to obtaining the estimated articulatory process from acoustic signals (as is done in AAI), the term "synthesis" in ACS emphasizes the resynthesis of an acoustic duplicate that matches the original utterance as closely as possible. Similarly, ACS involves the acquisition of natural and intelligible speech like in CSA; however, the term "copy" in ACS emphasizes the temporally exact imitation of the target speech. Hence, the results produced by ACS generally encompass those obtained by AAI or CSA.

Table 2.4 summarizes the representative literature on ACS. Like CSA, ACS also needs an articulatory synthesizer; the only exception is the work of Richmond, Ling, and Yamagishi (2013) who used a statistical synthesis model. ACS performance can be evaluated using reference utterances in the acoustic, articulatory, and perceptual domains since the results contain both estimated articulatory trajectories and reproduced acoustic signals.

Since the "copy" ability of ACS was achieved by reproducing reference speech, Prom-on and his colleagues attempted to use ACS to find the underlying articulatory configurations of Thai vowels (Prom-on, Birkholz, and Xu, 2013; Prom-on, Birkholz, and Xu, 2014). They used neutral positions of the articulators as the starting point for articulatory parameters of VocalTractLab. Specifically, the articulatory configuration of the “schwa” sound was used since it was closest to the neutral positions of articulators. They found the optimal articulatory configurations of Thai vowels using stochastic gradient descent; these configurations were subsequently fed into VocalTractLab to synthesize acoustic signals. The results demonstrated excellent agreement between the reproduced and reference utterances in both the acoustic and articulatory domains. A listening test was also used to evaluate the perceptual quality of the reproduced speech. Their ACS framework was refined by using a particle swarm optimization (PSO) algorithm to speed up the search process (Fairee, Sirinaovakul, and Prom-on, 2015).

Table 2.4: Summary table of literature on articulatory copy synthesis (ACS). The "SI" in the second column indicates whether the study is speaker-independent, i.e., "Y" for SII and "N" for SDI.

	SI	**Utterance**	**Synthesizer (No. of involved parameters)**	**Acoustic representation**	**Inversion algorithm**	**Evaluation**
Dang and Honda, 2002	Y	Japanese vowels	Dang & Honda synthesizer (9)	Formants	Difference method	Acoustic and articulatory distances
Panchapagesan and Alwan, 2011	N	11 English vowels or diphthongs	Maeda synthesizer (7)	Formants and cepstra	Codebook and quasi-Newton	Acoustic and articulatory distances
Nam et al., 2010; Nam et al., 2012	Y	English sentences	TADA synthesizer (8)	Log spectra and linear prediction spectra	DTW	Spectral distance and Correlation
Aryal and Gutierrez-Osuna, 2013	N	English sentences	GMM statistical synthesizer (6)	MFCC	RBF network	Correlation, RMSE, Mel cepstral distortion, and listening test
Richmond, Ling, and Yamagishi, 2013	N	English sentences	mngu0 and FSS-MRHMM synthesizer (12)	LSF	Linear projection, Codebook, MLP, and TMDN	Correlation and RMSE, LSF RMSE and listening test
Prom-on, Birkholz, and Xu, 2013	Y	English: 5 vowels and a word; German: 2 four-syllable utterances	VocalTractLab (18)	MFCC	Stochastic gradient descent	Correlation of formants and EMA trajectories
Prom-on, Birkholz, and Xu, 2014	Y	81 Thai vowel-vowel sequences	VocalTractLab (18)	MFCC	Stochastic gradient descent	Sum of Squared error (SSE), listening test
Fairee, Sirinaovakul, and Prom-on, 2015	Y	5 Thai vowel-vowel	VocalTractLab (15)	MFCC	PSO	Sum of Squared error (SSE)
Gao, Stone, and Birkholz, 2019	Y	41 German words	VocalTractLab (2 of each gesture)	MFCC, voicing of probability	Genetic algorithm	Acoustic distance, listening test
Gao, Steiner, and Birkholz, 2020	Y	German sentences	VocalTractLab (30)	MFCC, voicing of probability	LSTM	RMSE, correlation, word recognition accuracy
Sun and Wu, 2020	Y	English and Chinese sentences	TRM (42)	Mel-spectrograms	Convolutional BLSTM	Signal to noise ratio (SNR)

Unlike articulatory trajectories, articulatory processes can be organized at a higher level. For example, articulatory phonology defines the realization of a word as a constellation of vocal-tract constriction actions, referred to as gestures. The realization of an utterance involving multiple gestures is organized by a gestural score. Therefore, another type of ACS is the estimation of gestural scores from acoustic signals. This kind of speech inversion is based on the ABS procedure, in which inversion is conducted by iteratively adjusting the articulatory parameters (e.g., the duration, timing, and targets of gestures) to minimize the acoustic distance between the resynthesized and original utterances. Nam et al. (2012) proposed a framework for estimating gestural scores from acoustic signals. They first fed the canonical transcriptions of utterances into a linguistic gestural model which generated initial gestural scores using a segment-to-gesture dictionary and a set of syllable-based inter-gestural coupling (phasing) principles. The initial gestural scores were then refined via the ABS procedure which aimed to minimize the DTW-based spectral distance between synthetic speech generated with the gestural score and the reference speech. However, their work made two strong assumptions: (1) The content of utterances (e.g., the canonical transcription) was assumed to be known in advance; (2) Only the durations and relative timings of gestures were allowed to vary while the dynamical parameters (target and stiffness) of gestures were held constant from instance to instance.

In our previous study (Gao, Stone, and Birkholz, 2019), we also proposed an ACS approach for the estimation of gestural scores from acoustic signals using VocalTractLab. For a given natural utterance, an appropriate gestural score was obtained in two steps: initialization and optimization. In the first step, a rule-based method was employed to create an initial gestural score. This initial gestural score was optimized in the second step by using a genetic algorithm such that the cosine distance between the acoustic features of the synthetic and natural utterances was minimized. This study made assumptions similar to the first assumption presented in Nam et al. (2012). However, in addition to the duration and timings of gestures, "time constant" (a dynamical parameter similar to “stiffness”) was also estimated. Depending on the optimized values of the “time constant” parameters, this method allowed the articulators to approach their underlying target positions with different degrees, which accounted for the mechanism of coarticulation.

We also investigated ACS using VocalTractLab and an LSTM neural network based regression model, which were responsible for the articulatory-to-acoustic mapping and its inversion, respectively (Gao, Steiner, and Birkholz, 2020). We used a rule-based method to create gestural scores from texts, which were converted to articulatory trajectories and subsequently synthesized to produce the corresponding acoustic signals. To make the subsequent mapping more robust, the acoustic and articulatory spaces were expanded by manipulating the speaking effort, voice quality, pitch level, and vocal tract length of the created gestural scores or acoustic signals, producing 81 variants for each utterance. With acoustic features as inputs and articulatory trajectories as targets, we trained the LSTM models on the acoustic-to-articulatory inversion process. By providing the trained LSTM model with acoustic features of testing utterances, the estimated articulatory trajectories could be obtained; these were subsequently fed into VocalTractLab to acquire reproduced speech.

Sun and Wu proposed an embodied self-supervised learning (EMSSL) algorithm and successfully applied it to speech inversion (Sun and Wu, 2020). The algorithm used an ABS-based procedure to train a neural network that could infer articulatory parameters from speech signals. Instead of preparing all training data in advance before exclusively training the neural networks, they integrated the data sampling and neural network training into a single iterative framework. During the sampling step, the acoustic features of natural utterances served as the inputs for the neural networks. The outputs were sampled articulatory parameters which were subsequently fed into the Tube Resonance Model (TRM) articulatory speech synthesizer (Hill, Taube-Schock, and Manzara, 2017). In the training step, the sampled articulatory parameters and corresponding acoustic features of the synthetic TRM speech were used to update the weights of the neural networks. These two steps were performed alternatingly to obtain the final neural networks which could estimate articulatory parameters of reference utterance with arbitrary length and generalize well to unseen speakers or even new languages. However, they used the signal-to-noise ratio (SNR) of the mel-spectrograms between the synthetic and reference utterances as the metric of similarity. This work did not evaluate the reproduced speech in terms of intelligibility and naturalness, nor did they evaluate its similarity from an articulatory perspective.

In addition, some studies also attempted to introduce articulatory constraints to speed up the ACS process or make the estimated articulatory process more physiologically plausible. To this end, the regularization term, usually calculated as the sum of squares of articulatory parameters minus their nominal values, was designed to prevent vocal tract parameters from significantly deviating from their mean or neutral position. The continuity or smoothness term, usually calculated as the sum of squares of the first order time-derivatives of articulatory parameters, was designed to generate smoother articulatory trajectories. For example, in addition to the acoustic distance between the synthetic and reference utterances, Panchapagesan and Alwan (2011) incorporated the regularization and continuity terms into the loss function to address the non-uniqueness problem and increase the smoothness of estimated articulatory trajectories. Besides, to address the problem of nonlinear one-to-many mapping, Dang and Honda (2002) incorporated a physiological constraint (specifically, the quantitative relationship between frequency difference of the first two formants and tongue dorsum position) into the inversion procedure.

It should be mentioned that there is another type of ACS that uses articulatory data concurrently recorded together with the acoustic signals. For example, Steiner and Richmond (2009) and Steiner (2010) used EMA data as reference to resynthesized utterances, including German CV sequences and sentences, by creating gestural scores using the VocalTractLab synthesizer. Laprie et al. (2013) and Elie and Laprie (2016) derived glottal source parameters such as f_0, voicing feature, and formants, from original acoustic signals and nominal values found in literature; they also derived area functions from X-ray films. By using the Maeda synthesizer, they then accurately reproduced the formant trajectories as well as the prosodic and phonetic contrasts of the original utterances. Although this type of methods rely on articulatory data recorded in advance for each utterance to be reproduced, the findings and other modules of those studies may be beneficial to other speech inversion investigations.

2.4 Concluding Remarks

This chapter introduced three kinds of speech inversion, together with selected representative studies, from many aspects, mainly including tested utterances, articulatory data, acoustic features, articulatory synthesizers used for CSA and ACS, inversion algorithms, and evaluation metrics.

AAI relies on paired articulatory-acoustic datasets in which the articulatory movements and corresponding acoustic signals are simultaneously recorded. The primary downsides of speech inversion based on articulatory data are that data collection is expensive and laborious, and even sometimes invasive to subjects, especially when it involves the recording and labeling of data. In addition, since massive articulatory recordings can be too expensive to obtain, the AAI models are trained with data from a small number of subjects; hence these models are not robust to utterances from unknown speakers. Furthermore, methods that rely on recorded articulatory data have limited applications, especially in scenarios that require real-time responses. Therefore, methods that do not require recorded articulatory data, and that use synthetic data generated by articulatory synthesizers instead, are becoming increasingly prevalent.

CSA is usually conducted on short utterances and in a speaker-dependent fashion. The speech of the caregiver, regarded as ambient speech, is usually synthesized by the same synthesizer. To simulate the acoustic variations of the same phonemes produced by different speakers in the real world, researchers usually add random noise to predefined vocal tract configurations while keeping the anatomy of the model speaker constant before synthesizing different variants for each phone. However, this deviates significantly from real-world scenarios because the acoustic characteristics of human speech vary considerably from speaker to speaker, much less the difference between synthetic and natural speech.

ACS appears to be more challenging since it not only estimates articulatory processes but also reproduces acoustic duplicates for the given utterances. However, it mitigates the limitations of AAI and CSA to some extent by relying on the power of articulatory synthesizers and a large collection of synthetic articulatory-acoustic samples. In addition, ACS inherently provides more articulatory information than AAI and CSA. Most inversion studies based on recorded articulatory data only estimate the articulatory processes of the supraglottal region, i.e., vocal tract parameters. For example, since X-ray pellets or EMA coils are usually placed either in the oral cavity or on the face, AAI studies based on such recordings cannot provide information about the laryngeal region. For CSA studies involved in the learning of vowel production, the configuration of the vocal cords is usually set to that of the modal phonation in advance and is fixed during the learning process. Lacking articulatory information about the glottal region may limit their application to scenarios introduced in Section 1.1.

It is important to note that there is no real distinction between AAI, CSA, and ACS. Although these three research topics have different motivations, aims, and terminologies, they share many similar technical modules, such as the acoustic and articulatory representations of utterances and inversion algorithms. Generally speaking, most of the modules that are used in AAI or CSA tasks are also involved in ACS tasks. It should also be mentioned that the performance of different methods reviewed in this study are not directly comparable unless they use the same datasets and are evaluated using the same metrics. In general, speech inversion for longer utterances like sentences and phrases is generally more difficult than shorter ones like words and isolated vowels. Speaker-independent tasks are more difficult than speaker-dependent ones. Speech inversion to reproduce a target sound is more difficult than the one without that aim. Normally, speech inversion is also more difficult when more articulatory parameters are involved, e.g., larger degrees of freedom for articulatory synthesizers.

ABS-based approaches tend to be computationally expensive. Therefore, initialization strategies are used to speed up the optimization and search procedures, such as goal-directed babbling for CSA or rule-based methods for creating initial gestural scores for ACS. The use of regularization and continuity terms in the loss function is also effective at resolving the issues related to non-uniqueness and local optima in ABS-based inversion. This study investigated the use of an ABS-based method in conjunction with a genetic algorithm for ACS (this is presented in full in Chapter 4). It was further developed to not require the utterance transcriptions in advance. A new adaptive regularization method was also tested.

Another solution is to directly model the mapping of acoustic features to articulatory trajectories. To train such inversion models, a large number of training samples are required. In addition, the training samples need to be diverse enough to cover as much of the acoustic and articulatory space as possible. The review of the literature suggests that neural network based methods generally outperform traditional methods for training such mapping models. Among them, neural networks with a recurrent architecture are capable of modeling the temporal dependence of acoustic and articulatory data, allowing them to produce smoother articulatory trajectories for testing utterances. In addition, acoustic features processed by the neural networks with convolutional layers tend to be more distinctive for speech inversion. Therefore, LSTM neural networks as well as convolution LSTM neural networks, were examined for ACS (Chapter 5).

The speech produced by CSA and ACS can be evaluated by using perceptual experiments. Nevertheless, only a few studies used listening tests to assess the quality of their reproduced speech. Therefore, Chapter 5 presents the results of a perceptual experiment in which human listeners were recruited to rate the quality of the reproduced speech through a recognition task. In addition, the ACS systems built with neural network regression models were expected to be applicable to foreign languages, as articulatory synthesis is considered to be relatively language-independent compared to other synthesis techniques. Therefore, Chapter 5 also examined the generalization ability of the proposed method. More specifically, the ACS systems trained with German data were used to reproduce utterances of two foreign languages: English and Mandarin Chinese.

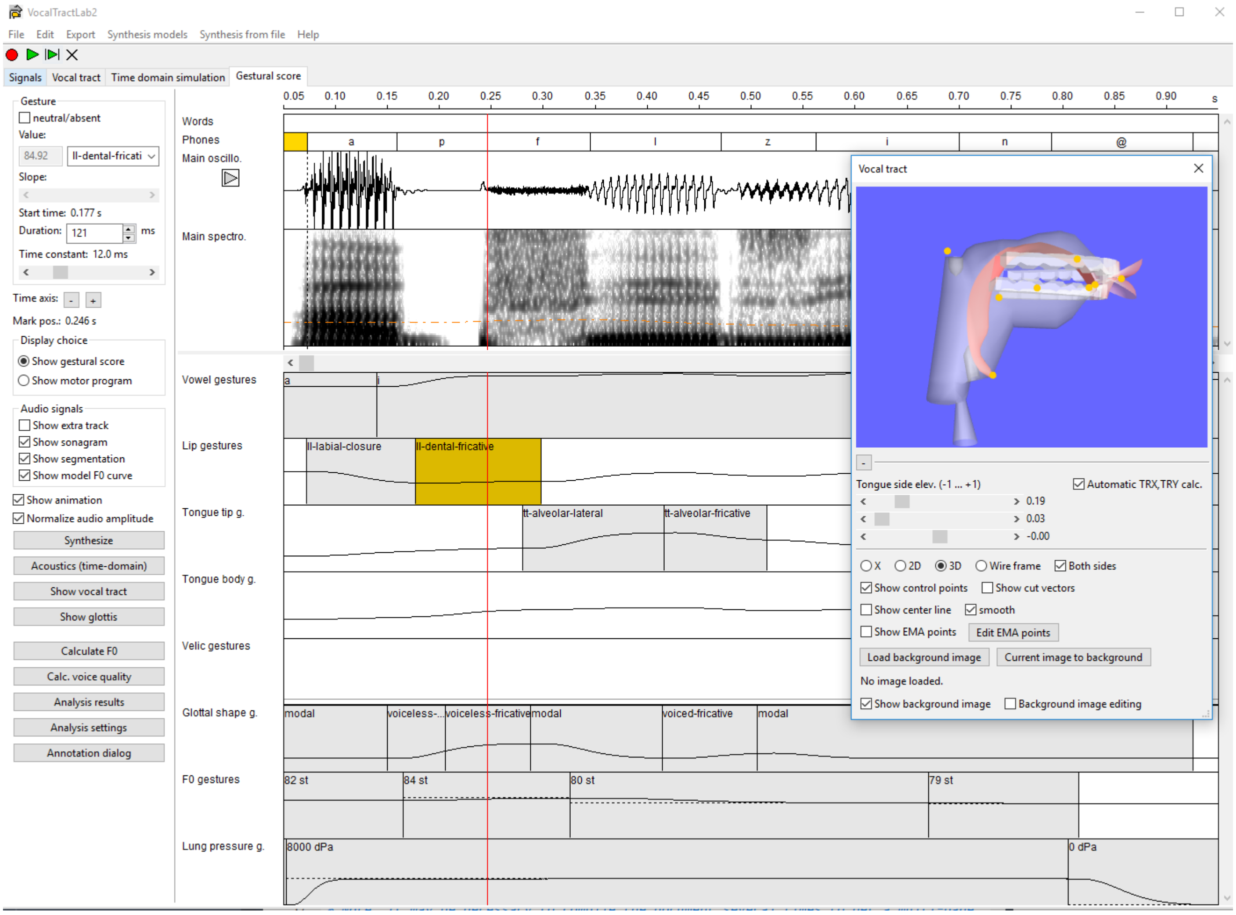
Chapter 3

Fundamentals

Before elaborating upon the ACS methods proposed in this study, some fundamentals must be introduced which will allow for the proposals to be validated experimentally. Section 3.1 introduces VocalTractLab, the articulatory speech synthesizer used in this study. It is the medium through which the articulatory and acoustic domains are connected. All articulatory parameters involved in this study adhere to the formats defined in VocalTractLab. Estimating articulatory parameters of VocalTractLab and reproducing acoustic signals for given utterances constitute the main body of this thesis. However, it should be mentioned that none of the methods proposed in this study are constrained to any specific articulatory synthesizers or datasets. Section 3.2 presents the datasets used to train and/or evaluate the speech inversion models. Section 3.3 introduces the acoustic features used in this study to represent utterances in the acoustic domain. Section 3.4 lists the main computation platform, software, and tools used to conduct the experiments. Section 3.5 provides some concluding remarks.

3.1 Articulatory Speech Synthesizer VocalTractLab

As reviewed in Section 2, lots of studies used articulatory speech synthesizers for speech inversion. Articulatory synthesizer produces human-like speech by modeling human vocal tract and vocal folds and simulating articulatory, phonatory, and control processes involved in speech production. Hence, it is used as the forward model in CSA and ACS, mapping articulatory parameters to acoustic signals. The commonly used synthesizers include the Maeda synthesizer (Maeda, 1990) as well as its implementation variant (Tourville and Guenther, 2011), Dand & Honda synthesizer (Dang and Honda, 2001), Task Dynamics and Application (TADA) model (Nam et al., 2004), Tube Resonance Model (TRM) (Hill, Taube-Schock, and Manzara, 2017), and VocalTractLab (Birkholz, 2013). Among these articulatory speech synthesizers, VocalTractLab (see Figure 3.1) is the most complete, flexible, feature-rich, and continuously developed one, which makes it the widely used articulatory synthesizer in speech inversion tasks, especially in recent years. Therefore, VocalTractLab was adopted as the synthesizer in this study.



VocalTractLab2

Gesture

□ neutral/absent

Value: 84.92 ll-dental-fricati

Slope:

Start time: 0.177 s

Duration: 121 ms

Time constant: 12.0 ms

Time axis: - +

Mark pos.: 0.246 s

Display choice

● Show gestural score

○ Show motor program

Audio signals

□ Show extra track

☑ Show sonagram

☑ Show segmentation

☑ Show model F0 curve

☑ Show animation

☑ Normalize audio amplitude

Vocal tract

Tongue side elev. (-1 ... +1) ☑ Automatic TRX,TRY calc.

0.19

0.03

-0.00

○ X ○ 2D ● 3D ○ Wire frame ☑ Both sides

☑ Show control points □ Show cut vectors

□ Show center line ☑ smooth

□ Show EMA points

No image loaded.

☑ Show background image □ Background image editing

FIGURE 3.1: The screenshot of VocalTractLab graphical user interface showing the gestural score page in the back with the 3D vocal tract model in front of it.

VocalTractLab (VTL) was originally developed by Birkholz (2005) and later continuously refined by Birkholz and his colleagues (Birkholz, Kröger, and Neuschaefer-Rube, 2011; Birkholz, 2013; Birkholz et al., 2017). VTL implemented various models, including a 3D vocal tract model representing the supraglottal airways and articulators, vocal fold models providing source excitation, and acoustic simulation models. The 3D articulatory defines the vocal tract and uses 19 degrees of freedom (vocal tract parameters) to control articulator states. VTL implemented three vocal fold models, among which the geometric model (Birkholz, Drechsel, and Stone, 2019) generally proved to provide a better synthesis quality. The geometric model is hence selected as the vocal fold model for this study. The articulatory parameters modeled in VTL are listed in Table 3.1.

TABLE 3.1: Articulatory parameters of VocalTractLab (version 2.3).

Model	Parameter	Description	Min.	Max.	Unit
Vocal tract model	HX	Hyoid horizontal position	0	1	cm
	HY	Hyoid vertical position	-6	-3.5	cm
	JX	Jaw position	-0.5	0	cm
	JA	Jaw angle	-7	0	deg
	LP	Lip protrusion	-1	1	cm
	LD	Lip (vertical) distance	-2	4	cm
	VS	Velum shape	0	1	-
	VO	Velum opening	-0.1	1	-
	TCX	Tongue body center horizontal position	-3	4	cm
	TCY	Tongue body center vertical position	-3	1	cm
	TTX	Tongue tip horizontal position	1.5	5.5	cm
	TTY	Tongue tip vertical position	-3	2.5	cm
	TBX	Tongue blade horizontal position	-3	4	cm
	TBY	Tongue blade vertical position	-3	5	cm
	TRX	Tongue root horizontal position	-4	2	cm
	TRY	Tongue root vertical position	-6	0	cm
	TS1	Tongue (root) side elevation	0	1	cm
	TS2	Tongue (back and dorsum) side elevation	0	1	cm
	TS3	Tongue (tip and blade) side elevation	-1	1	cm
Vocal fold model	f0	Fundamental frequency	40	600	Hz
	pressure	Subglottal pressure	0	20000	dPa
	x_bottom	Vocal fold lower displacement	-0.05	0.3	mm
	x_top	Vocal fold upper displacement	-0.05	0.3	mm
	chink_area	Chink area	-0.25	0.25	mm^2
	lag	Phase lag	0	3.1415	deg
	rel_amp	Relative amplitude	-1	1	-
	double_pulsing	Double pulsing	0	1	-
	pulse_skewness	Pulse skewness	-0.5	0.5	-
	flutter	Flutter	0	100	%
	aspiration_strength	Aspiration strength	-40	0	dB

There are 30 articulatory parameters in total (19 for vocal tract model and 11 for vocal fold model) if the geometric vocal fold model is used. Since TRX and TRY are usually calculated by linear regression from the remaining parameters, there are only 17 actual degrees of freedom for the vocal tract model. For each parameter, a sequence of samples (with an interval of 2.5 ms) over time constitutes an articulatory trajectory. VTL takes these articulatory trajectories as inputs and simulates acoustic signals as outputs. Therefore, ACS can be regarded as its inversion process that, for a given utterance, estimates the underlying articulatory trajectories whose corresponding VTL-synthesized speech matches the original one as closely as possible. In addition to the elementary level representation of articulatory configurations (i.e., the 30 control parameters), VTL also provides a higher level organization format, *gesture*, originally developed in the framework of articulatory phonology (Browman and Goldstein, 1992). According to articulatory phonology, gestures are the fundamental units that constitute an utterance. In VTL, a gesture describes the movement of participating articulators/parameters toward a target configuration of the vocal tract model or the vocal fold model.

Gestural score is an organization pattern of all gestures for an utterance to indirectly control the articulatory process. The realization of each phoneme is cooperatively governed by multiple gestures, each of which consists of three parameters (Birkholz, Steiner, and Breuer, 2007): a gesture *value*, a *duration*, and a *time constant*, which define target positions of articulators, their duration, and how quickly the participating articulators reach the targets (i.e., speaking effort), respectively. All involved gestures for the realization of an utterance are distributed over eight tiers of a gestural score. As shown in Figure 3.1, from top to bottom, they are vowel, lip, tongue tip, tongue body, velic, glottal shape, f_0, and lung pressure tiers. All gestures that are temporally arranged and coordinated over these tiers constitute a gestural score. In fact, the gestural score will be internally converted to articulatory trajectories in VTL, because the motions of articulators in response to discrete gestures are controlled by linear dynamical systems, thus producing articulatory trajectories along the time-axis (Birkholz, Kröger, and Neuschaefer-Rube, 2011). Subsequently, acoustic signals are simulated by a time-varying branched acoustic tube system.

3.2 Corpora

In order to evaluate the effectiveness of the proposed approaches, two corpora were deliberately designed. One was a speech corpus of **p**honemically **b**alanced **a**nd **c**ommonly **u**sed German words, referred to as PBACU corpus, which enabled to use least utterances to cover all German phonemes. This corpus contained the acoustic signals of a set of 160 German words produced by two male native speakers. In this study, the PBACU corpus was used as the target utterances for ACS in Chapters 4 and 5. Another one was the paired articulatory-acoustic dataset, referred to as VTL-Kiel corpus, which was created using VocalTractLab and Kiel Corpus. Together with the original natural utterances of Kiel corpus, the VTL-Kiel corpus was used to train and evaluate the neural network based ACS systems in Chapter 5.

3.2.1 PBACU Corpus

Although there are a lot of speech corpora available, they were usually designed for specific tasks, such as speech recognition, speech synthesis and speaker identification. They are not proper to evaluate the proposed method in Chapter 4 for several reasons. First, the method in Chapter 4 relies on the ABS-based procedure in which reproducing a given utterance is very time-consuming. The iteration-based method usually takes several minutes, on the condition of using multiple CPU cores, to finish the optimization of VTL parameters for a word. The longer the utterance is, the more gestures or articulatory parameters it has, and thus the more time it takes to optimize. Besides, the utterances produced in a spontaneous style are highly co-articulated, i.e., the phones are usually realized in a reduced way. Using such utterances with complicated linguistic phenomena makes it hard to focus on the algorithm itself. In this study, therefore, a speech corpus of phonemically balanced and commonly used German words, referred to as PBACU corpus, was constructed to validate the proposed methods. The corpus should be succinct, i.e. it should have as few utterances as possible, so that it can save not only the cost of recording speech but also the time of running algorithms. Furthermore, it should have a good coverage of phonemes and also be phonemically balanced so that it can provide enough phoneme combinations to fully analyze the performance of the algorithms.

Word List Design

First, a word list together with the canonical phonetic transcription was designed. The candidate words came from two sources: the pronunciation dictionary of the open source German ASR system proposed in Radeck-Arneth et al. (2015); the words together with their pronunciations used in the phonetically balanced BITS corpus (Ellbogen, Schiel, and Steffen, 2004). The words as well as their pronunciations from these two sources were merged before selection. Even though the proposed approaches can be applied to optimize any complex utterances in principle, in this study only two- and three-syllable words that contained "CV" or "CVC" syllables were used since their gestures or parameters could be optimized with acceptable computational effort and the results allowed us to analyze the effectiveness of proposed methods. Therefore, two lexicons were created from the merged candidate word list with canonical pronunciations: one for two-syllable words and another for three-syllable words. Next, for each lexicon, the entries of commonly used words with the frequency level higher than 15, by referring to the frequency-based ranking list "derewo-v-ww-bll-320000g-2012-12-31-1.0[1]", were selected, yielding a new lexicon of commonly used words, which was used as the initial lexicon in the next step.

After that, the minimal set of words covering all German phonemes at least once were further selected using a least-to-most-ordered algorithm (Wu et al., 2016). Assume that L was the initial lexicon containing all words for selection, where $W = \{w_1, ..., w_N\}$ and N was the number of candidate words. Let T is the target word set and $P = \{p_1, ..., p_M\}$ was the phoneme set to be covered by the pronunciations of the target words. Starting from the initial lexicon, a set of target words were selected as follows.

- Step-1 for initialization: The sub-list for each target phoneme $W(p_j) = \{w_1{}^j, ..., w_{n^j}{}^j\}$ was constructed in which the pronunciation of the word w_i^j contained the target phoneme p_j at least once and n^j was the number of the words containing this target phoneme in the lexicon L.

- Step-2 for phoneme selection: The phoneme with least frequency occurring in the lexicon L was selected,

$$p^* = \arg\min_{p_j \subset P} f(p_j) \tag{3.1}$$

  where $f(p_j)$ denoted the frequency of phoneme p_j occurring in the lexicon L.

- Step-3 for word selection: The word with highest score from the sub-list for the phoneme p^* was selected,

$$w^* = \arg\max_{w_i \subset W(p^*)} s_i \tag{3.2}$$

  where the score s_i was the entropy of phonemes when the word w_i was inserted into the target word set.

- Step-4 for update of target word and phoneme sets and the lexicon:

[1] https://www1.ids-mannheim.de/s/corpus-linguistics/projects/methods-of-analysis/corpus-based-lemma-and-word-form-lists.html?L=1

$$L \leftarrow \{L - w^*\} \tag{3.3}$$

$$T \leftarrow \{T + w^*\} \tag{3.4}$$

$$P \leftarrow \{P - P_{w^*}\} \tag{3.5}$$

where P_{w^*} was the phoneme set covered by the pronunciation of the word w^*.

- Step-5 for iteration: The above four steps were repeated until the target phoneme set P was empty.

Performing the above steps separately for each initial lexicon, the least-to-most-ordered algorithm selected a minimal set of words covering all German phonemes at least once. In Step-2, the phoneme least-frequently occurring in the lexicon was always preferentially selected. This selection criterion tended to select the words whose corresponding phonemes were hardest to cover, which conduced to the smallest target word set. In Step-3, the word was preferentially selected when the phoneme entropy of resulting target word set was highest. This entropy-based criterion tended to make the phoneme set of all selected words more phonemically balanced.

To cover more combinations of different consonants and vowels, this above selection procedure was repeated five time separately for the two-syllable word lexicon and three-syllable word lexicon. In addition, to examine the capability of the proposed method for dealing with consonant clusters, extra 20 words were manually selected, such as the words "zuviel", "Prototyp" and "Campingplatz" containing [ts], [pr] and [pl], respectively. Finally, these words selected from two lexicons were merged into one union set, resulting in a word list containing 160 words (80 two-syllable words and 80 three-syllable words), which is listed in the Appendix A.

Speech Recording

In order to make sure that the words were produced with a natural way, all the target words were further embedded in the carrier sentence *"Ich habe <word> gesagt"* (English: *"I have said <word>"*). By means of the carrier words, the articulators were placed in the resting position before and after target words. Two German male speakers (SPK-1: 31 years old; and SPK-2: 26 years old) produced the set of 160 words. They were native Germans studying in Chair of Speech Technology and Cognitive Systems of TU Dresden and had normal speech and hearing functions with no history of any communication disorders. The recordings took place in the Audio-Studio of Institute of Acoustics and Speech Communication equipped with a recording console (Behringer Eurorack MX1602). The studio microphone (Microtech Gefell M930) was placed at a distance of approximately 20 cm from the speaker's mouth. After a brief instruction and practice, the speaker was asked to read all of the 160 sentences containing the target words. All utterances were recorded with a sampling rate of 22.05 kHz and a quantization of 16-bit. The 160 utterances produced by each speaker were recorded in a long recording.

Post-processing of Recordings

An automatic forced-alignment was first carried out via the WebMAUS service (Kisler, Reichel, and Schiel, 2017) on both word and phoneme levels, outputting a TextGrid format annotation of Praat (Boersma and Weenink, 2019). Then, the boundaries were manually adjusted, taking into account both the acoustic landmarks and zero-crossing positions by referring to both visual and audio cues simultaneously. Based on the derived word-level boundaries, the acoustic signal and corresponding manual segmentations for each target word were cut out from the respective long recordings and TextGrid based annotation files.

The segmented signals sometimes did not taper to zero at borders, which were incongruent with the real-world signals. To reduce the impact of cutting signal on the statistical properties of the signal, one half of a Blackman window was applied to smooth both sides of each segmented signal, making the amplitude of it taper to zero at the borders. Its effect is similar to that of a Tukey (tapered cosine) window. The window of length N is defined mathematically as:

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right), \quad n = 0, 1, ..., N-1. \tag{3.6}$$

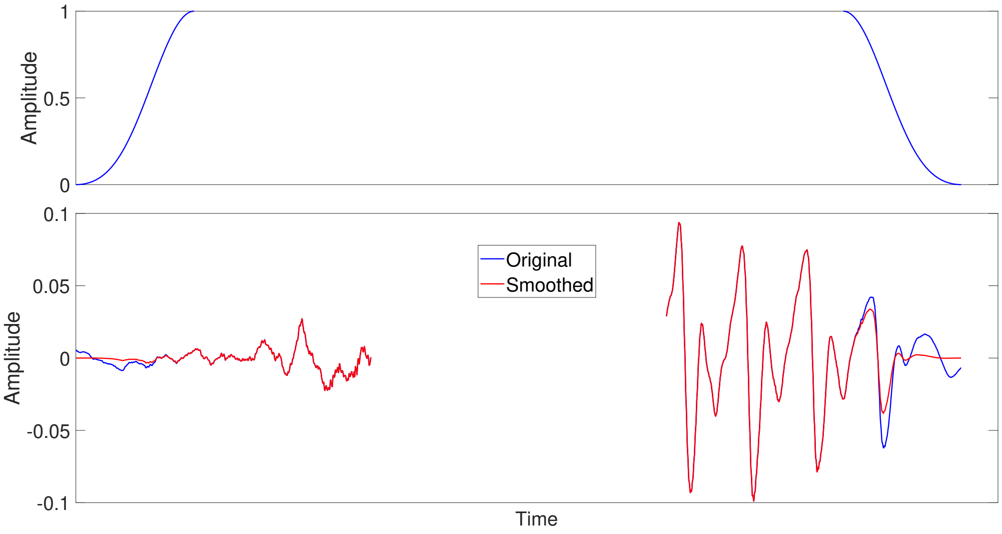


FIGURE 3.2: Smoothing borders of segmented signal using Blackman window.

A Blackman window of 40 ms was used to smooth each segmented acoustic signal. The window was split into two parts, as shown in the upper subplot of Figure 3.2, which were applied to the beginning and end parts of the segmented spoken word. The blue lines in the lower subplot of Figure 3.2 represents two parts of original signal (here, only 50 ms is shown for each side). By multiplying each part of the original signal with half of the Blackman window (20 ms for each side, respectively), the smoothed signals (the red curves) go to zero at the borders.

After that, an extra silence of 100 ms was added before and after the obtained waveform for each word, so that there was some leeway in real use, e.g., creating the gestural scores based on acoustic signals for ACS. Accordingly, extra silent segments of 100 ms were also inserted into the phone tier of TextGrid files. In the last step, each utterance was also normalized to -3 dB (amplitude normalization) to adjust the volume. Finally, the two sets of 160 target spoken words containing acoustic signals and corresponding manual segmentations storied in TextGrid files constitute the PBACU corpus. It should be noted that the proposed ACS methods only needed acoustic signals as inputs. The annotation files were not used as inputs of the proposed ACS methods, but only for evaluating the performance of some processing modules.

3.2.2 VTL-Kiel Corpus

Introduction of Kiel Corpus

The VTL-Kiel corpus was designed using VTL and the *Kiel Corpus of Spoken German* (Kohler, Peters, and Scheffers, 2018) which contains (read or spontaneous speech) recordings of sentences, short stories, and dialogues. The text used in read speech part came from seven text corpora, which are listed in Table 3.2. In this study only the sentence recordings of the read speech part were used and the very long recordings of stories ("Butter" and "The North Wind and the Sun") were excluded.

TABLE 3.2: Text corpora used in read speech part of Kiel corpus.

Text corpus	Text	Abbreviation
Berlin	100 Berlin sentences	be
Marburg	100 Marburg sentences	mr
Butter	The Butter Story in 3 parts	butt1/2/3
Nordwind	The North Wind and the Sun in 2 parts	nord1/2
Restkorp	20 CNET sentences	cn
	63 Kohler sentences	ko
	45 SEL sentences	s1
	25 additional SEL sentences	s2
	15 Schiefer/Sommer sentences	s3
	30 Tillmann/Kohler sentences	tk
Erlangen	100 Erlangen sentences	er
Siemens	100 Siemens sentences	si

Each sentence was spoken by one or several speakers who were seated in a sound-treated booth and read the sentence through a window from a monitor. The original Kiel corpus also provides users with detailed annotations. Listing 3.1 show an example of the annotation file. The first line indicates the file name itself. The second line is the orthographic text. The fourth line contains the canonical transcription automatically generated via G2P tools. The sixth line contains the segmental labels. The remaining lines contain the segmentation information created by manual time alignment of segmental labels. The prosodic labels, not shown in this listing, are also provided in the annotation files with the "*.s2" extension.

```
dlmer044.s1
ja , gut das wars dann , vielen Dank .
oend
 j 'a: , g 'u: t d a s+ v 'a: 6 s d a n+ , f 'i: l @ n d 'a N k .
kend
 c: j 'a: , -p: %g -gh 'u: t- d -dh a s+ v 'a:6 s d -dh a n+ , -p: f 'i: l @- n d -dh 'a N k -kh .
hend
    13786 #c:          0.8615625
    13786 ##j          0.8615625
    14739 $'a:         0.9211250
    16646 #,           1.0403125
    16646 #-p:         1.0403125
    19501 ##%g         1.2187500
    20249 $-gh         1.2655000
    20676 $'u:         1.2921875
    21539 $t-          1.3461250
    21539 ##d          1.3461250
    22348 $-dh         1.3966875
    22614 $a           1.4133125
    23667 $s+          1.4791250
    24526 ##v          1.5328125
    25049 $'a:6        1.5655000
    26885 $s           1.6802500
    ...
    36871 ##d          2.3043750
    37024 $-dh         2.3139375
    37180 $'a          2.3236875
    39165 $N           2.4477500
    39752 $k           2.4844375
    40147 $-kh         2.5091250
    40424 #.           2.5264375
```

LISTING 3.1: Example of an annotation file of Kiel corpus.

Gestural Score Creation

For this study, the utterances produced by female speakers were excluded since the model speaker ("JD2.speaker") of VTL was created based on a German male speaker. The resulting 1998 utterances (corresponding to 598 distinct sentences) produced by 14 male speakers were used as the basis for creating paired articulatory-acoustic utterance samples. The procedure of creating gestural scores based on utterances of Kiel corpus consisted of the following steps:

1. **Pre-processing of annotation files**. This step was to get the clean SAMPA based segmentation from the annotation files. As can be seen from Listing

3.1, the annotation file contains special symbols other than SAMPA symbols (see segmentations from Line seven), such as the sentence beginning marker "#c", segmental label markers like "##" and "$#", "$", stress marker "'", and punctuation markers like "," and ".". These special symbols together with the very first few lines (i.e., filename, orthographic and canonical transcriptions) were removed.

2. **Removing non-verbal segments and padding fixed-length silence**. This step was conducted for both segmentation files and waveforms. The very long non-verbal segments at the beginning and the end, and pause segments marked by "-p" and breathing segments marked by "-h" in the middle of recordings were removed from the annotation files. Accordingly, the corresponding acoustic signals were trimmed. A silent segment of 100 ms was padded to both ends of each resulting signal.

3. **Segmentation to segment sequence file of VTL**. The processed segmentation files were converted to the segment sequence files that were supported by VTL. Below is the content of a segment sequence fie ("dlmer044.seg"). Each line defines one segment in terms of its name (in SAMPA) and duration. The first and last lines indicate the 100 ms silent segments padded to the beginning and end of speech.

```
    name = ; duration_s = 0.1;
    name = j; duration_s = 0.059562;
    name = a:; duration_s = 0.11919;
    name = g; duration_s = 0.073438;
    name = u:; duration_s = 0.053937;
    name = d; duration_s = 0.067187;
    name = a; duration_s = 0.065813;
    name = s; duration_s = 0.053687;
    ...
    name = d; duration_s = 0.019313;
    name = a; duration_s = 0.12406;
    name = N; duration_s = 0.036687;
    name = k; duration_s = 0.042;
    name = ; duration_s = 0.1;
```

4. **Segment sequence file to gestural score file**. This step was done using a rule-based method implemented in VTL application programming interface (VTL-API), which will be introduced in greater detail in Section 4.1.2.

5. **Replacing the pitch tier of gestural score with estimated pitch targets**. The gestural scores created from last step contained only pseudo gestures for the f_0 tier. Therefore, the f_0 related gestures had to be estimated separately, which was performed by TargetOptimizer 2.0 (Krug et al., 2021). The estimated f_0 gestures were then merged to the gestural scores by replacing the pseudo ones. The detailed description of gestural score creation will be introduced in Section 4.1.2.

Data Augmentation

As mentioned in Section 3.1, for acoustic simulation, VTL will internally convert gestural scores into articulatory trajectories over time. For a single time step, the articulatory configuration is a vector of 30 articulatory parameters. Figure 3.3 shows an example of gestural score created from the last section and its corresponding synthetic speech above it. Acoustic features were extracted form the synthetic speech. The frame shift for extracting acoustic features was also 2.5 ms that equaled the time interval of between two adjacent articulatory parameters vectors. Hence, the articulatory representations (i.e., the articulatory parameter vectors) and the acoustic representation (i.e., extracted acoustic features) temporally matched frame-by-frame. They constituted the paired articulatory-acoustic sample for an utterance. A large scale corpus of was paired articulatory-acoustic data constructed using the above procedure with Kiel corpus, which could be then used to train supervised speech inversion systems.

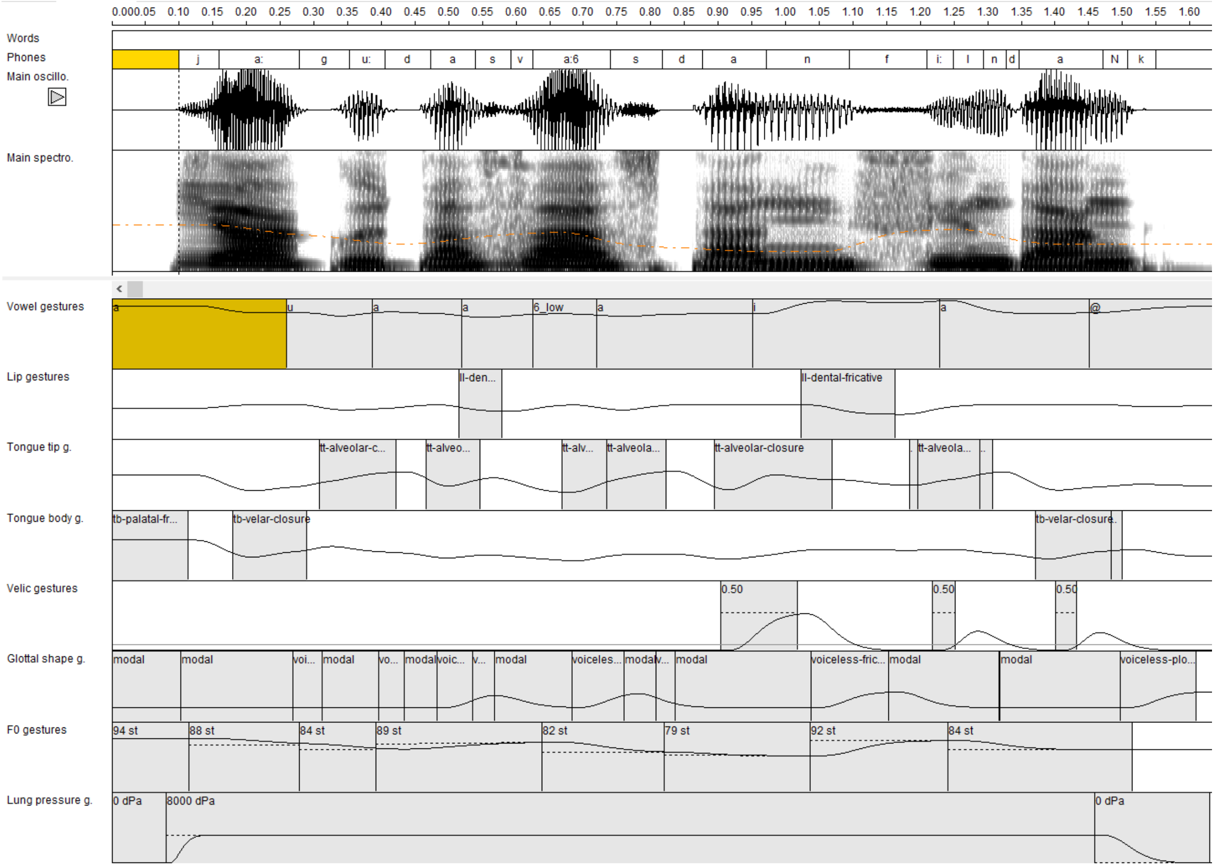


FIGURE 3.3: Example of gestural score for Kiel corpus and corresponding acoustic signal.

It should be pointed out that the arrangement and coordination of gestures involved in a gestural score were determined by the rule-based method (see Section 4.1.2 for details). The corresponding synthetic speech was normally natural and intelligible. However, such utterances only covered the most common articulatory configurations (and thus the corresponding synthetic speech with less acoustic variations) because the rule-based method converting segment sequences to gestural scores always used the same default values for some gesture parameters. For example, the "time constant" parameter reflecting speaking effort was always set to 12 ms (0.012 second) for supra-glottal gestures which, in turn, determined the same transition pattern from one phone to another for all instances. This is obviously inconsistent with that occurring in natural speech, because people usually speak with different speaking effort. Even for a specific speaker, different speaking effort will be used when he/she speaks for example with different speaking rates and/or emotions. Also, the voice quality (phonation) varies from speaker to speaker. Moreover, people have different vocal tract lengths, resulting in higher f_0 and formant frequencies normally for a shorter vocal tract length or lower ones for a longer vocal tract length.

A gestural score created with default values were referred to as the *prototype* one. In order to sufficiently cover the articulatory and acoustic space, which in turn makes the ACS models more robust, 12 variants for an utterance were created by manipulating the default configurations. First, its glottal gestures associated with voiced phones were always set to the "modal" phonation type by default, which was replaced by either "pressed" or "breathy" glottal gestures, thus producing different voice quality for voiced segments. Then, the default value (0.012 second) for time constant of supra-glottal gestures of the prototype gestural score was replaced by a random value sampled from a normal distribution with a mean of 0.012 and a standard deviation of 0.002. It should be mentioned that this manipulation was individually performed for each gesture, which could create more transition patterns. If the randomly sample values fell outside the valid range of time constant (here, [0.01, 0.0396] second defined in VTL), the sampling was repeated again. Up to this point, there were six gestural scores (three phonation types × two sets of time constant) in which the prototype one was included. Next, VTL synthesized six acoustic signals using these gestural scores. After that, the vocal tract length (to be more precise, the formant frequencies) was manipulated using the "change gender" functionality of Praat (Boersma and Weenink, 2019). The formant shift ratio for an acoustic signal was set to a random value sampled from a normal distribution with a mean of 1 and a standard deviation of 0.1 with a limited range of [0.8, 1.2]. Because the manipulation of vocal tract length was only operated in acoustic domain, the corresponding articulatory trajectories of the original six gestural scores were used again to form the articulatory-acoustic samples. In other words, each of six sets of articulatory trajectories corresponded to two acoustic signals: one with VTL model speaker vocal tract length and another with randomly manipulated vocal tract length. Up to this point, 12 articulatory-acoustic samples (three phonation types × two sets of time constant × two vocal tract lengths) were created based one original utterance of Kiel corpus. Finally, a set of 23 976 paired articulatory-acoustic samples were obtained with VTL and original utterances of Kiel corpus and referred to as the VTL-Kiel corpus.

3.3 Acoustic Features

Acoustic features can serve as two main functions in the context of ACS. Firstly, by encoding audio signals, they represent utterances in a compact format. Secondly, they are used as parameters for measuring similarity of utterances, for example, by comparing extracted acoustic features between reproduced and original utterances.

Both human natural speech and articulatory synthetic speech are produced by time varying vocal tract systems excited by time varying sources. Hence, acoustic signals are essentially non-stationary. However, the engineering way of processing speech is to assume the signal to be stationary in short time segments. More specifically, speech signal is assumed to be stationary when it is processed in frames of 20∼40 ms. A lot of acoustic features can be extracted for various tasks by applying short term processing (STP) to such short segments. This section introduces the acoustic feature used in this study, including mel-frequency cepstral coefficients (MFCC) together with spectrogram and some other complementary features.

3.3.1 Spectrogram and MFCC

Human speech is comprised of many frequency components. A speech spectrogram, calculated by repeatedly applying Fourier Transform to overlapped frames of a signal, shows the magnitude of the frequency components as it varies with time. The phones within utterances are characterized by the time-frequency structure of spectrogram. Therefore, spectrogram as well as its derivatives can be used as acoustic representation of utterances and widely used in speech signal processing. Mel-frequency cepstral coefficients (MFCC) are one of its derivatives and are the most widely used features in speech-related applications, such as recognition systems (Dave, 2013; Ittichaichareon, Suksri, and Yingthawornsuk, 2012), speaker recognition (Tiwari, 2010; Ganchev, Fakotakis, and Kokkinakis, 2005), and music information retrieval (Müller, 2007).

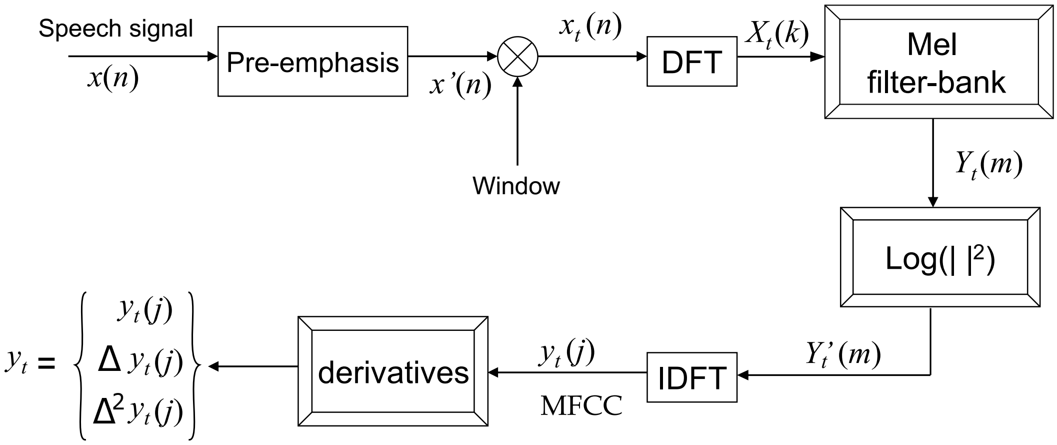


FIGURE 3.4: The general diagram of extracting MFCC.

The generic procedure of extracting MFCC is shown in Figure 3.4. Since the spectrogram is an intermediate product of extracting MFCC, its extraction will not be individually introduced. The extraction of MFCC features consists of the following steps:

Pre-emphasis

Voiced speech usually has a negative spectral slope (attenuation) due to the physiological characteristics of the speech production system (Picone, 1993). Pre-emphasis refers to emphasizing the higher frequency components so that the high frequency components have similar magnitude with respect to low frequency components. The following equation converts the original signal $x(n)$ into the pre-emphasized signal $x^{'}(n)$.

$$x^{'}(n) = x(n) - 0.97x(n-1) \tag{3.7}$$

Framing and Windowing

Human speech is a time-varying signal and usually quasi-stationary. Therefore, speech is usually split into successive overlapping short segments, each of which is assumed to be acoustically stationary. Short-term spectral analysis is typically carried out over $20 \sim 40$ ms frames and shifted every $10 \sim 20$ ms. In Figure 3.4, the subscript t is the time index of frames. For each analysis frame, a window is applied to taper the signal toward the frame boundaries. The most commonly used window is Hamming window, which is defined as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad n = 0, 1, ..., N-1, \tag{3.8}$$

where N is the length (in number of samples) of the Hamming window.

Discrete Fourier Transform (DFT)

Discrete Fourier transform (DFT) converts each Hamming-windowed frame into magnitude spectrum as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi nk}{N}}, \quad k = 0, 1, ..., N-1. \tag{3.9}$$

Mel-frequency Filter-bank Processing

Since human perception of frequencies is not linear in Hz scale but proportional in logarithm scale, the spectral representation is usually further processed in mel-frequency scale. A typical approximation of Hz-to-Mel conversion is defined as:

$$f[\text{Mel}] = 2595\log_{10}\left(1 + \frac{f[\text{Hz}]}{700}\right) \tag{3.10}$$

To further simulate human auditory sensitivity to critical band, triangular filter-banks equally spaced in mel scale are applied to the magnitude spectrum.

$$Y(m) = \sum_{k=0}^{N-1} |X(k)|H_m(k), \quad m = 0, 1, ..., M-1, \tag{3.11}$$

where M is the number of triangular filter-banks. $H_m(k)$is the weight given to the k-th energy spectrum bin contributing to the m-th output band (Huang et al., 2001; Rao and Vuppala, 2014). In addition, the logarithmic power spectrum, $\log|Y(m)|^2$, is usually used to approximate roughly the sensitivity of the human ears.

Discrete Cosine Transform (DCT)

Since the vocal tract is smooth, the energy levels in adjacent bands tend to be correlated. Besides, the envelope of the vocal tract changes slowly, while the excitation changes much faster. Therefore, discrete cosine transform (DCT) is required to reduce the correlation between the transformed mel-frequency coefficients, which produces a set of cepstral coefficients.

$$y(j) = \sum_{m=0}^{M-1} \log\left(|Y(m)|^2\right) \cos\left[j\left(m - \frac{1}{2}\frac{\pi}{M}\right)\right], \quad j = 0, 1, ..., J-1, \tag{3.12}$$

where J is the number of cepstral coefficients. Normally the first $12 \sim 13$ dimensions are used in speech recognition systems.

Dynamic Features: Deltas and Delta-deltas of MFCC

As time-varying signal, human speech is a sequence of transitions between phonemes. However, MFCC features are calculated for each given frame, which can only reflect the static information. A common method of capturing the dynamic characters of phoneme transitions is to compute the first and second derivatives of cepstral coefficients, which are defined as follows (Rao and Vuppala, 2014), respectively:

$$\Delta y_t(j) = \frac{\sum_{i=-T}^{T} i \cdot y_{t+i}}{\sum_{i=-T}^{T} i^2} \tag{3.13}$$

$$\Delta^2 y_t(j) = \frac{\sum_{i=-T}^{T} i \cdot \Delta y_{t+i}}{\sum_{i=-T}^{T} i^2} \tag{3.14}$$

where the $\Delta y_t(j)$ and $\Delta^2 y_t(j)$ denote the delta and delta-delta of the j-th dimension feature of the t-th time-indexed frame, respectively. T, normally taken from $\{1, 2, 3\}$, is the number of successive frames considered for dynamic feature computation.

3.3.2 Other Features

Short Time Energy

Short time energy refers to the energy associated with short term region of speech, which is calculated based on the amplitude of the speech signal. As shown in Equation 3.15, the short time energy of a frame with N samples is defined as the sum of the squared values of these samples.

$$SE = \sum_{n=0}^{N-1} x_n^2 \tag{3.15}$$

The signal of an utterance is composed of a chain of successive segments which can be voiced, unvoiced or silent (Atal and Rabiner, 1976). Figure 3.5 shows the waveform and spectrogram as well as the annotation of all phones for the word "durchhalten" from the PBACU corpus. As we can see from the upper panel, the amplitude of the waveform varies with time, which characterizes different segments to some extent. The silent segments can be the regions in the beginning and end of the utterance or short pauses between spoken words. They usually contain least or negligible energy compared to the real speech parts. The silent segments can also be the closure phases of plosive consonant, e.g., the closure for the phone [t]. Although this kind of silent segments are relatively short and appear within the spoken words, they are still distinctive from segments of their surrounding phones. Moreover, the voiced segments tend to sound louder than the unvoiced segments, i.e., the energy of voiced speech is generally higher than that of unvoiced speech (Bachu et al., 2010). Therefore, the features related to amplitude variations, in particular in the form of energy, are commonly used in speech analysis.

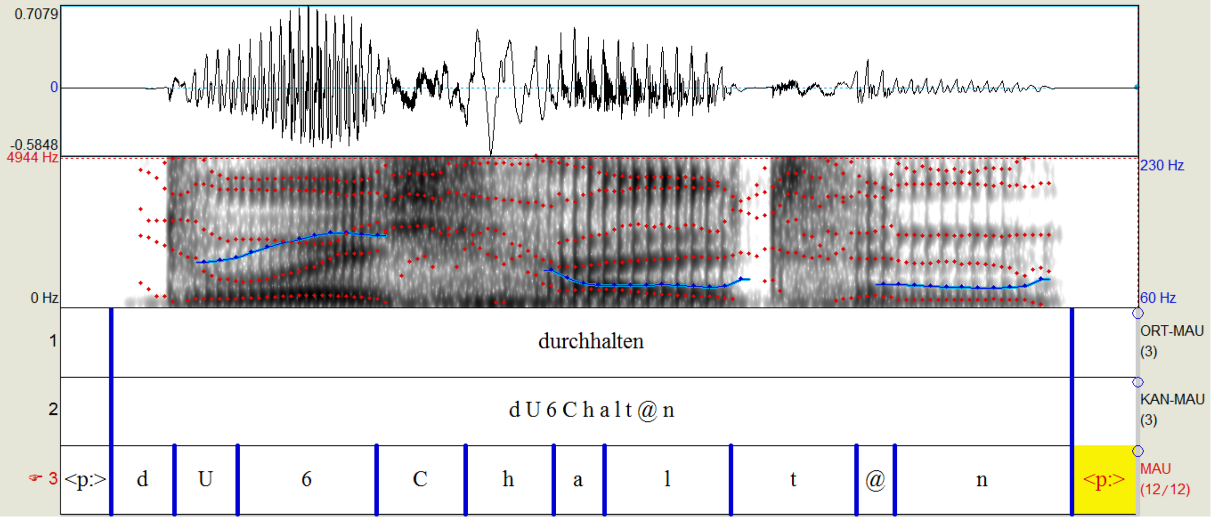


FIGURE 3.5: The waveform and spectrogram as well as the annotation of all phones for the word "durchhalten".

Short Time Zero Crossing Rate

The zero crossing rate (ZCR) refers to the rate at which a signal changes its algebraic sign from positive to negative or vice versa within a given signal, which reflects how fast the signal changes over time. It can be calculated according to Equation 3.16,

$$ZCR = \frac{1}{2N} \sum_{n=1}^{N} |sgn(x_n) - sgn(x_{n-1})| \tag{3.16}$$

where sgn is the sign function. The factor "2" in the denominator is usually responsible for reflecting the fact that there are two zero crossings per cycle of a signal.

ZCR can be used as an indirect indicator of the frequency information of the signal. A high ZCR means the signal is changing rapidly and accordingly it may contain high frequency information while a low ZCR indicates the signal is changing slowly and hence it may contain low frequency information.

Voiced speech is excited by the periodic air flow from the glottis and usually shows a low ZCR while unvoiced speech is excited by the noise-like source and usually shows a high ZCR (Bachu et al., 2010). As can be seen from Figure 3.5, the waveform of the unvoiced palatal fricative [C] has a very high ZCR. The ZCRs for the /h/ sound and the closure release phase of the [t] sound also show distinctive patterns from other segments.

Fundamental Frequency and Probability of Voicing

Voiced sounds are produced with the vocal folds vibrating and air going out through the vocal tract. As the result of vocal folds' regular opening and closing, the peak of air pressure in the sound wave appears periodically. The fundamental frequency (f_0) of a speech signal refers to the approximate frequency of the periodic structure of voiced speech signals, i.e., the number of complete repetitions (cycles) of a pattern of air pressure variation occurring in a second (Ladefoged and Johnson, 2014). Another concept closely related to f_0 is pitch, which describes how f_0 is perceived by human ears and brains in terms of periodicity. f_0 conveys important linguistic and paralinguistic information, such as forming stress and accents, distinguishing lexical meaning in tonal languages, expressing emotions and attitudes and so on. In the task of ACS, reproducing the original f_0 is very important since it affects how similar the reproduced utterance is to the reference one in both the acoustic and perceptual domains. Therefore, it is also considered in this study.

Another feature together with f_0 is probability of voicing (POV) that describes the voicing status of speech signals. In addition to the binary status (voiced vs. unvoiced), voicing status can be quantified using continuous values. It can be used not only in speech analysis but also in speech recognition (e.g., Zorila, Kandia, and Stylianou, 2012; Ghahremani et al., 2014). The POV feature is also important in the current study in that the realization of each phoneme in VTL is separately controlled by the vocal tract and vocal fold parameters. For example, a tongue-tip alveolar plosive consonant vocal tract configuration will produce the [d] sound with a voiced vocal fold configuration while it will produce the [t] sound with an unvoiced one.

The f_0 extraction from original utterances and f_0 reproduction in synthetic utterances are performed with Praat (Boersma and Weenink, 2019) and TargetOptimizer (Krug et al., 2021), respectively, which will be introduced in Chapter 4. The POV feature was extracted with TANDEM-STRAIGHT (Kawahara et al., 2008). The TANDEM-STRAIGHT first calculates the temporally stable power spectrum of a periodic signal, TANDEM spectrum, by adding two power spectra using a pair of time windows temporally separated for half of the fundamental period. Next, the interference-free spectrum, STRAIGHT spectrum, is calculated upon the TANDEM spectrum using the consistent sampling method. The fluctuation spectrum as well as its weighted Fourier transform are subsequently defined by combining these two spectra. Finally, for each f_0 candidate, a periodicity score is calculated by integrating all weighted Fourier transforms of the fluctuation spectrum. The detailed technical implementations are introduced in Kawahara et al. (2008) and Kawahara and Morise (2011).

3.4 Experiment Platform, Software, and Tools

All experiments in this study were conducted on the High Performance Computing (HPC) system[2] of TU Dresden. The HPC system are equipped with about 60,000 CPU cores, 448 GPU accelerators, and a shared storage containing about 2000 high capacity disks. Therefore, it is efficient for compute-intensive and data-intensive experiments of various scientific research. The account of the author was equipped with a "/home" directory of 50 GiB storage and a project directory of 300 GiB fast speed storage, and a shared computation storage. The experiment jobs were managed by the Slurm job scheduler. The Listing 3.2 shows an example of Slurm based experiment job file.

```
#!/bin/bash
#SBATCH --time=10:00:00    # walltime
#SBATCH --nodes=1    # number of nodes
#SBATCH --ntasks=1       # limit to one node
#SBATCH --cpus-per-task=24  # number of processor cores (i.e.
    ↪ threads)
#SBATCH --partition=haswell
#SBATCH --mem-per-cpu=4096M    # memory per CPU core
#SBATCH --mail-user=yingming.gao@mailbox.tu-dresden.de    #
    ↪ email address
#SBATCH --mail-type=BEGIN,END
#SBATCH --output=log.main.txt

module load modenv/scs5
module load MATLAB/2019b
matlab -nodisplay -nodesktop -nosplash -r
    ↪ main_genetic_algorithm
```

LISTING 3.2: Example of Slurm based experiment job file.

[2] https://tu-dresden.de/zih/hochleistungsrechnen

As for the software and tools, the experiments of Chapter 4 were conducted mainly in the Matlab-2019b environment[3]. The parallel pool of 24 workers (i.e., 24 CPU cores) were used to speed up the running of algorithms. The experiments of Chapter 5 were conducted mainly in the Python-3.8 environment. The neural network models were implemented with PyTorch-1.6 (Paszke et al., 2019). The graphics processing unit (GPU) was used to speed up computation. The NVIDIA Tesla V100 or A100 GPUs were used, depending on the available resources during job submission.

3.5 Concluding Remarks

This chapter laid the foundations for the development and evaluation of ACS systems. The VTL synthesizer was introduced with a particular focus on the organization pattern of the articulatory process, such as articulatory parameters, gestures, and gestural scores, which were estimated from acoustic signals in this study.

Furthermore, two corpora were deliberately designed for this study. The PBACU corpus was designed to cover all phonemes with as few words as possible. To this end, a set of 160 words that were both phonemically-balanced and commonly used were selected from a large list of words using an entropy-based greedy selection algorithm. The words were spoken by two German native speakers to allow researchers to test the effect of speaker variation on ACS. In Chapter 4, this corpus was used as the target utterances and reproduced using an ACS system based on a genetic algorithm. The VTL-Kiel corpus was designed to create paired articulatory-acoustic data which could be used to train and evaluate supervised speech inversion models. The segment sequence files were obtained by processing the annotation files of the Kiel corpus. The segment sequence files were then converted to gestural scores with a rule-based method implemented in VTL-API; these could then be further converted to articulatory trajectories using the acoustic simulation model of VTL to conduct the articulatory-to-acoustic conversion, thus producing the corresponding synthetic acoustic signals. In addition, to sufficiently cover both the articulatory and acoustic space, each original utterance was augmented by creating 12 variants which, to some extent, accounted for variations in speech production. The acoustic features extracted from synthetic speech, together with the articulatory trajectories generated from gestural scores, constituted the paired articulatory-acoustic data. These were used in Chapter 5 to build the neural network based regression models that mapped acoustic features to articulatory trajectories.

Finally, the acoustic features and the respective extraction methods were introduced. Since the genetic algorithm based ACS systems calculated acoustic similarity frame-by-frame, the acoustic features used in Chapter 4 included MFCC, short time energy, zero crossing rate, and probability of voice. The neural network based ACS systems used all frames of an utterance as a whole and could account for the temporal dependence between frames of inputs and outputs. Therefore, Chapter 5 only employed and compared MFCC and spectrogram.

[3] `https://www.mathworks.com/products/matlab.html`

Chapter 4

Articulatory Copy Synthesis Based on a Genetic Algorithm

This chapter investigates an ACS approach based on a genetic algorithm. For a given utterance to be copy-synthesized, a two-step procedure for estimating the underlying gestural score reproducing the target utterance is required. Section 4.1 introduces the first step, gestural score initialization, which generates the initialized gestural scores from given acoustic signals. Section 4.2 introduces the second step, gestural score optimization, which is based on a genetic algorithm. The articulatory parameters ("duration" and "time constant") of the initialized gestural scores from the first step were iteratively adjusted to minimize the acoustic distance between the reproduced and reference acoustic signals. In addition, the regularization of the deviation of the "time constant" parameter from its preferred values was also investigated to mitigate the non-uniqueness problem. Section 4.3 describes how the experiments were conducted and how the proposed methods were evaluated. The experiments and analysis are presented in Section 4.4. Section 4.5 provides some concluding remarks.

4.1 Gestural Score Initialization

The proposed ACS method for given utterances was based on the ABS procedure. The articulatory parameters of gestural scores were iteratively adjusted until the synthetic speech of the final gestural score had the least acoustic distance to the reference speech. A good starting state could reduce the time required by this ABS based procedure. To be more precise, if the initial values of gesture parameters (duration and time constant) within a gesture score were already close to the final optimal ones, the less effort was needed to further optimize them. This section introduces a rule-based method of initializing gestural scores for given utterances, the framework of which is shown in Figure 4.1.

4.1.1 Speech Transcription and Segmentation

Unlike the studies (Nam et al., 2012; Gao, Stone, and Birkholz, 2019) that assumed the transcriptions of utterances were known in advance, the proposed method only needed acoustic signals of utterances to generate initial gestural scores. As shown

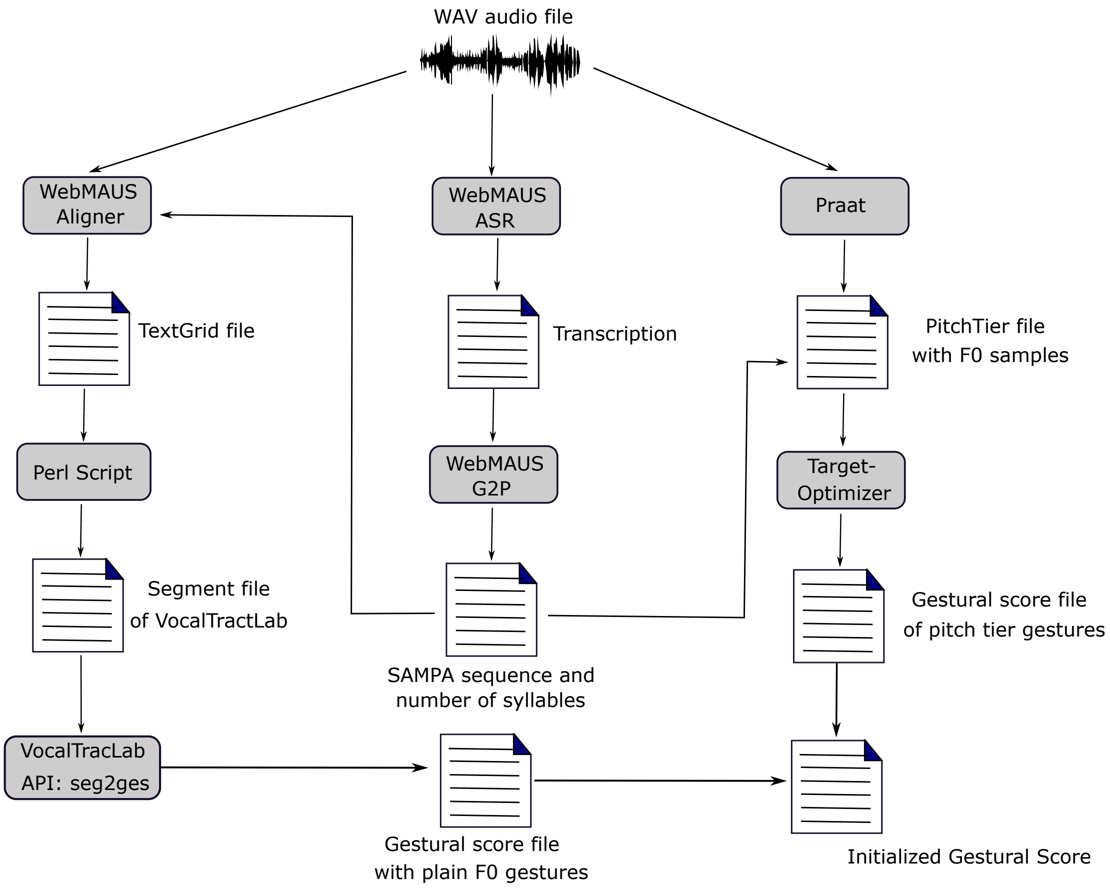

WAV audio file
WebMAUS Aligner
WebMAUS ASR
Praat
TextGrid file
Transcription
PitchTier file with F0 samples
Perl Script
WebMAUS G2P
Target-Optimizer
Segment file of VocalTractLab
SAMPA sequence and number of syllables
Gestural score file of pitch tier gestures
VocalTracLab API: seg2ges
Gestural score file with plain F0 gestures
Initialized Gestural Score

FIGURE 4.1: Information flow diagram for initial gestural score creation.

in the middle column of Figure 4.1, for a given utterance, the audio file was first transcribed using the WebMAUS ASR service (Kisler, Reichel, and Schiel, 2017). This online service accepted speech signal (e.g., in the format of ".wav" file) as input, internally called the speech recognition API, and returned the recognized words as output (e.g., in the format of ".txt" file). In this study, the Google Cloud Speech-to-Text API was selected as the backend recognizer[1].

The recognized words were further mapped into the phoneme level sequence using the WebMAUS G2P conversion service (Reichel, 2012; Reichel and Kisler, 2014). This online service accepted the orthographic text as input and internally used statistically trained decision trees to estimate the most likely string of phonemes, and finally returned the corresponding canonical phonological transcript (standard pronunciation) as output. In this step, the speech assessment methods phonetic alphabet (SAMPA) was used as the output symbol inventory. Besides, the option of 'Syllabification' was set to 'yes', which made the SAMPA sequence contain the syllable boundaries. The output format was set to the BAS Partitur Format (BPF) file with the orthographic (ORT) and canonical (KAN) tiers. Applying the

[1] https://cloud.google.com/speech-to-text

G2P conversion to the recognized orthographic text yielded the canonical SAMPA transcriptions.

The third step was to segment the audio files using the WebMAUS aligner, more specifically with the "General" mode. As shown in the left column of Figure 4.1, this online service took audio files and corresponding phonological transcriptions created by the G2P service as inputs, and generated the TextGrid based segmentation files containing the alignment between acoustic signals and segmental labels. Figure 4.2 shows an annotation file (in the TextGrid format of Praat) for the utterance "besonders" segmented by the WebMAUS aligner. The upper two panels display the waveform and spectrogram. Below them are the annotation tiers. The tiers "ORT-MAU" and "KAN-MAU" show the orthographic and canonical transcriptions respectively for the word level segmentation. The "MAU" tier shows the phone level segmentation. Silence or short pauses (represented as "<p:>" in the WebMAUS system) could be automatically inserted into the input phonological transcription if they existed, especially at the beginning and end of the audio file.

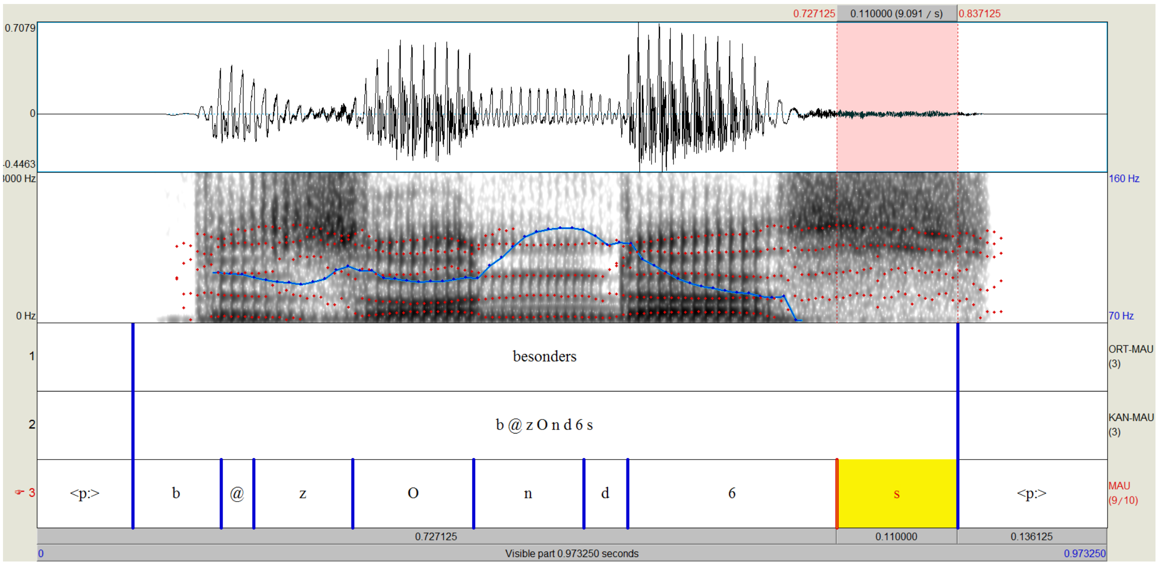


FIGURE 4.2: Annotation for the utterance "besonders" segmented by the WebMAUS aligner. Inaccurate segmentation can be found for the onset of the [s] sound.

4.1.2 Initial Gestural Score Creation

Gestures control the movement of participating articulators/parameters toward target configurations of the vocal tract model or the vocal fold model. The arrangement and coordination of multiple gestures involved in a gestural score determine the acoustic results. Creating a gestural score which can synthesize natural-sounding speech faces two main challenges. The first is how to arrange the timing of gestures. This involves not only the timing of gestures on the same tier but also the coordination of gestures distributed over multiple tiers. For example, the [t] sound in the word "guten" should be realized by the combination of a "tongue-tip-alveolar-closure" gesture on the tongue tip tier and a "voiceless-plosive" gesture on the glottal tier. If they do not coordinate well, the resulting sound might be perceived like [d] since its neighboring phones are voiced and have the "modal" gestures on the glottal tier. Directly using the duration of acoustic segments is not a satisfactory solution, as the gestures and acoustic boundaries do not temporally coincide. The acoustic results usually appear with a delay compared to the corresponding gestures. How much earlier the gestures should be placed is difficult to predict because it is influenced by factors like phonetic context, speaking rates and speaking effort. The second is how to determine the starting and finishing time for lung pressure which particularly depends on the types of utterance-initial and -final phones. The poor coordination between the lung pressure gesture and other gestures will lead to unnatural utterance, even the initial and/or final phones might totally disappear. Therefore, the proper way of creating initial gestural scores is of great importance. This study used a rule-based method implemented in VTL-API to create initial gestural scores, whose corresponding synthetic speech was generally natural and intelligible.

Segment Sequence to Gestural Score

The TextGrid files generated from Section 4.1.1 were first parsed by a Perl language script, which collected the segments in the "MAU" tier of TextGrid file and assembled them into the segment sequence file (*.seg) with the formats defined in VTL. The segment sequence file represented the structure and metadata of an utterance. Although the "*.seg" files supported many additional attributes to describe the utterance, such as information about syllables, words, phrases and sentences, only phones and corresponding duration were used in this study. The example below shows the segment sequence file for the utterance "besonders". Each line defines one segment in terms of its name in SAMPA symbol and duration in second.

```
name = ; duration_s = 0.087125;
name = b; duration_s = 0.08;
name = @; duration_s = 0.03;
name = z; duration_s = 0.09;
name = O; duration_s = 0.11;
name = n; duration_s = 0.1;
name = d; duration_s = 0.04;
name = 6; duration_s = 0.19;
name = s; duration_s = 0.11;
```

The next step was to convert the segment sequence file (*.seg) to the gestural score (*.ges) file using the function `vtlSegmentSequenceToGesturalScore` of VTL-API, referred to as `seg2ges` in Figure 4.1. This function was implemented based on a set of rules made by Peter Birkholz. The major steps are summarized as follows.

1. To pre-process the input SAMPA sequence including:
    - To split affricates into a plosive and a fricative.

- To set flag for the alveolars [d, t, n, l] whether they should be realized as post-alveolar consonants. The flag was set as `True` when they were next to [S, Z] in the same consonant cluster.
- To traverse all segments and find the beginning and the end of the valid (non-pause) segments, respectively.
- To use the "modal" phonation type as the default setting for the gestures of the glottal tier.

2. To create the basic lung pressure gesture from the beginning of the first valid segment to the end of the last valid segment.
    - The lung pressure started to rise 50 ms earlier than the first valid segment when it was a plosive, otherwise it started 20 ms earlier.
    - The lung pressure started to drop 70 ms earlier than the end of the last valid segment when it was a plosive, otherwise it started to drop 120 ms earlier.
3. To traverse all segments and create gestures for the vowel tier.
    - To replace all diphthongs by monophthongs, e.g., [aI] by [a] plus [e], [aU] by [a] plus [o], [OY] by [O] plus [e].
    - To append all vowel gestures.
    - To append a final schwa gesture that filled the gap until the end of the segment sequence.
4. To traverse all segments and create fricative gestures.
    - To set the onset 72 ms and the offset 47 ms earlier than the beginning and the end of the segment, respectively, when it was a fricative.
    - To further set the onset 5 ms earlier than the beginning when it was voiced.
    - To set the onset to the very beginning of the utterance when the segment was the first valid phone.
    - To set the onset and offset for glottal gestures like the corresponding supraglottal gestures.
    - To extend the glottal gesture by 30 ms when the current segment was a voiceless fricative followed by a voiceless plosive.
5. To traverse all segments and create the plosive, nasal, lateral, glottal stop, and glottal fricative gestures.
    - To determine voice onset times for plosives with the reference values (Klatt, 1975).
    - To set the closure duration not shorter than half the segment duration and not shorter than 20 ms when it was a plosive but not the very first segment.

- To set the glottal gesture of plosive 10 ms and 40 ms earlier for onset and offset than the beginning and end of the segment, respectively.
- To shorten the glottal gesture of plosive by 20 ms if the next segment was a voiced fricative.
- To further shorten the glottal gesture of plosive by 80 ms when the plosive was followed by a nasal or lateral with the same place of articulation, otherwise by 55 ms.
- To extend the glottal gesture by 100 ms when a voiceless plosive was the last segment.
- To set the onset of the velic gesture 20 ms earlier than the beginning of the nasal segment when it was followed by a plosive.
- To set the onset and offset of the velic gesture 5 ms earlier and 5 ms later than beginning and end of the nasal segment respectively, when it was followed by a fricative.
- To set the onset and offset of the velic gesture 35 ms earlier and 5 ms later than beginning and end of the nasal segment respectively, when it was followed by the later segment [l].
- To shift the onset and offset of the [h] and glottal stop gestures to the left by 60 ms.
- To shift the onset and offset of gestures of [l] to the left by 65 ms and 55 ms, respectively.

The segment sequence files (*.seg) were converted to the gestural score files (*.ges) with the function `vtlSegmentSequenceToGesturalScore`. In this step, the f_0 tiers of gestural scores were initialized with pseudo f_0 gestures, corresponding to a plain intonation, and hence had to be replaced by real gestures separately estimated from the given utterances.

Pitch Target Estimation for Gestural Score

The intonation realization affects not only the acoustic simulation but also the perceptual similarity between the reproduced synthetic speech and the given natural utterance. In this study, the reproduction of intonation was performed with the software TargetOptimizer-2.0 (Krug et al., 2021). It used the pitch targets to control the intonation under the theory of target approximation model (TAM). TAM assumed one underlying pitch target for each syllable of an utterance, and the surface pitch contour was the sequential realization of all pitch targets of the utterance. TargetOptimizer-2.0 estimated the underlying pitch targets from the surface pitch contours in the sense of least cost that was the sum of the Euclidean distance of the f_0 samples between the original and the reproduced pitch contours and the deviation of estimated parameters from their preferred values. TargetOptimizer-2.0 accepted the f_0 values in the format of PitchTier files (*.PitchTier) of Praat and either syllable boundaries or number of syllables as inputs and returned the estimated pitch targets in the format of gestural score file (*.ges) or CSV files (*.csv) or the modeled f_0 in the format of PitchTier files (*.PitchTier).

The f_0 range usually varied from speaker to speaker. To make the f_0 extraction more accurate, the f_0 of the original utterances was extracted by a two-pass procedure following the strategy proposed by Hirst (Hirst, 2011). In the first pass, f_0 was extracted using the default range of 75 $\sim$ 600 Hz of Praat. Then, the first and third quartiles (i.e. q_1 and q_3) were calculated across all f_0 samples for each speaker. In the second pass where the f_0 floor and ceiling for each speaker were set to $0.75 \cdot q_1$ and $1.5 \cdot q_3$, respectively, f_0 was extracted again. Using a personalized search range greatly reduced the estimation errors of f_0 extraction, which was confirmed by comparing speakers' f_0 histograms. In this way, long tails disappeared and f_0 samples were more centralized around the mean values. In this study, the f_0 samples of the original utterances were extracted every 10 ms and saved as PitchTier spreadsheet files (*.PitchTier) of Praat. Below is an example of the PitchTier file.

```
"ooTextFile"
"PitchTier"
0 0.97212500000000002 41
0.1660625        113.28215490025224
0.17606249999999998        113.45882776753524
0.18606249999999999        118.31278340139376
0.1960625        123.86812703177232
0.20606249999999998        127.23721853633627
0.21606249999999999        128.78126272026506
...
0.79606250000000001        98.299397422904164
0.80606250000000002        98.730611991176247
0.81606250000000002        98.63015808792251
0.82606250000000003        97.107997238043481
0.83606250000000004        97.011675162327009
0.84606250000000005        97.993257607234327
```

The first two lines are the header of the PitchTier file. The third line indicated the beginning and end of the utterance (in second), and the number of voiced frames, i.e., the duration of the processed speech is 0.97 second and 41 f_0 samples in Hz were extracted within this period. The remaining lines show the locations and f_0 values with one line for each pair.

Another input for the TargetOptimizer-2.0 was the number of syllables which came from the G2P conversion step as shown in Figure 4.1. The G2P conversion yielded not only the canonical transcriptions but also the syllabification information when the option `Syllabification = True`. The pitch targets for each utterance were estimated using the TargetOptimizer-2.0 command line mode with the name of PitchTier file, the corresponding number of syllables, and the "-g" option of output format as arguments. The resulting file was the gestural score which contained only the gestures of the f_0 tier.

As shown in Figure 4.1, in the final step of creating the initial gestural score, the gestures of the f_0 tier of gestural scores initialized by VTL-API were replaced by those estimated by the TargetOptimizer-2.0. An example of initialized gesture score is shown in Listing 4.1. It should be noted that since the gestures of the f_0 tier were separately estimated, their controlling parameters were already optimal and hence would be fixed during gestural score optimization. Only gestures of other tiers needed to be optimized in the next step.

```
<gestural_score>
  <gesture_sequence type="vowel-gestures" unit="">
    <gesture value="i" slope="0.000000" duration_s="0.363055"
       ↪  time_constant_s="0.012" neutral="0" />
    <gesture value="o" slope="0.000000" duration_s="0.354069"
       ↪  time_constant_s="0.012" neutral="0" />
    <gesture value="@" slope="0.000000" duration_s="0.020000"
       ↪  time_constant_s="0.012" neutral="0" />
  </gesture_sequence>
  <gesture_sequence type="lip-gestures" unit="">
  </gesture_sequence>
  <gesture_sequence type="tongue-tip-gestures" unit="">
    <gesture value="" slope="0.000000" duration_s="0.318055"
       ↪ time_constant_s="0.012" neutral="1" />
    <gesture value="tt-alveolar-lateral" slope="0.000000"
       ↪ duration_s="0.104069" time_constant_s="0.012"
       ↪ neutral="0" />
  </gesture_sequence>
     ...
  <gesture_sequence type="f0-gestures" unit="st">
    <gesture value="85.387572" slope="0.000000" duration_s
       ↪ ="0.180481" time_constant_s="0.010" neutral="0" />
    <gesture value="83.243764" slope="-0.392107" duration_s
       ↪ ="0.156684" time_constant_s="0.015" neutral="0" />
    <gesture value="79.039295" slope="-5.588275" duration_s
       ↪ ="0.394679" time_constant_s="0.025" neutral="0" />
  </gesture_sequence>
  <gesture_sequence type="lung-pressure-gestures" unit="dPa">
    <gesture value="0.000000" slope="0.000000" duration_s
       ↪ ="0.047125" time_constant_s="0.012" neutral="0" />
    <gesture value="8000.000000" slope="0.000000" duration_s
       ↪ ="0.569999" time_constant_s="0.005" neutral="0" />
    <gesture value="0.000000" slope="0.000000" duration_s
       ↪ ="0.170000" time_constant_s="0.012" neutral="0" />
  </gesture_sequence>
</gestural_score>
```

LISTING 4.1: Example of an initialized gestural score (in the format of XML file) for the word "Kilo" spoken by the speaker SPK-1.

4.2 Gestural Score Optimization Based on a Genetic Algorithm

Feeding the initialized gestural scores into VTL could synthesize highly natural and intelligible speech. However, the goal of ACS was to accurately reproduce the real articulatory processes and acoustic signals for given utterances. The current gestural scores were not good enough for two reasons. First, the initialization procedure might introduce errors such as the G2P conversion errors and misalignment by the WebMAUS aligner. Second, the rule-based method of creating initial gestural scores by VTL-API used the default value (12 ms) for the "time constant" parameters of all supra-glottal gestures. That means it used a fixed transition pattern for all phones. However, different speakers, even a particular speaker under the conditions of different speaking rates or emotions, use different speaking effort for implementing phone transition. Therefore, the fine-tuning adjustments of gestural scores (to be more precise, duration and time constant of gestures) were often needed.

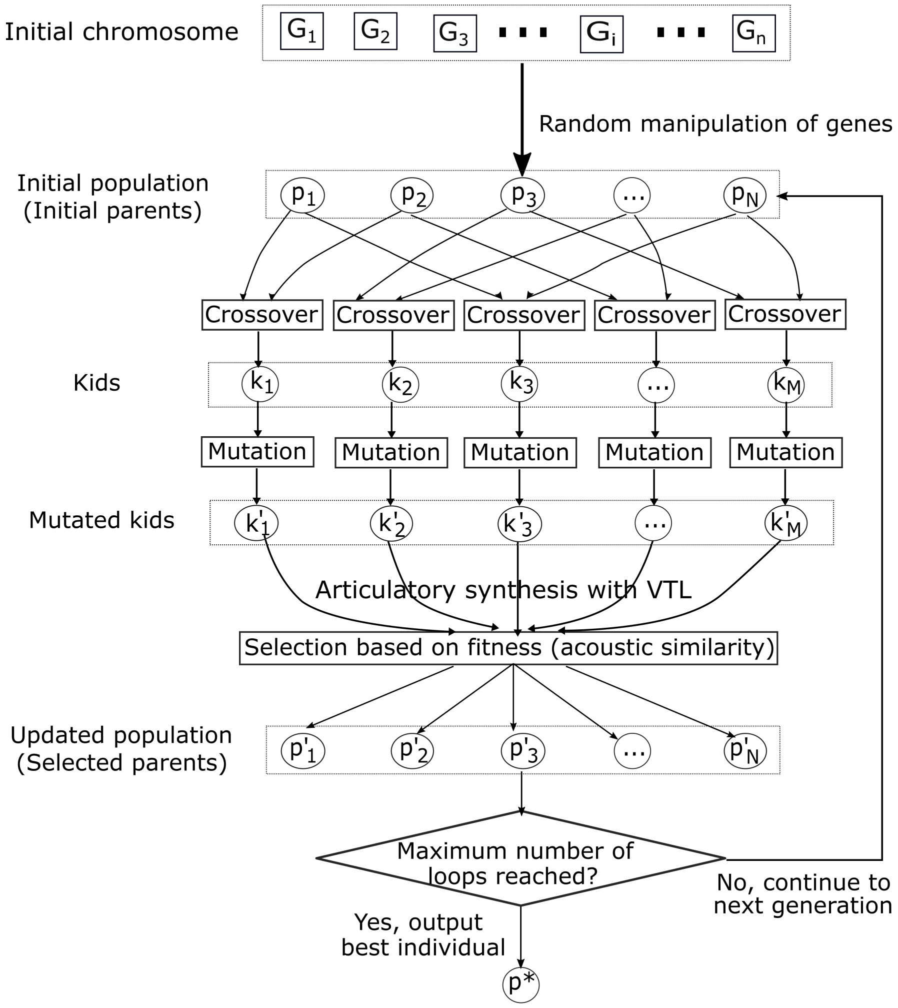

Initial chromosome
G1
G2
G3
Gi
Gn
Random manipulation of genes
Initial population
(Initial parents)
p1
p2
p3
...
pN
Crossover
Kids
k1
k2
k3
kM
Mutation
Mutated kids
k'1
k'2
k'3
k'M
Articulatory synthesis with VTL
Selection based on fitness (acoustic similarity)
Updated population
(Selected parents)
p'1
p'2
p'3
p'N
Maximum number of loops reached?
No, continue to next generation
Yes, output best individual
p*

FIGURE 4.3: Schematic diagram of gestural score optimization using genetic algorithm.

Genetic algorithm has been demonstrated to be effective for real parameter optimization, especially for non-differentiable problems (Wright, 1991). The introduction of genetic algorithms and their applications can be found in Tang et al. (1996) and Katoch, Chauhan, and Kumar (2021). This section proposed a gestural score optimization method based on the genetic algorithm. Figure 4.3 shows the schematic diagram of gestural score optimization using the genetic algorithm where the gestural score was represented as a chromosome, and the parameters of the gestures were encoded as genes. Through the loop consisting of "crossover", "mutation", and "selection", a population of individuals (candidate gestural scores) evolved under the law of survival of the fittest. The fittest gestural score corresponded to the synthetic utterance with the least acoustic distance to the reference utterance. The final gestural score with best fitness was selected as the solution.

Encoding Gestural Score as Chromosome

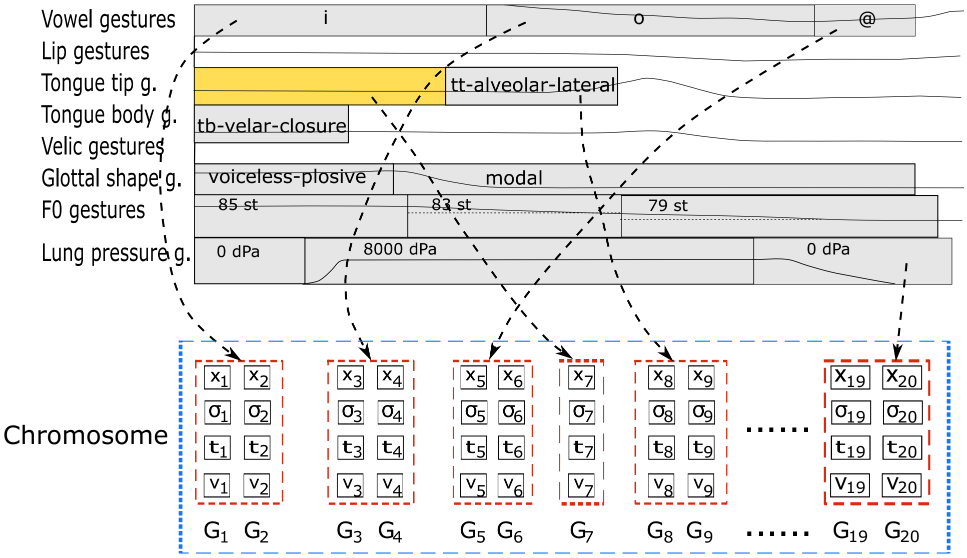

Vowel gestures
Lip gestures
Tongue tip g.
Tongue body g.
Velic gestures
Glottal shape g.
F0 gestures
Lung pressure g.
i
o
@
tt-alveolar-lateral
tb-velar-closure
voiceless-plosive
modal
85 st
83 st
79 st
0 dPa
8000 dPa
0 dPa
Chromosome
x_1 x_2 σ_1 σ_2 t_1 t_2 v_1 v_2
x_3 x_4 σ_3 σ_4 t_3 t_4 v_3 v_4
x_5 x_6 σ_5 σ_6 t_5 t_6 v_5 v_6
x_7 σ_7 t_7 v_7
x_8 x_9 σ_8 σ_9 t_8 t_9 v_8 v_9
......
x_{19} x_{20} σ_{19} σ_{20} t_{19} t_{20} v_{19} v_{20}
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8 G_9 ...... G_{19} G_{20}

FIGURE 4.4: Diagram of creating the initial chromosome from the initialized gestural score (with the word "Kilo" as the example).

The proposed method started from encoding gestural score as chromosome so that the optimization of gestural score could follow the generic framework of genetic algorithms. Figure 4.4 shows the creation of the initial chromosome from the initialized gestural score (with the word "Kilo" as the example). In this study, what should be optimized by genetic algorithms was the "duration" and "time constant" parameters of gestures, which were encoded as genes using a real-valued encoding scheme (Wright, 1991; Janikow and Michalewicz, 1991). A gesture parameter (either duration or time constant) was encoded as a gene. Therefore, a valid gesture was encoded as two genes: one for duration and another for time constant. For example, the gesture for vowel "i" in Figure 4.4 was represented by two genes: G_1 and G_2. For an empty gesture, only one gene was used since only its duration was valid. For example, the first gesture in the tongue tip tier was encoded as gene G_7. A gene was represented by a 4-dim vector. The first dimension, x, was the value of duration or time constant to be optimized. The second dimension, σ, was the standard deviation associated with x which would be used as the step size during gene mutation. The third dimension, t, was the tag to indicate the current gene was for a duration or time constant parameter. The fourth dimension, v, was the preferred value for this gesture parameter. The gestural score shown in Figure 4.4 was encoded as 20 genes, each of which was a 4-dim vector. All encoded genes constituted the chromosome for the gestural score indicated by the dashed blue rectangular.

As one of the evolutionary algorithms, the genetic algorithm operated on a group of individuals, which was referred to as *population*. As shown in Figure 4.3, before the optimization steps, an *initial population* (or *initial parents*) of a certain size was generated by randomly manipulating genes of the initial chromosome. To be more precise, the x value of each gene was changed by adding a random value sampled from a normal distribution while the other variables of genes (σ, t, and, v) were fixed. Up to this point, a population comprising a large number of individuals was initialized. By decoding genes (the x value) into duration or time constant, a new gestural score with manipulated values could be reconstructed. That is to say, each individual of the population represented a candidate gestural score to be optimized.

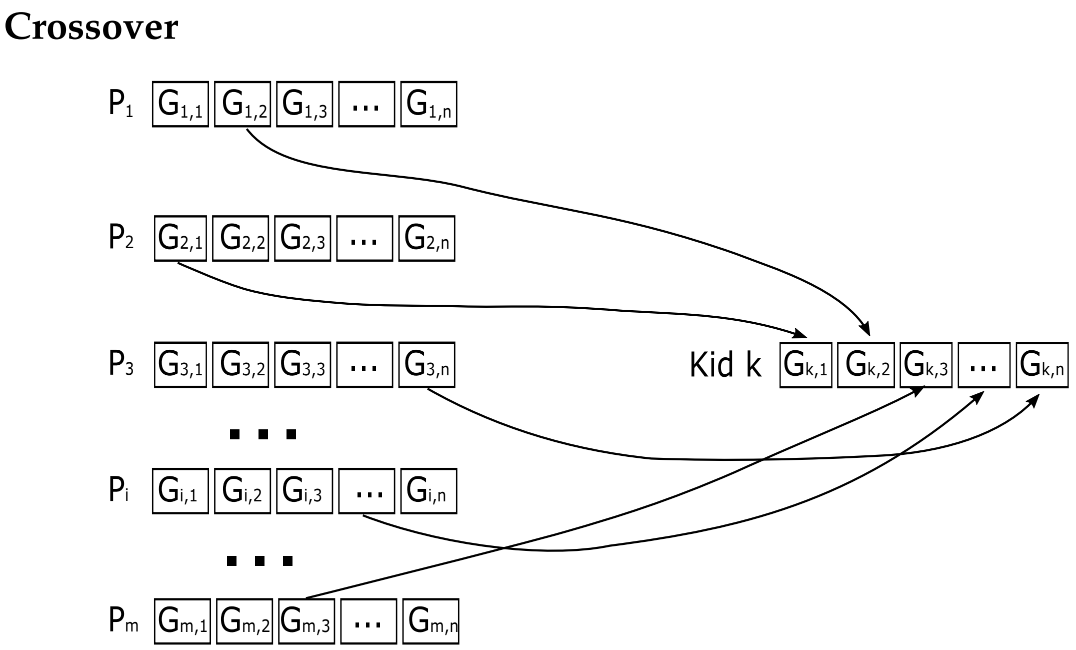

Crossover
P1 G1,1 G1,2 G1,3 ··· G1,n
P2 G2,1 G2,2 G2,3 ··· G2,n
P3 G3,1 G3,2 G3,3 ··· G3,n
Kid k Gk,1 Gk,2 Gk,3 ··· Gk,n
Pi Gi,1 Gi,2 Gi,3 ··· Gi,n
Pm Gm,1 Gm,2 Gm,3 ··· Gm,n

FIGURE 4.5: Diagram of gene crossover operation.

The *crossover* operation was applied to the initial population/parents to obtain new combinations of genes, the new *Kids*. Figure 4.5 shows the gene crossover operation. Unlike the regular strategy where the crossover occurs between two selected parents, the crossover in this study was performed in a *global* form, allowing randomly taking genes for one new individual from potentially all individuals available in the parent population. This manner is also termed *recombination* of evolutionary algorithms (Bäck and Schwefel, 1993), which could increase the gene randomization for the population. Each gene of a new kid came from a gene of a randomly selected parent. In the example shown in Figure 4.5, for the k-th new kid, its first gene $G_{k,1}$ came from the second parent, P_2, its second gene $G_{k,2}$ came from the first parent, P_1, its third gene $G_{k,3}$ came from the m-th parent, P_m, and so on.

Mutation

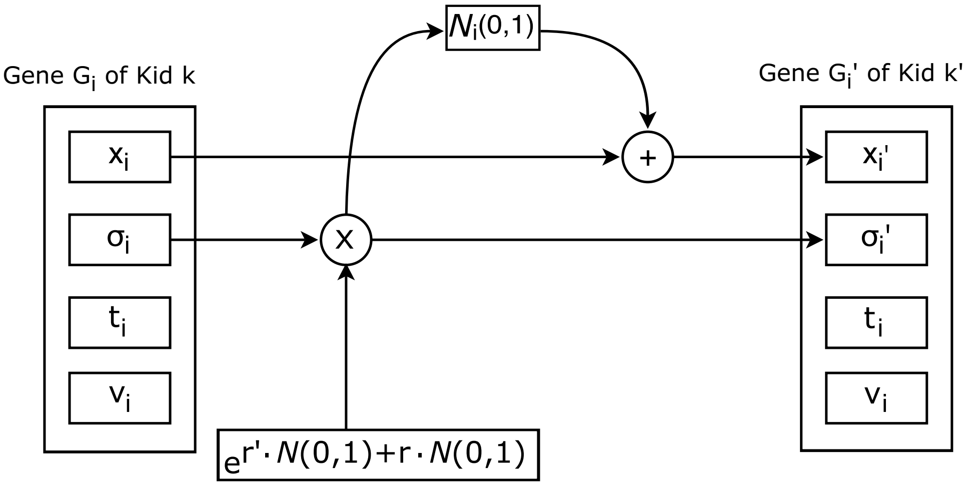


FIGURE 4.6: Diagram of gene mutation operation.

In this step, each gene of a new kid would mutate by adding a random value. Figure 4.6 shows the mutation operation for a gene of a kid. The mutation operation was individually performed for each gene x_i ($1 < i < n, n$ is the total number of the "duration" and "time constant" parameters of a gestural score to be optimized) by adding a normally distributed random value with a mean of zero and a standard deviation σ_i. The mutated gene x_i' was defined as:

$$x_i' = x_i + \sigma_i \cdot N_i(0,1) \tag{4.1}$$

where σ_i is the so-called step size, which is the second dimension of the gene vector. This study used an adaptive step size that changed in each generation over time. The mutation was then rewritten as (Back, Hammel, and Schwefel, 1997):

$$\sigma_i' = \sigma_i \cdot \exp\left(r' \cdot N(0,1)\right) + r \cdot N_i(0,1)) \tag{4.2}$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1) \tag{4.3}$$

where $r' = \frac{c}{\sqrt{2n}}$ and $r = \frac{c}{\sqrt{2\sqrt{n}}}$ are the learning rates ($c = 1$ is a reasonable choice when the number of evolution generations is between 20 and 100, as suggested in Beyer and Schwefel (2002)).

Selection

For the assessment of the fitness of mutated kids, their genes were decoded into duration or time constant to reconstruct gestural scores, which were further fed into VTL to synthesize acoustic signals.

This study adopted the acoustic distance between the synthetic and reference utterances as the similarity metric. More specifically, cosine distance of acoustic features (MFCC, energy, ZCR, and POV) between the synthetic and reference speech was used since it was less sensitive to feature magnitudes than Euclidean distance, which was proved by Bulla (2017). The smaller the distance is, the more similar the reproduced and original utterances are. Therefore, the cosine distance was expressed as the loss function during optimization which was calculated as:

$$\mathcal{L}_g = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{\sum_{j=1}^{M} X_{i,j}\hat{X}_{i,j}}{\sqrt{\sum_{j=1}^{M} X_{i,j}^2}\sqrt{\sum_{j=1}^{M} \hat{X}_{i,j}^2}}) \tag{4.4}$$

where g is the generation index of the population, N is the frame number of the reference utterance, and M is the dimension of acoustic features per frame. $X_{i,j}$ and $\hat{X}_{i,j}$ are the j-th dimensional features of the i-th frame of the reference and synthetic utterances, respectively. Therefore, the term in the parenthesis was the cosine distance of acoustic features for the i-th frame. The cosine distances of all frames were then averaged.

As mentioned in Chapter 1, speech inversion suffered from the problem of non-uniqueness due to motor equivalence phenomena in speech production (Perrier and Fuchs, 2015). For ACS, different gestural scores might result in synthetic signals that had same degree of similarity relative to the reference speech. As a result, the combinatorial explosion made the search space of optimization grow rapidly. Nevertheless, the problem could be alleviated by introducing additional articulatory and phonological constraints (Dusan and Deng, 2000; Dang and Honda, 2002). Here, by incorporating deviations of time constants from their *preferred* values into the loss function, gestural score parameters could be constrained to plausible values:

$$\mathcal{L}_g = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\sum_{j=1}^{M} X_{i,j}\hat{X}_{i,j}}{\sqrt{\sum_{j=1}^{M} X_{i,j}^2}\sqrt{\sum_{j=1}^{M} \hat{X}_{i,j}^2}}\right) + \frac{1}{K}\sum_{k=1}^{K}\exp\left(\frac{\alpha}{\sqrt{2g}}|\hat{\tau}_k - \tau_k|\right) \tag{4.5}$$

where K is the number of time constants to be optimized. $\hat{\tau}_k$ is the current value of the k-th time constant at the g-th generation while τ_k is preferred value which is the fourth dimension of the gene vector. The first term is the cosine distance of acoustic features and can be referred to as acoustic loss while the second term reflects the deviations of time constants from their preferred values and can be referred to as articulatory loss. α is a weight to balance the magnitudes of these two kinds of losses. The exponential function used in the second term is to exaggerate the loss when some specific time constants deviate too much from their preferred values.

That is to say, the penalty of time constant deviation is non-linear. It should be noted that the generation index, g, is also included in the second term which modulates the magnitude of the articulatory loss. Its effect will be explained in Section 4.4.

For each mutated kid, the genes were decoded to reconstruct a gestural score which was further fed into VTL. Based on these equations, the fitness of the mutated kid (i.e., the similarity between its synthetic and reference speech) was obtained. The mutated kids that had better fitness (i.e., closer distance to the reference utterance) would be selected as new parents of the next generation, forming the new *population*. The population evolved until the maximum number of iterations was reached. The aim of the evolutionary strategy was to produce increasingly better individuals over time, so finally, the optimal kid corresponded to the best gestural score for the target utterance.

4.3 Experiments

The proposed methods introduced in Sections 4.1 and 4.2 were evaluated using the PBACU corpus. Three ACS methods were implemented in this section: (1) coordinate descent without regularization of time constants, (2) genetic algorithm without regularization of time constants, and (3) genetic algorithm with regularization of time constants. During gestural score optimization, the first two methods used the loss function defined by Equation 4.4 while the third method used the loss function defined by Equation 4.5. This section introduces how the experiments were conducted and analyzes corresponding intermediate results. The reproduced utterances of ACS and analysis will be given in the next section.

These three methods were used to estimate the articulatory processes, in the form of gestural scores, of the 320 human utterances of the PBACU corpus. Because the PBACU corpus contained only acoustic signals, it was impossible to directly compare the estimated articulatory trajectories and the real ones. Therefore, synthetic utterances were created with VTL. 16 words (10 %) out of the whole word-list of the PBACU corpus produced by the speaker SPK-1 were selected: "bayrisch", "besonders", "Chemie", "dadurch", "genauso", "Hierarchie", "Komiker", "Lehrerin", "nebenbei", "neugierig", "Performance", "rigoros", "symbolisch", "Teufel", "Verkauf", "wunderbar". Based on these acoustic signals produced by the speaker SPK-1, a set of gestural scores were created by manually adjusting the duration and time constant of gestures so that their corresponding synthetic speech exactly matched the original acoustic signals. That is to say, *manual* copy synthesis was performed to obtain the optimal gestural scores for human utterances. Finally, the manually created gestural scores and their corresponding synthetic speech were used the complementary evaluation data of the PBACU corpus.

Initial Gestural Score Creation

Although the 320 utterances of the PBACU corpus have been manually segmented, only the acoustic signals were used to create initial gestural scores. Their transcriptions were obtained by recognizing the acoustic signals via the WebMAUS ASR service with Google engine as its backend. Comparing the recognized transcriptions with the ground truth (i.e., the corresponding words) achieved a word recognition accuracy of 98.44%, indicating the very high intelligibility of these natural utterances. There were five incorrectly recognized words as shown in Table 4.1. A close observation revealed that most of these recognition errors were due to the transient acoustic property of incorrectly recognized phones. For example, the first syllable in “bezahlen” and the plosive consonant in “Pole” were very short and weak. Besides, the very similar pronunciation (e.g, "Journal" vs "Ronal"), the same articulation place (e.g., tongue-tip-alveolar consonants: [l] of “Kilo” and [n] in “Kino”) and the same articulation manner (e.g., plosive consonants: [p] of “Pole” and [k] of “Kohle”) might also cause the recognition errors. These recognition errors were manually corrected before the next step.

TABLE 4.1: Incorrectly recognized utterances by WebMAUS ASR.

	Speaker	Reference	Recognized
1	SPK-1	"bezahlen"	"zahlen"
2	SPK-1	"Pole"	"hole"
3	SPK-1	"Kilo"	"Kino"
4	SPK-2	"Journal"	"Ronal"
5	SPK-2	"Pole"	"Kohle"

Conducting the G2P conversion for the recognized transcriptions (i.e., 160 unique words) yielded their canonical SAMPA transcriptions. Two conversion errors were found: the [t] of “mitmachen” was converted to [p] and the [d] of “besonders” was converted to [n]. When an input word had different pronunciation variants, the WebMAUS G2P service always returned the most likely phoneme sequence. However, speakers might produce the given word with another pronunciation variant. For example, both speakers produced the [C] sound for the initial consonant of the word “Chirurg” while the WebMAUS G2P yielded the phoneme [k]. These G2P conversion errors were manually corrected before the next step.

The 320 audio files and their phonological transcription (encoded in SAMPA) were fed into the WebMAUS aligner, which output the segmentation, i.e., TextGrid files. The derived boundaries were compared with those manual annotations of the 320 utterances. Table 4.2 lists the average deviation of boundaries between the WebMAUS alignment and manual segmentation. The WebMAUS aligner caused the average segmentation deviation of approximately 20 ms for both speakers. A close observation revealed that the WebMAUS aligner suffered from inaccurate segmentation (an example of segmentation is shown in Figure 4.2) due to two reasons. First, the boundaries of manual segmentation were continuous values in the time axis while the frame shift of calculating acoustic features in WebMAUS aligner was 10 ms. That is to say, the searched boundaries were always integral multiples of 10 ms. Second, the segmentation for plosives and fricatives was not accurate, especially for utterance-initial or -final ones. The closure phase of plosives was acoustically similar to the silence. And the acoustic characteristics of voiceless fricatives were similar to that of background noise. Hence, locating the boundaries for the utterance-initial or -final plosives and fricatives was not accurate. For example, very poor alignments were found for fricatives in the utterances of "Physik", "Parteilos", and "Positiv".

TABLE 4.2: The average deviation per boundary of the WebMAUS alignment from manual segmentation in ms.

Speaker	SPK-1	SPK-2	Total
Avg. deviation	19.36	19.20	19.28

The gestures on the f_0 tier of the 320 gestural scores were estimated by TargetOptimizer-2.0. Its performance of modeling f_0 contour was evaluated in terms of RMSE and Pearson correlation coefficient (ρ) between the original f_0 samples and the modeled ones. It should be mentioned that TargetOptimizer-2.0 internally converted the f_0 values from Hertz scale (Hz) to Semitone scale (St) with the reference of 1 Hz. Table 4.3 shows the average values for the 320 utterances. When RSME was larger than 1 St or ρ was smaller than 0.9, the f_0 samples extracted by Praat were manually checked and then the pitch targets were estimated again by TargetOptimizer-2.0.

TABLE 4.3: Average RSME and correlation coefficient (ρ) between the original and modelled f_0 by TargetOptimizer.

	SPK-1	SPK-2	Total
RMSE	0.396	0.412	0.404
ρ	0.947	0.98	0.964

Acoustic Features

As introduced in Section 4.2, the acoustic distance between the synthetic and reference speech was calculated in the frame-by-frame fashion (see Equations 4.4 and 4.5). The following acoustic features were used to measure the acoustic similarity: MFCC, energy, ZCR, and POV (see Section 3.3 for details). The 13-dim MFCC, 1-dim short time energy, 1-dim ZCR, and 1-dim POV were extracted by a window size of 20 ms shifted every 10 ms. This manner did not consider the relation between adjacent frames. Therefore, the first-order derivatives of these extracted features were calculated. Concatenating them yielded a 32-dim acoustic feature vector for each frame.

Dimension-wise Z-score normalization was performed after feature extraction for two considerations. First, different dimensions of acoustic features had different magnitudes. Those dimensions with relative higher magnitudes were dominant during distance calculation. Normalizing all dimensions resulted in their comparable magnitudes. Second, due to speaker variation, the acoustic observations of the same phone produced by different speakers might also show differences, even on the same feature dimension. Therefore, each dimension of extracted acoustic features was Z-score normalized, which is similar to the technique of cepstral mean and variance normalization (CMVN) widely used in ASR systems. The generic CMVN is individually performed for each frame with mean and standard deviation estimated from its current utterance or all utterances of its current speaker. Such a way is suitable for long utterances but the performance usually degrades when the utterance is too short or there is no insufficient data for parameter estimation. Considering that the target utterances of the PBACU corpus were isolated words, parameter estimation by calculating acoustic features individually for each utterance was improper. Therefore, global mean and standard deviation of acoustic features were separately estimated from another corpus, the BITS corpus (Ellbogen, Schiel, and Steffen, 2004). BITS corpus is phonetically balanced and contains 6732 sentences produced by four German native speakers (two males and two females). The acoustic statistics of each dimension of acoustic features were estimated from the BITS corpus in advance. During gestural score optimization of an utterance, each dimension of acoustic features of both synthetic and reference speech were Z-score normalized by the same global statistics of that dimension.

Optimization Based on Coordinate Descent Algorithm

In order to compare the performance of genetic algorithms with that of other algorithms, a baseline optimization method based on a coordinate descent algorithm (Wright, 2015) was implemented. Similar to genetic algorithms, the coordinate descent algorithm also minimized the loss via an iterative procedure. At each time step, this algorithm chose a coordinate via a coordinate selection rule (e.g., a random or fixed order), then a line search along the selected coordinate direction was performed to determine the appropriate step size to move from the current position. During the search, the other parameters were fixed. One iteration was done until every coordinate was selected once. This procedure was repeated until the maximum number of iterations was reached.

The difference between them is that the genetic algorithm was performed on a population of individuals (i.e., a group of candidate gestural scores) while the coordinate descent algorithm was performed only on one gestural score. As shown in Algorithm 1, the input was the initial gestural score which was fed into VTL to synthesize acoustic signal. The initial global loss variable, *global_loss*, was calculated by Equation 4.4 between the reference speech and the synthetic speech of the initial gestural score. After that, the initial gestural score was iteratively optimized. In each iteration, all parameters to be optimized (i.e., duration and time constant) were traversed. For each parameter, an appropriate step was searched from a step-size array while fixing all other parameters. If a small change of the selected parameter, in the form of moving a step (i.e., adding a value to the current parameter value), resulted in a new gestural score that had smaller loss than *global_loss*, the current parameter was updated. In this study, the maximal iteration was 100 which was in line with the number of evolution generation of gene population. The steps were searched from eight candidate values around the current positions (i.e., the parameter values). The step-size array was set to $\{-20, -15, -10, -5, 5, 10, 15, 20\}$ (in ms) for the "duration" parameters or $\{-4, -3, -2, -1, 1, 2, 3, 4\}$ (in ms) for the "time constant" parameters.

Algorithm 1 Coordinate Descent

```
Input: Initial gestural score
Output: Optimized gestural score and the minimal global loss, $global\_loss$
$global\_loss \leftarrow$ Loss of initial gestural score
for $iteration = 1, 2, \ldots$ do
    for $parameter = param_1, param_2, \ldots$ do
        for $step = size_1, size_2, \ldots$ do
            Temporarily update $parameter$ by adding $step$
            Calculate temporary loss, $temp\_loss$, of the temporary gestural score
            if $temp\_loss < global\_loss$ then
                Update $parameter$ by adding current $step$
                $global\_loss \leftarrow temp\_loss$
            end if
        end for
    end for
end for
```

Optimization Based on Genetic Algorithm

For each gestural score to be optimized, a population with a certain size (*size of population*) of candidate gestural scores were created. During crossover, a group of new kids were created via gene recombination from the whole population. After gene mutation, each new kid was decoded into a gestural score which was further fed into VTL. The new kids who had better fitness (i.e., smaller loss) were selected as the new parents of the next generation. This procedure was repeated until the maximal number of evolution generations (here, 100 used in this study) was reached. It should be noted that the step size σ for gene mutation was assigned with a fixed value for all "duration" genes and another fixed value for all "time constant" genes in *initial population* and that the step sizes of all genes of all chromosomes individually adapted in each generation over time (Back, Hammel, and Schwefel, 1997).

TABLE 4.4: Valid parameter ranges and their preferred values defined in VocalTractLab (in ms).

Parameter	Min	Max	Preferred
Time constant of glottal gestures	10	39.6	12
Time constant of the first valid lung pressure gesture	5	39.6	5
Time constant of other gestures	5	39.6	12
Duration	1	-	-

VTL defined valid ranges and preferred values for the "time constant" and "duration" parameters, which are listed in Table 4.4. During the step of gene mutation for gestural score optimization, the random values from normal distributions were used (see Equation 4.3) which might cause the new values of genes to fall outside the valid parameter ranges of VTL. In this cases, the mutated genes were set to the closest valid boundary. For example, if the mutated gene of a glottal gesture had a value smaller than 5 ms, then it was reset to 5 ms. The preferred values of gesture parameters were used for loss calculation of the optimization method with time constant regularization (see Equation 4.5).

The hyper-parameters of the genetic algorithm were determined by the grid search strategy for all combination of the size of population, the number of new kids, the initial step size for gene mutation (see Equation 4.1), and the weight α (see Equation 4.5). The grid search for hyper-parameter was performed with a maximal of 50 iterations on 10 out of the 320 utterances of the PBACU corpus. The optimal values of hyper-parameters whose combination obtained the smallest average loss were selected. Table 4.5 lists the search ranges and resulting optimal values of hyper-parameters. After that, the gestural score of each of the 320 utterances was individually optimized by the genetic algorithm with the optimal hyper-parameters. The number of evolution generations for each utterance was set to 100.

TABLE 4.5: Hyper-parameters of genetic algorithm based methods for gestural score optimization.

Hyper-parameter	**Search Range**	**Optimal**
Size of population	{10, 15, 20, 25, 30}	15
No. of new kids	{100, 110, ..., 200}	150
Initial σ for duration	{0.001, 0.002, ..., 0.01}	0.005
Initial σ for time constant	{0.001, 0.002, ..., 0.01}	0.002
Weight α	{10, 20, 50, 100, 200}	100

4.4 Evaluation of the Optimization Results

The aim of ACS was not only to obtain the actual articulatory process of given natural utterance, but also to reproduce a synthetic speech which acoustically matched the original signal as closely as possible. The performance of the optimization methods could be measured in three domains: acoustic domain, articulatory domain, and perceptual domain.

4.4.1 Acoustic Evaluation

During optimization, the three optimization methods used different loss functions: two without and one with time constant regularization. However, to fairly compare their performance in the evaluation stage, only acoustic loss term was used, i.e., the cosine distance of acoustic features as defined by Equation 4.4. Figure 4.7 shows the average acoustic losses of three methods for the 320 utterances over 100 generations of genetic population or iterations of coordinate descent algorithm. "CD", "GA w/o Reg", and "GA w/ Reg" indicate the reproduced utterances by coordinate descent algorithm, genetic algorithm without time constant regularization, and genetic algorithm with regularization, respectively. Within the first five generations or iterations, the acoustic distances between synthetic speech of optimized gestural scores and the original speech sharply decreased. After the first five generations or iterations, the acoustic distance remained almost constant for the coordinate descent

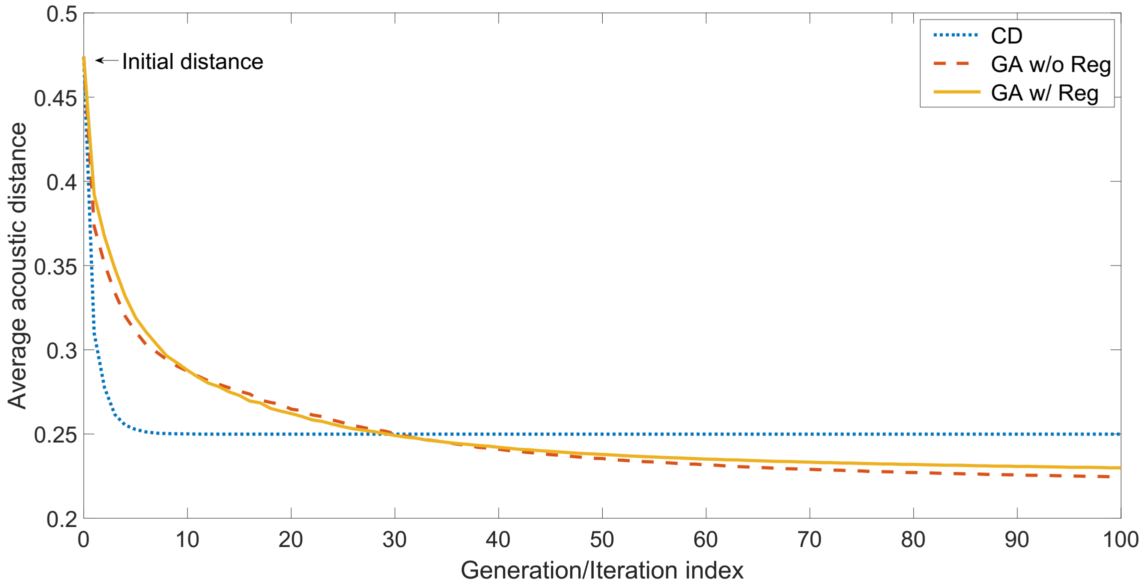


FIGURE 4.7: Comparison of acoustic distance among different optimization methods after each generation or iteration. "CD", "GA w/o Reg", and "GA w/ Reg" indicate the reproduced utterances by coordinate descent algorithm, genetic algorithm without regularization, and genetic algorithm with regularization, respectively.

method while it continued to decrease for the other two methods. After 50 generations, the genetic algorithm without regularization slightly outperformed that with regularization in terms of acoustic distance. However, this did not necessarily mean that the method without regularization could reproduce good-quality utterances since they might have unnatural segments. The benefit of regularizing time constant was prominent in terms of articulatory and perceptual metrics.

The optimized gestural scores were fed into VTL to synthesize speech, which was referred to as the reproduced speech. For the PBACU corpus, the resulting gestural scores and corresponding reproduced speech together with original utterances as well as a few animation examples of reproduced utterances can be found in Google Drive repository[2, 3]. In the acoustic domain, the performances of different optimization methods were measured in terms of acoustic similarity between the reproduced and original speech.

Figure 4.8 shows the spectrograms of the natural and reproduced speech for the word "bayrisch" originally produced by the speaker SPK-1. For the sake of convenient comparison, the spectrogram of the natural utterance is used as the reference and plotted in the middle subplot. More specifically, from top to bottom are the spectrograms of the synthetic speech of initialized gestural score ("Init"), the synthetic speech of optimized gestural score by coordinate descent algorithm ("CD"),

[2] All gestural scores and audio files for the PBACU corpus: https://drive.google.com/drive/folders/1e3lXjVDVxihtxQvOT8jdOmfVCboQQ9KS?usp=sharing

[3] A few animation examples: https://drive.google.com/drive/folders/1coFeI2u-NutYYv5SlS1972Lgjj2YP6_3?usp=sharing

the original speech produced by human speaker ("Orig"), the synthetic speech of optimized gestural score by the genetic algorithm without time constant regularization ("GA w/o Reg"), and the synthetic speech of optimized gestural score by the genetic algorithm with time constant regularization ("GA w/ Reg"), respectively. The "Init" utterance has the worst acoustic alignment with the "Orig" utterance while the utterances of optimized gestural scores have relative good alignment. The reproduced utterances by genetic algorithms have better similarity to the "Orig" utterance than that by coordinate descent algorithm (e.g., compare the high frequency regions of the segment between 0.2 second and 0.4 second and the formant transitions occuring around 0.22 second).

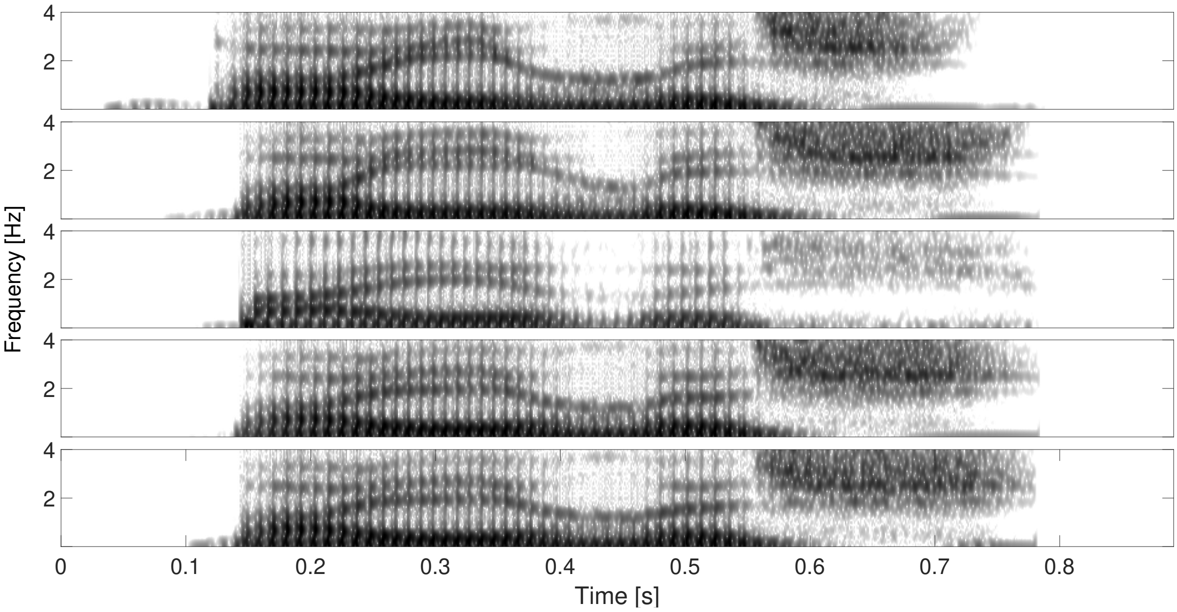


FIGURE 4.8: Spectrograms of the original and reproduced utterances for the word "bayrisch" produced by the speaker SPK-1. From top to bottom are the "Init", "CD", "Orig", "GA w/o Reg" and "GA w/ Reg", respectively.

Figure 4.9 shows the frame-wise cosine distances of between acoustic features of the reproduced utterances and the original utterance for the word "bayrisch" produced by the speaker SPK-1. Again, the "Init" utterance has the largest distance relative to the original utterance, especially for the beginning and end parts. The "CD" utterance also has a poor reproduction of the beginning part and a larger distance compared to those of "GA" utterances. Moreover, the acoustic distances of two "GA" utterances are comparable for most frames with the exception of specific segments (e.g., see the segment around 0.44 second), indicating the usefulness of regularizing time constant of gestures.

4.4.2 Articulatory Evaluation

In the articulatory domain, the performances of different optimization methods were measured in terms of statistics of optimized VTL parameters, articulatory loss,

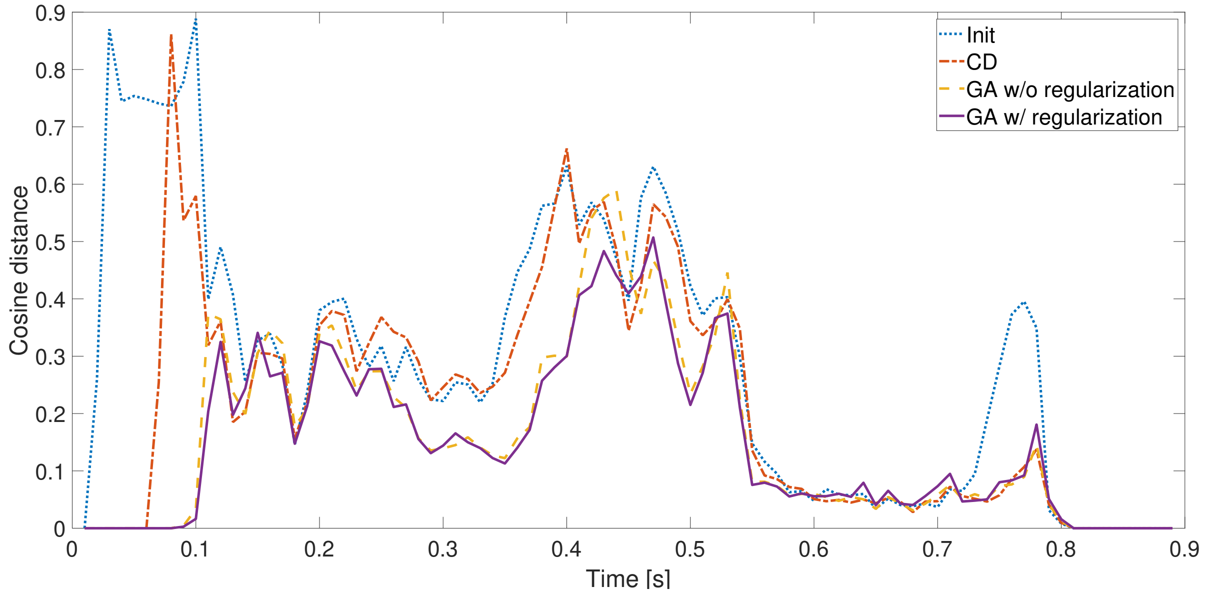


FIGURE 4.9: Frame-wise cosine distances between the reproduced and original utterances for the word "bayrisch" produced by the speaker SPK-1.

and articulatory similarity between the original and reproduced articulatory trajectories.

Table 4.6 shows the mean values and standard deviations of time constant of gestural scores optimized by different methods. The unit used here is millisecond, which is indicated in squared brackets. All gestures are roughly divided into two categories: vocal tract gestures and glottal gestures. The "Vocal tract" category contains vowel gestures and consonant gestures while the "Glottis" category contains the glottal gestures of common phonation types. Other glottal gestures for special phonation type like "pressed", "breathy", "hoarse" and "whisper" were not involved in this chapter. Since all words of the PBACU corpus have the "CV[C]" syllable structure, the glottal "stop" gesture was not involved.

TABLE 4.6: Mean values (standard deviation in parenthesis) of time constant of gestures of optimized gestural scores by different methods.

	Gesture name	**Time constant [ms]**		
		CD	**GA w/o Reg**	**GA w/ Reg**
	a	11.1 (3.2)	13.8 (6.4)	12.1 (1.3)
	e	12.1 (3.9)	15.1 (7.9)	12.1 (1.8)
	i	10.3 (3.4)	10 (4.3)	11.6 (1.4)
	o	11.9 (3.5)	13.5 (6.1)	12.3 (1.2)
	u	10.9 (2.8)	12.6 (4.6)	12 (2.2)
	2	11 (1.4)	13.3 (4.2)	12 (0.3)
	y	11.3 (3)	11.8 (3.9)	12.4 (1)
	I	10.7 (4.1)	15.3 (10)	11.7 (1.6)
	E	12.1 (3.6)	14.6 (3.8)	12.3 (1.3)

Vocal tract

Table 4.6 continued from previous page

	Gesture name	Time constant [ms]		
		CD	GA w/o Reg	GA w/ Reg
	O	11 (3.3)	11 (6.2)	12.2 (1.8)
	U	10.9 (3.4)	11.3 (5.4)	10.9 (2.6)
	9	7 (1.4)	15.1 (6.8)	12.2 (4.8)
	Y	13.5 (4.3)	15.1 (4.3)	11.5 (0.4)
	@	12.2 (4.5)	19.2 (10.4)	11.9 (0.9)
	6_low	15.4 (9.2)	20.1 (12)	13.4 (4.7)
	ll-dental-fricative	11.7 (1.3)	14.2 (7.9)	11.9 (1)
	ll-labial-closure	11.6 (2.3)	14.9 (8.2)	11.9 (1.6)
	tt-alveolar-closure	12.9 (3.3)	17.4 (9.6)	12.6 (2.7)
	tt-alveolar-fricative	11.3 (2.8)	11.6 (6.2)	11.7 (1.6)
	tt-alveolar-lateral	13 (4.8)	15.9 (9.4)	12.4 (1.7)
	tt-postalveolar-fricative	11.5 (2.4)	12 (6.2)	11.8 (1.2)
	tb-palatal-fricative	12.5 (2.5)	12.9 (6.2)	12.1 (1.1)
	tb-uvular-fricative	13.1 (3.8)	17.4 (9.1)	12.7 (2.4)
	tb-velar-closure	13.4 (3.1)	17.5 (9.6)	12.9 (2.3)
	velic-opening	14 (3.2)	16.2 (6.4)	12.3 (1.1)
Glottis	modal	13.2 (3.2)	16.3 (7.8)	12 (1.2)
	voiced-fricative	13.5 (3.1)	18.3 (8.5)	12.2 (1.5)
	voiced-plosive	13.7 (3.2)	21.8 (10.3)	12.1 (0.8)
	voiceless-fricative	13.3 (3.5)	15.9 (7.3)	12 (0.7)
	voiceless-plosive	12.9 (2.7)	16.4 (8)	11.8 (1.1)
	h	14.3 (5.6)	17.1 (9.6)	12 (2)

The "time constant" parameters should have smaller values for gestures realized with more speaking effort which corresponds to faster transitions between adjacent phones. On the contrary, the gestures realized with less speaking effort should have larger time constant value which corresponds to slow transitions. It is worth mentioning that the gestures are not equivalent to phones although they are bound tightly. The realization of the latter usually involves the coordination of several gestures. The default value for time constant defined in VTL is 12 ms for all gestures with the exception for lung pressure gestures. Time constant should vary not only from gesture to gesture but also from speaker to speaker. Table 4.6 shows the mean values for each type of gestures across two speakers' gestural scores optimized by three different methods. There are several findings regarding time constant from this table:

- The mean values of time constant of gestures optimized by the genetic algorithm with regularization were closest to the initial default value (12 ms) specified by VTL-API, followed by the other two methods without regularization: the one by the coordinate descent algorithm and the one by the genetic algorithm without regularization.
- The effect of regularization could also be reflected by the standard deviation of time constant of optimized gestures. Among the two methods based on genetic algorithms, regularizing the search range around the default value (i.e.,

the preferred one) led to smaller standard deviations of time constant of gestures. Among the two methods without using time constant regularization, their standard deviations were still different. This could be explained by the different ways of searching the optimal values. The genetic algorithm searched the optimal time constants by randomly sampling values from normal distributions while the coordinate descent algorithm searched the values in a limited set. More specifically, the coordinate descent algorithm exhausted the eight candidate values from the step-size array $\{-4, -3, -2, -1, 1, 2, 3, 4\}$ (in ms).

- Time constant of glottal gestures were generally larger than those of vocal tract gestures. The values of time constant were larger than the default values (12 ms) for all glottal gestures with the exception of the "voice-plosive" gestures optimized by the genetic algorithm with regularization. The mean values of time constants for vocal tract gestures after optimization were distributed around the initial default value. This was mainly due to the valid VTL parameter ranges. All gestures shared the same upper bound (39.6 ms) for time constants while they had different lower bounds (5 ms for vocal tract parameters and 10 ms for glottal gestures).
- The values of the open schwa gesture "6_low" were most variable compared to those of other gestures. The "6_low" gesture defined one of the vocalized allophones of the phoneme [r], which appeared in the final part of diphthongs. Its realization tended to be mostly influenced by its preceding vowels, which was also suggested in Ulbrich and Ulbrich (2007).

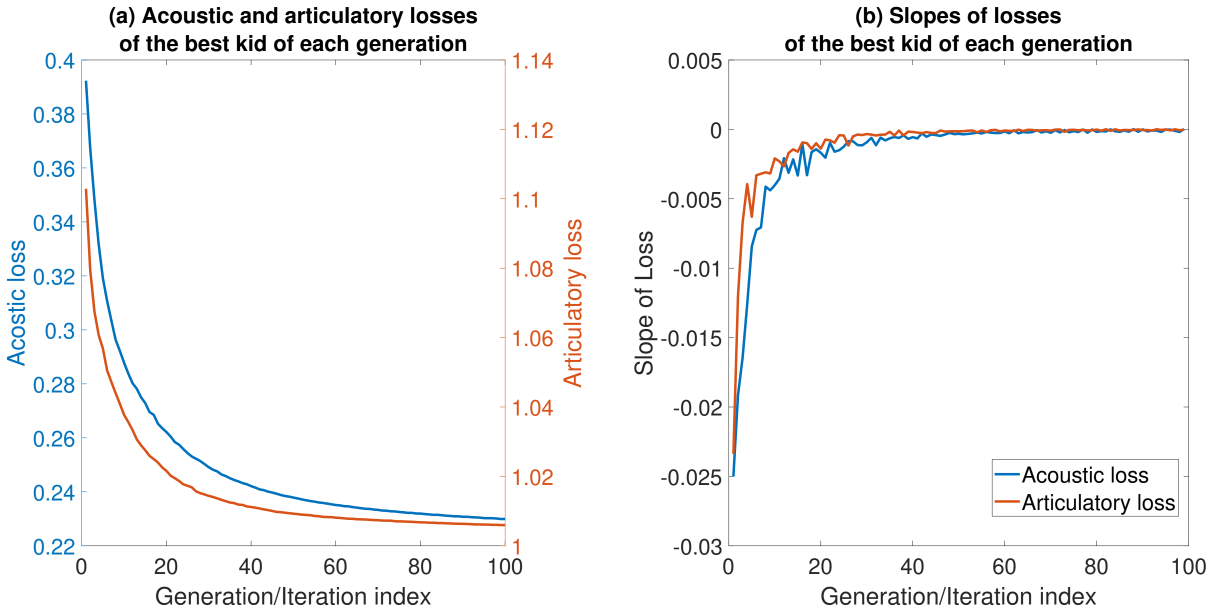


FIGURE 4.10: Comparison of two losses of the best kid after each generation (averaged across all utterances): (a) acoustic and articulatory losses; (b) corresponding slopes.

The effect of regularizing time constant could also be examined in terms of optimization losses over time. When time constant regularization was introduced during gestural score optimization based on the genetic algorithm, there were two kinds of losses (see Equation 4.5): the acoustic loss (i.e., cosine distance of acoustic features) and the articulatory loss (i.e., exponential deviation of time constants from their preferred values). In Figure 4.10, (a) shows the two losses of the best kid after each generation (averaged across all utterances) for the method using the genetic algorithm with regularization while (b) shows their rates of changes for these two losses over time. The acoustic and articulatory losses demonstrate the same tendency that they dramatically decrease in the first ten generations, continue to decrease until about the 40th generation, and approximate the plateaus. Besides, although they have slightly different magnitudes, their amplitudes of variation (i.e., the maximal minus the minimal) are comparable.

Figure 4.11 shows the average deviation of time constants of the best kid and the articulatory loss after each generation (averaged across all utterances) for the method using the genetic algorithm with regularization. The orange curves in Figures 4.10 and 4.11 are the same. Here, the articulatory loss is the whole second term of the loss function defined by Equation 4.5 while the deviations of time constants is the absolute values of differences between the optimized and their preferred values. As can be seen from Figure 4.11, the average absolute deviation of time constant (i.e., the blue curve) first increases and then decreases after about the sixth generation. However, the articulatory loss has a monotone decreasing curve. This was realized by the adaptive weight α used in Equation 4.5. Otherwise, the articulatory loss would have non-monotonic curve like that for the absolute deviation of time constant, and then influence the total loss, which might in turn result in local optimum or other side effects for optimization.

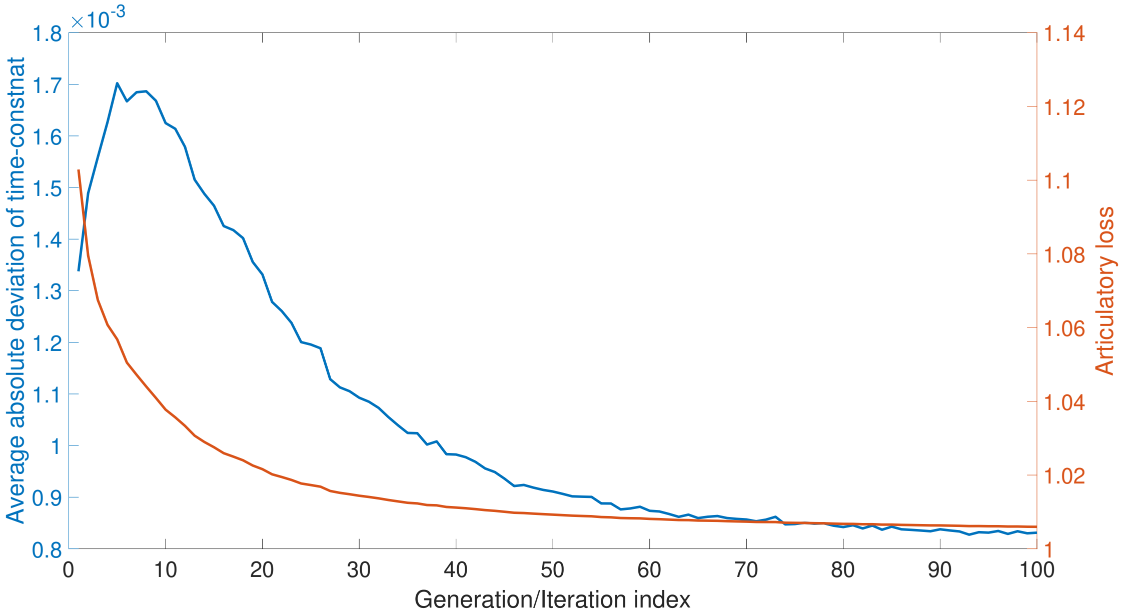


FIGURE 4.11: Average absolute deviation of time constant (in second) and articulatory loss of the best kid of each generation (averaged across all utterances).

Figure 4.12 shows the two losses between the best kid and the population after each generation (averaged across all utterances) for the method using the genetic algorithm with regularization. The blue and orange curves in the subplot (a) demonstrate the acoustic losses for the best kid and the whole population after each generation, respectively. Both of them decreased with the population evolving and the best kids had smaller losses than the whole population. However, the loss gap between the best kids and the whole population gradually became narrower with population evolving. That is to say, more *good* genes were created and kept while poor genes were eliminated (i.e., durations and time constants gradually approached the optimal values). The articulatory losses shown in the subplot (b) show a similar pattern.

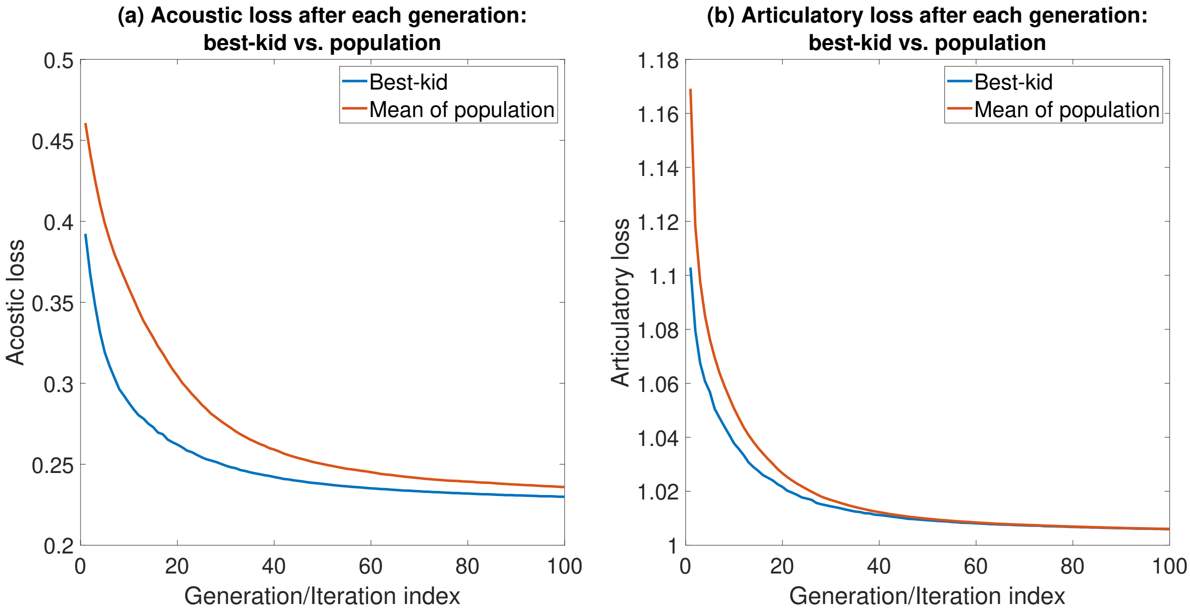


FIGURE 4.12: Comparison of losses between the best kid and the population after each generation: (a) acoustic loss; (b) articulatory loss.

The "duration" parameters of gestures could also be examined in terms of their changes over time. Figure 4.13 shows the average deviation of duration parameters from their *initial* values after each generation in second (averaged across all utterances). Subplot (a) shows the average absolute deviation of duration parameters from their initial values for the best kids and the whole population. With the population evolving, the duration parameters increasingly moved away from their initial values and approached their optimal ones. Unlike time constant, the duration deviation curves for the best kids and the whole population overlapped from the beginning to the end. During optimization, the duration parameters of gestures within a tier may become smaller or larger while the total duration may not change. In other words, the duration reduction of a gesture might be "absorbed" by its adjacent gestures or other gestures within its belonging tier. Therefore, the sum of duration deviations within the gestural score was also calculated by summing up all differences (i.e., the new duration of each generation minus the initial duration, which is not the *absolute* value). The sum of duration deviation could roughly reflect the duration change of the whole gestural score. Subplot (b) shows the sum of

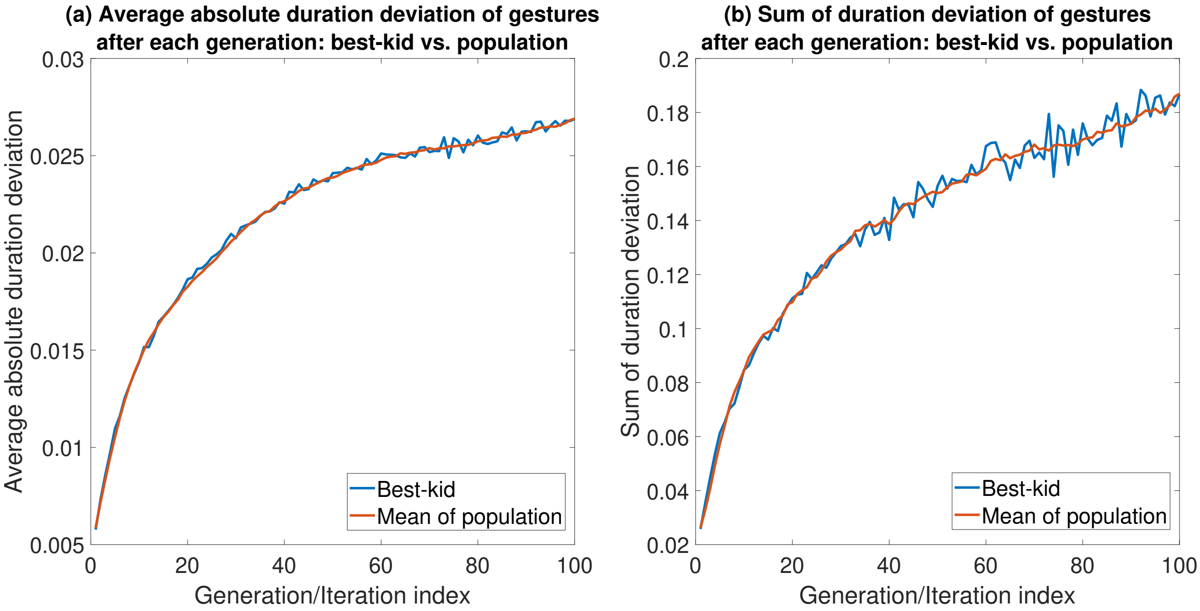


FIGURE 4.13: Duration deviation of gestures from initial values after each generation in second (averaged across all utterances): (a) average absolute duration deviation of gestures; (b) sum of duration deviation of gestures.

duration deviation (averaged across all utterances) which has the same trend like the absolute deviation but with greater magnitudes. After 100 generations of optimization, the average value for the duration deviations of the whole gestural scores is still smaller than 0.2 second. It can be speculated that the duration change mainly occurred in the form of adjusting the duration parameters among gestures, i.e., the boundary shift between gestures.

The performances of different optimization methods were further measured in terms of articulatory similarity between the estimated and reference articulatory trajectories. Since there are no ground truth articulatory trajectories for the 320 human utterances of the PBACU corpus, the 16 synthetic utterances created in Section 4.3 were used to measure the articulatory similarity. The gestural scores of these utterances were manually created with human speech as reference. Then, the corresponding *synthetic* utterances were used as the target of ACS. Repeating the steps of gestural score initialization and optimization like the procedure conducted for natural speech, 16 optimized gestural scores were obtained. The resulting gestural scores and corresponding reproduced speech together with original utterances can be found in Google Drive repository[4]. Both the ground-truth and optimized gestural scores were then converted to articulatory trajectories. The correlation coefficients (averaged across all dimensions) between the estimated and the ground-truth articulatory trajectories were calculated for different optimization methods. The cosine distance for each frame was also calculated and then averaged over all frames. Table 4.7 shows the articulatory similarity of the original and reproduced

[4] All gestural scores and audio files for 16 synthetic utterances: https://drive.google.com/drive/folders/1sstYowFSxX5vRY5ZCtZ6INMxeNWzdZVt?usp=sharing

utterances. Again, optimization by the genetic algorithm with time constant regularization achieved the highest correlation coefficients and smallest cosine distance. The genetic algorithms outperformed the coordinated descent algorithm. It can be imagined that the estimated and ground-truth articulatory trajectories almost exactly match each other, which was confirmed by visual examination (not shown here).

TABLE 4.7: Articulatory similarity in terms of correlation coefficient and cosine distance between the estimated and reference articulatory trajectories for speaker-dependent ACS.

	Correlation coefficient	**Cosine distance**
Init	0.90642	30.08×10^{-4}
CD	0.95987	11.39×10^{-4}
GA w/o Reg	0.97518	6.98×10^{-4}
GA w/ Reg	0.98473	2.38×10^{-4}

4.4.3 Perceptual Evaluation

In the perceptual domain, the performances of different optimization methods were measured in terms of the intelligibility and speech quality of reproduced utterances. These two metrics were not assessed by human listeners but by computer programs. Although the automatic evaluation by machines could not fully replace the judgement by human listeners, it could be used as a complementary indicator to compare the performance of different methods. Besides, it was particularly convenient and efficient when a large number of utterances had to be evaluated.

Evaluation by ASR

ASR accuracy was used an indicator of the intelligibility of utterances in this study. Using the same procedure adopted in Section 4.1.1, the original and reproduced speech was recognized by Google speech recognizer. For each utterance, the recognized transcription was compared with the reference one. Here, the target transcription was the word, the spoken utterance of which was expected to be reproduced by ACS methods.

TABLE 4.8: Overall utterance intelligibility measured in terms of Google ASR accuracy for ACS of human speech.

Utterance types	**SPK-1**	**SPK-2**	**Average**
Original	98.13	98.75	98.44
Init	81.88	91.25	86.56
CD	76.88	85	80.94
GA w/o Reg	76.88	83.13	80
GA w/ Reg	85	87.5	86.25

Table 4.8 shows the overall ASR accuracy of the 320 PBACU utterances reproduced by different methods. The original utterances spoken by human speakers and the

initialized utterances created by the VTL rule-based method were also recognized by the Google ASR program and their accuracies are listed in the table for the sake of comparison. The detailed recognition results for each word distinguished by the way how it was created are shown in Appendix B.

As we can see from Table 4.8, the "Original" utterances had the best intelligibility with an accuracy of 98.44% since they were produced by human speakers. The other four types of utterances were synthetic speech created from either initialized or optimized gestural scores. A close observation of the incorrectly recognized original utterances revealed that most phones of them could be correctly recognized and only a few segments were recognized to similar phones of the reference ones. Among the synthetic utterances, the "Init" type had the generally best intelligibility with an accuracy of 86.56%.

The reproduced utterances by optimizing gestural scores with coordinate descent had a relatively lower accuracy of 80.94% compared to those of the "Init" type. This was because the gestural scores were initialized using the VTL rule-based method, which could generate utterances with generally high intelligibility on the condition that the segment durations were similar to those of natural speech. However, the gestural score optimization aimed at adjusting parameters of gestures so as to make the synthetic and reference utterances temporally match and obtain the least acoustic distance between them. This process may destroy the articulators' physiological constraints, resulting to abnormal movements of articulators. With time constant regularization, the utterances optimized by the genetic algorithm achieved a comparable accuracy (i.e., similar level of intelligibility) compared to that of the "Init" utterances and simultaneously had better acoustic match with the reference utterances.

Furthermore, the intelligibility of utterances also shows the difference resulting from speaker variation. Although the original utterances were similarly intelligible for the two human speakers in terms of Google ASR accuracy, the utterances spoken by the speaker SPK-1 appeared to be more difficult to reproduce by the ACS methods than those by the speaker SPK-2. To be more precise, the recognition accuracies of utterances related to the speaker SPK-2 were generally higher than those related to the speaker SPK-1.

The above ASR results reflect the performance of the ACS methods on utterances produced by human speakers, i.e., the speaker-independent ACS. To measure their performance in the speaker-dependent scenario that was what most previous studies did, these methods were applied to VTL speech, i.e., the reference utterances to be reproduced were VTL synthetic speech. Table 4.9 shows the ASR results for the original and reproduced synthetic speech. Since both the original and reproduced utterances were synthetic speech of VTL, (i.e., the speaker-dependent ACS), the reproduction of reference utterances was much easier. Only three tokens were not correctly recognized which were found in the "Init" or "GA w/o Reg" types.

TABLE 4.9: Detailed utterance intelligibility measured in terms of Google ASR accuracy for ACS of synthetic speech. "1" indicates the utterance was correctly recognized.

Utterance	Original	Init	CD	GA w/o Reg	GA w/ Reg
bayrisch	1	1	1	1	1
besonders	1	1	1	1	1
Chemie	1	1	1	1	1
dadurch	1	1	1	1	1
genauso	1	1	1	1	1
Hierarchie	1	1	1	1	1
Komiker	1	1	1	1	1
Lehrerin	1	1	1	0	1
nebenbei	1	1	1	1	1
neugierig	1	1	1	1	1
Performance	1	1	1	1	1
rigoros	1	0	1	0	1
symbolisch	1	1	1	1	1
Teufel	1	1	1	1	1
Verkauf	1	1	1	1	1
wunderbar	1	1	1	1	1

Evaluation by PESQ and STOI

In addition to the evaluation by ASR, there are several objective metrics to measure speech quality in perceptual domain. Among them, perceptual evaluation of speech quality (PESQ) (Rix et al., 2001) and short term objective intelligibility (STOI) (Taal et al., 2011) are two popular metrics to evaluate speech quality and intelligibility, respectively. They are widely used for the speech assessment in the field of speech enhancement (Tan and Wang, 2018; Fu et al., 2019). This study also adopted them to evaluate the quality of reproduced speech. We used the python implementations for measuring PESQ[5] and STOI[6]. The PESQ and STOI scores range from -0.5 to 4.5 and from 0 to 1, respectively. Both of the two metrics are the higher the better. For each measurement, both of them accepted two inputs: the clean speech as the reference and the processed speech. In the ACS task, the reference speech was always the original utterance and the processed was one of the reproduced utterances.

Figure 4.14 shows the boxplots of PESQ scores of pooled words and human speakers for different methods. The values next to the boxes indicate the mean scores. Compared to the "Init" utterances (whose gestural scores were created using the VTL rule-based method), the utterances produced by the optimized gestural scores obtained higher PESQ scores. The "GA" methods also outperformed the "CD" method and had a slight difference between with or without regularization. The PESQ scores were further subjected to a two-way analysis of variance (ANOVA) with factors of OptimizationMethod and HumanSpeaker, showing that the main effect of OptimizationMethod ($F(3,1272) = 36.29$, $p < 0.001$) was significant. Also, the main

[5] https://github.com/ludlows/python-pesq

[6] https://github.com/mpariente/pystoi

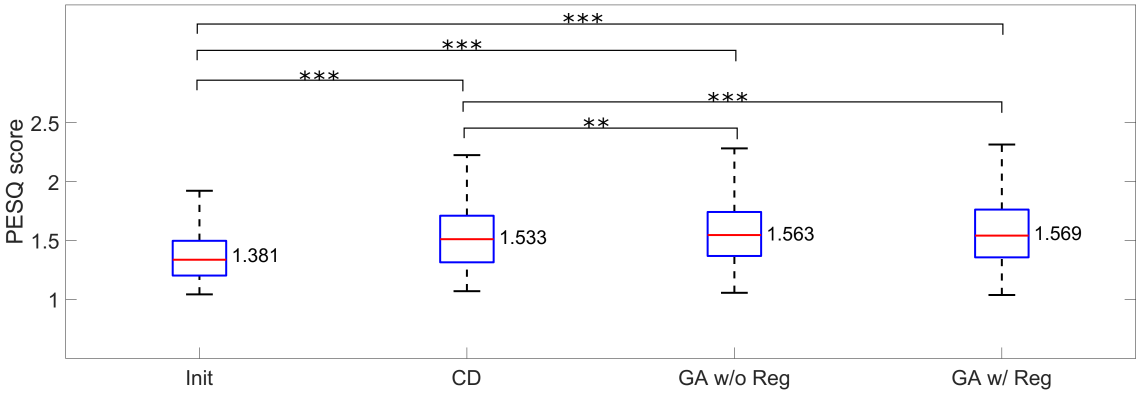


FIGURE 4.14: Boxplots of PESQ scores for different optimization methods. The numbers next to the boxes indicate the mean values. (Considering Bonferroni correction: $^{**}p < 0.0017$; $^{***}p < 0.0002$).

effect of HumanSpeaker was also significant ($F(1, 1272) = 9.37,\ p = 0.002$) with the speaker SPK-2 having a higher average PESQ score than the speaker SPK-1 (1.534 vs. 1.489), which means the utterances produced by the speaker SPK-2 were relatively easier to reproduce in terms of the PESQ metric. Their interaction effect was not significant ($F(3, 1272) = 0.33,\ p = 0.803$). The post-hoc tests were applied to each specific pairs. The paired t-test showed that the differences of PESQ scores between any pair of optimization methods were significant at a level of 0.001 except for the last pair (i.e., "GA w/o Reg" vs. "GA w/ Reg"). The lines with asterisks between two groups indicate the difference of PESQ scores was statistically significant. It should be mentioned that the significant level was adjusted using the Bonferroni correction.

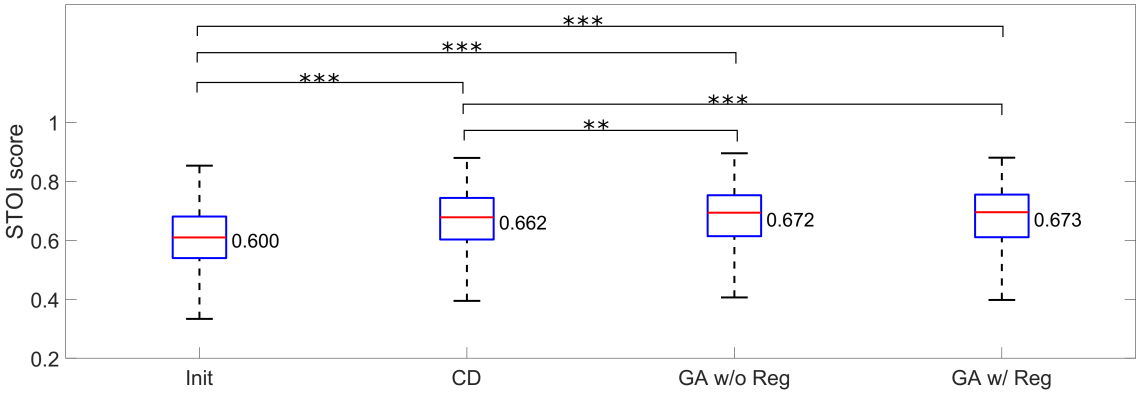


FIGURE 4.15: Boxplots of STOI scores for different optimization methods. The numbers next to the boxes indicate the mean values. (Considering Bonferroni correction: $^{**}p < 0.0017$; $^{***}p < 0.0002$).

Figure 4.15 shows the boxplots of STOI scores of pooled words and human speakers for different methods. The pattern of STOI scores among different optimization methods are similar to that of PESQ. The STOI scores were subjected to a two-way analysis of variance (ANOVA) with factors of OptimizationMethod and HumanSpeaker, showing that the main effect of OptimizationMethod ($F(3, 1272) =$

28.82, $p < 0.001$) was significant. Also, the main effect of HumanSpeaker was significant ($F(1, 1272) = 49.33$, $p < 0.001$) with the speaker SPK-2 have a higher average STOI score than the speaker SPK-1 (0.675 vs. 0.629), which means the utterances produced by the speaker SPK-2 were relatively easier to reproduce in terms of the STOI metric. Their interaction effect was not significant ($F(3, 1272) = 0.07$, $p = 0.975$). The post-hoc t-tests were also applied to each specific pairs. The pairs with a significant difference are indicated by connecting lines with asterisks. Three optimization methods achieved significantly higher scores than the "Init" method. The "GA" methods performed significantly better the "CD" method. Likewise, no significant difference in terms of STOI score was found for the "GA" methods between with and without regularization.

4.5 Concluding Remarks

This chapter investigated an ACS approach based on the genetic algorithm. For each of the given utterances, the articulatory process (in the form of gestural score) was estimated in an iterative ABS procedure which consisted of two steps: gestural score initialization and optimization. The method proposed in this study was similar to those described in Nam et al. (2012) and Gao, Stone, and Birkholz (2019), such as using the ABS based procedure and gestural scores for organizing articulatory process. However, the method proposed in this study did not rely on the assumption that transcriptions of the target utterances were known in advance; this was circumvented with the help of ASR. In the first step, the acoustic signals were first transcribed via ASR systems, yielding orthographic text; these were then converted to canonical phonetic transcriptions. A forced-alignment tool was used to generate the annotation files with coarse segmentation using acoustic signals and canonical transcriptions. The VTL rule-based method was applied to create an initial gestural score, which was used as a starting point for further optimization. The duration and time constant parameters of the initialized gestural score were encoded as genes during optimization. As the population evolved (a loop consisting of crossover, mutation, and selection), the reproduced utterances became increasingly similar to the reference ones.

Furthermore, unlike Nam et al. (2012) where the stiffness and the target of gestures were not allow to change, this study could "copy" the speaking effort of target utterances. The dynamical parameter "time constant" (similar to the "stiffness" parameter in their study) was jointly estimated with duration and timing of gestures. Consequently, this method allowed articulators to approach their underlying "target" positions with different degrees since the target realization of VTL followed the target approximation model (Prom-On, Xu, and Thipakorn, 2009; Birkholz, Kröger, and Neuschaefer-Rube, 2011). In this sense, the "target" parameters of gestures were allowed to vary to some extent. For example, if the optimized time constant of a gesture was large, then the participating articulators would not fully reach their underlying positions, which could simulate the mechanisms of coarticulation and lenition; this was true even in the case of disappeared phones where the targets of gestures were completely undershot due to very large time constants.

There were other extensions. To address the problem of one-to-many mapping, the regularization of the time constant was introduced into the loss function by adopting an adaptive hyperparameter that was used to control the trade-off between the two loss terms over time. The regularization of the time constant constrained the genetic algorithm's search for the optimal values to specific ranges, which not only reduced the number of generations that the genetic algorithm needed to run, but also allowed it to find more plausible values for the parameters to be optimized. In addition, more acoustic features were used to measure the frame-wise similarity between the reproduced and original speech, and more metrics were used to evaluate the model's performance.

The proposal was validated using the 320 utterances produced by two human speakers. To make it possible to directly compare the articulatory similarity between the estimated and reference articulatory trajectories, 16 synthetic utterances were also used. The evaluation in the acoustic, articulatory, and perceptual domains showed that the best performance was achieved by the genetic algorithm with time constant regularization; this was in comparison to the coordinate descent algorithm and the genetic algorithm without regularization.

Figures 4.10, 4.11, and 4.13 revealed another finding. When the loss of optimization decreased, the gestural scores gradually approached their final states. At the start of the gestural score optimization process, the loss reduction came from the changes in both the "duration" and "time constant" genes. However, as the population evolved, the changes in the time constant parameter were slower than those in the duration parameter. This was reflected by the relation between articulatory loss and the average absolute deviation of the time constants parameter (see Figure 4.11). Although the time constant parameters first increasingly deviated from their preferred values in the initial stages, they gradually returned to their preferred values. The average absolute deviation of the time constant parameters was smaller than 0.001 second.

Finally, there were still some general problems with the method proposed in this chapter. First, although the genetic algorithm appeared to converge (i.e., the loss function quickly plateaued) after approximately 30 generations, it was still very time-consuming. This was because a significant amount of synthetic speech (i.e., candidate kids) had to be synthesized for each generation. In addition, the iterative nature of the ABS based method made it difficult to speed up the optimization procedure. Furthermore, this method relied on the recognition results and G2P conversion. Incorrect recognition or G2P conversion may result in poor initial gestural scores which could increase the difficulty of reproducing the target utterances. Chapter 5 attempted to address these issues using the neural network based ACS methods.

Chapter 5

Articulatory Copy Synthesis Based on Deep Neural Networks

In Chapter 4, articulatory processes were estimated in an iterative loop in which the optimal gestural scores were obtained by an ABS strategy using a genetic algorithm. Although this approach had advantages such as its non-reliance on training data, it had some disadvantages such as its relatively time-consuming nature and its lack of generalization ability. To circumvent the problems of the approach described in Chapter 4, this chapter investigated another ACS approach based on deep neural networks. The premise of this new approach was to train a regression model that accepted acoustic features as inputs and produced estimated articulatory trajectories as outputs. The ACS systems built with deep neural networks and VTL are introduced in Section 5.1. To make the estimated trajectories smoother and be more articulatorily preferred by VTL, two strategies of regularizing articulatory trajectories were attempted in Section 5.2. The proposals were experimentally validated in Section 5.3, which was followed by results and analysis in Section 5.4. Section 5.5 presents a perceptual experiment in which human listeners evaluated the intelligibility of the reproduced speech. To validate the generalization ability of the trained ACS models, a complementary evaluation using the utterances of other languages was performed in Section 5.6. Section 5.7 provides some concluding remarks.

5.1 ACS Based on Deep Neural Networks

As introduced in Section 2.3, ACS aims at: (1) estimating the actual articulatory processes from given natural utterances; (2) producing synthetic speech that resembles the original speech as closely as possible. This section represents a novel ACS approach built with VTL and artificial neural networks. Figure 5.1 demonstrates the schematic diagram, which consists of three main stages. The first stage was to create paired articulatory-acoustic samples. It started from processing human speech of available corpora (in the bottom middle of the schematic diagram), over creating gestural scores, to converting them to articulatory and acoustic representations. The second stage was to train a supervised neural network (e.g., LSTM) regression model using the training samples created in the first stage. More specifically, the acoustic features and articulatory trajectories of the training samples were used as inputs and targets of the regression model. The third stage was to estimate articulatory process for new utterances. The acoustic features were first extracted from the speech signal of a testing utterance, and then fed into the trained neural network regression model. For each frame of speech, the trained model estimated a vector of articulatory parameters. The estimated articulatory parameters of all frames constituted articulatory trajectories, which were further fed into VTL to produce the synthetic speech that was considered as a duplicate of the original speech.

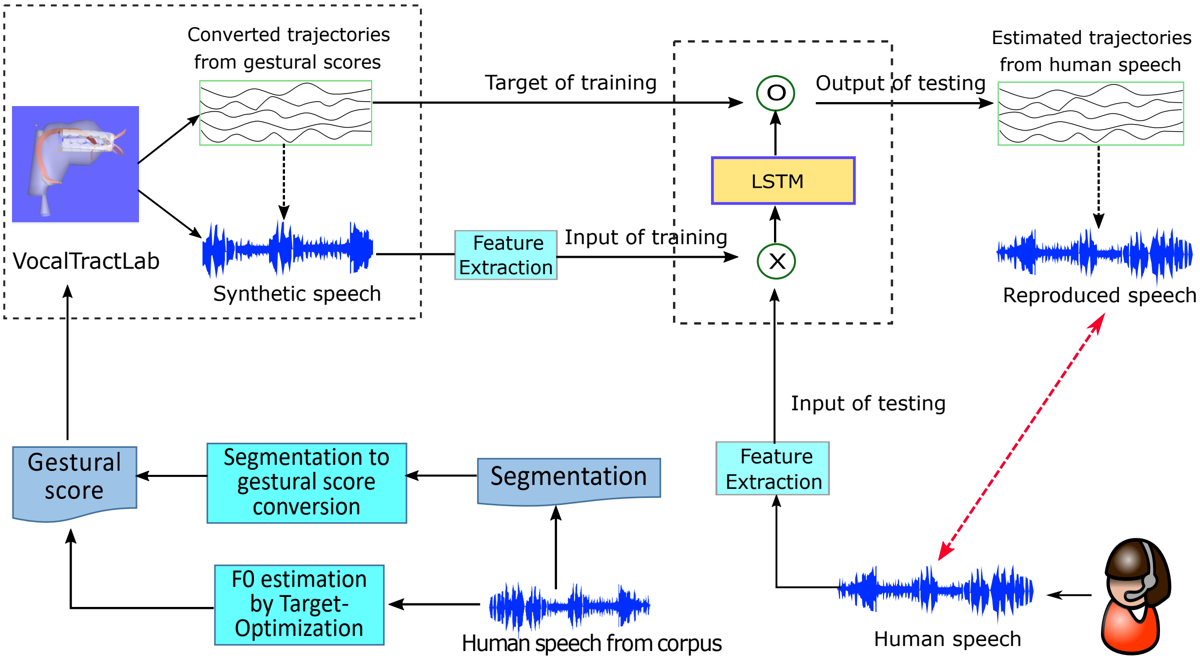


FIGURE 5.1: Schematic diagram of articulatory copy synthesis using VocalTractLab and artificial neural networks.

From the perspective of speech production, the articulatory synthesizer, VTL, plays the role of the *forward* model that maps the articulatory parameters to synthetic speech via acoustic simulation. Estimating articulatory parameters from given acoustic signals could then be defined as learning a *backward* model that inverses the mapping of the forward model. In this study, the well-developed articulatory synthesizer, VTL, was used as the forward model. Therefore, the focus of this chapter was to train a robust backward model based on neural networks.

5.1.1 LSTM Neural Networks

As reviewed in Section 2.4, lots of studies suggests that the neural network based methods generally outperform the traditional methods for training the mapping models of acoustic features to articulatory representations. Among the neural networks, the ones with a recurrent architecture are considered to be able to model the temporal dependence of acoustic and articulatory data, thus producing smooth articulatory trajectories for testing utterances.

Recurrent neural network (RNN) (Hopfield, 1982; Rumelhart, Hinton, and Williams, 1986) is a kind of artificial neural networks. Compared to the feedforward neural networks, RNNs use their internal state (memory) to retain information from past input, thus exhibiting temporal dynamic behavior. Besides, this recurrent mechanism allows them to process variable length sequences of inputs. The most basic RNN is called vanilla RNN. In theory, the vanilla RNN can maintain information in "memory" over time. However, due to the vanishing and exploding gradient problems, it is difficult to train vanilla RNNs to solve practical tasks that have long-term temporal dependencies. Therefore, other variants of RNNs are proposed, such as RNNs with gated recurrent unit (Cho et al., 2014) and long short-term memory unit (Hochreiter and Schmidhuber, 1997), usually just called "GRU" and "LSTM", respectively. By introducing a set of gates to control the flow of information, these RNN variants can learn long-term dependencies. The key difference between GRU and LSTM is that, the GRU unit has two gates ("reset" and "update") while the LSTM unit has three gates ("input", "output", and "forget"). Accordingly, LSTM has more parameters and generally outperforms better than GRU. The LSTM neural networks have been successfully applied to speech inversion (Liu et al., 2015; Shahrebabaki et al., 2019; Gao, Steiner, and Birkholz, 2020; Sun and Wu, 2020). Therefore, this chapter adopted LSTM neural networks to build the backward model for ACS.

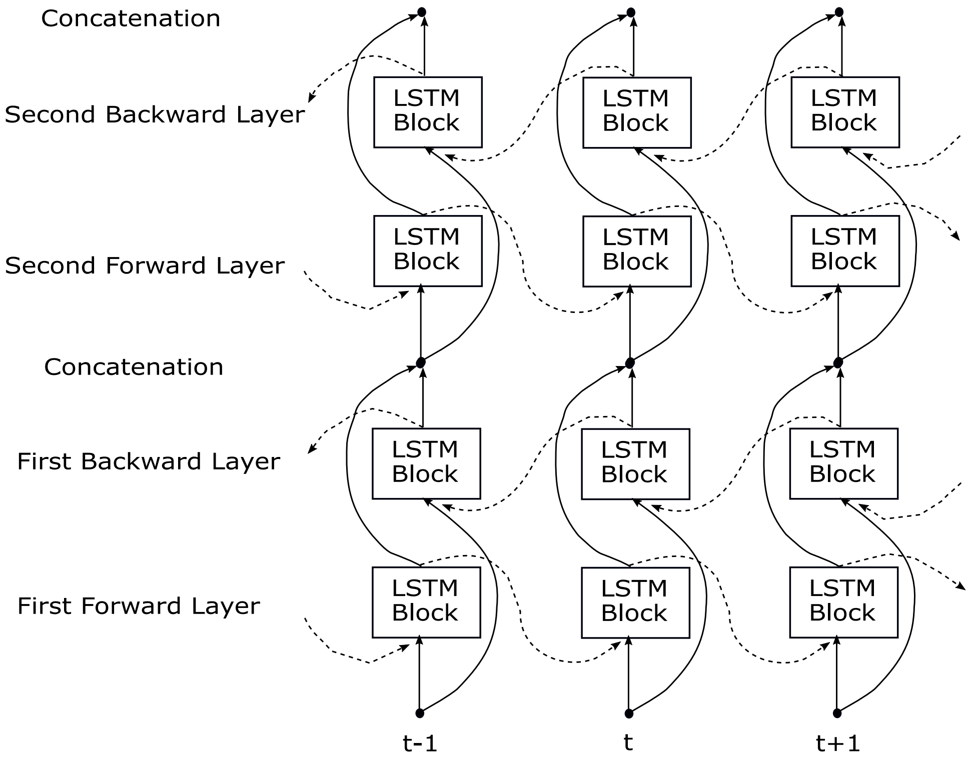


FIGURE 5.2: Diagram of a two-layer bi-directional long short-term memory (LSTM) neural networks.

As a variant of vanilla RNN, LSTM has the recurrent architecture. Figure 5.2 shows the diagram of a two-layer LSTM neural networks, which is unfolded in three time steps over the input sequence. The inner architecture of a LSTM layer is abstracted as a block of LSTM units, which is also termed "cell" (depicted by a rectangle). The solid arrows indicate the information flow from lower layers to higher layers while the dotted arrows indicate the information flow along time axis (i.e., between adjacent time steps). There are two kinds of dotted arrows: one with the left-to-right direction and another with the right-to-left direction, which connect the forward and backward layers, respectively. The unidirectional LSTM, indicated by forward layers, only preserves information from *past* inputs. For example, at the time step t, the input of each "LSTM Block" is the concatenation of the output of its lower layer (or original features) at the current time step and the output of the current "LSTM Block" from the *previous* time step $t-1$. On the contrary, the backward layers deal with input sequences from the opposite direction so that they can make full use of *future* information. Therefore, each layer of a bi-directional LSTM (BiLSTM) has two layers side-by-side: the forward layer accepts the input sequence as it is and the backward layer accepts a reversed copy of the input sequence. Besides, their outputs are concatenated together and then passed to the next/higher layer.

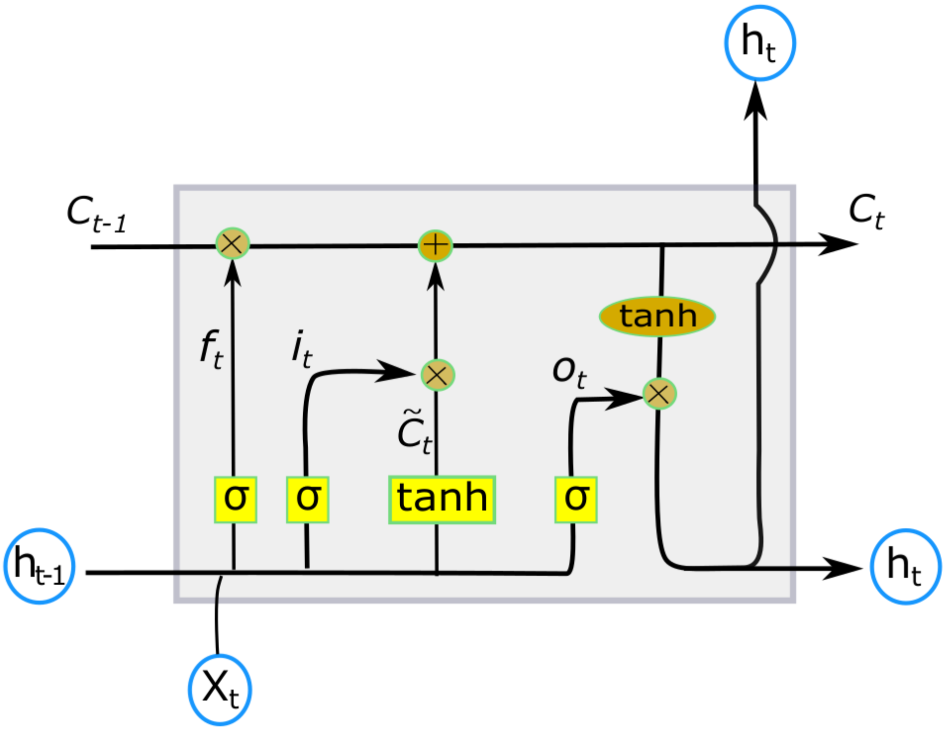


FIGURE 5.3: The inner architecture of a LSTM block/cell.

Figure 5.3 shows the inner architecture of a LSTM block/cell at the time step t. Its internal flow of information is regulated by the *gate* mechanism. There are three gates (forget gate, input gate, and output gate) which learn the importance of information and determine how much it should be kept or thrown away. The input of the LSTM cell at current time step is the concatenation of the output from its previous time step and the current input features (or output of previous layers), $[h_{t-1}, x_t]$. The concatenated inputs are used to create gates controlling the flow of information. The LSTM cell is mathematically implemented by the following equations:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{5.1}$$

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right) \tag{5.2}$$

$$\widetilde{C}_t = \tanh\left(W_C[h_{t-1}, x_t] + b_C\right) \tag{5.3}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \tag{5.4}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{5.5}$$

$$h_t = o_t \circ \tanh\left(C_t\right) \tag{5.6}$$

where f_t, i_t, and o_t represent the forget, input, and output gates, respectively. W and b with subscripts (f, i, or o) denote the weight and bias, respectively, for the corresponding gates. σ is the sigmoid function and $\circ$ represents element-wise multiplication.

The forget gate f_t decides what information should be thrown away or kept. The input gate i_t decides what new information should be stored in the cell state. Here, the new information is $\widetilde{C}_t$, which is calculated from the current concatenated inputs. Then, the current state of the cell can be updated by Equation 5.4 in which its previous state C_{t-1} is multiplied by the forget gate f_t and the new information $\widetilde{C}_t$ is multiplied by the input gate i_t. After that, the updated cell state C_t serves two purposes. It will be compressed to between -1 and 1 and then multiplied by the output gate o_t, producing the output of the cell h_t. In the meanwhile, the updated cell state C_t will be directly passed to next time step.

5.1.2 Convolutional LSTM Neural Networks

Convolutional neural network (CNN) (Fukushima and Miyake, 1982; Ciregan, Meier, and Schmidhuber, 2012) is another kind of neural network which is widely used in image processing field due to its powerful capability of capturing spatial structure and hierarchical pattern of grid-like features like images. CNN and LSTM taken together constitute the new architecture, referred to as convolutional LSTM, which can capture not only the spatial structure of features but also the temporal consistence among consecutive inputs. Convolutional LSTM is first proposed by Shi et al. (2015) to predict the future rainfall intensity from radar echo map sequences. In speech research field, it has been successfully used, for example, in emotion recognition (Ma et al., 2018) and speech enhancement (Tan and Wang, 2018). The idea behind this strengthened neural network is to let the convolutional layers capture the spatial structure (mainly along the frequency axis) of acoustic features, while the LSTM layers capture the temporal dynamics (i.e., along the time axis).

We also adopted the convolutional LSTM neural networks for ACS. Its architecture implemented in this study is shown in Figure 5.4. The acoustic features, in the form

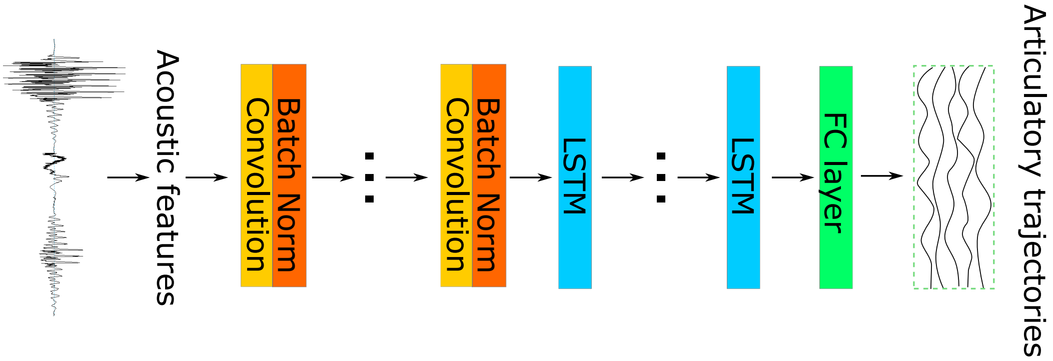


FIGURE 5.4: Architecture of convolutional LSTM neural networks for ACS.

of spectrogram, were first extracted form audio signals. Then, the convolutional layers were constructed to effectively learn spatial patterns of spectrogram. After that, the LSTM layers were used to model the temporal dependence among consecutive frames. Finally, the fully-connected (FC) layers were used to estimate articulatory trajectories. Besides, the batch normalization technique (Ioffe and Szegedy, 2015) was applied by attaching a batch normalization layer to each convolutional layer, which normalized the feature maps before they were fed into the next convolutional layer.

Convolutional Neural Network

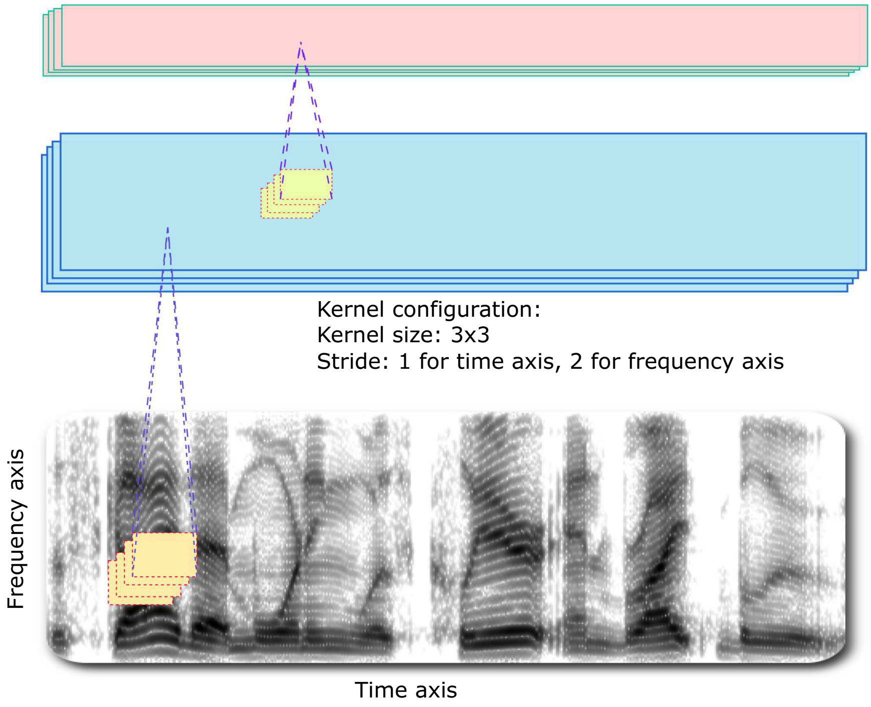


FIGURE 5.5: The process of learning features from spectrogram by convolutional neural networks (CNN).

A convolutional neural network (CNN) consists of an input layer, a convolutional layers and an output layer. The convolutional layer is its core component which consits of one or multiple kernels (or filters). Each convolution kernel slides along the input matrix for the layer, generating a feature map. The feature maps generated by all kernels are then used as inputs for the next layer. Figure 5.5 shows the process of learning features from spectrogram by CNN. Unlike the kernel configuration used in image processing tasks, the stride of kernels used in this study was always set to 1 for the time axis and 2 for the frequency axis. This assured that, through each convolutional layer, the dimension of learned features was halved while the number of frames was unchanged. That is to say, the CNN only transformed the spectrogram to a compressed, distinctive representations but did not change its width (i.e., the convolution operation kept the length of original inputs).

Batch Normalization

During the neural network training stage, when parameters of previous layers are updated, the distribution of following layers' inputs also change. This fact complicates the training of neural network models, such as a slow training speed resulting from the required smaller learning rate and careful parameter initialization. This phenomenon is usually referred to as internal covariate shift. Ioffe and Szegedy (2015) proposed the batch normalization method to address this problem by incorporating normalization as a part of the neural network architecture. This technique can not only speed up the training process by allowing higher learning rates and less careful parameter initialization, but also regularize the model and/or reduce the need for dropout operation. Algorithm 2 shows the procedure of batch normalization. For a mini-batch samples, the statistics (mean μ and standard deviation σ) were first estimated based on all values within this mini-batch. Next, each value was normalized in a Z-score fashion with the estimated statistics and then scaled and shifted by γ and β, which were learnable parameters.

Algorithm 2 Batch Normalization

```
Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
          Parameters to be learned: $\gamma, \beta$
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ // mini-batch mean

$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$ // mini-batch variance

$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$ // normalize

$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$ // scale and shift
```

Batch normalization is widely used for processing the outputs of convolutional layers. Figure 5.6 illustrates the diagram of collecting outputs of convolutional layers for statistical parameter estimation used in batch normalization. Each cuboid represented the output of the convolutional layer for a training sample. It consisted of multiple feature maps (i.e., output channels) produced by the respective number of convolutional kernels. A particular feature map indicated by a colored slice of a cuboid was a matrix whose elements were the result of applying a particular kernel of the convolutional layer to its inputs. All feature maps corresponding to a particular convolutional kernel were collected into a group. The mean μ and standard deviation σ for this channel were then estimated using all elements of this group. The following steps (Lines 6 and 7 in Algorithm 2) were further performed to produce transformed outputs.

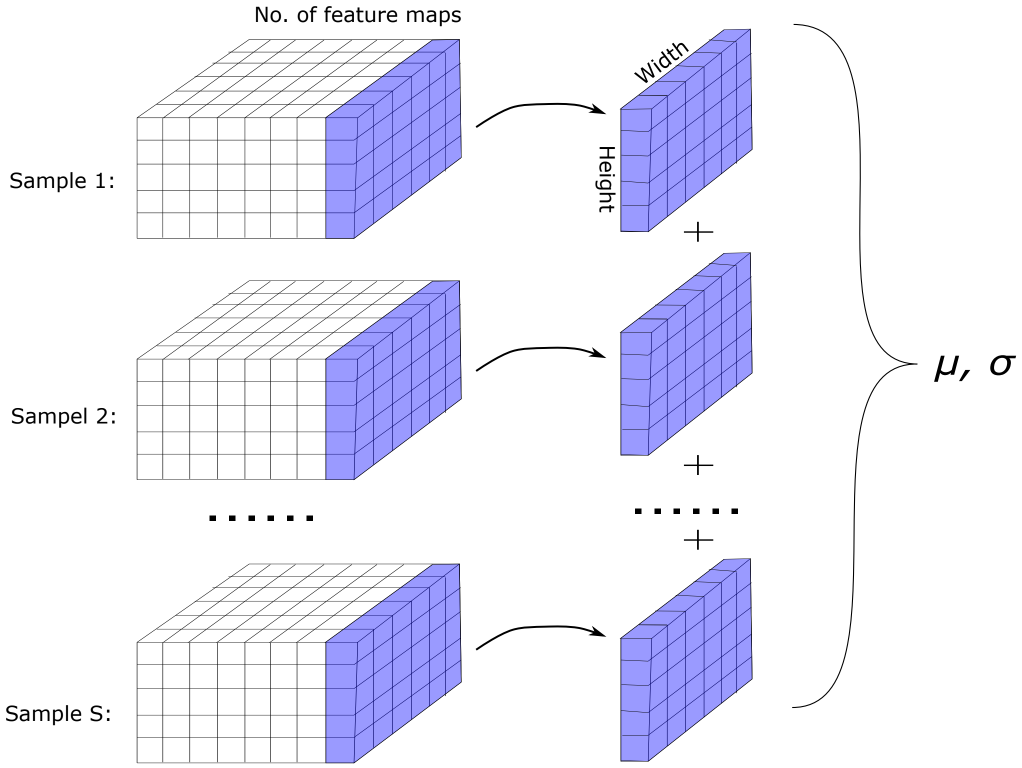


FIGURE 5.6: Diagram of collecting outputs of convolutional layers for statistical parameter estimation used in batch normalization.

5.2 Regularization Applied to Loss Function

Because the articulatory parameters of VTL were continuous values, estimating articulatory trajectories by neural network models was a regression task. Therefore, the mean squared error (MSE) between the target and estimated articulatory trajectories was used as the loss function for training neural network regression models, which was written as follows:

$$\mathcal{L}_{art} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{T}\sum_{t=1}^{T}(y_{s,m,t} - \hat{y}_{s,m,t})^2 \tag{5.7}$$

where S is the number of samples (i.e., utterances) in a mini-batch, M is the number of articulatory trajectories of utterances (i.e., the number of VTL parameters modeled in the regression task), and T is the number of time steps of utterances (i.e., the number of frames). The variable $\hat{y}_{s,m,t}$ denotes the estimated articulatory parameter at the t-th time step for the m-th parameter of the s-th sample while the variable $y_{s,m,t}$ denotes the corresponding target. Accordingly, $\mathcal{L}_{art}$ is referred to as articulatory loss.

Some speech inversion studies statically estimated a vector of articulatory parameters for a frame of acoustic features with no regard for the dynamical properties or continuity of the articulatory trajectories over time. Therefore, to make the estimated articulatory trajectories smoother, researchers usually employed filters to smooth the output of inversion models, e.g., the 15-point median filter used in Howard and Huckvale (2005) and the Kalman filter used in Mitra et al. (2010) and Sivaraman et al. (2016). Moreover, some studies directly incorporated additional constraints into the process of speech inversion. For example, Panchapagesan and Alwan (2011) incorporated the regularization term, calculated as the sum of squares of articulatory parameters (the relative values to neutral ones), and the continuity/smoothness term, calculated as the sum of squares of the first order time-derivatives of articulatory parameters into the loss function so that the articulatory parameters were smooth and not going far from the mean or neutral positions. Dang and Honda (2002) incorporated a physiological constraint (specifically, the quantitative relationship between frequency difference of the first two formants and tongue dorsum position) into the inversion procedure. These strategies solved the one-to-many problem of speech inversion to some extent and increased smoothness of estimated articulatory trajectories.

In this study, the regularization or constraint techniques were also used. Chapter 4 incorporated the deviations of the "time constant" parameters from their preferred values into the loss function. This chapter investigated another two regularization or constraint terms (in the form of additional losses) in the loss function.

5.2.1 Smoothness Loss

In this study, the MSE between target and estimated articulatory trajectories was used as the loss function for training neural network regression models. One can imagine a case there are two sets of estimated articulatory trajectories, both of which have the same MSE loss. However, the first set of trajectories are relatively smooth while the second set of trajectories frequently fluctuate around the target ones. Obviously, the first set of estimated trajectories are better than the second. Therefore, the usefulness of smoothness loss was investigated in this study, which was motivated by the task of planning motion trajectory for robots (Dai et al., 2020a). Hogan (1984) pointed out that the smoothness of a trajectory could be quantified as a function of *jerk*. Mathematically, jerk is the third order time derivative of location (i.e., position or displacement). In other words, jerk is the time derivative of acceleration. In this study, the squared *jerk* of articulatory trajectories was calculated with the following equation:

$$\mathcal{L}_{smooth} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{T}\sum_{t=1}^{T}\left(\frac{y_{s,m,t+2} - 2y_{s,m,t+1} + 2y_{s,m,t-1} - y_{s,m,t-2}}{2h^3}\right)^2 \tag{5.8}$$

where h is the time interval between two adjacent frames. In fact, the *jerk* is the central finite difference for the third order time-derivative of articulatory trajectories. Accordingly, $\mathcal{L}_{smooth}$ is referred to as smoothness loss. It can be further incorporated with the articulatory loss, $\mathcal{L}_{art}$, forming a new loss function as follows:

$$\mathcal{L}_{artsmo} = \mathcal{L}_{art} + \alpha\mathcal{L}_{smooth} \tag{5.9}$$

where the weight α is used to balance the two loss terms.

5.2.2 Acoustic Loss

Incorporating the smoothness loss $\mathcal{L}_{smooth}$ with the articulatory loss $\mathcal{L}_{art}$ into the loss function might further make the estimated articulatory trajectories smoother, thus obtaining higher correlation between the target and estimated articulatory trajectories. However, this is still insufficient to effectively train neural network models for ACS. Although some speech inversion studies (especially for the AAI task) can achieve very high correlation on the articulatory recordings, it is still unclear whether the estimate articulatory parameters are fully consistent with those of natural articulatory process of human speakers. One can imagine another case there are two sets of estimated articulatory trajectories that are parallel to the target ones and hence have the same correlation coefficient. However, the first set is very close to the target while the second set has a big overall shift (i.e., an offset value) relative to the target. There is no doubt that the synthetic speech of the first set of estimated articulatory trajectories has better quality than that of the second set. Therefore, another regularization method was proposed for ACS in this study.

As mentioned in Section 5.1, VTL played the role of the forward model that mapped the articulatory parameters to synthetic speech via acoustic simulation while the neural networks played the role of backward model that inversely mapped the acoustic representations to the articulatory representations. The forward model was only used to create training samples or synthesizing speech from estimated articulatory trajectories during testing. In order to regularize the ACS system, the forward model was also implemented as a part of the neural networks. Figure 5.7 shows the diagram of regularizing the model by concatenating the backward and forward neural networks and jointly training them to minimize the sum of articulatory and acoustic losses. The LSTM based backward model in the subplot (a) accepted acoustic features as inputs and used articulatory trajectories as targets. The same training samples were used to train the forward model, demonstrated in the subplot (b), which accepted the articulatory trajectories as inputs and used the acoustic features as targets. Not directly using VTL as the forward model was due to that it was hard to obtain the gradients of acoustic loss with respect to the articulatory parameters.

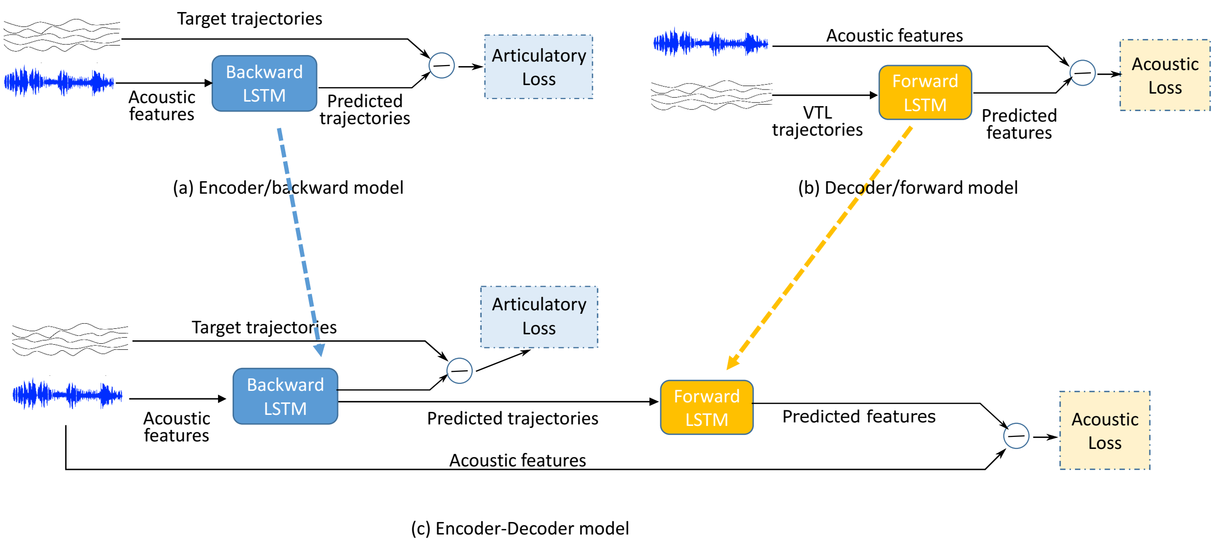


FIGURE 5.7: Diagram of neural network regression model for ACS with an acoustic loss regularization: (a) the encoder/backward model; (b) the decoder/forward model; (c) the encoder-decoder (backward-forward) model.

The MSE between estimated "acoustic features" and the real acoustic features was calculated by the following equation:

$$\mathcal{L}_{aco} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}(x_{s,n,t} - \hat{x}_{s,n,t})^2 \tag{5.10}$$

where N is the dimension of acoustic features and $x_{s,n,t}$ and $\hat{x}_{s,n,t}$ are the real and estimated acoustic features, respectively. Accordingly, $\mathcal{L}_{aco}$ is referred to as the acoustic loss. It can be further incorporated with the articulatory loss, $\mathcal{L}_{art}$, into a new loss function as follows:

$$\mathcal{L}_{artaco} = \mathcal{L}_{art} + \beta\mathcal{L}_{aco} \tag{5.11}$$

where the weight β is used to balance the two loss terms. The model in the subplot (c) was the concatenation of the backward model and the forward model. To speed up the training process, the forward and backward were separately trained in advance using the same samples but with an opposite way of determining inputs and targets. After that, they were jointly trained once again using the articulatory-acoustic samples to minimize the sum of the articulatory loss and the acoustic loss, $\mathcal{L}_{artaco}$. The original inputs were acoustic features and the outputs of the forward model were estimated acoustic features. In this sense, the whole neural network could also be regarded as an encoder-decoder model. In the testing stage, only the trained backward model was used to estimate articulatory trajectories. The estimated articulatory trajectories by the backward model (i.e., the encoder) were the latent representations of the original inputs and the forward model (i.e., the decoder) could recover the original inputs from the encoded features.

5.3 Experiments

5.3.1 Dataset

To train a robust neural network based regression model, a large number of paired articulatory-acoustic samples are required, which are expected to cover acoustic and articulatory space as fully as possible. The text to be spoken should cover as many linguistic phenomena as possible and be produced by as many speakers as possible, thus yielding different acoustic and articulatory realizations for the same linguistic units. Acquiring such data is very hard and even impractical. One way to circumvent this problem is to use synthetic data generated by articulatory synthesizers. This strategy has been attempted in Mitra et al. (2013) and Mitra et al. (2014) where synthetic articulatory-acoustic data were created with the TADA synthesizer and then used to train speech inversion models. This study also created a synthetic articulatory-acoustic dataset using the VTL synthesizer with a deliberate coverage on the acoustic and articulatory diversity.

Creation of Articulatory-Acoustic Samples

As shown in Figure 5.1, the left part demonstrates the stage of creating articulatory-acoustic samples. It started from a standard speech corpus. In this study, the *Kiel Corpus of Spoken German* (Kohler, Peters, and Scheffers, 2018) was used as the basis. Its text consisted of seven text corpora that had a good coverage of German phoneme combinations. Those sentences of the text corpora used in this study sentence were spoken by 14 speakers, thus covering, for example, different speaking rates and pitch levels. The speech segmentation was already done via force-alignment followed by manual adjustment in the original corpus. The segmentation files were processed and then converted to gestural score files via VTL-API. The f_0 gestures were separately estimated from acoustic signals and then used to replace correspond pseudo f_0 gestures in the gestural scores created via VTL-API. Next, the "merged" gestural scores were fed into VTL which internally converted them into articulatory parameters and then synthesized speech via the acoustic simulation model. Finally, acoustic features (MFCC or spectrogram) were extracted from synthetic speech with the frame rate equal to that of VTL articulatory parameter vectors. Figure 5.8 shows the diagram of splitting an acoustic signal into successive overlapped frames and their alignments with VTL parameter frames. The frame rate of VTL was 400 frames/second and the window length used in this study to extract acoustic features (MFCC or spectrogram) was 20 ms. A frame of VTL parameters described the articulatory state at a specific time point while a frame of acoustic signal was a segment (i.e., an interval) of speech. Therefore, in this study, a frame of acoustic signal was aligned with the VTL parameter frame (i.e., a vector) whose time point was equal to the center of the acoustic frame, thus forming an articulatory-acoustic data frame. For example, the center of the first acoustic frame was at 10 ms position which was aligned with the fifth VTL parameter frame, and the center of the second acoustic frame was at 12.5 ms which was aligned with the sixth VTL parameter frame, and so on. All such frames of an utterance constituted an articulatory-acoustic sample. In this fashion, the first and last four VTL frames were discarded. In practice, the silence frames in the beginning and end of utterances were excluded from training samples.

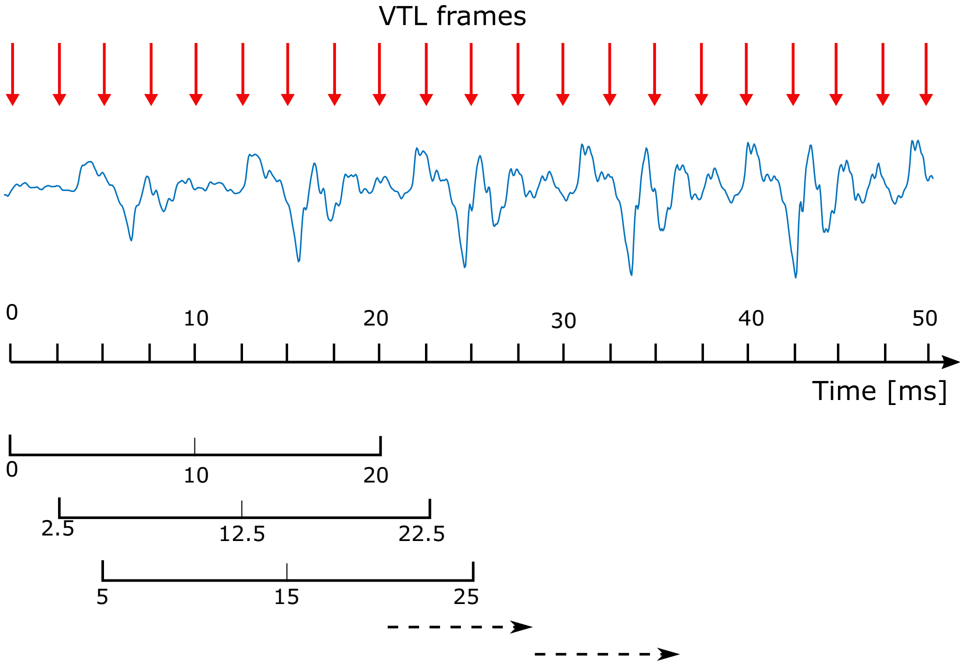


FIGURE 5.8: Diagram of splitting acoustic signal into overlapped frames and their alignments with VocalTractLab parameter frames.

However, the training samples created by this method only covered limited articulatory and acoustic variations since VTL-API used a rule-based method to convert segment sequences to gestural scores, during which the time constant was always set to 12 ms for supra-glottal gestures, thus producing almost same transition patterns for the phoneme combinations. Nevertheless, for an utterance, different speakers, even for a particular speaker under different conditions (e.g., speaking rates or emotions), have different realizations of articulatory processes and acoustic results. Therefore, upon the created gestural scores and synthetic speech, more variants of synthetic articulatory-acoustic samples were created so as to cover larger articulatory and acoustic space.

First, for the created gestural score of each utterance of the Kiel corpus, the glottal gestures with the "modal" phonation type were *globally* replaced by either the "breathy" or "pressed" gestures. Up to this point, for an utterance, there were three gestural scores that differed in the phonation type. It should be noted that, only the "modal" glottal gestures that were mainly related to sonorants were replaced and the other gestures related to fricatives, plosives, and glottal-stop were fixed. After that, for each of the three gestural scores, the "time constant" parameter of each gesture was *individually* manipulated by sampling values from a normal distribution. This step yielded another three new gestural scores that differed in the "time constant" parameters from their corresponding ones. Then, the six gestural scores for an utterance were fed into VTL, producing acoustic signals. Next, the vocal tract length of the speaker was manipulated not by changing anatomical parameters of the VTL model speaker but by transforming acoustic features of synthetic speech while keeping the corresponding gestural scores unchanged. This operation was motivated by vocal tract length perturbation (VTLP), a technique used in speech recognition (Jaitly and Hinton, 2013). To increase the robustness of ASR systems, the training samples can be augmented by transforming inputs in a way that does not change the label. VTLP should not be confused with vocal tract length normalization (VTLN) in ASR research. Unlike the VTLN technique that reduces speaker variation by fitting a warp factor to acoustic features of each training and testing speaker (or utterance), VTLP increases the input diversity for each label by manipulating acoustic features of training samples with a random perturbation factor. In this study, the vocal tract length (to be more precise, the formant frequencies of synthetic speech of gestural scores) were manipulated with the "change gender" functionality of Praat (Boersma and Weenink, 2019), during which a random value controlling the formant shift ratio was sampled from a normal distribution.

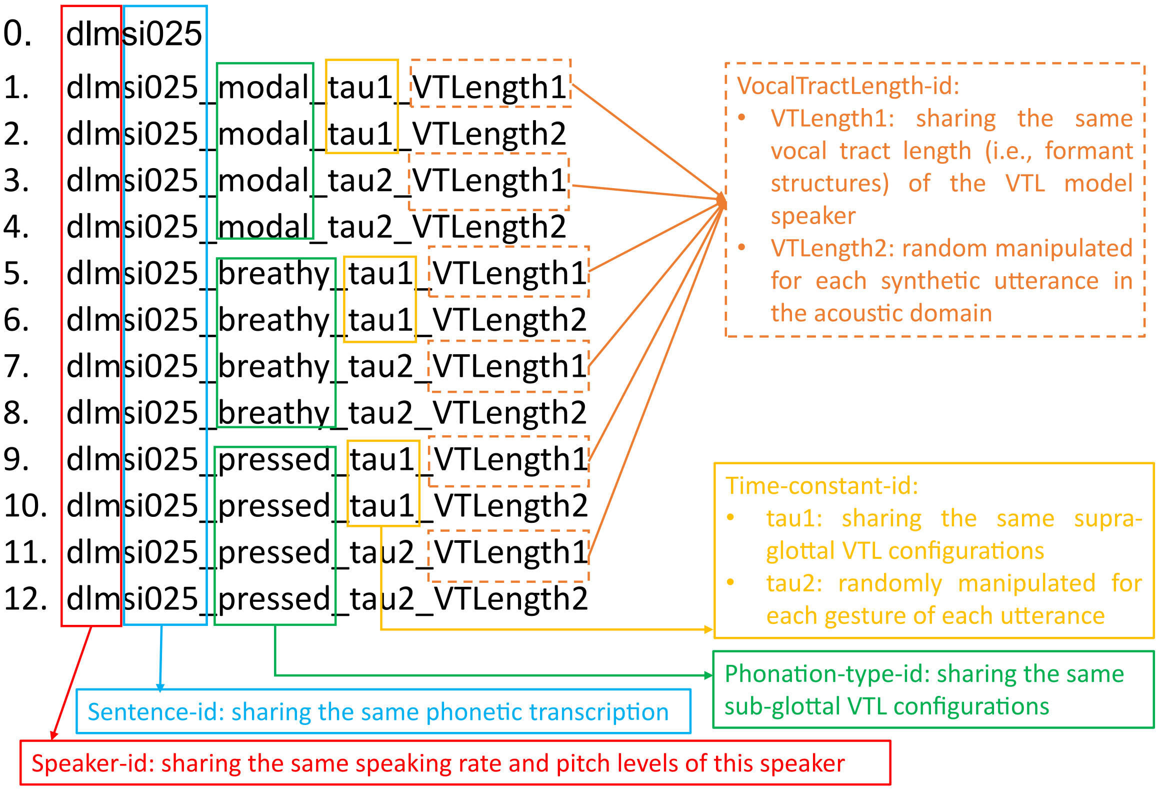

0. dlmsi025
1. dlmsi025_modal_tau1_VTLength1
2. dlmsi025_modal_tau1_VTLength2
3. dlmsi025_modal_tau2_VTLength1
4. dlmsi025_modal_tau2_VTLength2
5. dlmsi025_breathy_tau1_VTLength1
6. dlmsi025_breathy_tau1_VTLength2
7. dlmsi025_breathy_tau2_VTLength1
8. dlmsi025_breathy_tau2_VTLength2
9. dlmsi025_pressed_tau1_VTLength1
10. dlmsi025_pressed_tau1_VTLength2
11. dlmsi025_pressed_tau2_VTLength1
12. dlmsi025_pressed_tau2_VTLength2
VocalTractLength-id:
• VTLength1: sharing the same vocal tract length (i.e., formant structures) of the VTL model speaker
• VTLength2: random manipulated for each synthetic utterance in the acoustic domain
Time-constant-id:
• tau1: sharing the same supra-glottal VTL configurations
• tau2: randomly manipulated for each gesture of each utterance
Phonation-type-id: sharing the same sub-glottal VTL configurations
Sentence-id: sharing the same phonetic transcription
Speaker-id: sharing the same speaking rate and pitch levels of this speaker

FIGURE 5.9: The relationships of 12 VTL-Kiel samples corresponding to an original utterance of the Kiel corpus (with "dlmsi025" as an example).

The detailed description of creating paired articulatory-acoustic samples was given in Section 3.2.2. Applying the above-mentioned method to utterances of the Kiel corpus generated a paired articulatory-acoustic corpus, which was referred to as VTL-Kiel corpus. For each original utterance, there were 12 articulatory-acoustic samples (3 phonation-types × 2 time-constants × 2 vocal-tract-lengths). Figure 5.9 demonstrates the relationships of these 12 VTL-Kiel samples corresponding to an original utterance of the Kiel corpus. Sample-0 ("dlmsi025") is the original utterance of the Kiel corpus and the other 12 samples are synthetic utterances. Sample-1 ("dlmsi025_modal_tau1_VTLength1") corresponds to the "prototype" gestural score which was created by the VTL-API rule-based method with the utterance "dlmsi025" as reference. The other 11 samples were its variants created by manipulating the "Sample-1" sample in the articulatory or acoustic domain. The first three letters of the sample names indicate the speaker index. Therefore, all these 13 samples had the same speaking rate and pitch level of the speaker "dlm" since the duration and pitch information of the original utterance were not manipulated. The following five letters or digits indicate the sentence index, suggesting that they had the same phonetic transcription of the sentence "si025". The remaining fields delimited by underscores indicate how the commonalities and differences among the 12 samples. The samples falling into one of the green rectangles had the same sub-glottal articulatory configurations ("modal", "breathy", or "pressed"). The samples whose time constant parameters are within the yellow rectangles ("tau1") had the same supra-glottal articulatory configurations. However, the supra-glottal parameters of other samples are different because the time constant parameters of their gestural scores were *individually* manipulated with random factors. The samples with the "VTLength1" indicated by the dashed orange rectangles had the same vocal tract length of the model speaker of VTL. The vocal tract lengths of other samples were *individually* manipulated with random factors in the acoustic domain.

Data Partition

Table 5.1 gives a summary of the numbers of sentences, speakers, and utterances of the Kiel corpus. There were 598 unique sentences, each of them was produced by one or several speakers, yielding 1998 natural utterances in total. Using the method proposed in this study created 12 articulatory-acoustic samples for each original natural utterance. Therefore, the VTL-Kiel corpus contained 23976 samples in total.

TABLE 5.1: Summary of the numbers of sentences, speakers, and utterances of the Kiel corpus.

	Text Corpus (No. of sentences)	**Speakers**	**No. of total utterances**
	Berlin (100)	K01, k03, k05, k61, k63, k65	600
	Marburg (100)	K07, k09, k11, k61, k67, k69	600
	Erlangen (100) + Siemens (100)	dlm, hpt, kko	600
	CNET (20) + Kohler (63) + SEL (70) + Schiefer/Sommer (15) + Tillmann/Kohler (30)	k61	198
Total	598	14	1998

The 598 sentences of the Kiel corpus were partitioned into three groups with the proportions of 80%, 10%, and 10%, the corresponding created articulatory-acoustic samples of which formed the training, validation, and test subsets of the VTL-Kiel corpus, respectively. That is to say, there was speaker-overlap but no text-overlap among these three subsets. Table 5.2 gives a summary of utterances of the three subsets of the synthetic VTL-Kiel corpus. The detailed dataset partition for training and validating the neural network based ACS systems is given in Appendix C.

TABLE 5.2: Summary of the number of utterances of VTL-Kiel corpus.

Subset	**No. of sentences**	**No. of natural utterances**	**No. of synthetic utterances**
Training	478	1601	19212
Validation	60	199	2388
Test	60	198	2376

5.3.2 Voice Activity Detection for Lung Pressure Trajectory

The subglottal pressure is need for acoustic simulation of articulatory synthesis. In VTL, this is controlled by the *lung pressure* parameter. Compared to other articulatory parameters, the lung pressure parameter is less variable. The lung pressure trajectory can be roughly divided into three phases: (1) lung pressure's ramp up from 0 dPa to 8000 dPa, which corresponds to the utterance initial part (i.e., voice activity onset); (2) lung pressure's plateau with the fixed value of 8000 dPa, which corresponds to the long stable part during speech production; (3) lung pressure's ramp down from 8000 dPa to 0 dPa, which corresponds to the utterance final part (i.e., voice activity offset). Therefore, only three valid lung pressure gestures are usually enough to describe the lung pressure trajectory of an utterance.

In this study, the lung pressure parameter was not modeled in the neural network regression models but separately estimated from acoustic signals. Here, the voice activity detection (VAD) technique, also known as speech activity detection or speech detection, was used to determine the onsets and offsets of three lung pressure phases. VAD refers to the automatic determination of whether an acoustic segment is speech or silence, i.e., the presence of speech. This technique is widely used in the preprocessing stage of many applications such as speech coding and speech recognition.

In this study, VAD was done using the python version interface[1] of the VAD implement of the WebRTC project of Google, which is one of the best available, fast, and free VAD tools. WebTRC-VAD accepts a frame of speech (10, 20, or 30 ms) as input and returns the binary decision ("0" for silence and "1" for speech). It provides users with an option to specify the aggressiveness mode which is an integer between 0 and 3. "0" is the least aggressive about filtering out non-speech, "3" is the most aggressive. VAD for utterance-initial or -final plosives and fricatives are vulnerable since they are easily confused with the background noise. In this study, the aggressiveness mode was set to "2" which was a compromise choice that dealt with well for all phone types. The frame size was set to 10 ms. For an utterance, WebTRC-VAD was applied to a sequence of non-overlapped frames, outputting a binary string of

[1] https://github.com/wiseman/py-webrtcvad

decisions. After that, post-processing was done to determine the voice activity onsets and offsets. A sliding window of 100 ms (10 audio frames) scanned the binary decision string. Since extra silences (100 ms) were padded to the utterances in this study, the VAD results were always "0" for the beginning and end parts of the utterance. The voice activity onset was triggered when the number of speech frames within the sliding window was above nine (90% of the frames). Likewise, the voice activity offset was triggered when 90% of the frames in the sliding window were non-speech. Figure 5.10 shows examples of WebTRC-VAD results: one for a natural utterance ("Badminton" produced by the speaker SPK-1 in the PBACU corpus) and another for a synthetic utterance ("dlmsi036_modal_tau1_VTLength1" in the VTL-Kiel corpus). The red vertical lines indicate the detected onsets or offsets. It should be mentioned that the detected utterance onsets and offset were always integer multiples of 10 ms.

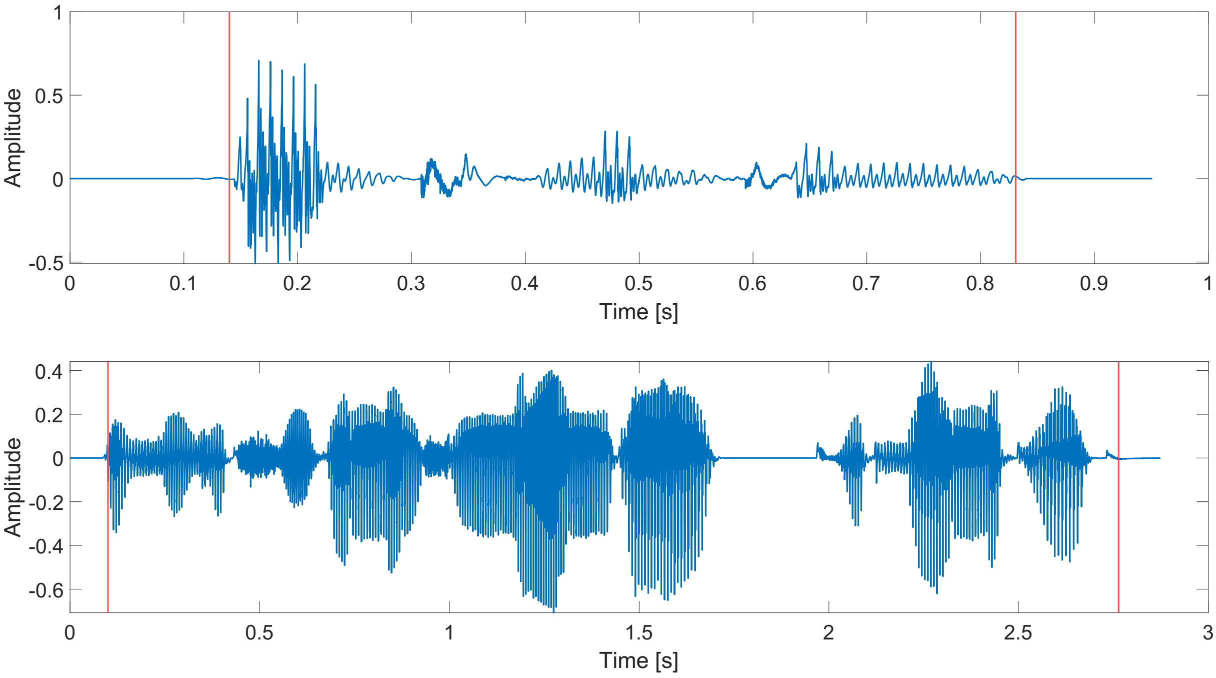


FIGURE 5.10: Examples of voice activity detection results. The upper and lower subplots are the results for a natural utterance of the PBACU corpus and a synthetic utterance of the VTL-Kiel corpus, respectively. The red vertical lines indicate the detected onsets or offsets.

Based on the VAD results, the lung pressure trajectory was created using a rule-based method. During the conversion of segment sequences to gestural scores, VTL used the values of 5 ms and 12 ms for time constants of the first and last valid lung pressure gestures, respectively. That is to say, the ramp up phase of lung pressure from 0 dPa to the stable state (8000 dB) was achieved faster than the ramp down phase from the stable state to 0 dPa. Therefore, different durations were used for specifying the ramp phases in this study. The left half of a Blackman window (30 ms) and the right half of another Blackman window (70 ms) were used to create the lung pressure ramp up and ramp down phases, respectively. Figure 5.11 shows an example of VAD based lung pressure trajectory created with Blackman windows.

The time points, T_1 and T_2, are the detected voice activity onset and offset, respectively.

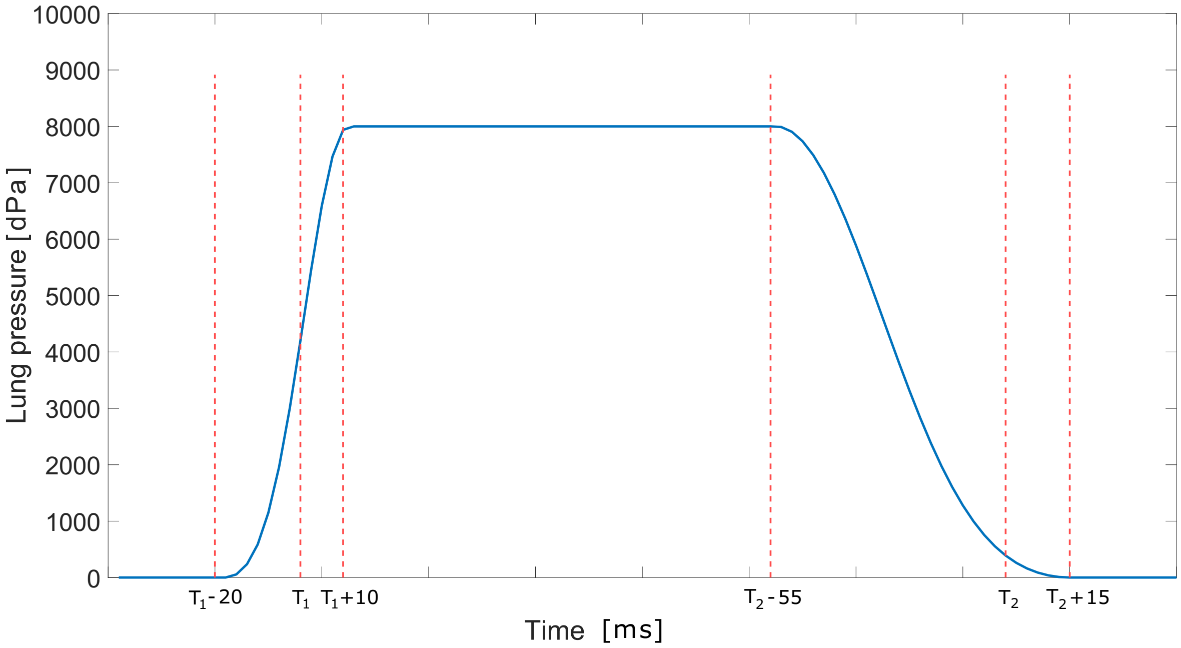


FIGURE 5.11: VAD based lung pressure trajectory using Blackman windows. The time points, T_1 and T_2, are the detected voice activity onset and offset, respectively

To make the utterance-initial and -final phones sound natural, the lung pressure usually starts earlier than the utterance VAD onset while it ends later than the utterance VAD offset. Improper alignments between detected VAD onset/offsets and lung pressure boundaries will either make the utterance-initial or -final phones sound very weak or disappear, or introduce extra unexpected sounds due to the combination of valid lung pressure and open vocal tract. In this study, the proper time interval between VAD and lung pressure boundaries for each side was determined by searching the optimal number of shifted frames. This was done upon 50 randomly selected words from the PBACU corpus. Their initialized gestural scores were converted to articulatory trajectories. WebTRC-VAD was applied to the corresponding *synthetic* speech. Based on the detected VAD boundaries, the new lung pressure trajectories were created with a particular frame shit size, which then replaced the original ones. The resulting articulatory trajectories were fed into VTL to produce new synthetic speech. After that, WebTRC-VAD was applied to the new synthetic speech, the results of which were compared with those of original synthetic speech. The optimal number of VTL shifted frames was searched from the array $\{1, 2, \ldots, 20\}$ (each VTL frame interval corresponds to 2.5 ms) with the aim of minimizing the average deviation of detected boundaries of between synthetic speech created with prototype gestural scores and that created with manipulated gestural scores whose lung pressure trajectories were generated based on Blackman windows. It should be mentioned that, searching optimal value for boundary shift can be done upon any other utterances other than the 50 *synthetic* speech created based on the PBACU utterances. This would not influence performance measurement, since the proposal would be evaluated using *original natural* utterances. The final optimal boundary shift is shown in Figure 5.11. To be more precise, the lung pressure should start 8 frames (20 ms) earlier than the detected VAD onset while it should end 6 frames (15 ms) later than the detected VAD offset.

5.3.3 Pitch Target Estimation for f_0 Trajectory

Like lung pressure trajectory, the f_0 trajectory was also not modeled in the neural network regression models but separately estimated from acoustic signals. Again, this was done using TargetOtimizer-2.0 (Krug et al., 2021). The raw f_0 samples were extracted by Praat (Boersma and Weenink, 2019) and stored in the PitchTier files. The number of pitch targets was determined by counting the number of vowels from the phonological transcription of the given utterance. In this sense, the transcription was assumed to be known. However, this can be avoided by performing ASR like that used in gestural score initialization in Section 4.1. The number of pitch targets for a given utterance can also be roughly specified based on utterance duration, e.g., by dividing the total duration by an average German phoneme duration calculated from other corpora. For simplicity, this study directly counted the number of vowels from annotation files and used them as the number of targets for pitch target estimation by TargetOptimizer-2.0. Except for this, the annotation files were never used in other steps. The average RSME and correlation coefficient (ρ) between the original and reproduced f_0 by TargetOptimizer-2.0 were 0.575 st and 0.966 for the natural utterances of the Kiel corpus, respectively.

5.3.4 Training Neural Networks

Two kinds of neural networks (LSTM and convolutional LSTM) and two kinds of features (MFCC and spectrogram) were compared in this study. For each utterance of the VTL-Kiel corpus, MFCC and spectrogram were extracted from the acoustic signal with a frame size of 20 ms and a frame rate of 2.5 ms. The MFCCs consisted of the original 13-dim features and their first order derivatives. The MFCC features were Z-score normalized individually for each utterance. The spectrograms used in this study were the log power spectrograms extracted by the 1024-point discrete Fourier transform (DFT), thus having 513 frequency bins. When log power spectrograms were used as input features, no explicit normalization was applied. It was expected that the convolutional layers together with batch normalization layers could deal with feature normalization. However, for the encoder-decoder model that introduced acoustic loss, the log power spectrograms were also used as the targets of the forward model. The magnitude of log power spectrograms was much larger than the output of forward model. It was very hard for the neural network models to converge if acoustic loss was directly calculated between them. Therefore, the backward model used the *raw* log power spectrograms as inputs while the forward model used the *normalized* log power spectrograms as targets. Here, the Z-score normalization was individually applied to each spectrogram.

Although VTL had 30 articulatory parameters, ACS was simplified by fixing some parameters or separately estimating some parameters from acoustic signals. In this study, the lung pressure trajectory was determined by VAD together with a rule-based method and Blackman windows. The f_0 trajectory was estimated by TargetOptimizer-2.0. The tongue root parameters ("TRX" and "TRY") were excluded from modeling because they were automatically determined by VTL when all other parameters were given. The upper vocal fold displacement was always set to be the value of the lower counterpart plus a fixed value (0.01). There were also five fixed vocal fold parameters that used the VTL default values: phase-lag, double-pulsing, pulse-skewness, flutter, and aspiration-strength. The remaining 20 articulatory parameters were modeled by the neural network regression models. Each VTL articulatory parameter, y_{vtl}, was scaled to the range $[-1, 1]$ with the following equation:

$$y_{vtl\_scaled} = 2\frac{y_{vtl} - y_{min}}{y_{max} - y_{min}} - 1 \tag{5.12}$$

where, y_{min} and y_{max} are constants (see Table 3.1). They are the minimal and maximal values of the VTL parameter that the variable y_{vtl} corresponds to. The neural network models used the scaled articulatory parameters as targets for training. At the testing stage, each estimated value y_{esti} was scaled back to its corresponding original VTL parameter range with Equation 5.13. The articulatory trajectories composed of estimated VTL articulatory parameter y_{esti_vtl} were fed into VTL to synthesize speech.

$$y_{esti\_vtl} = \frac{1}{2}(y_{max} - y_{min})(y_{esti} + 1) + y_{min} \tag{5.13}$$

Since the convolutional layers usually accepts spectrogram as inputs in other studies, this study mainly trained two kinds of ACS systems: one using LSTM with MFCC as input features and another using convolutional LSTM with log power spectrogram as input features. Some complementary systems (e.g. the one using LSTM with spectrogram as input features) were also trained to support the findings of this study. All neural network models were implemented with PyTorch (Paszke et al., 2019). The models were trained using the mini-batch gradient descent method (Dekel et al., 2012). Within a mini-batch, zeros were padded to short utterances so that all samples had the same length. It should be noted that those padded segments were excluded using the masking technique during loss calculation. The weights of neural network models were updated by the Adam optimizer (Kingma and Ba, 2017) with default values for all parameters in PyTorch except for the learning rate. Moreover, to prevent the neural networks from getting over-fitted to the training data, the *dropout* technique (Hinton et al., 2012) was applied to the output of hidden layers of LSTM (except for the last layer). The probability of an element to be zeroed used the default value (0.5) in PyTorch. Finally, the linear layer was attached to the last layer of LSTM, which transformed the LSTM outputs to the final outputs, i.e., the estimated articulatory trajectories.

Table 5.3 lists the feature dimensions and the candidate hyper-parameters that were searched for two kinds of neural network models. The optimal hyper-parameters indicated by the bolded values were determined by a grid-search strategy with regard to the MSE loss. There were many combinations of hyper-parameters, which resulted in comparable MSE losses. The same set of optimal configurations for all systems were determined in terms of running time, performance, and comparability between different systems.

TABLE 5.3: Feature dimensions and candidate hyper-parameters of two kinds of neural networks. The bold text indicates the optimal values after grid search, which were used as the configuration for subsequent neural network models.

	LSTM + MFCC	**ConvLSTM + Spectrogram**
Feature dimension	26	513
Convolution layers	–	No. of layers: {4, 5, **6**}
		No. of kernels per layer: {16, **32**, 64}
LSTM layers	Direction of LSTM: {Unidirectional, **Bidirectional**}	
	No. of layers: {2, **3**, 4}	
	Nodes per layer: {100, **200**, 300, 400}	
Learning-rate	{0.0005, **0.001**}	

Using the same set of hyper-parameters, a series of ACS systems were trained in this study in order to ascertain which features and neural network architectures were more useful, how to use the training data, what strategies were useful to improve the performance, and so on. Each system was trained for a maximum of 50 epochs with the early stopping strategy by monitoring the validation loss with a tolerance of five epochs. Figure 5.12 shows an example of using the early stopping strategy during training a neural network model. Since the MSE loss of validation subset achieved a minimal value after 22 epochs and never decreased again in the following five epochs, the training stopped after 27 epochs. The model saved after 22 epochs was used as the final model in this example.

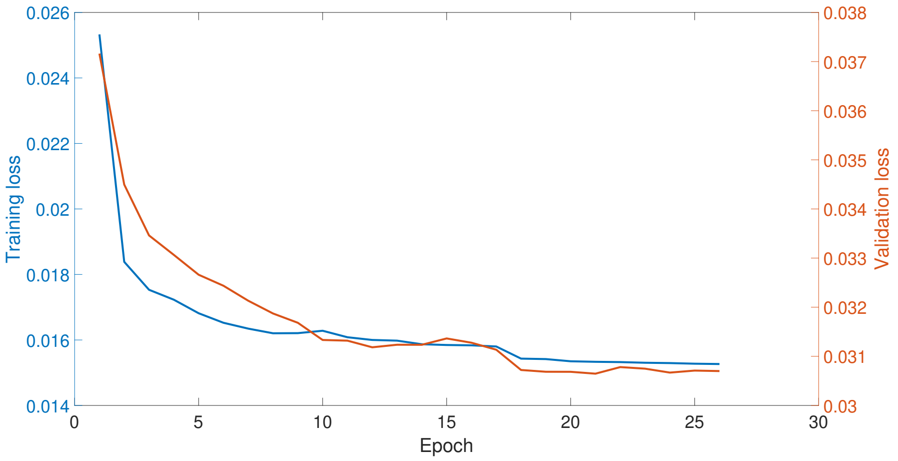


FIGURE 5.12: Example of using early stopping strategy with a tolerance of five epochs during training a neural network model. The training was stopped after 27 epochs, because the validation loss did not get smaller in the following five epochs since it obtained the smallest value after 22 epochs.

5.4 Results and Analysis

Different ACS systems were built and compared in terms of neural network architectures, acoustic features, normalization methods, regularization strategies and other aspects. The experiments always started from the basic configurations and were incrementally extended to other aspects. If some attempts did not work, they would be discarded in subsequent experiments.

The first ACS system was trained using LSTM neural networks with MFCC as inputs. The MFCC features were extracted from VTL synthetic speech of the VTL-Kiel corpus whereas the final goal of ACS was to apply the trained model to natural speech produced by human speakers. The mismatch of acoustic characteristics between synthetic and natural speech undoubtedly affected the testing performance in the speaker-independent scenario. Therefore, the MFCC features extracted from the original natural speech of the Kiel corpus were also employed to examine whether they were useful to increase the system robustness against unknown speakers. Since the Kiel corpus did not contain *real* articulatory data, the "prototype" articulatory trajectories of the VTL-Kiel corpus were used as the corresponding targets for the MFCC features extracted from natural speech of the Kiel corpus. One can take the Figure 5.9 as an example to understand such an articulatory-acoustic training sample. The MFCC features were extracted from the natural utterance "dlmsi025" while the articulatory trajectories of Sample-1 ("dlmsi025_modal_tau1_VTLength1") were used as corresponding targets/labels which were generated from the "prototype" gestural score. Accordingly, 1998 new samples (VTL articulatory trajectories of prototype gestural scores and acoustic features of human speech of the Kiel corpus) were created. Such samples were referred to as the "VTL-Human" samples with a total length of 1.28 hours. The samples whose articulatory trajectories and acoustic signals were created by VTL were referred to as "VTL-VTL" samples with a total length of 15.39 hours. Their combination was referred to as the "Mixed" samples with a total length of 16.67 hours.

The second ACS system was built using LSTM neural networks with the Mixed samples. Although the human speakers' MFCC features and VTL articulatory trajectories did not perfectly match, they may be beneficial to training robust ACS models. These samples were also divided into three subsets (80% for training, 10% for validation, and 10% for testing) according to the data partition used in Section 5.3.1 (see Table 5.2 for a summary and Appendix C for details). After the training of these two ACS systems, their performances were evaluated on the test subset in terms of correlation coefficients between the estimated and the target articulatory trajectories. Table 5.4 lists the average correlation coefficients for each VTL parameter across all testing utterances of the VTL-Kiel corpus. Even though the second ACS system was trained with the Mixed samples, the average correlation coefficients (i.e., the third column in this table) were calculated only for the testing utterances of the VTL-Kiel corpus, i.e., the same test subset used for the first ACS system. As can be seen from Table 5.4, the average correlation coefficients are very close to each other not only between different articulatory parameters but also for these two systems. It should be noted that this table only shows the results for the 20 VTL parameters that were

TABLE 5.4: Average correlation coefficients between the original and estimated VTL parameters across all testing utterances of the VTL-Kiel corpus for two LSTM-MFCC based ACS systems trained with different training samples.

	System trained with VTL-VTL samples	**System trained with Mixed samples**
Horizontal Hyoid position (HX)	0.983	0.98
Vertical Hyoid positon (HY)	0.985	0.984
Horizontal jaw position (JX)	0.985	0.984
Jaw angle degree (JA)	0.986	0.985
Lip protrusion (LP)	0.976	0.976
Lip distance (LD)	0.978	0.982
Velum shape (VS)	0.982	0.982
Velic opening (VO)	0.991	0.99
Tongue body X (TCX)	0.99	0.989
Tongue body Y (TCY)	0.986	0.984
Tongue tip X (TTX)	0.985	0.983
Tongue tip Y (TTY)	0.987	0.987
Tongue blade X (TBX)	0.979	0.982
Tongue blade Y (TBY)	0.983	0.987
Tongue side elevation 1 (TS1)	0.971	0.973
Tongue side elevation 2 (TS2)	0.986	0.985
Tongue side elevation 3 (TS3)	0.982	0.983
Lower displacement	0.985	0.988
Chink area	0.965	0.965
Relative amplitude	0.99	0.989

modeled by neural networks. As mentioned before, f_0 and *lung pressure* were separately estimated from acoustic signals and not modeled by neural networks. All of these 20 VTL parameters achieved very high correlation coefficients. Among them, the average correlation coefficient of the "chink_area" parameter that appeared to be more difficult to estimate than other parameters still had a high value (0.965).

Table 5.5 gives a summary of the average correlation coefficient and cosine distance between the original and estimated VTL articulatory parameters across all dimensions and all testing utterances of the VTL-Kiel corpus for the two LSTM-MFCC based ACS systems trained with different training samples. These two systems achieved very similar results in terms of articulatory trajectory similarity. The system trained with Mixed samples performed slightly better than the one trained with VTL-VTL samples alone. The performance of the ACS systems based on LSTM neural networks was comparable with that of the genetic algorithm based ACS approach proposed in Chapter 4 (c.f. Table 4.7). It should be noted that the results in Table 4.7 was computed on a set of 16 utterances of isolated words while the results in Table 5.5 were computed on a set of 198 utterances of sentences.

Figure 5.13 shows the articulatory trajectories for an utterance of the VTL-Kiel test subset. The blue solid curves indicate the original articulatory trajectories while the

TABLE 5.5: Average correlation coefficients and cosine distance between the estimated and target articulatory parameters across all dimensions and all testing utterances of the VTL-Kiel corpus for two LSTM-MFCC based ACS systems trained with different training samples.

	System trained with VTL-VTL samples	**System trained with Mixed samples**
Average correlation coefficients	0.98315	0.98329
Average cosine distance	4.998×10^{-4}	4.821×10^{-4}

dashed red curves indicate the estimated ones. This figure also illustrates the good consistence between the original and estimated articulatory trajectories.

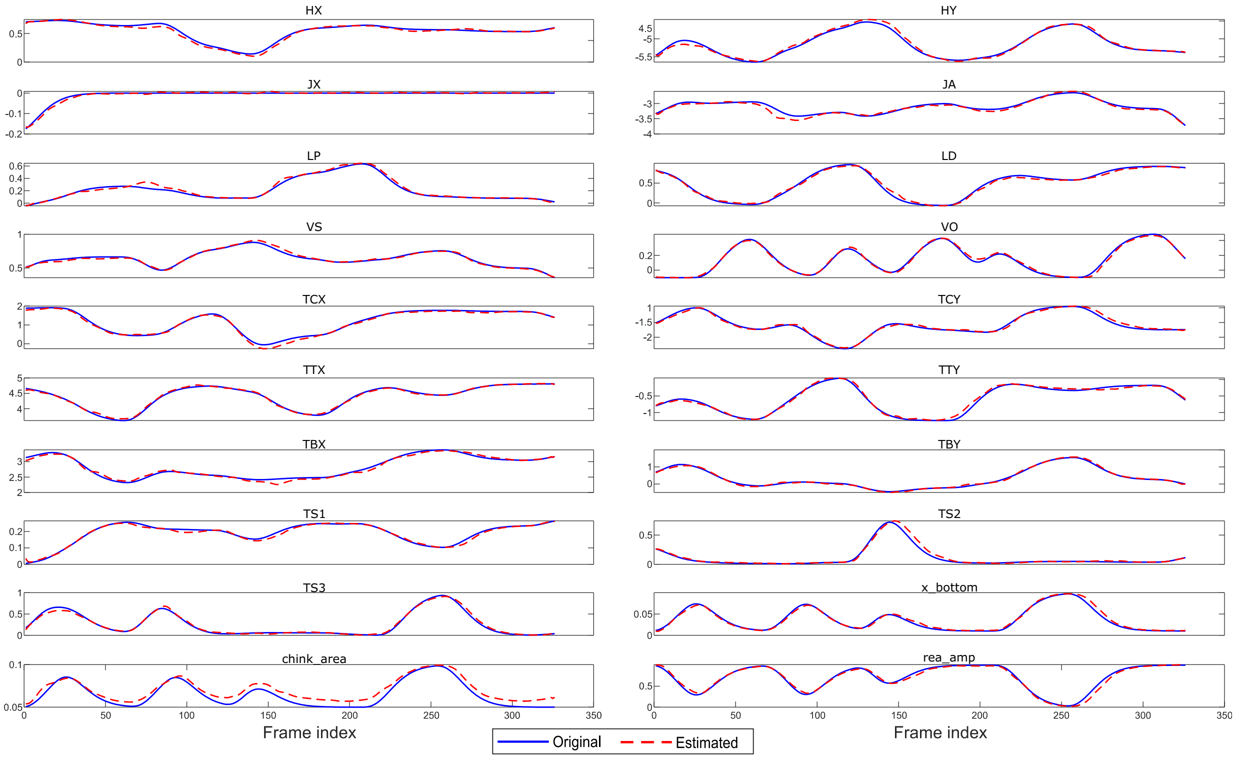


FIGURE 5.13: Comparison of articulatory parameter trajectories for the utterance "dlmsi025_modal_tau1_VTLength1" of VTL-Kiel test subset. The blue solid curves indicate the original articulatory trajectories while the dashed red curves indicate the estimated ones.

The above results suggested that the ACS systems based on LSTM neural networks achieved very high correlation coefficients and very small cosine distance between the estimated and target articulatory trajectories. However, both the training and testing utterances (except for the "VTL-Human" samples) were produced by the model speaker of VTL. That is to say, the above results can only reflect the speaker-dependent performance of the trained models. Therefore, it was more expected to evaluate the trained models in the speaker-independent scenario, for example, by using the human utterances.

Since there were no real articulatory trajectories together with the human speech used in this study, evaluating the trained models on natural utterances could not be done in the articulatory domain. Hence, the performance of the trained ACS systems on human speech was evaluated in the perceptual domain in terms of the intelligibility of reproduced utterances. Like the method used in Chapter 4, the Google ASR accuracy was adopted as the metric to measure the intelligibility of reproduced utterances. The 320 natural utterances of the PBACU corpus were used to compare the performance among different ACS systems. There were three reasons for using utterances of the PBACU corpus instead of the testing utterances of the Kiel corpus. First, conducting ASR on reproduced utterances of the PBACU corpus was less expensive because the PBACU corpus had fewer and shorter utterances than the test subset of the Kiel corpus. Second, as mentioned in the step of data partition, the utterances of the Kiel corpus were divided into three subsets according to the text. As a result, there was speaker-overlap between the training and test subsets although there was no text-overlap. Nevertheless, both the text and the speakers of the PBACU corpus could be considered unknown. Third, the utterances of the PBACU corpus were isolated words while the utterances of the Kiel corpus were sentences. Therefore, recognizing isolated words instead of sentences could disentangle the speech intelligibility measurement from the influence of language model of the speech recognizer, thus giving a relatively fair performance measurement. Therefore, the Google ASR accuracy of the reproduced utterances of the PBACU corpus was used in the subsequent experiments to compare different ACS systems. Of course, to make the study more complete, the performance on the testing utterances of the Kiel corpus was also evaluated (only for the best ACS systems) at the end of this section.

TABLE 5.6: Google ASR accuracies (in %) of reproduced PBACU utterances by LSTM-MFCC ACS systems trained with different training samples.

Training samples	**Total duration (in hours)**	**Accuracy**
VTL-VTL	15.39	28.13
VTL-Human	1.28	43.75
Mixed	16.67	49.06

To examine the benefit of including VTL-Human samples to training neural networks, another ACS system was trained with the VTL-Human samples alone. Table 5.6 lists the Google ASR accuracies of the reproduced PBACU utterances for LSTM-MFCC based ACS systems trained with different training samples together with respective total duration of them. The ACS system trained with the Mixed samples improved the accuracy by approximately 21% compared to the one trained with the VTL-VTL samples, which resulted from the inclusion of the additional VTL-Human samples. Furthermore, the ACS system trained using the VTL-Human samples alone obtained an accuracy of 43.75%, which was still much higher than that of the ACS system trained with VTL-VTL samples. That is to say, with fewer utterances, the former achieved a much higher ASR accuracy than the latter (1.28 hours vs. 15.39 hours). This further suggested the effectiveness of including acoustic features of natural utterances into training the ACS systems.

Although the ACS system trained with the Mixed samples performed better than the other two systems, its performance in terms of ASR accuracy was still lower than that of the genetic algorithm based ACS method proposed in Chapter 4. The following experiments aimed to enhance the ACS performance using new neural network architectures or regularization methods. Since the VTL-Human utterances were more useful to train the neural networks for ACS than the VTL-VTL samples, hereafter, other new ACS systems were only trained with the two types of samples: VTL-Human and Mixed.

Another two systems (one with the VTL-Human samples and another with the Mixed samples) were built using the convolutional LSTM neural networks with log power spectrograms as inputs. They used the same configurations for the LSTM and fully-connected layers like those used in the previous ACS systems. The main difference was that six convolutional layers were inserted between acoustic features and the LSTM layers. Unlike the LSTM-MFCC based ACS systems that used the handcrafted acoustic features, the ConvLSTM-Spectrogram based ACS systems automatically learned features from spectrograms with convolutional layers. The ACS systems trained only with articulatory loss (i.e., without other losses) were referred to as the baseline systems. As mentioned in Section 5.2, to make the estimated articulatory trajectories smoother, some studies usually applied low-pass filters to the results in the post-processing stage. In this study, for the sake of comparing the proposed regularization methods with the low-pass filter based method, the articulatory trajectories estimated by the baseline systems were smoothed by performing zero-phase digital filtering (in both the forward and reverse directions) with a 12-order Butterworth filter before being fed into VTL to synthesize speech.

After that, another four ACS systems were trained with the smoothness loss regularization and they differed in the model architecture (LSTM-MFCC or ConvLSTM-Spectrogram) and training samples (VTL-Human or Mixed). The hyper-parameter α for the weight of smoothness loss in Equation 5.9 was separately specified for different ACS systems since the acoustic loss and smoothness loss might have different magnitudes resulting from the different acoustic features and training samples used in these ACS systems. The values for the weight α were set to 10 and 100 for the ACS systems trained with VTL-Human samples and Mixed samples, respectively. That is to say, the latter type used a larger weight. This is because the Mixed samples contained the VTL-VTL samples (the acoustic features and articulatory trajectories are matched) and their articulatory trajectories were easy to estimate. Hence, their smoothness loss had a smaller magnitude than that of the ACS systems trained with only the "mismatched" VTL-Human samples. Similarly, another four ACS systems were trained with the acoustic loss regularization. The hyper-parameter β for the weight of acoustic loss in Equation 5.11 was also separately specified for different ACS systems. The value for the ConvLSTM-Spectrogram based ACS system trained with Mixed samples was set to 0.001 while it was set to 0.05 for the other three ACS systems.

Table 5.7 summarizes the Google ASR accuracies of the 16 sets of reproduced utterances of the PBACU corpus. Each set contained 320 utterances reproduced by a specific ACS system. For each ACS system, a capital letter before its accuracy is assigned for the sake of convenient reference. In the "Baseline" row, System-A

TABLE 5.7: Google ASR accuracies (in %) of reproduced utterances of the PBACU corpus by different ACS systems.

	VTL-Human		**Mixed**	
	LSTM-MFCC	**ConvLSTM-Spec**	**LSTM-MFCC**	**ConvLSTM-Spec**
Baseline	(A) 43.75	(B) 52.81	(C) 49.06	(D) 59.06
Baseline (low-pass filter smoothed)	(A) 42.5	(B) 54.38	(C) 50.94	(D) 58.75
Smoothness loss regularized	(E) 44.69	(F) 54.38	(G) 51.88	(H) 60.31
Acoustic loss regularized	(I) 46.88	(J) 55.31	(K) 63.13	**(L) 64.69**

and System-C are the above-mentioned LSTM-MFCC systems trained with VTL-Human and Mixed samples (their results are already reported in Table 5.6) while System-B and System-D were convolutional LSTM based systems trained with log power spectrograms as inputs. This new model architecture increased the accuracy by approximately 10%, suggesting the effectiveness of learning distinctive features from spectrograms by convolutional layers. The effect on speech intelligibility of smoothing estimated articulatory trajectories by the low-pass filter was not obvious since the ASR accuracies of two cases (when applied to System-B and System-C) increased while the ASR accuracies of another two cases (when applied to System-A and System-D) decreased a little. However, the regularization methods proposed in this chapter consistently outperformed the systems without regularization as well as the post-processing method by the low-pass filter.

Moreover, the regularization of acoustic loss performed better than that of smoothness loss. This may be explained as follows. For each of the acoustic loss regularized systems ("I", "J", "K", and "L"), there were a forward model and a backward model. They were separately trained in advance using the same articulatory-acoustic samples but with an opposite way of determining inputs and outputs. In this pre-training stage, the forward model perhaps had learned the mapping of articulatory trajectories to acoustic representations, which was similar to what VTL did. After that, they were concatenated and further jointly trained using the same samples once again. In this stage, the concatenated backward-forward model was trained to minimize the weighted sum of the articulatory loss and the acoustic loss. Only when the estimated articulatory trajectories were more *preferred* by the forward model, the acoustic loss tended to decrease. In turn, this would make the estimated articulatory trajectories be more preferred by VTL, thus producing more natural and intelligible

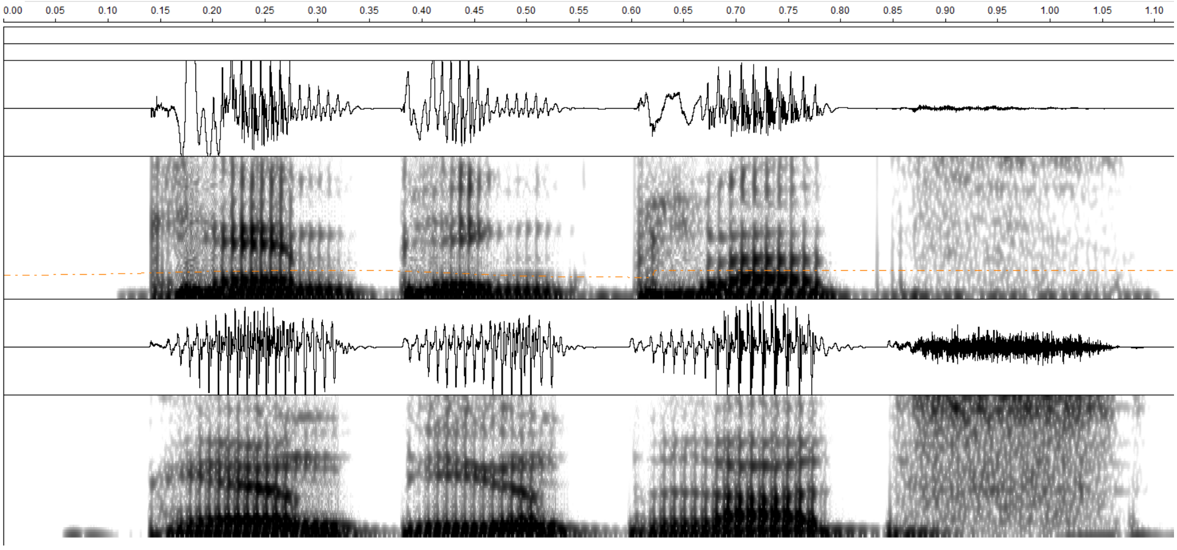


FIGURE 5.14: The waveforms and spectrograms for the word "Campingplatz". The upper part is the original utterance produced by the speaker SPK-1 in the PBACU corpus while the lower part is the reproduced utterance by ConvLSTM-Spectrogram ACS system.

speech. For the PBACU corpus, the estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive repository[2, 3]. Figure 5.14 gives an example of waveforms and spectrograms for the word "Campingplatz". The upper part is the original utterance produced by the speaker SPK-1 in the PBACU corpus while the lower part is the reproduced utterance by the best ACS system ("L").

Next, it was worth examining what the convolutional and bath normalization layers learned. Figure 5.15 shows the log power spectrogram and learned features by CNN of the System-D with the utterance "besonders" produced by the speaker SPK-2 as an example. The obvious difference was their magnitudes. The values of log power spectrograms usually ranged from -300 dB to -20 dB. It was very hard for models to reach convergence if such values were directly used as inputs of neural networks. By inserting CNN (convolutional and batch normalization layers) before LSTM layers, the log power spectrogram were transformed to new features with a much smaller range. The CNN-learned features almost fell into the range from -1 to 1, except for the very beginning part of the waveform which was a silence segment. The horizontal strips of CNN-learned features appeared to be straighter than those of log power spectrogram.

One may think that the effect of CNN used in this study (i.e., convolutional and batch normalization layers) was just to normalize the input features to a narrow

[2] The estimated articulatory trajectories and reproduced speech for the PBACU corpus: https://drive.google.com/drive/folders/1oGub00-bdNnNGFl-N-WnVoRn3Jc9ADka?usp=sharing

[3] A few animation examples: https://drive.google.com/drive/folders/1coFeI2u-NutYYv5SlS1972Lgjj2YP6_3?usp=sharing

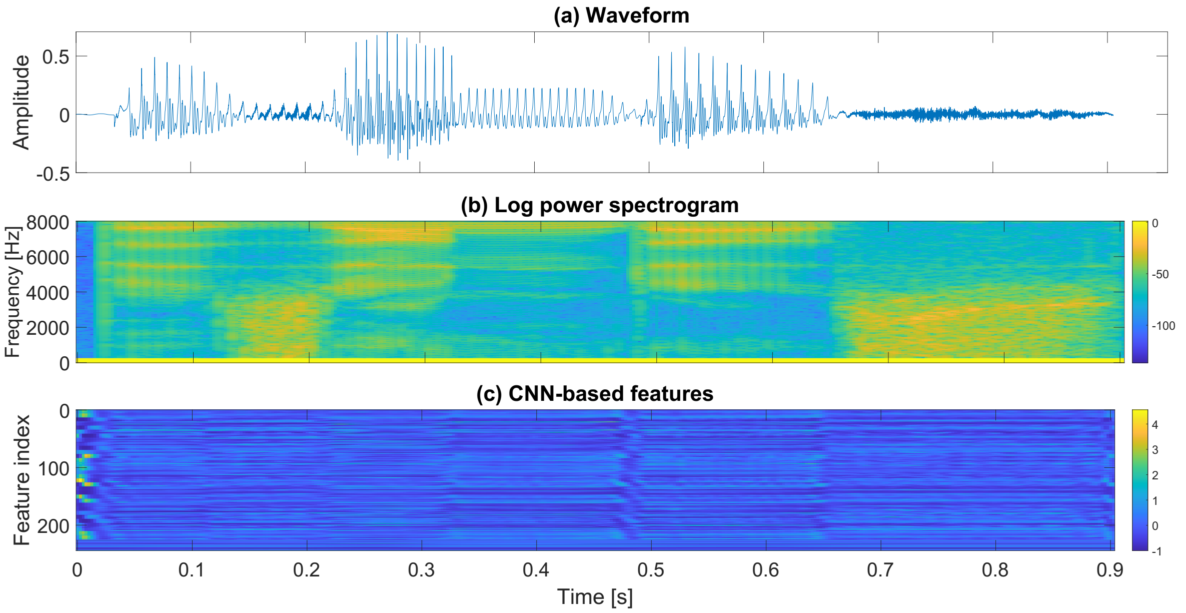


FIGURE 5.15: Waveform (a), log power spectrogram (b) and CNN-based features (c) of the utterance "besonders" produced by the speaker SPK-2.

range around 0. Therefore, four additional ACS systems were built to examine whether CNN could be replaced by the normalization techniques. The idea was to use the combination of LSTM and *normalizd spectrogram*, an example of which is shown in Figure 5.16. In the subplot (a), Z-score normalization was globally applied to the whole spectrogram. In the subplot (b), Z-score normalization was inidvidually applied to each frequency bin. In the subplot (c), Z-score normalization was inidvidually applied to each frame. Another one did not use any normalization technique.

To save the training time, these four systems were trained with the VTL-Human features and no regularization was used. Therefore, the accuracy of System-B (52.81%) should be used as the reference for comparison. Table 5.8 lists the Google ASR accuracies of the reproduced PBACU utterances for LSTM-Spectrogram systems with different normalization methods. As can be seen from this table, the system without normalizing the log power spectrograms hardly reproduced the natural utterances. Applying Z-score normalization to the whole spectrograms performed better than that performed individually along time or frequency axis. However, all of them were inferior to that of System-B where the features were automatically learned and implicitly normalized by the convolutional and batch-normalization layers.

The ACS systems trained with the Mixed samples were also tested using the sentence-level utterances. They were used to reproduce the utterances of the VTL-Kiel test subset (i.e., the speaker-dependent scenario). The reproduced utterances sounded very similar to the original ones. The resulting estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive

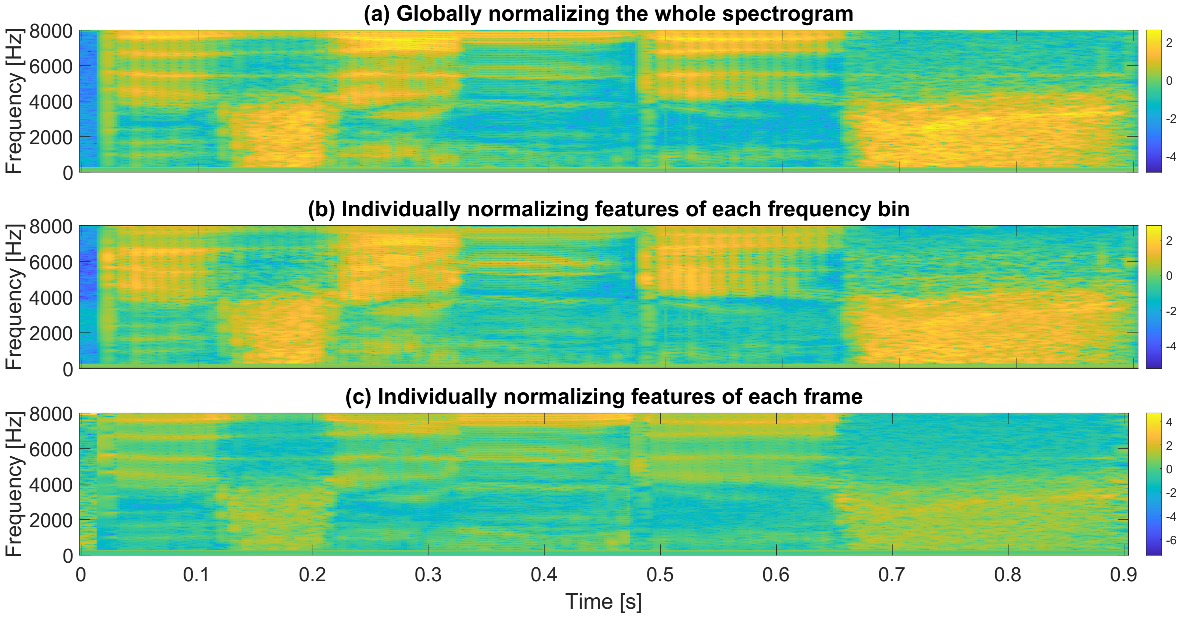


FIGURE 5.16: Z-score normalized log power spectrogram of the utterance "besonders" produced by the speaker SPK-2 in the PBACU corpus: (a) globally for the whole utterance; (b) individually for each frequency bin; (c) individually for each frame.

TABLE 5.8: Google ASR accuracies (in %) of reproduced PBACU utterances for LSTM-Spectrogram systems with different normalization methods.

	Accuracy
Global normalization for the whole spectrogram	28.13
Individual normalization for each frequency bin	22.5
Individual normalization for each frame	14.06
Without normalization	3.13

repository[4]. Next, they were also used to reproduce the utterances of the Kiel test subset (i.e., the speaker-independent scenario). The estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive repository[5, 6]. Figure 5.17 gives an example of waveforms and spectrograms for the utterance "k03be019" of the Kiel corpus. The upper part is the original utterance while the lower part is the reproduced utterance by System-L.

Table 5.9 shows the Google ASR accuracies of reproduced utterances of the Kiel test

[4] The estimated articulatory trajectories and reproduced speech for VTL-Kiel test subset: https://drive.google.com/drive/folders/1iHfAk7sGyNc26b616G21dZ_WAi2iJHcP?usp=sharing

[5] The estimated articulatory trajectories and reproduced speech for the Kiel test subset: https://drive.google.com/drive/folders/14bBAJEeo9tYYp2p0mLrScaFh2WpJsF_Y?usp=sharing

[6] A few animation examples: https://drive.google.com/drive/folders/1DXehX8iLWL_vc7YwoDvsr72LSUsomTPv?usp=sharing

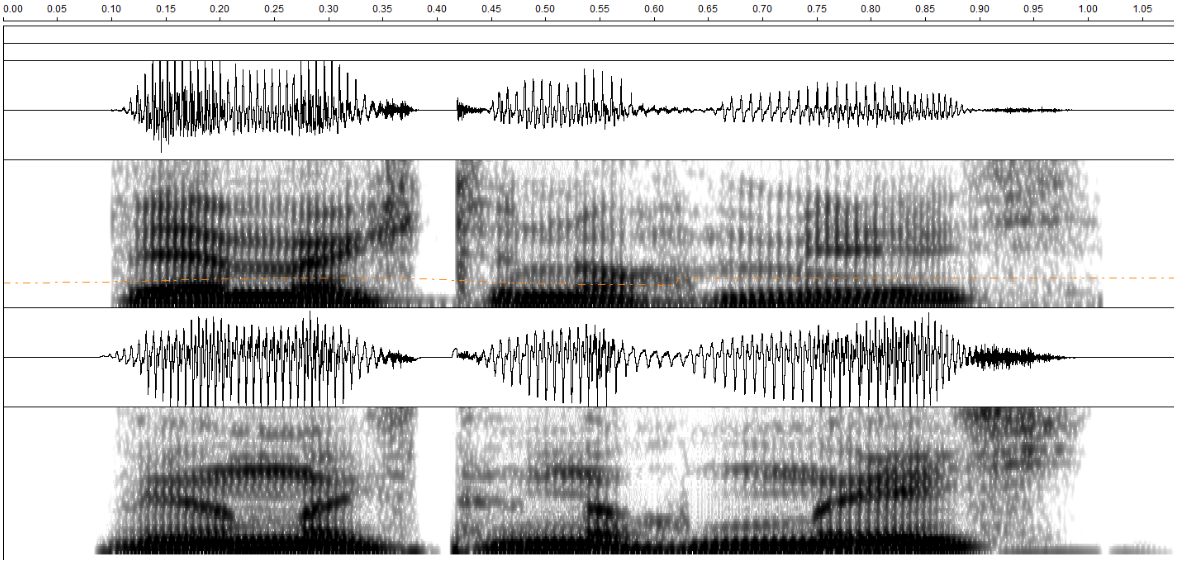


FIGURE 5.17: The waveforms and spectrograms for the utterance "k03be019" of the Kiel corpus. The upper part is the original utterance while the lower part is the reproduced utterance by System-L.

TABLE 5.9: Google ASR accuracies (in %) of reproduced utterances of the Kiel test subset by different ACS systems.

	LSTM-MFCC	**ConvLSTM-Spec**
Reproduced without Regularization	(C) 50.32	(D) 53.50
Reproduced with smoothness loss regularization	(G) 56.73	(H) 57.29
Reproduced with acoustic regularization	(K) 73.88	(L) 58.11
Original natural utterances	92.78	
Synthetic utterances of prototype gestural scores	68.52	

subset. The ASR accuracies of original natural utterances and the synthetic utterances created by the prototype gestural scores are also listed. There is no doubt that the natural utterances produced by human speakers had the highest intelligibility. The utterances synthesized with prototype gestural scores also had very high intelligibility since their gestural scores were created using the VTL-API rule-based method with manual segmentation as reference. The systems with regularization outperformed those without regularization. On the same conditions, the ConvLSTM-Spec based systems performed better than the LSTM-MFCC based systems except for the System-L. For ACS of the Kiel utterances, the LSTM-MFCC based system with acoustic loss regularization achieved the best performance (73.88% for System-K), the accuracy of which even exceeded that of synthetic utterances of prototype gestural scores. The System-L did not work well probably due to the fact that the inputs of the backward models were log power spectrogram whereas the targets of the forward model were normalized log power spectrogram. This might violate the idea of the encoder-decoder architecture, resulting in a degraded performance. For the test subset of the Kiel corpus, the estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive repository[7], [8].

5.5 Evaluation by Perception Experiment

As mentioned above, ASR by machines was convenient and could be used as a complementary manner to human assessment. The subjective assessment of the author suggested that some reproduced utterances sounded intelligible whereas they were not correctly recognized by the Google speech recognizer. Likewise, some reproduced utterances sounded unnatural and unintelligible, whereas the Google speech recognizer correctly recognized them. Hence, to examine the reliability of the Google speech recognizer, the reproduced utterances by ACS systems were further recognized by the IBM speech recognizer. Table 5.10 shows ASR accuracies of the two speech recognizers on the reproduced PBACU utterances by Systems "K" and "L" trained. The last column "Together" indicates the percentage of the utterances that were correctly recognized by either of two speech recognizers or both, i.e., they were considered recognizable.

TABLE 5.10: Google and IBM ASR accuracies (in %) of reproduced PBACU utterances by different ACS systems.

	Google	**IBM**	**Together**
System-K	61.88	55	71.56
System-L	63.44	53.43	72.5

As can be seen from Table 5.10, the Google speech recognizer performed generally better than the IBM speech recognizer in this study. However, using them together always achieved the highest recognition accuracy. This reflected the difference in intelligibility measurement by different machine recognizers. Therefore, to investigate the intelligibility of reproduced utterances by human listeners and the reliability of evaluation by machines, it was necessary to conduct a perception experiment.

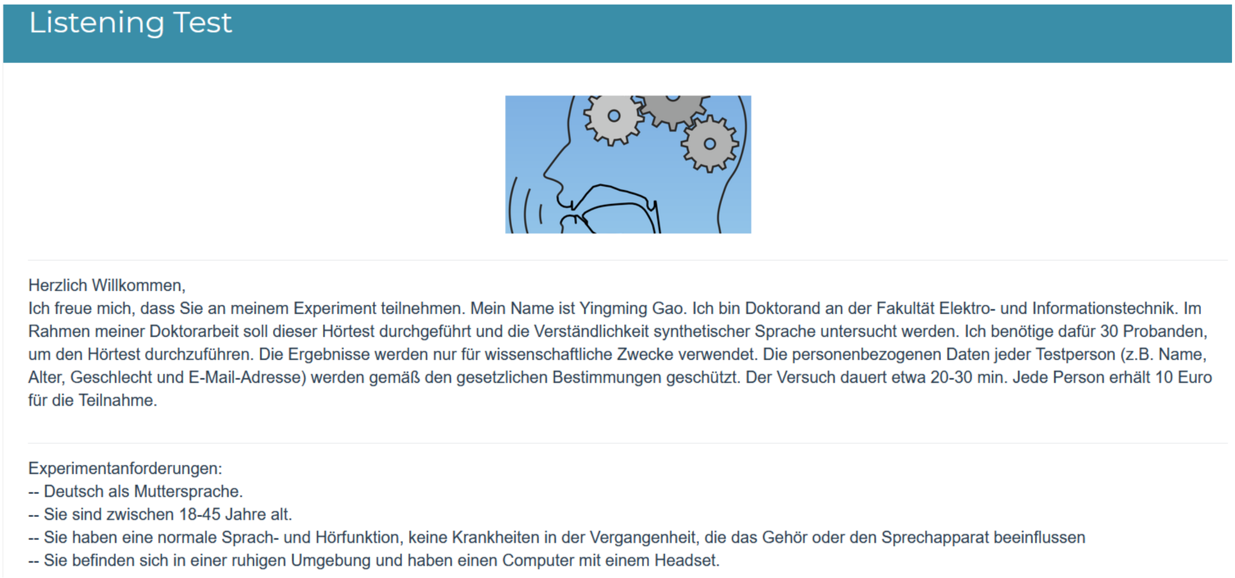
5.5.1 Experiment Design

To reduce people's physic contact during COVID-19 pandemic, the perception experiment was conducted in the form of online listening test. Alchemer[9], formerly named SurveyGizmo, was used as the tool to create the listening test. Alchemer is an online survey platform which provides users with various question types and flexible controls of survey design. With Alchemer, users can create an unlimited number of surveys, each of which has no limit of the number of questions and respondents. The tool can also automatically collect statistical information such as the starting and submit time, city and IP address of subjects, and detect unreliable answers based on

[7] The estimated articulatory trajectories and reproduced speech for the Kiel test subset utterances: https://drive.google.com/drive/folders/14bBAJEeo9tYYp2p0mLrScaFh2WpJsF_Y?usp=sharing

[8] A few animation examples: https://drive.google.com/drive/folders/1DXehX8iLWL_vc7YwoDvsr72LSUsomTPv?usp=sharing

[9] https://www.alchemer.com/



Herzlich Willkommen,
Ich freue mich, dass Sie an meinem Experiment teilnehmen. Mein Name ist Yingming Gao. Ich bin Doktorand an der Fakultät Elektro- und Informationstechnik. Im Rahmen meiner Doktorarbeit soll dieser Hörtest durchgeführt und die Verständlichkeit synthetischer Sprache untersucht werden. Ich benötige dafür 30 Probanden, um den Hörtest durchzuführen. Die Ergebnisse werden nur für wissenschaftliche Zwecke verwendet. Die personenbezogenen Daten jeder Testperson (z.B. Name, Alter, Geschlecht und E-Mail-Adresse) werden gemäß den gesetzlichen Bestimmungen geschützt. Der Versuch dauert etwa 20-30 min. Jede Person erhält 10 Euro für die Teilnahme.

Experimentanforderungen:
-- Deutsch als Muttersprache.
-- Sie sind zwischen 18-45 Jahre alt.
-- Sie haben eine normale Sprach- und Hörfunktion, keine Krankheiten in der Vergangenheit, die das Gehör oder den Sprechapparat beeinflussen
-- Sie befinden sich in einer ruhigen Umgebung und haben einen Computer mit einem Headset.

FIGURE 5.18: Welcome page of the perception experiment.

the response time. Another feature is to support uploading and playing of audio clips which especially meets the demand of the current study. Therefore, Alchemer was chosen as the tool to conduct the listening test.

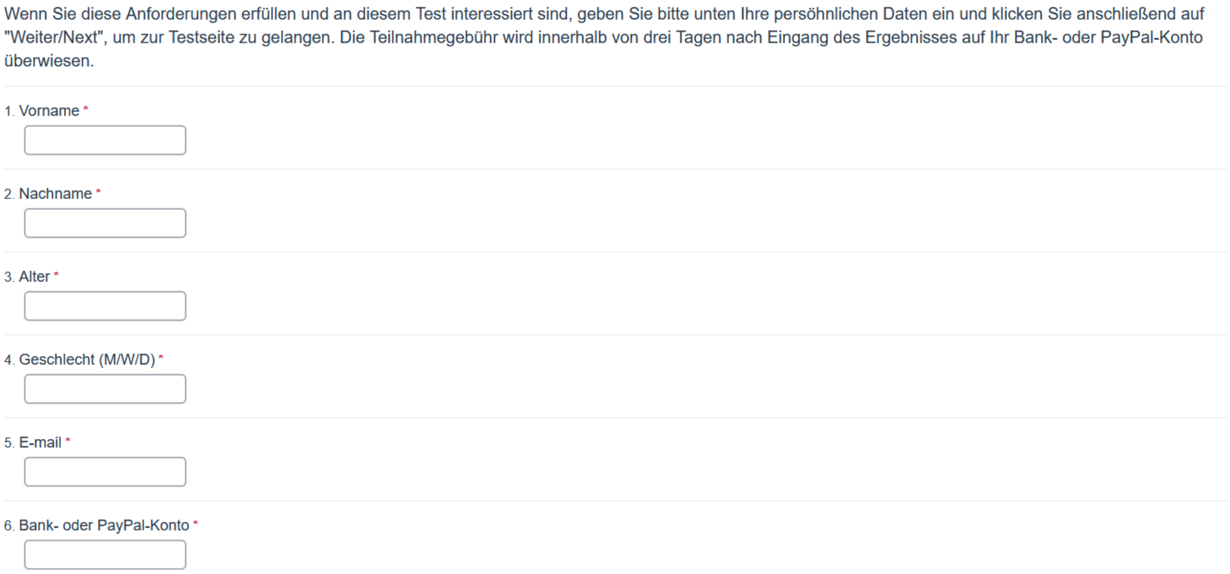
Figure 5.18 shows the welcome page of the listening test. It first introduced the basic information of the experimenter and the purpose of this listening test. Then four experiment conditions that subjects should meet were explained. Their translations are listed below:

- German is your mother tongue.
- Your age is between 18-40 years.
- You have normal speech and hearing functions with no history of any communication disorders.
- You have a quiet environment and a computer together with a headset.

When the subjects met these conditions according to their self-report, they could click the "Next" button to jump to next page where their basic information would be collected. As shown in Figure 5.19, the subjects needed to fill in their given names, surnames, age, gender, email address, and bank or PayPal account.

After filling in their basic information, they jumped to the real test part shown in Figure 5.20. Before listening to the stimuli, the subjects were reminded again that they should sit in front of computer and wearing a headset in a quiet room. They were also instructed that "There are 280 synthetic sounds in total. All of them are simple and common words. Your task is to play each of them and recognize which word it is." The translations of three additional instructions are listed below:

- Please write down only one word for each sound.
- Please do not write punctuation.



Wenn Sie diese Anforderungen erfüllen und an diesem Test interessiert sind, geben Sie bitte unten Ihre persöhnlichen Daten ein und klicken Sie anschließend auf "Weiter/Next", um zur Testseite zu gelangen. Die Teilnahmegebühr wird innerhalb von drei Tagen nach Eingang des Ergebnisses auf Ihr Bank- oder PayPal-Konto überwiesen.

1. Vorname *

2. Nachname *

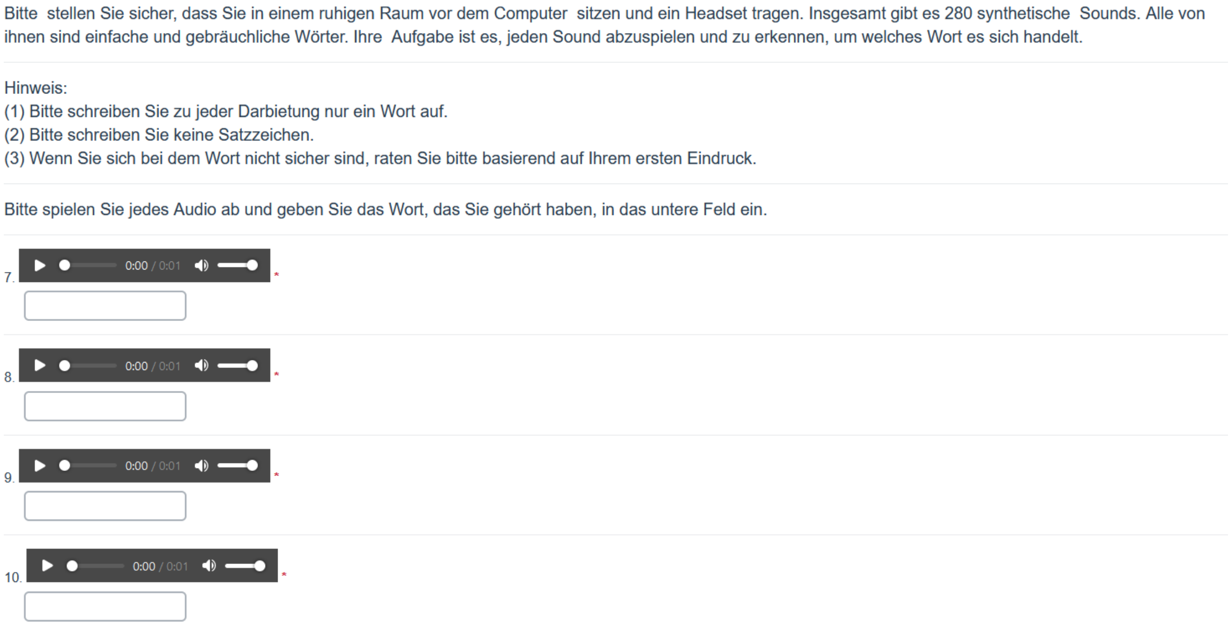
3. Alter *

4. Geschlecht (M/W/D) *

5. E-mail *

6. Bank- oder PayPal-Konto *

FIGURE 5.19: Information collecting page of the perception experiment.



Bitte stellen Sie sicher, dass Sie in einem ruhigen Raum vor dem Computer sitzen und ein Headset tragen. Insgesamt gibt es 280 synthetische Sounds. Alle von ihnen sind einfache und gebräuchliche Wörter. Ihre Aufgabe ist es, jeden Sound abzuspielen und zu erkennen, um welches Wort es sich handelt.

Hinweis:
(1) Bitte schreiben Sie zu jeder Darbietung nur ein Wort auf.
(2) Bitte schreiben Sie keine Satzzeichen.
(3) Wenn Sie sich bei dem Wort nicht sicher sind, raten Sie bitte basierend auf Ihrem ersten Eindruck.

Bitte spielen Sie jedes Audio ab und geben Sie das Wort, das Sie gehört haben, in das untere Feld ein.

7. *

8. *

9. *

10. *

FIGURE 5.20: Instruction and stimulus page of the perception experiment.

- If you are not sure about the word, please guess based on your first impression.

After that, they began the real recognition part. Their task was to play each audio and write the word they heard. Each pair of audio and answer field were numbered. The subjects could replay the stimuli as many times as they wanted. The volume of each audio was also adjustable.

The reproduced speech by the best system ("L") was used as stimuli for the listening experiment. The 232 utterances that were correctly recognized by either Google or IBM ASR systems or both were included in the listening test. To reduce the burden of human listeners, the quality of the other 88 utterances was first subjectively judged by the author. 48 of them having the chance of being recognized were also included in the listening test. 30 subjects (15 females and 15 males) were recruited to participate in the listening test. They were students of TU Dresden with average ages of 25.2 and 25.6 years for females and males, respectively. The whole listening test lasted for approximately 28 minutes for each subject. After the listening test, they were financially compensated for their participation.

5.5.2 Results

Although the subjects were instructed that they should write only one word for each audio, they sometimes typed two words or strings delimited by white space for specific stimuli. They sometimes split a word into a stem together with a prefix or a suffix. For example, some subjects wrote "zuwenig" as "zu wenig", "zuviel" as "zu viel", and "hinterher" as "hinter her". Their responses sometimes had obvious typos like writing "Rhythmus" as "Rhytmus" and "nirgendwo" and "niergendwo". Such cases accounted for 1.67% of all responses. They were manually corrected and regarded as correctly recognized by the listeners. However, there were some cases where the subjects missed or inserted letters compared to the correct answers. For example, they typed "vorhandene" for "vorhanden" and "gebilde" for "gebildet". In these cases, it was hard to judge whether this was due to typos or they really perceived the additional schwa sound for the reproduced "vorhanden" stimulus but no /t/ sound for the reproduced "gebildet", respectively. Because these responses were meaningful words in German, they were not regarded as typos but as real responses. Accordingly, the corresponding stimuli were considered to be utterances that were not correctly reproduced by the ACS system.

The recognition accuracy was used as the metric, for each word, which was calculated as the ratio of the number of correct responses to the number of all responses. During accuracy calculation, all responses and the reference words were converted to lower cases. The histogram of recognition accuracies for the 320 reproduced utterances of the PBACU corpus is shown in Figure 5.21. The group in blue indicate the 232 utterances that were correctly recognized by the ASR systems. Most of them had a recognition accuracy higher than 90% in terms of judgement by human listeners. The group in dark orange indicate the 88 utterances that were not correctly recognized by the ASR systems. It should be noted that the 48 reproduced utterances that had the chance of being recognized were included in the listening test. Even though they were neither correctly recognized by Google nor IBM ASR systems, more than half of them had recognition accuracies higher than 50%. The remaining

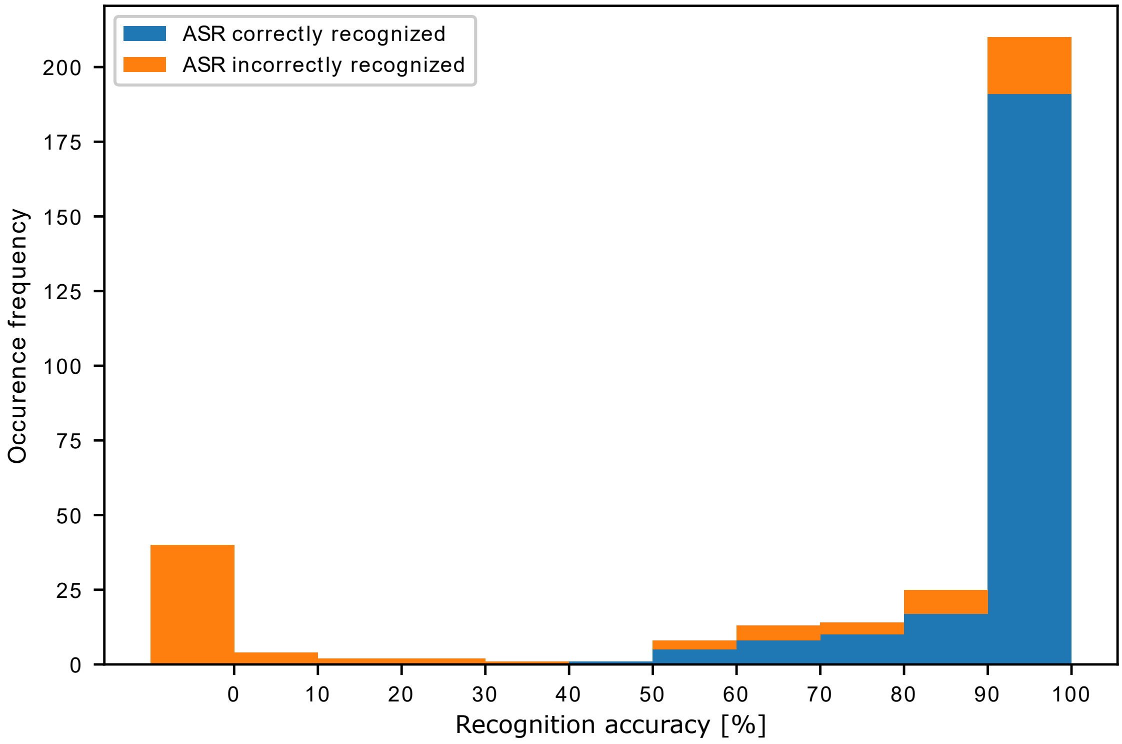


FIGURE 5.21: Histogram of recognition accuracies evaluated by perception experiment.

40 utterances, represented by the first bar in Figure 5.21, were excluded from the listening experiment due to the low intelligibility. They are marked with recognition accuracies lower than 0% so as to distinguish from those utterances included in the listening test.

5.6 Evaluation on Utterances of Other Languages

Since the articulatory synthesis is implemented by modeling human vocal tract, vocal folds, and the articulatory processes of speech production, it is language independent to some extent. To investigate the generalization of the trained models, they were tested using English and Mandarin Chinese utterances in terms of intelligibility of reproduced speech.

5.6.1 Oxford-805 Corpus

The English utterances were isolated words selected from the word list "Oxford 3000"[10], which contained the 3000 core words to learn in English. They were carefully chosen by language experts and experienced teachers as the words that had the priority to be mastered during vocabulary study. According to the description of Common European Framework of Reference (CEFR), these words were at A1-B2 levels, i.e., elementary and pre-intermediate levels of ability. Therefore, the words

[10] https://www.oxfordlearnersdictionaries.com/about/oxford3000

of Oxford 3000 were considered simple and common, and they were used as the candidate words in this study.

The words having two, three or four syllables were first selected, resulting in a subset of 1907 words. The audio files of their pronunciations came from Google Dictionary which is an online dictionary service of Google. The definition of a word together with its pronunciation can be accessed via a search query like "define:example" in Google Search. Users can also directly play and download the pronunciation audio file. Below is the link of the audio file of the word "example".

https://ssl.gstatic.com/dictionary/static/sounds/oxford/example--_gb_1.mp3

Users can replace "example" in this link with any of English words to look up. The accent of the searched word can be specified during querying, e.g. "gb" or "us" for Great Britain or American English, respectively. The pronunciation audio files (in MP3 format) of Great Britain accent of the 1907 words were downloaded using a Python script, which were subsequently converted to the WAV format.

The pronunciation audios provided by Google Dictionary were sometimes produced by female speakers while the VocalTractLab model speaker was recorded from a male speaker. Hence, the audios produced by female speakers were excluded. The remaining 805 audio files came from one male speaker based on the judgement of his voice quality. These utterances of isolated English words, referred to as Oxford-805 corpus, were used in this study for measuring the generalization ability of the trained model to English utterances.

The six ACS systems trained with Mixed samples conducted ACS on the Oxford-805 utterances. The Google and IBM speech recognizers with the "gb" model were used to recognize reproduced utterances. Table 5.11 shows their ASR accuracies together with that of the original natural utterances (the detailed recognition results can be found in Appendix D.1). This table shows a similar accuracy pattern among these systems. The acoustic loss regularization performed better than the smoothness loss regularization. Besides, except for System-L, the other systems based on ConvLSTM and Spectrogram achieved higher accuracies than those based on LSTM and MFCC. For the Oxford-805 corpus, the estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive repository[11], [12].

5.6.2 CAPL-1592 Corpus

The Chinese Interlanguage corpus for computer assisted pronunciation learning (CAPL) (Cao and Zhang, 2009) was used for testing the ACS systems on Mandarin Chinese utterances. This corpus consisted of six sub-corpora distinguished by the number of syllables of the words. This study used the sub-corpus of 1592 disyllabic words produced by the male speaker "LH", referred to as CAPL-1592 corpus

[11] The estimated articulatory trajectories and reproduced speech for Oxford-805 corpus: https://drive.google.com/drive/folders/1i4n88qhVLsxaZfx5iZN3EoCmgFOw2fFT?usp=sharing

[12] A few animation examples of reproduced Oxford-805 utterances: https://drive.google.com/drive/folders/1t0txitw9hYtocHPUkg2qxcDfjxtID0s1?usp=sharing

TABLE 5.11: ASR accuracies (in %, by Google and IBM) of reproduced Oxford-805 utterances by different ACS systems.

	LSTM-MFCC	**ConvLSTM-Spec**
Reproduced without Regularization	(C) 31.06	(D) 46.34
Reproduced with smoothness loss regularization	(G) 35.90	(H) 49.94
Reproduced with acoustic regularization	(K) 52.92	(L) 50.93
Original natural utterances	98.63	

in this study. The same six ACS systems conducted ACS on the 1592 Chinese spoken words. During the performance measurement, the recognition accuracies for character and base syllable were used as the metrics instead of word recognition accuracy. The reason can be explained with the example shown in Figure 5.22.

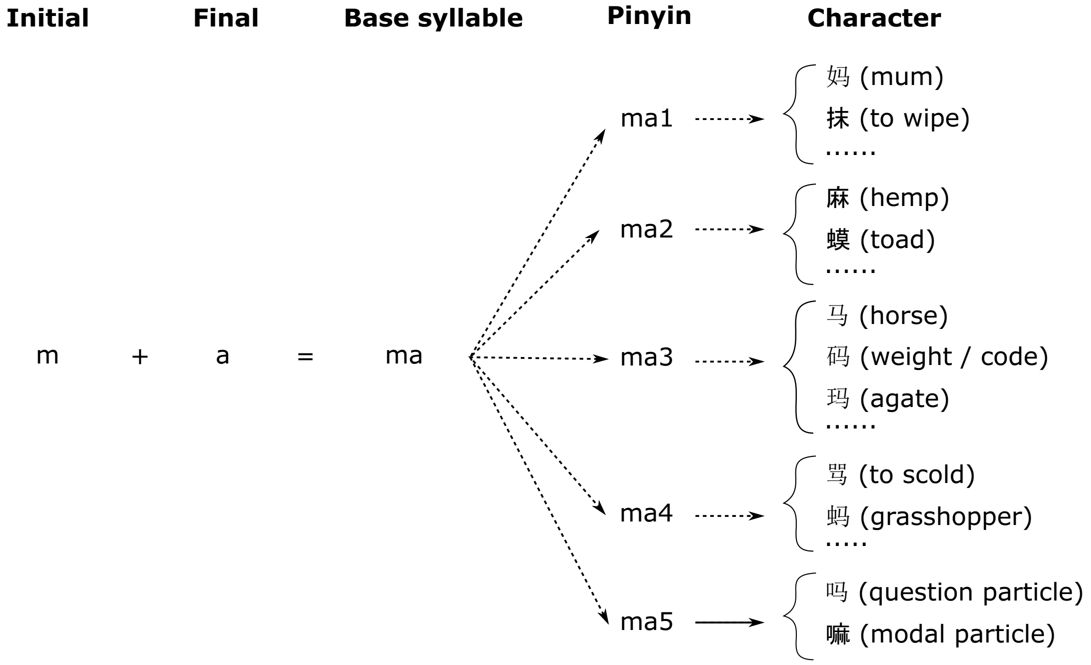


FIGURE 5.22: Example of relation between Chinese Pinyin and character.

Mandarin Chinese is a well-known syllable-based tone language. Each base syllable consists of an optional initial and a mandatory final. A base syllable associated with one of five pitch tones is termed Pinyin. Pitch tones play crucial phonemic roles so that the same syllable with different tones has different lexical meanings. Even for a specific Pinyin, there might exist one or several corresponding characters (i.e., homophones). Likewise, a disyllabic utterance may have different possible character combinations. Without a context, it is sometimes hard to determine which character is really spoken. In the current study, the ACS system perhaps correctly reproduced the base syllable but the tone might be incorrect due to possible errors during pitch (or pitch target) estimation. Even in the case both base syllable and tone were correctly reproduced (i.e., the reproduced speech sounded the same to the reference one), the speech recognizer did not necessarily produce the same characters without a context (here, the language model was nonfunctional for isolated word utterances). Therefore, the recognition accuracies for both character and base syllable were used as the metrics for measuring the performance of the trained ACS systems on Mandarin Chinese utterances.

In this study, the recognized disyllabic words were split into two characters which were further converted to the base syllables. Table 5.12 shows the accuracies for characters and base syllables reproduced by the six ACS systems together with those for the original natural utterances (the detailed recognition results can be found in Appendix D.2). Compared to German and English isolated words, the Chinese character recognition accuracy of natural utterances (86.33%) was much lower while the recognition accuracy of base syllables was on the similar level. Again, the best performance on Chinese CAPL-1592 was obtained by System-K. In the case of using acoustic loss regularization, the LSTM-MFCC based system outperformed the ConvLSTM-Spectrogram based system. In other two cases (without regularization or with the smoothness loss regularization), the pattern was the other way around. For the CAPL-1592 corpus, the estimated articulatory trajectories and reproduced utterances by System-K and System-L can be found in Google Drive repository[13, 14].

TABLE 5.12: ASR accuracies (in %, by Google and IBM) of reproduced CAPL-1592 utterances by different ACS systems.

	Character		**Base syllable**	
	LSTM-MFCC	**ConvLSTM-Spec**	**LSTM-MFCC**	**ConvLSTM-Spec**
Reproduced without Regularization	(C) 25.92	(D) 26.86	(C) 34.17	(D) 39.24
Reproduced with smoothness loss regularization	(G) 26.8	(H) 27.71	(G) 35.18	(H) 39.78
Reproduced with acoustic loss regularization	(K) 44.06	(L) 33.77	(K) 52.41	(L) 42.68
Original natural utterances	86.33		96.47	

5.7 Concluding Remarks

This chapter investigated ACS by training neural network based regression models which accepted acoustic features as inputs and estimated articulatory trajectories as outputs. To this end, the paired articulatory-acoustic samples were created using

[13] The estimated articulatory trajectories and reproduced speech for CAPL-1592 corpus: https://drive.google.com/drive/folders/1N-ZHu7MmHG3npRMQX3uMcVw4_ZESNhdI?usp=sharing

[14] A few animation examples: https://drive.google.com/drive/folders/1GF8lwRwJcoCjJKtVSeIFivJZGLGcXc8a?usp=sharing

VTL and the Kiel natural utterances. To ensure that the training samples covered as much of the articulatory and acoustic space as possible, 12 versions of the gestural scores or waveforms for an original utterance were created; these versions differed in terms of phonation types, speaking effort, and vocal tract length. The methods proposed in this study could be used to create as many articulatory-acoustic samples as needed to train supervised models. In addition, the diverse speaking rates and pitch levels were also covered since the original utterances of the Kiel corpus were produced by different speakers.

Two kinds of neural network architectures were compared: LSTM and convolutional LSTM. As a variant of RNN, LSTM efficiently modeled the temporal dependencies of input sequences using a bi-directional recurrent architecture. In addition to the input features at a specific time step, the forward layers and backward layers were allowed to make full use of past and future information stored in the memory cells. The LSTM was adopted in this study to capture the temporal dependence of the input sequence while a CNN was used to capture the spatial structure of acoustic features. This chapter investigated the use of a combination of convolutional and batch normalization layers to automatically learn distinctive features from log power spectrograms for speech inversion.

The speaker-dependent ACS was relatively simple. When both the training and testing samples were produced by VTL, the ACS systems obtained very high correlation coefficients and small cosine distance between the estimated and target articulatory trajectories. This chapter then focused on training robust ACS systems against speaker variation. To this end, the articulatory trajectories converted by the prototype gestural scores and acoustic features of human utterances of the Kiel corpus were combined and used as "paired" articulatory-acoustic samples to train neural networks. Although they did not match perfectly with respect to the time axis, incorporating them could make the trained models more robust to the utterances of unknown speakers. The results showed that the systems trained with fewer "mismatched" samples even achieved much better performance in terms of ASR accuracy than the systems trained with a large number of synthetic samples in which both the articulatory trajectories and the acoustic features were from VTL.

The results in this chapter suggested that the ConvLSTM-Spectrogram based systems generally outperformed the LSTM-MFCC based systems. The high ASR accuracy confirmed that the ConvLSTM could not only capture the temporal and spatial patterns of acoustic features but also make the trained models more robust to speaker variations. Complementary experiments were also conducted to ascertain the effectiveness of CNN. Additional ACS systems were trained using LSTM with normalized spectrograms as inputs. However, all manual normalization strategies were found to be inferior to automatic feature learning by using a combination of convolutional and batch normalization layers.

Furthermore, a large amount of effort was spent on investigating regularization methods. In addition to smoothing the estimated articulatory trajectories in the post-processing stage, two regularization methods were proposed which were incorporated into the loss function during training neural networks. The first method used smoothness loss, which measured the smoothness of estimated articulatory trajectories. The second method used acoustic loss, which measured the preference of estimated articulatory trajectories by VTL. This was implemented by training both a backward model and a forward model using the same training samples in advance. These two pre-trained models were then concatenated and jointly trained again to minimize the weighted sum of the articulatory loss and the acoustic loss. The acoustic loss tended to decrease only when the estimated articulatory trajectories were more "preferred" by the forward model. Consequently, this made the estimated articulatory trajectories be more preferred by VTL, thus producing more natural and intelligible speech.

The ASR results showed that the systems with regularization performed better than their counterparts that did not include regularization. The acoustic loss regularization generally outperformed the smoothness loss regularization. The only exceptions were systems that used spectrograms as inputs. This study used log power spectrograms as inputs that were not explicitly normalized in advance but were implicitly normalized by the convolutional and batch normalization layers. However, in the case of the encoder-decoder based ACS system, the magnitudes of the spectrograms were so large that using them directly as targets for the forward models made it difficult for the neural networks to converge. Therefore, the explicitly normalized spectrograms were used as targets for the forward models. As this backward-forward model architecture was inspired by the idea of encoder-decoder, the mismatch between original inputs and the targets may degrade the performance of such ACS systems.

A web-based online perception experiment was conducted to investigate the intelligibility of reproduced utterances and the reliability of automatic evaluation by machines. 320 PBACU utterances reproduced by the best-performing ACS system (System-L for German speech) were first assessed by the author. 280 of them were judged to be recognizable by human listeners; these were used as the stimuli for 30 participants in the experiment. The recognition results showed that most of the stimuli presented to human listeners were intelligible. In addition, some utterances that were not correctly recognized by machines were correctly recognized by most human listeners. Likewise, there were some utterances that were recognized by machines but difficult for human listeners to recognize.

Finally, English and Mandarin Chinese utterances were used to investigate the generalization ability of the trained models in terms of intelligibility of reproduce speech. Although the ACS systems were trained using German utterances, they were capable of estimating articulatory trajectories from the acoustic signals for utterances of other languages.

Chapter 6

Conclusions and Future Work

6.1 Summary

Speech is the most common mode of human communication and is thus widely investigated. The process of speech communication involves three main events: the production of speech, the propagation of speech, and the perception of speech. Together, these three events constitute the speech chain. Accordingly, speech can be investigated in three distinct but interdependent domains: articulatory, acoustic, and auditory/perceptual domains. This thesis studied the field of speech inversion, which is closely associated with and built upon the three aspects of the speech chain.

Chapter 2 formulated the generic concept of speech inversion and divided it into three research topics: acoustic-to-articulatory inversion (AAI), computational speech acquisition (CSA), and articulatory copy synthesis (ACS). The relevant literature for each of these three research topics was reviewed in terms of the acoustic and articulatory representations of utterances, inversion algorithms, evaluation metrics, etc. Through the comparison of the similarities, differences, and existing challenges of these three research topics, this study narrowed its focus to the field of ACS, which emphasized the reproduction of reference utterances, and involved both the physiological articulation processes and the corresponding acoustic results.

Chapter 3 laid the foundations for ACS research of this thesis. The articulatory synthesizer VocalTractLab (VTL) as well as its organization patterns of articulatory processes, acoustic representations of utterances, experiment platform and tools were introduced. More importantly, two corpora were deliberately designed to validate the proposals in this thesis.

Chapter 4 presented an ACS method based on a genetic algorithm. This method involved two steps: gestural score initialization and optimization. It also attempted to regularize deviations of the "time constant" parameters. Automatic evaluations in the acoustic, articulatory, and perceptual domains were also conducted. Compared to most previous studies, this proposal's main contribution to the literature was that the articulatory processes and acoustic signals of speaker-independent word-level utterances could be reproduced without the need for transcriptions or the manual segmentation of target utterances.

Chapter 5 presented another ACS method based on artificial neural networks and VTL. Using the synthetic dataset generated by VTL, LSTM and convolutional LSTM neural network models were trained to estimate VTL articulatory parameters from acoustic inputs. Compared to the handcrafted acoustic features (MFCC) that were explicitly normalized before being used as inputs, the features that were automatically learned by CNN (specifically, convolutional and batch normalization layers) from spectrograms were more distinctive for ACS. Besides, two additional regularization methods were also proposed. The smoothness or acoustic loss was directly incorporated into the loss function during the training stage of ACS systems. In terms of the ASR accuracy of the reproduced utterances, both of these regularization methods outperformed smoothing the estimated articulatory trajectories in the post-processing stage. In addition to making the estimated articulatory trajectories smoother and reducing the training time, acoustic loss regularization resulted in the estimated articulatory trajectories being more preferred by the forward model, thus producing more natural and intelligible utterances. Furthermore, a perceptual experiment by human listeners was conducted to complement the automatic evaluations by machines. The high speech recognition accuracy further verified the intelligibility of reproduced utterances and also confirmed that some utterances that were not correctly recognized by machines were still possibly recognizable by human listeners. Finally, this chapter also confirmed the generalization ability of the ACS models trained on German utterances by testing utterances of other languages (English and Mandarin Chinese).

This thesis addressed some of the limitations of other studies. The ACS methods proposed in this study did not require transcription and/or segmentation in advance. The proposals were applicable to both speaker-dependent and speaker-independent scenarios. The utterances to be reproduced were words and sentences instead of isolated vowels or simple "CV[C]" syllables. The estimated articulatory trajectories were more consistent with the actual articulatory process of human speakers since they could be fed into VTL to synthesize natural and intelligible speech.

6.2 Discussion

To reduce the impact of the different magnitudes of feature dimensions on distance calculations, normalization of acoustic features is required before the similarity calculation. The cepstral mean and variance normalization (CMVN) technique is widely used in ASR systems. Normally, CMVN is individually performed for each utterance using the statistics estimated from the current utterance or speaker. This approach is suitable for long utterances but the performance usually degrades for short utterances due to insufficient data. Therefore, in Chapter 4, four kinds of acoustic features (MFCC, energy, ZCR, and POV) of both synthetic and natural speech were normalized using the same set of statistics that were estimated from another corpus in advance. However, this method did not necessarily reflect the acoustic similarity for all segments. Using a few utterances, we conducted a pilot experiment in which the conventional CMVN method was used. The results showed that, compared to the current method used in Chapter 4, the conventional CMVN method worked better for some of the testing utterances while it worked worse for the rest. Hence, the results of the pilot experiment were not reported here. How to normalize acoustic features before similarity calculation is worth systematically investigating.

In addition, the common global statistics used for Z-score normalization in Chapter 4 were calculated based on the BITS corpus. Although it was phonetically balanced, all utterances were produced by only four speakers (two males and two females). Using a corpus with more speakers or only using utterances from male speakers might increase the robustness of the acoustic statistics. Besides, cosine distance was used since it was less sensitive to feature magnitudes than Euclidean distance. However, the cosine distance was not always optimal. We also conducted a pilot experiment using a few utterances of the PBACU corpus. The results showed that some of the reproduced utterances whose gestural scores were optimized using Euclidean distance were of better quality than their counterparts optimized using cosine distance.

The synthesizer VTL provided three vocal fold models: a geometric model, a two-mass model, and a triangular glottis model. The vocal fold model used in this study was the geometric model. The degree of complexity of the speech inversion depended on the parameters chosen to represent both the articulatory and acoustic space. To simplify the model, some articulatory parameters were not included in the neural network models described in Chapter 5. For example, only the lower vocal fold displacement was modeled while the upper vocal fold displacement was determined by adding 0.01 to the estimated value of the lower vocal fold displacement. Five other vocal fold parameters were set to VTL's default values. Properly modeling them together with the other parameters that were already included might improve the quality of the reproduced utterances.

Furthermore, voice quality is not usually considered in speech inversion tasks. The anatomical values of the VTL model speaker were fixed. If the voice quality similarity was also included as a goal of ACS, then optimizing the default values of some parameters as well as modifying the anatomical parameters of the model speaker might be beneficial. A preliminary experiment was conducted but not included in this study. Some stable segments of German vowels were extracted from natural utterances and used as the targets of ACS. The shapes of corresponding vowels defined in VTL were used as the initial states. A particle swarm optimization algorithm was then employed to adjust the articulatory parameters. This step was similar to the study that identified the underlying articulatory targets of Thai vowels from acoustic data (Prom-on, Birkholz, and Xu, 2014). The speech synthesized with the resulting vowel shapes had much smaller acoustic distances than the speech synthesized without optimizing vowel shapes. A further attempt was made to optimize the configurations of the VTL model speaker by fixing the vowel shapes and adjusting the anatomical parameters of the model speaker. The results showed that the acoustic distance continued to decrease while the speech synthesized with the optimized speaker sounded unnatural. Due to insufficient knowledge about the anatomical differences between the two speakers and the lack of physiological constraints during optimization, this naive strategy did not work well. However, these attempts hinted at the possibility of further reducing the acoustic distance and increasing the perceptual similarity between synthetic and reference speech. More factors should be considered if voice quality is used as an additional goal of ACS.

6.3 Future Work

In Chapter 4, several handcrafted acoustic features like MFCC and POV were used to measure the frame-wise acoustic similarity between the reproduced and reference speech. These features may not work well for the acoustic variations of all phones. The other generic features extracted from acoustic signals might improve the overall performance for acoustic distance measurement. One possible solution is acoustic word embedding (AWE). The AWE technique converts an audio signal into a new representation in the AWE space where acoustically and/or linguistically similar utterances are clustered together. Hence, future work involving ACS tasks can calculate acoustic similarity based on the distance in AWE space. In addition, the acoustic signals can be converted into a fixed dimensional representation regardless of the length of utterances, making it appropriate for CSA.

Speech intelligibility determines how easily and accurately individual words within an utterance can be recognized by a listener. The intelligibility of the utterances reproduced in this study was evaluated via speech recognition (by both machines and human listeners) and in terms of PESE and STOI in the perceptual domain. Naturalness is another metric that describes the quality of the speech in terms of its timing structure, pronunciation, rendering of emotions, etc. The naturalness of the reproduced speech was not assessed in this study. In future work, other automatic evaluations like NISQA-TTS (Mittag and Möller, 2020) can be used to assess the naturalness of synthetic speech.

The gestural scores were optimized with respect to the acoustic distance between synthetic and reference utterances while the neural network models were trained in terms of articulatory loss and/or acoustic loss. However, neither of these models considered perceptual similarity during optimization or model training, even though the quality of the reproduced utterances were evaluated from a perceptual perspective. Therefore, future work should investigate the incorporation of other quantitative metrics of acoustic, articulatory, and perceptual similarity or losses simultaneously when training the systems or optimizing gestural scores; this may further increase the similarity between the reproduced and reference utterances.

In Chapter 5, during the training of backward models that mapped acoustic representations to articulatory representations, the loss function was based on mean square error (MSE), which treated all articulatory parameters equally. However, perceptual features vary in a highly nonlinear manner with respect to changes in articulatory parameters. Fang (2020) argued that the training samples of different articulatory channels were usually unbalanced and played different roles in AAI. Therefore, he proposed that each articulatory channel at each time instant could be classified as critical or noncritical according to its role in the formation of constrictions along the vocal tract for a given phoneme. In this way, each articulatory channel for a given utterance would be composed of a combination of critical articulatory portions (CAP) and non-critical articulatory portions (NCAP). He modified the MSE-based cost function by selectively discarding the loss of NCAP samples such that the trained model reduced the CAP prediction errors (Fang, 2020). In light of this, future work could investigate the relative impact of different articulatory parameters on both acoustic and perceptual similarity. Furthermore, using a set of weights for different articulatory parameters in MSE-based loss calculation could improve the performance of neural network models.

The acoustic data used for validating the proposed methods were produced by human speakers. However, no real articulatory data were used in this study. The articulatory trajectories of the testing utterances were created by VTL. In future work, it is also worth evaluating the similarity of articulatory trajectories using real, paired articulatory-acoustic data as testing utterances. For example, the ACS systems trained with neural networks in Chapter 5 can be applied to standard articulatory-acoustic datasets like "mngu0" and "MOCHA-TIMIT". The estimated articulatory trajectories can be compared with the real articulatory variables contained in EMA data.

Appendix A

The Word List of PBACU Corpus with Canonical SAMPA Transcriptions

TABLE A.1: The word list of PBACU corpus with canonical SAMPA transcriptions.

Orthography	SAMPA Transcription	Orthography	SAMPA Transcription
Badminton	/b E t . m I n . t @ n/	Mensa	/m E n . z a:/
bayrisch	/b aI . r I S/	Milieu	/m I l . j 2:/
beheben	/b @ . h e: . b m/	mitmachen	/m I t . m a . x @ n/
Belohnung	/b @ . l o: . n U N/	Monarchie	/m o: . n a 6 . C i:/
Berater	/b @ . r a: . t 6/	monatlich	/m o: . n a t . l I C/
Bereich	/b @ . r aI C/	Monitor	/m o: . n i: . t o:6/
Besitz	/b @ . z I t s/	Monopol	/m o: . n o: . p o: l/
besonders	/b @ . z O n . d 6 s/	moralisch	/m o: . r a: . l I S/
bevor	/b @ . f o:6/	Motiv	/m o: . t i: f/
bezahlen	/b @ . t s a: . l @ n/	Motorrad	/m o: . t o:6 . r a: t/
Bezirk	/b @ . t s I6 k/	Mythos	/m y: . t O s/
Buchung	/b u: . x U N/	Nachbarin	/n a x . b a: . r I n/
Bundeswehr	/b U n . d @ s . v e:6/	nachholen	/n a: x . h o: . l @ n/
Campingplatz	/k E m . p I N . p l a t s/	nebenbei	/n e: . b @ n . b aI/
Chemie	/C e: . m i:/	nehmen	/n e: . m @ n/
Chirurg	/C i: . r U6 k /	Neubau	/n OY . b aU/
Cowboy	/k aU . b OY/	neugierig	/n OY . g i: . r I C/
dadurch	/d a: . d U6 C/	nirgendwo	/n Ir . g @ n t . v o:/
dagegen	/d a: . g e: . g @ n/	parteilos	/p a 6 . t aI . l o: s/
Deutung	/d OY . t U N/	Parteitag	/p a 6 . t aI . t a: k/
Doktor	/d O k . t O6/	Performance	/p 96 . f O6 . m E n s/
durchhalten	/d U6 C . h a l . t @ n/	Physik	/f y . z i: k/
Dynamo	/d y . n a: . m o:/	Pole	/p o: . l @/
Fauna	/f aU . n a:/	Politik	/p o: . l i . t i: k/
Firma	/f I6 . m a:/	politisch	/p o: . l i: . t I S/
Forderung	/f O6 . d @ . r U N/	Polizei	/p o: . l i: . t s aI/
gebildet	/g @ . b I l . d @ t/	positiv	/p o: . z i . t i: f/
Geburt	/g @ . b U6 t/	produziert	/p r o: . d u: . t s i:6 t/

Table A.1 continued from previous page

Orthography	SAMPA Transcription	Orthography	SAMPA Transcription
Gegensatz	/g e: . g @ n . z a t s/	Prototyp	/p r o . t o: . t y: p/
Gegenteil	/g e: . g @ n . t aI l/	reduziert	/r e: . d u: . t s i:6 t/
gehalten	/g @ . h a l . t @ n/	Regie	/r e: . Z i:/
genau	/g @ . n aU/	relativ	/r E . l a . t i: f/
genauso	/g @ . n aU . z o:/	Rhythmus	/r Y t . m U s/
genug	/g @ . n u: k/	rigoros	/r i: . g o: . r o: s/
gerade	/g @ . r a: . d @/	Risiko	/r i: . z i . k o/
Geruch	/g e: . r U x/	rosa	/r o: . z a:/
Gesetz	/g @ . z E t s/	sagen	/z a: . g @ n/
Gesuch	/g @ . z u: x/	Seminar	/z e . m i: . n a: 6/
Gipfel	/g I p . f @ l/	Senator	/z e . n a: . t o:6/
haben	/h a: . b @ n/	sensibel	/z E n . z i: . b @ l/
Hamburger	/h a m . b U6 . g 6/	Service	/s 96 . v I s/
Handy	/h E n . d i:/	sogar	/z o: . g a:6/
Harmonie	/h a 6 . m o . n i:/	sowieso	/z o: . v i: . z o:/
Hierarchie	/h i: . r a 6 . C i:/	symbolisch	/z Y m . b o: . l I S/
hierher	/h i:6 . h e:6/	System	/z Y s . t e: m/
Hindernis	/h I n . d 6 . n I s/	Tango	/t a N . g o:/
hinterher	/h I n . t 6 . h e:6/	Teufel	/t OY . f @ l/
hinzu	/h I n . t s u:/	Thema	/t e: . m a:/
historisch	/h I s . t o: . r I S/	tierisch	/t i: . r I S/
Hobby	/h O . b i:/	Toleranz	/t o . l @ . r a n t s/
Honig	/h o: . n I C/	Tournee	/t U6 . n e:/
human	/h u: . m a: n/	typisch	/t y: . p I S/
Humor	/h u: . m o:6/	Variante	/v a . r i a n . t @/
Jahrhundert	/j a: 6 . h U n . d 6 t/	vergleichbar	/f E6 . g l aI C . b a: 6/
jedoch	/j e: . d O x/	Verkauf	/f E6 . k aU f/
Journal	/Z U r . n a: l/	Verlag	/f E6 . l a: k/
Jubel	/j u: . b @ l/	vermuten	/f E6 . m u: . t @ n/
Jura	/j u: . r a/	verrichten	/f E6 . r I C . t @ n/
Kardinal	/k a 6 . d i: . n a: l/	Versuch	/f E6 . z u: x/
keinerlei	/k aI . n 6 . l aI/	vertretbar	/f E6 . t r e: t . b a: 6/
Kilo	/k i: . l o/	vorbei	/f o:6 . b aI/
Kino	/k i: . n o/	vorhanden	/f o:6 . h a n . d @ n/
Kirche	/k I6 . C @/	wahrlich	/v a: 6 . l I C/
Koma	/k o: . m a/	wenig	/v e: . n I C/
Komiker	/k o: . m i . k 6/	wiederum	/v i: . d @ . r U m/
komisch	/k o: . m I S/	wieso	/v i . z o:/
Komitee	/k o . m i: . t e:/	Wirklichkeit	/v I6 k . l I C . k aI t/
Kopie	/k o . p i:/	Wirkung	/v I6 . k U N/
Kultur	/k U l . t u:6/	wobei	/v o: . b aI/
kulturell	/k U l . t u: . r E l/	Wohnung	/v o: . n U N/
Laptop	/l E p . t O p/	worauf	/v o: . r aU f/
lediglich	/l e: . d I g . l I C/	wozu	/v o: . t s u:/

Table A.1 continued from previous page

Orthography	**SAMPA Transcription**	**Orthography**	**SAMPA Transcription**
Lehrbuch	/l e:6 . b u: x/	wunderbar	/v U n . d 6 . b a: 6/
Lehrerin	/l e: . r @ . r I n/	zugute	/t s u: . g u: . t @/
Liebhaber	/l i: p . h a: . b 6/	Zuhause	/t s u: . h aU . z @/
logisch	/l o: . g I S/	Zulassung	/t s u: . l a . s U N/
Logo	/l o: . g o/	zuviel	/t s u: . f i: l/
machbar	/m a x . b a: 6/	zuvor	/t s u: . f o:6/
Malerei	/m a: . l @ . r aI/	zuwenig	/t s u: . v e: . n I C/
Medizin	/m e: . d i . t s i: n/	zynisch	/t s y: . n I S/

Appendix B

Detailed Recognition Results of Original and Reproduced PBACU Utterances by Google ASR system

TABLE B.1: Utterance intelligibility measured in terms of Google ASR accuracy. The "1" indicates that the recognized content was consistent with the word in the first column of the table, i.e., the target word was correctly reproduced by the copy synthesis methods in terms of intelligibility. The "0" indicates the recognized content was inconsistent with the expected word. The last row and the last column are the total numbers of recognized utterances.

	SPK-1					SPK-2					Sum
Utterance	Orig	Init	CD	GA w/o Reg	GA w/ Reg	Orig	Init	CD	GA w/o Reg	GA w/ Reg	
Badminton	1	1	0	0	0	1	1	0	0	1	5
bayrisch	1	1	1	1	1	1	1	1	1	1	10
beheben	1	0	0	1	1	1	1	0	0	1	6
Belohnung	1	0	1	1	1	1	1	1	1	1	9
Berater	1	1	1	1	1	1	1	1	1	1	10
Bereich	1	1	1	1	1	1	1	1	1	1	10
Besitz	1	1	0	0	1	1	1	1	1	1	8
besonders	1	1	1	1	1	1	1	1	1	1	10
bevor	1	1	1	1	1	1	1	1	1	1	10
bezahlen	0	1	0	0	1	1	1	1	1	1	7
Bezirk	1	1	0	1	1	1	1	1	0	0	7
Buchung	1	0	1	1	1	1	0	1	1	0	7
Bundeswehr	1	0	0	0	1	1	0	1	1	0	5
Campingplatz	1	1	1	0	0	1	1	1	1	0	7
Chemie	1	1	1	1	1	1	1	1	1	1	10
Chirurg	1	1	1	1	1	1	0	0	0	0	6
Cowboy	1	1	1	1	1	1	1	1	1	1	10
dadurch	1	1	1	1	1	1	0	1	1	1	9
dagegen	1	1	1	1	1	1	1	1	1	1	10
Deutung	1	0	0	1	1	1	0	0	0	1	5
Doktor	1	0	0	1	1	1	1	1	1	1	8
durchhalten	1	1	1	1	1	1	1	1	1	1	10
Dynamo	1	0	0	1	1	1	1	1	1	1	8
Fauna	1	0	0	0	0	1	0	1	1	1	5
Firma	1	0	0	0	0	1	1	0	0	0	3
Forderung	1	1	1	1	1	1	1	1	1	1	10
gebildet	1	1	1	1	1	1	0	1	0	0	7
Geburt	1	0	1	1	1	1	1	1	1	1	9
Gegensatz	1	1	1	1	1	1	1	1	1	1	10
Gegenteil	1	1	1	1	1	1	1	1	1	1	10
gehalten	1	0	1	1	1	1	1	0	0	1	7

genau	1	1	1	1	1	1	1	1	1	1	10
genauso	1	1	0	1	1	1	1	1	1	1	9
genug	1	1	1	1	1	1	1	0	0	0	7
gerade	1	1	1	1	1	1	1	1	1	1	10
Geruch	1	1	1	1	1	1	1	0	0	0	7
Gesetz	1	0	0	1	1	1	1	1	1	1	8
Gesuch	1	1	0	1	1	1	1	1	1	1	9
Gipfel	1	0	0	1	1	1	1	0	0	0	5
haben	1	1	1	1	1	1	1	1	1	1	10
Hamburger	1	1	1	1	1	1	1	1	1	1	10
Handy	1	1	0	1	1	1	1	1	1	1	9
Harmonie	1	1	1	1	1	1	1	1	1	1	10
Hierarchie	1	1	1	1	1	1	1	1	1	1	10
hierher	1	1	0	1	1	1	0	0	0	0	5
Hindernis	1	1	1	0	0	1	1	1	1	1	8
hinterher	1	1	1	0	0	1	1	1	1	1	8
hinzu	1	1	1	1	1	1	1	1	1	1	10
historisch	1	1	1	1	1	1	1	1	1	1	10
Hobby	1	1	1	1	1	1	1	0	1	1	9
Honig	1	1	0	0	1	1	1	1	1	1	8
human	1	0	0	0	0	1	1	0	1	0	4
Humor	1	1	1	0	1	1	1	1	0	1	8
Jahrhundert	1	1	1	1	1	1	1	1	1	1	10
jedoch	1	1	1	1	1	1	1	1	1	1	10
Journal	1	0	0	1	1	0	1	1	0	1	6
Jubel	1	1	1	1	1	1	1	1	1	1	10
Jura	1	1	1	1	1	1	1	1	1	1	10
Kardinal	1	1	1	1	1	1	1	0	1	1	9
keinerlei	1	1	1	1	1	1	1	1	1	1	10
Kilo	0	0	0	0	0	1	1	0	0	1	3
Kino	1	1	1	1	1	1	0	1	1	1	9
Kirche	1	1	1	1	0	1	1	1	1	1	9
Koma	1	1	1	1	1	1	1	1	0	0	8
Komiker	1	1	1	1	1	1	1	1	1	1	10
komisch	1	1	1	1	1	1	1	1	1	1	10
Komitee	1	1	1	1	1	1	1	1	1	1	10
Kopie	1	0	0	0	1	1	1	0	0	0	4
Kultur	1	1	1	1	1	1	1	1	0	1	9
kulturell	1	1	1	0	1	1	1	1	1	1	9
Laptop	1	1	1	1	1	1	1	1	1	1	10
lediglich	1	1	1	1	1	1	1	1	1	1	10
Lehrbuch	1	0	0	1	1	1	1	1	1	1	8
Lehrerin	1	1	1	0	0	1	1	0	1	1	7
Liebhaber	1	1	0	1	0	1	0	1	1	1	7
logisch	1	1	1	1	1	1	1	1	1	1	10
Logo	1	1	1	1	1	1	1	1	0	1	9
machbar	1	0	1	1	1	1	0	1	1	1	8
Malerei	1	1	1	1	1	1	1	1	1	1	10
Medizin	1	1	1	1	1	1	1	1	1	1	10
Mensa	1	1	1	1	1	1	1	1	1	1	10
Milieu	1	1	1	1	1	1	1	1	1	1	10
mitmachen	1	1	1	1	1	1	1	1	1	1	10
Monarchie	1	1	1	1	1	1	1	1	1	1	10
monatlich	1	1	1	1	1	1	1	1	1	1	10
Monitor	1	1	1	1	1	1	1	1	1	1	10
Monopol	1	1	1	1	1	1	1	1	1	1	10
moralisch	1	1	1	1	1	1	1	1	1	1	10
Motiv	1	1	0	0	0	1	1	1	1	1	7
Motorrad	1	1	1	1	1	1	1	1	1	1	10
Mythos	1	1	1	1	1	1	1	1	1	1	10
Nachbarin	1	1	0	0	0	1	1	1	1	1	7
nachholen	1	1	1	1	1	1	1	1	1	1	10
nebenbei	1	1	1	1	1	1	1	1	1	1	10
nehmen	1	1	1	1	1	1	1	1	1	1	10
Neubau	1	1	1	1	1	1	1	1	1	1	10
neugierig	1	1	1	1	1	1	1	1	1	1	10
nirgendwo	1	1	1	1	0	1	1	1	1	1	9
parteilos	1	1	1	0	1	1	1	1	1	1	9

Parteitag	1	1	1	0	1	1	1	1	1	1	9
Performance	1	1	1	1	1	1	1	1	1	1	10
Physik	1	1	1	1	1	1	1	1	1	1	10
Pole	0	0	0	0	0	1	1	1	0	1	4
Politik	1	0	1	0	0	0	1	1	1	1	6
politisch	1	1	1	0	1	1	1	1	1	1	9
Polizei	1	1	1	1	1	1	1	1	1	1	10
positiv	1	1	1	1	1	1	1	1	1	1	10
produziert	1	1	1	1	1	1	1	0	0	1	8
Prototyp	1	1	1	1	1	1	1	1	1	1	10
reduziert	1	1	1	1	1	1	1	1	1	1	10
Regie	1	0	1	1	1	1	1	0	1	1	8
relativ	1	1	1	1	1	1	1	1	1	1	10
Rhythmus	1	1	1	1	1	1	1	1	1	1	10
rigoros	1	1	1	1	1	1	1	1	1	1	10
Risiko	1	1	1	1	1	1	1	1	1	1	10
rosa	1	1	1	1	1	1	1	1	1	1	10
sagen	1	1	1	0	0	1	1	1	1	1	8
Seminar	1	1	1	0	0	1	1	1	1	1	8
Senator	1	1	1	1	1	1	1	1	1	1	10
sensibel	1	1	1	1	1	1	1	1	1	1	10
Service	1	1	1	1	1	1	1	1	1	1	10
sogar	1	0	1	1	1	1	1	1	1	1	9
sowieso	1	1	1	1	1	1	1	1	1	1	10
symbolisch	1	1	1	1	1	1	1	1	1	1	10
System	1	1	1	1	1	1	1	1	1	1	10
Tango	1	1	0	0	1	1	1	1	1	1	8
Teufel	1	1	1	1	1	1	1	1	1	1	10
Thema	1	0	0	0	0	1	1	1	1	1	6
tierisch	1	0	0	0	0	1	0	0	0	0	2
Toleranz	1	1	1	0	1	1	1	0	0	0	6
Tournee	1	1	1	1	1	1	1	1	0	1	9
typisch	1	1	1	1	1	1	1	0	1	1	9
Variante	1	1	1	1	1	1	1	1	1	1	10
vergleichbar	1	1	1	1	1	1	1	1	1	1	10
Verkauf	1	1	0	1	1	1	1	0	1	1	8
Verlag	1	1	1	0	1	1	1	1	1	1	9
vermuten	1	1	0	0	0	1	1	1	1	1	7
verrichten	1	1	0	0	0	1	1	1	1	1	7
Versuch	1	1	1	1	1	1	1	1	1	1	10
vertretbar	1	1	1	1	1	1	1	1	1	1	10
vorbei	1	0	0	1	1	1	1	1	1	1	8
vorhanden	1	1	1	0	1	1	1	1	1	1	9
wahrlich	1	0	0	0	0	1	1	0	0	0	3
wenig	1	1	1	1	1	1	1	1	1	1	10
wiederum	1	1	1	1	1	1	1	1	1	1	10
wieso	1	1	1	1	1	1	1	1	1	1	10
Wirklichkeit	1	1	1	0	1	1	1	1	0	0	7
Wirkung	1	1	1	1	1	1	1	1	1	1	10
wobei	1	1	1	1	1	1	1	1	1	1	10
Wohnung	1	1	1	1	1	1	0	1	1	1	9
worauf	1	1	1	1	1	1	1	1	1	1	10
wozu	1	1	1	1	1	1	1	1	1	1	10
wunderbar	1	1	1	1	1	1	1	1	1	1	10
zugute	1	1	1	1	1	1	1	1	1	1	10
Zuhause	1	1	1	1	1	1	1	1	1	1	10
Zulassung	1	1	1	1	1	1	1	1	1	1	10
zuviel	1	0	0	0	0	1	0	0	1	0	3
zuvor	1	1	1	1	1	1	1	1	1	1	10
zuwenig	1	1	1	1	1	1	1	1	0	0	8
zynisch	1	0	0	0	1	1	1	1	1	1	7
	157	**131**	**123**	**123**	**136**	**158**	**146**	**136**	**133**	**140**	**1383**

Appendix C

Text Partition of Kiel Read Speech Corpus

TABLE C.1: Text partition of Kiel Read Speech Corpus

Subset	Sentence ID
Training	be001; be002; be003; be004; be005; be007; be008; be009; be010; be011; be012; be016; be020; be021; be022; be023; be025; be026; be027; be028; be029; be030; be031; be033; be036; be037; be038; be039; be040; be041; be043; be045; be046; be047; be048; be049; be050; be051; be052; be054; be055; be056; be059; be060; be061; be062; be063; be065; be066; be069; be071; be072; be073; be074; be075; be076; be077; be078; be079; be080; be081; be082; be083; be084; be085; be086; be087; be088; be089; be092; be093; be095; be096; be097; be098; be099; be100; cn001; cn002; cn003; cn004; cn006; cn007; cn008; cn009; cn010; cn011; cn012; cn013; cn015; cn017; cn018; cn019; er001; er002; er004; er005; er007; er009; er010; er011; er012; er014; er015; er017; er018; er019; er020; er021; er023; er024; er025; er027; er030; er031; er033; er035; er036; er037; er038; er039; er040; er041; er042; er043; er044; er045; er046; er047; er049; er050; er051; er053; er054; er055; er056; er057; er058; er059; er061; er062; er063; er064; er066; er067; er068; er069; er074; er075; er077; er078; er079; er080; er081; er082; er084; er085; er086; er087; er088; er089; er090; er091; er092; er093; er094; er095; er096; er097; er098; er100; ko001; ko003; ko006; ko007; ko008; ko009; ko010; ko011; ko012; ko013; ko014; ko015; ko019; ko020; ko021; ko023; ko024; ko025; ko026; ko027; ko028; ko029; ko030; ko031; ko034; ko035; ko036; ko037; ko038; ko039; ko040; ko041; ko042; ko043; ko044; ko045; ko046; ko047; ko048; ko049; ko050; ko051; ko052; ko053; ko054; ko055; ko056; ko057; ko058; ko060; ko061; ko063;

Table C.1 continued from previous page

Subset	Sentence ID
Training	mr001; mr002; mr003; mr005; mr007; mr008; mr009; mr010; mr012; mr013; mr014; mr016; mr017; mr018; mr019; mr020; mr021; mr022; mr023; mr024; mr025; mr026; mr027; mr028; mr029; mr030; mr031; mr033; mr034; mr035; mr036; mr038; mr040; mr042; mr043; mr044; mr045; mr047; mr048; mr049; mr050; mr051; mr052; mr053; mr054; mr055; mr056; mr057; mr058; mr059; mr060; mr061; mr062; mr063; mr065; mr066; mr067; mr068; mr069; mr070; mr071; mr073; mr074; mr076; mr077; mr078; mr079; mr081; mr082; mr083; mr084; mr085; mr086; mr087; mr089; mr090; mr091; mr093; mr094; mr095; mr096; mr097; mr098; mr100; s1001; s1002; s1003; s1004; s1005; s1006; s1008; s1009; s1010; s1011; s1012; s1015; s1016; s1017; s1018; s1019; s1020; s1021; s1022; s1023; s1024; s1025; s1026; s1027; s1028; s1030; s1031; s1032; s1033; s1034; s1036; s1037; s1038; s1040; s1041; s1042; s1043; s1044; s1045; s2002; s2003; s2004; s2005; s2006; s2008; s2009; s2010; s2011; s2013; s2014; s2017; s2018; s2021; s2022; s2023; s2024; s2025; si001; si002; si003; si005; si006; si008; si009; si010; si011; si012; si013; si014; si015; si016; si017; si018; si019; si020; si021; si022; si023; si026; si027; si028; si029; si030; si031; si032; si033; si034; si035; si037; si040; si041; si042; si043; si044; si045; si047; si049; si050; si051; si052; si053; si055; si056; si057; si059; si061; si062; si063; si064; si065; si066; si067; si068; si070; si071; si072; si073; si074; si076; si077; si078; si079; si080; si082; si085; si086; si088; si089; si090; si091; si092; si093; si094; si095; si097; si098; si099; si100; sr001; sr002; sr004; sr005; sr007; sr010; sr011; sr012; sr013; sr014; sr015; tk002; tk003; tk005; tk006; tk008; tk010; tk012; tk013; tk015; tk016; tk017; tk018; tk021; tk022; tk023; tk024; tk025; tk026; tk027; tk028; tk029; tk030
Validation	be014; be017; be018; be032; be035; be044; be057; be090; be091; cn014; er003; er006; er013; er028; er034; er052; er060; er065; er071; er073; er076; ko002; ko004; ko018; ko022; ko033; ko059; ko062; mr011; mr032; mr039; mr041; mr046; mr072; mr075; mr088; mr092; mr099; s1007; s1014; s1035; s1039; s2001; s2015; s2016; si004; si007; si036; si038; si046; si048; si054; si060; si084; si087; si096; sr008; tk004; tk019; tk020
Test	be006; be013; be015; be019; be024; be034; be042; be053; be058; be064; be067; be068; be070; be094; cn005; cn016; cn020; er008; er016; er022; er026; er029; er032; er048; er070; er072; er083; er099; ko005; ko016; ko017; ko032; mr004; mr006; mr015; mr037; mr064; mr080; s1013; s1029; s2007; s2012; s2019; s2020; si024; si025; si039; si058; si069; si075; si081; si083; sr003; sr006; sr009; tk001; tk007; tk009; tk011; tk014

Appendix D

Detailed Recognition Results of Reproduced Oxford-805 and CAPL-1592 Utterances by Google and IBM ASR systems

TABLE D.1: Oxford-805 word list (the asterisk behind a word indicates that its reproduced utterance was correctly recognized by ASR systems).

according*	badly	balance*	banana*	baseball
basically*	basic	basis	basketball*	battle
beautiful	beauty	because*	become*	beginning*
begin	behave	behind*	being	belief
believe	belong*	below	benefit	between*
beyond*	bicycle*	billion	biology*	birthday*
biscuit*	body	bottle	bottom	bubble
budget*	building*	bullet	businessman*	business*
busy	button	calculate	cartoon*	ceiling
certainly*	certain*	chairman*	challenge*	champion*
channel*	cheerful*	chemical	chicken*	childhood
chocolate	cinema*	circle*	circumstance	citizen*
city	civil*	classical	classic	clearly
client	climate	closely	clothing	coffee
collapse	colleague	collection*	collect	college*
coloured	column	combination	comedy*	comfortable
comfort*	command*	comment*	commercial*	commission*
commitment*	committee	commit	commonly*	common*
communicate*	community*	company	compete	competition
competitive*	completely	complete	complex	dirty
disadvantage*	disappointed*	disappointing*	discipline*	discount
discussion*	discuss	disease	dishonest	dislike
dismiss	display	distance*	divide	division
divorced	document*	domestic	dominate	donate
double	download*	downstairs*	downwards	dozen

duty early earthquake* easily* eastern
easy economic* economy* edition edit
educated* educate education effectively effective
effect efficient effort eighteen* eighty
elderly* election* elect* element* elephant*
eleven* emergency* emerge* emotional* emotion*
emphasis emphasize* employee employ empty
enable* ending* enemy energy engaged
engage engine* enhance enjoy* enormous*
enough* entertainment* entertain* entirely* equally
equal escape especially essay essential*
establish estate estimate ethical evaluate*
evening* event* even evidence* evil*
exactly* exact examine* example* exam*
excellent* except* exchange* excited* excitement*
exciting* excuse* executive exercise* exhibition*
existence* exist* expand* expectation expected*
expect* expedition* expense* expensive* expert*
explain* explanation* explode* explosion export
expose* extend* extent* external* fuel
fully function fundamental* funding funny*
garden gentleman* gentle giant global*
goodbye government* govern guilty habit
handle happen happily happiness happy
hardly harmful headache headline healthy*
heating heaven heavily* heavy hello*
helpful* herself hesitate* highlight* highly*
himself hobby hockey holiday* hollow
holy homework honest hospital hotel*
household* housing* human* hunting husband*
ideal* idea* identify* identity* illegal*
illness* image imagine* immediate* impact*
impatient* imply importance* important* import
impose* impossible incident* include including*
income indeed* independent* indicate* indoors*
infection* influence* informal* information* inform
initially* initial* initiative injured innocent*
insect inside* insight* insist* install
instance* instead* institute institution* intelligence*
intelligent* intended* intend* intense* intention
internal* internet* interview* invention* invent
investigate* investment* invest* invitation* invite*
involved* involve* island* issue* item*
itself* jacket journalist* journal journey
judgement* july justice* justify* keyboard*
killing* kitchen* knowledge* label* lady
landscape* language* laptop largely latest

lazy	leadership*	leading	learning*	legal*
lemon	lesson	level*	licence*	lifestyle
likely	limited*	limit*	lion*	liquid
listen*	little*	lively*	living	local*
located*	locate*	location*	logical	lonely*
loudly*	lovely*	lucky*	machine*	modern
modify	moment*	monday*	money*	monitor*
monkey	morning*	mostly	motorcycle*	mountain
movement*	movie*	multiple	multiply	muscle
museum	musical*	musician	music	myself*
national*	nation*	native	nearly*	needle*
negative*	neighbourhood	nervous*	network	nevertheless*
nineteen*	ninety	nobody*	noisy*	normally
normal	northern	nothing*	notice	notion
novel	obey*	objective*	object	obligation
observation*	observe	obtain	obviously*	obvious*
occasion	ocean	offence	offend*	offensive*
office*	official*	often	ok*	online*
only*	opening*	open*	opinion*	opponent*
opposed	oppose	opposite	opposition	option
organized*	organize	organ	otherwise*	ourselves
outcome	outdoors	outline*	outside	oven
package	painful	painter	painting*	palace
panel	parking	parliament	participant	participate*
partly	party	passage	passion	passport
patient*	pattern	payment	peaceful	pencil
penny	pension	people	percentage*	perfectly
perfect*	performance*	perform	perhaps	permanent*
permission*	permit	personally*	personal*	person*
perspective	persuade*	phenomenon*	philosophy	photo
physical	physics	piano*	pilot	planet
planning	plastic*	platform	pleasant	plenty
pocket	poem	poet	pointed*	poisonous*
poison	policeman*	police*	policy	polite
political*	politician	politics	pollution	population
position*	positive*	possession	possess	possible*
possibly	potato*	potential*	poverty	powerful
psychologist*	psychology*	publication*	public	publish*
punishment*	punish	pupil	purchase*	purple
purpose	pursue	qualified*	qualify	quality
quantity	question*	quickly	quietly	quiet
quotation	sadly	safety	sailing	salad
sample	sandwich*	satellite	satisfied	satisfy
saturday*	saving	schedule*	science	scientific*
scientist	season	secondly*	second*	sector
selection*	select	sensible*	sensitive*	sentence
sequence	servant*	service*	session*	setting*

settle*
seventeen*
seventy*
seven*
sexual*
shadow*
shallow*
shiny
shooting
shopping*
signal*
significant*
silence
silent
silly*
similarly*
simple*
simply
singing
single*
situation*
sixteen*
sixty
skiing
slightly
slowly*
smoking*
social*
society*
solid
solution*
somebody*
someone*
something*
sometimes*
somewhat
southern
speaker
specialist*
special*
species
specifically*
specific
spelling*
spending*
spicy
spoken*
stable*
stadium*
standard*
statement
station*
statistic*
statue
status*
steady
sticky
stomach
student*
studio
study*
stupid*
subject
submit*
substance*
succeed
successfully*
successful*
success*
suddenly*
sudden*
suggestion*
suggest
suitable*
sunday*
supermarket*
supply
support*
suppose*
surely*
surface
survey
survive*
suspect*
swimming
symbol*
sympathy*
symptom
system*
tablet
table*
talented*
talent
target
taxi
teaching*
technical*
technique
technology*
teenage
telephone*
television*
tennis
themselves*
thinking
thirsty
thirteen*
thirty
thousand*
thursday*
ticket
tidy
tiny
tired
title
today
toilet
tomato
tonight*
topic
totally
total
towards*
towel
tuesday
tunnel
tv*
twenty
typically
typical*
ugly
ultimately*
unable*
uncle
uncomfortable*
unconscious*
understanding*
understand*
unemployed*
unemployment*
unexpected*
unhappy*
uniform*
union*
unique
united*
unit*
universe*
unknown*
unless*
unlikely
unlike
unpleasant*
until
unusual*
update
upon
upset*
upstairs*
upwards*
urban
useful*
usually*
usual*
vacation
valley*
valuable
value
vegetable*
vehicle*
venue
version
via
victim
video*
village*
violence
violent*
virtual*
vision
visit
visual
vital
vitamin
volume
warning
washing
weakness*
wealthy
weapon
website*
wedding*
wednesday*
weekend*
welcome*
western*
widely
wildlife*
willing
window*
within*
without*
witness
woman*
wonderful*
wooden
working
yellow*
yesterday*
yourself*

TABLE D.2: CAPL-1592 word list (the number behind each word indicates the number of base syllables of its reproduced utterance that were correctly recognized by ASR systems).

除了2	水果2	电视1	状况0	尽量2	困难2	从事2	争取0	赶紧1	导演2
本子0	有用1	发现2	加快1	纪念0	退休2	来自2	中医0	不但0	网站0
有限1	广告2	家具2	孩子2	手段0	只有0	地下1	景色2	走路0	突然0
适应1	入门1	还是2	上网1	将来2	快要0	调查0	外国0	保存1	相互1
生存2	人民0	主要0	沙发2	民族2	自己0	唱歌1	原料2	价钱0	国王0
大夫1	它们2	好事1	现在2	打开2	达到0	一块1	感谢0	飞机1	日报0
能够2	舒服2	半年1	专门1	着急1	任何0	关于0	假如0	配合2	医院0
无法0	便宜1	人数2	加强1	主人2	确保0	修改2	周围1	还有2	真的2
欢乐0	自主1	和平0	好像0	完善0	平时1	短期0	从前2	理解2	打球1
可能2	概括0	否认1	想到2	形成2	不用1	准备2	至少1	公园0	领先2
比例0	推进0	表示1	之内0	忘记0	形状2	基础1	图画0	退出0	下车0
必须2	前天2	车辆1	作为1	认真0	凉快1	学院2	米饭0	没用2	地图1
小时2	光明0	全国2	问路0	由于2	以下0	报纸1	观看0	分钟0	回到1
一样2	价值1	实验1	搞好0	红茶1	作文2	印象2	难道1	女生2	说话0
限制0	游泳1	使用0	内容0	字典0	打工0	出租0	卫生2	晚报0	先进0
京戏0	完美2	变成2	根本2	本事0	迎接2	青年1	特色0	原来2	睡觉2
适用1	获得0	早餐0	文化2	科研0	我们1	京剧0	发出1	汉语0	发生1
上次2	桌子2	好看2	市长2	公民0	快餐1	谈判0	干吗1	主任0	程度0
直接0	停车1	以为2	手续0	丰富2	成就1	革命2	开展1	相信2	实行1
旁边0	黑板0	时间0	必要2	小姐2	当时0	胜利1	国外2	部门1	例如1
上车1	使得1	晚会0	对待2	春天2	精神2	公众0	答应0	部长0	大人0
危机1	输入1	预防2	这些0	持续1	下课0	经过0	及时1	相机2	交给1
白酒0	花园2	结婚2	愿望1	方法0	基金1	基本0	状态0	外边2	受伤2
原因2	出生0	工业0	味道2	据说2	木头1	难以2	爱人0	起来1	政治1
责任0	极了1	大姐2	信号0	方便1	保留0	双方0	书店2	开会1	名字2
出现0	半夜1	前进2	喜欢2	指导0	节日0	之间0	队长1	一切1	认为1
合同0	水平1	词典1	照顾0	把握0	普及2	时代0	好听1	然而1	整天1
生产1	牌子1	排队2	那些2	思考1	不行2	缺少2	成立1	日常0	介绍0
加工1	刚才0	大小1	队员2	检查2	重新1	听说0	反复1	上街2	传说1
帮助0	失去0	见到0	形象2	许多2	等待1	男人2	从来1	大衣1	重复2
转变0	内地2	做法1	安排2	安全1	食物1	男生2	月亮2	强调0	实现2
成为0	车票0	分配2	消费2	新闻2	破坏0	经济1	后边0	出国1	下次2
防止0	一会2	当中0	优势2	什么2	动作0	国庆0	不安1	绿色0	前面0
武术1	过年0	火车0	帮忙0	小孩3	不光0	共有1	复印2	人工0	交易0
小声2	夜里2	预计0	改革1	安装2	另外2	玩具0	存款2	活动0	事情0
那时1	十分2	故事2	飞行2	性格0	护照1	这时1	老人2	走向0	坚持2
场所1	男友2	见过1	足够0	离开2	应用0	明天2	费用0	白天2	赶快0
承认2	上班2	生病0	整个2	南边2	今年2	合理2	晚饭0	手表2	药水2
前往0	劳动1	书包1	至今1	食品0	团结1	客人1	体会2	节能2	理由2
成功0	来到2	作业2	白色0	交流2	时候1	回去2	愿意2	早晨2	请客0
那么0	放下1	保护0	马路2	对于0	写作1	节约0	亲人1	文字2	上来2
要求2	难过2	杂志1	照相1	美元1	铁路0	打车0	华人2	分别2	文件2
随时2	包括0	教育2	表明1	东边2	处理2	不同0	并且2	是否2	高中1
以外0	声音2	学校2	人类1	绝对1	满意1	上午1	欢迎0	秋天0	电器1

显然0 选手1 近期0 关系2 任务0 推广0 这里1 路口0 支持1 进去0
最后2 团体0 政府2 建立2 爱情2 超过0 商业2 联合1 房子2 上课2
报名2 太阳2 体育0 出来2 简直1 比赛0 将近2 表演2 同事1 资金1
广大0 生日1 环境2 表达2 场合0 属于1 态度2 农村0 哈哈0 教师2
强大2 各地0 同意2 生动1 动物0 意外2 考验2 平安0 红色2 部分2
南方2 初中1 学习0 家乡2 面前1 上学2 忽然1 看病2 收看0 满足2
定期0 起飞0 精彩2 划船1 麻烦1 到底0 中午0 碰到0 座位0 开心2
儿女2 生活1 合适1 专业2 演员2 为了2 成员0 制定0 这样1 组成1
司机1 真正0 就是2 方式0 设备2 照片2 去世0 庆祝0 自然0 确定0
生气2 停止2 干杯0 改进1 正确0 制度1 文学1 地铁1 下面0 白人1
网络0 紧急2 成果0 年纪2 温暖2 后果0 一致1 动力1 不对2 法院1
一直0 天上1 游戏2 自身2 速度1 路线2 证据0 宣布0 民主0 正是1
规模2 发展2 一路1 熟人2 角度0 绿茶1 批评1 分数2 功夫1 人生0
今后1 目标0 规范2 广播0 请假2 实在1 中心1 主意1 集中0 剧场0
听见1 信息2 日子0 男孩1 决定0 黑人2 接受2 提出0 回家2 现实2
药片1 女性2 训练2 同时2 增长2 一般1 球鞋1 样子2 话剧0 大约2
影视2 放假0 试验1 动人2 制造0 爱心2 天空2 黑色2 作用0 觉得0
当做1 意义0 传播2 互相1 一共1 高速2 个性0 因此2 上周2 古代2
虽然2 球迷1 看法1 行人2 随便2 努力2 实习1 那样2 根据1 舞台2
适合2 结实0 自由0 准确2 美术2 后来1 开业0 短处1 北京0 请教1
就业1 放在1 商量2 负责2 放学2 世纪0 非常2 最好0 抓住0 此外1
特点0 暖和1 大概1 故意0 职业0 看来2 放心2 结果2 右边0 石头0
按照1 中国2 期间0 优点2 球场2 出去0 难受2 通知0 果然0 姑娘2
通信1 脑子2 坚决2 里面0 农业2 进步0 气候1 路上1 生命2 以前2
加油2 有时2 算是2 西北1 招手0 好久2 感觉0 行为2 小组2 明确1
少年2 专题1 年底2 等于0 公司1 可是1 回答2 接近0 假期2 保持0
管理0 中餐0 无数1 长处0 春节2 谈话0 记得0 达成2 作家2 之后1
志愿0 年初0 运动1 读书0 首先2 汉字2 练习2 流行2 邮局2 在家1
告别2 显得0 下边0 力量0 交警2 创新2 明显2 话题0 结合2 会员2
衣架2 老年2 送到0 全体0 草地0 老头1 身上2 中年1 推动0 项目2
国内2 改变0 下去0 目的2 出发1 进口1 发送0 类似2 眼前2 价格2
广场0 影片1 不管0 重点0 播放0 上升1 从此1 里边1 仅仅2 周年2
上边1 就要0 计算0 请问2 休息2 形式2 判断2 录音0 树林2 组合1
收到0 日期2 房屋1 石油1 决赛0 跟着2 同学2 其实2 确实1 好奇2
入口0 概念0 回来2 夏天2 有效2 离婚2 急忙1 打听0 预报1 坐下2
超级0 信任2 教练1 网友0 一起0 完整1 报道1 工夫1 接到2 建议0
后面2 没事2 机会2 世界0 多久2 说明2 组织0 赶到2 评价2 变化0
病人2 普遍0 谢谢2 自觉2 球拍1 继续2 点头0 子女1 多数2 军人1
西南2 今天2 总是0 亲自1 人口1 反正0 种子1 华语1 外语2 旅行2
心情1 之前0 住房1 得出1 合法1 一生2 收费1 相当1 会议2 外地2
送给0 成熟2 去年1 功课2 一点2 大学2 网上1 男子2 压力2 到达2
不要2 经历0 篮球2 咱们0 左边2 消息2 被子0 曾经0 体验1 吃饭2
中学2 肯定2 也许2 住院2 左右2 心中0 明年2 做到2 信心0 午饭1
认识1 交费2 气温0 应该2 目前2 听众1 全身1 只能1 留下2 研制1
常见1 海关0 打印1 东南1 自我2 义务0 例子0 冬天0 太太2 证明2
强烈1 这边0 明白2 自从1 有名1 取消2 变为1 白菜2 电脑2 记住0

会谈0	老师2	长大2	到处0	缺点1	事故0	范围1	发明0	进行2	整齐0
时期1	年代2	人才2	这么0	大妈2	多少2	实际0	现场1	但是0	代表2
办法1	前后0	楼上2	不断2	平等0	一半0	屋子1	开始2	上面1	容易0
调整0	只要1	了解2	不如0	商品1	同志0	当初2	一些0	危险1	具备0
进来1	直到0	相对1	知道0	下来2	只是0	旅客0	批准1	创作2	妈妈2
先生2	大象2	家电2	爸爸2	反对2	记者0	当然0	经常0	哪里2	收获1
公斤1	下午2	联系2	需要0	参加0	全面0	合作1	校园2	比如2	现金2
安静2	艺术0	可爱1	对方2	带来1	瓶子1	美好2	卫星2	夫人1	名单2
中级2	推开1	汽车1	里头1	依靠1	充满0	害怕0	下周2	理论0	全球1
整整0	先后2	叫做0	小心2	开发0	考生2	别人2	高级1	绝不0	东方1
文明1	认可1	大声2	地区0	节目1	农民0	对话2	星期1	指出1	土地0
感受1	必然0	女人2	普通1	短信0	健康0	不必2	等到0	能力2	增加2
有空1	故乡2	英文0	以上1	笑话2	错误0	具体1	告诉0	乐观0	新年1
比方2	筷子2	办理1	书架2	收入2	总统2	受到1	正式1	服务0	女孩2
幸运2	休假0	教学1	流利2	排球0	影响2	其它2	看见1	之下0	校长2
经验0	经理0	连续1	面包0	相同1	举办2	坚强2	得分2	奇怪2	歌手1
月份0	主题0	考试2	深入1	刚刚2	宣传2	不久2	地上2	进展1	颜色2
思想1	只好1	城市1	大量2	数量0	永远2	初步1	意见0	得到2	请坐0
随着2	学生2	亲切0	正常0	他们2	懂得1	北方2	改造2	成绩2	取得2
值得1	客观2	服装1	发表2	合格1	面积1	下班2	带领0	国际2	习惯2
主动2	效果1	经营0	父母0	高兴0	地球2	百姓0	身体1	作者2	行动1
观众2	球队0	内心1	加上1	常用0	对手2	常常2	收听1	开放2	挺好0
公布0	一定2	决心2	个人2	沙子1	公路1	怎么0	东西0	杯子2	教室2
生意1	游客1	大家2	重视0	创造0	找到2	突出2	无论2	消失1	东部0
主张2	手术2	道路0	实现0	积极0	采取1	顺利1	北边0	女士1	然后2
语言0	演出2	网球0	语音1	现代1	发言0	正好0	父亲1	纪录2	实力1
课堂0	讨论0	学费2	显示2	东北0	西部0	过去2	观察2	领导0	相似2
称为1	更加2	知识0	民间2	报告0	其次0	公平0	建设2	主席2	歌迷2
空气0	出门2	打破1	应当0	痛苦0	句子1	提前2	存在2	红酒2	因为2
观点2	通常0	志愿0	好玩0	专家1	作品2	著名2	所长0	晚安0	过去2
热烈0	不满2	其他0	高度1	家属2	不大0	年级2	可靠0	记录0	手指2
历史1	少数2	午餐0	著作1	特别0	才能1	好多2	功能0	一边1	设计2
有利1	好吃2	跳舞0	饺子1	草原1	因而0	分开0	老板1	皮包0	礼物0
前边0	同样0	全家2	马上2	西餐1	提高2	万一0	母亲0	西医0	总结2
阳光1	黄色0	哥哥0	比较2	午睡1	想起2	或者1	不仅2	早已2	行李2
公开2	接着2	早就2	共同0	电台2	事件1	不足2	产量1	对象0	职工0
女子2	以来1	国家2	头脑0	包子2	读者1	半天0	旅游0	非法2	浪费2
人家2	顾客0	电话0	之中0	军队2	金牌2	长期0	所有2	工作0	已经0
起床0	握手1	伟大2	中间2	分之0	皮鞋0	反应0	干净0	危害0	哪些2
集体0	直播0	各种0	面条0	方向2	主持2	有些0	好处0	干活0	本领0
否定0	中央1	以后2	歌声0	长城2	文章2	可以2	工人2	补充0	总理1
科学2	地方2	中文1	热情2	员工1	开车0	品种0	最近2	落后0	深刻2
院子2	体现0	确实0	传来2	需求0	复杂2	不过1	方面1	公共0	明星2
外交2	制作0	友好1	展开2	音乐0	技术0	终于2	地点1	西边1	底下0
快乐1	证件2	人员2	现象2	小学2	菜单2	提到0	发动0	本来1	克服2
北部0	风险2	背后1	快速2	市场0	楼下1	于是1	各自0	交通1	后天1

全部0 银行2 规定1 早饭0 关注2 举行2 大大2 传媒2 标题1 完成0
南部1 家里2 不少2 空调2 利用1 课程1 车主2 家庭2 研究2 女友2
跳远1 商人2 命运0 重大0 结束2 一片1 上去2 而且0 对面0 请进2
条件0 警察2 平常0 门票0 运输2 造成2 整理1 仍然0 奶奶2 小说2
环保0 有的1 关心2 商店1 人物2 家人2 昨天0 各位2 保证0 请求2
整体0 男性2 不错1 清楚1 机票2 表现0 同时1 饭店2 如何2 连忙2
见面0 改善0 英雄0 听到0 导致0 难度2 晚餐0 公里0 面对2 手机1
衣服2 彩色1 讲话1 名称0 武器1 别的1 漂亮2 一下2 交往0 伤害1
播出0 过程0 重要0 计划1 相比1 发达0 想法2 立刻2 群众0 产生1
生长2 有点2 钱包0 初级2 加入2 解决0 牛奶1 年轻2 早上2 足球2
那里2 英语2 多么1 幸福2 她们1 热爱0 保险2 全年1 始终0 具有0
不好2 标准2 参观1 房间2 得意1 跳高1 过来2 资格2 营养0 紧张0
招生2 此时0 之外0 千万0 西方0 掌握0 排名0 设立1 如果1 朋友0
意思2 自动2 声明0 工资1 没有2 你们1 创业2 解开2 完全2 全场1
带动0 所以1 时刻1 车站0 考察0 事实0 大使1 爱好1 出口1 情况0
温度2 商场0 社会2 超市0 保安0 正在2 风景2 理想1 窗子0 接待1
毛病0 通过0 数字0 不够2 情感1 怎样1 可怕0 简单2 伤心2 建成1
儿子2 选举0 医生1 晚上0 学者2 大众1 事业2 之一0 要是2 心里0
注意1 装修2 打算2 天气2 门口0 其中0 表面2 有钱2 希望0 进入0
不论1 依据0 成长0 鸡蛋0 访问1 感情0 院长2 外文0 感动0 感到0
预期0 再见2 从小0 电影0 老是2 班长0 机场0 观念0 机器0 道理1
画家0 往往0 外面0 护士2 睡着0 相关2 单位2 真实0 问题2 采用1
科技2 资料2 邮票1 那会1 媒体1 区别0 姐姐0 看到2 工具2 酒店2
人们1 笑话0

Appendix E

List of Publications

Refereed Journal Articles

1. Yingming Gao, Hongwei Ding, Peter Birkholz, and Yi Lin (2021). "Comparing fundamental frequency of German vowels produced by German native speakers and Mandarin Chinese learners". In: *The Journal of the Acoustical Society of America Express Letters* 1.7, p. 075203
2. Yingming Gao, Hongwei Ding, and Peter Birkholz (2020). "An acoustic comparison of German tense and lax vowels produced by German native speakers and Mandarin Chinese learners". In: *The Journal of the Acoustical Society of America Express Letters* 148.1, EL112–EL118
3. Ju Lin, Yingming Gao, Wei Zhang, Linxuan Wei, Yanlu Xie, and Jinsong Zhang (2020). "Improving pronunciation erroneous tendency detection with multimodel soft targets". In: *Journal of Signal Processing Systems* 92.8, pp. 793–803
4. Ju Lin, Wei Li, Yingming Gao, Yanlu Xie, Nancy F Chen, Sabato Marco Siniscalchi, Jinsong Zhang, and Chin-Hui Lee (2018). "Improving Mandarin tone recognition based on DNN by combining acoustic and articulatory features using extended recognition networks". In: *Journal of Signal Processing Systems* 90.7, pp. 1077–1087

Refereed Publications in Proceedings

1. Wenjie Peng, Yingming Gao, Binghuai Lin, and Jinsong Zhang (2021). "A practical way to improve automatic phonetic segmentation performance". In: *Proceedings of the 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, pp. 1–5
2. Yingming Gao, Xinyu Zhang, Yi Xu, Jinsong Zhang, and Peter Birkholz (2020). "An investigation of the target approximation model for tone modeling and recognition in continuous Mandarin speech". In: *Proceedings of the Interspeech 2020*, pp. 1913–1917
3. Wang Dai, Jinsong Zhang, Yingming Gao, Wei Wei, Dengfeng Ke, Binghuai Lin, and Yanlu Xie (2020b). "Formant tracking using dilated convolutional networks through dense connection with gating mechanism". In: *Proceedings*

of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020. Ed. by Helen Meng, Bo Xu, and Thomas Fang Zheng. ISCA, pp. 150–154

4. Yingming Gao, Peter Steiner, and Peter Birkholz (2020). "Articulatory copy synthesis using long-short term memory Networks". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*. Ed. by Ronald Böck, Ingo Siegert, and Andreas Wendemuth. TUDpress, Dresden, pp. 52–59
5. Yingming Gao, Simon Stone, and Peter Birkholz (2019). "Articulatory copy synthesis based on a genetic algorithm". In: *Proceedings of the Interspeech 2019*, pp. 3770–3774
6. Yingming Gao, Hongwei Ding, Peter Birkholz, Rainer Jäckel, and Yi Lin (2019). "Perception of German tense and lax vowel contrast by Chinese learners". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 25–32
7. Konstantin Sering, Niels Stehwien, Yingming Gao, Martin V. Butz, and Harald Baayen (2019). "Resynthesizing the GECO speech corpus with VocalTractLab". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 95–102
8. Susanne Drechsel, Yingming Gao, Jens Frahm, and Peter Birkholz (2019). "Modell einer Frauenstimme für die artikulatorische Sprachsynthese mit VocalTractLab". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 239–246
9. Yingming Gao and Peter Birkholz (2018). "Speaking rate changes affect phone durations differently for neutral and emotional speech". In: *Proceedings of the EUSIPCO 2018, 26th European Signal Processing Conference*. IEEE, pp. 2070–2074
10. Longfei Yang, Yanlu Xie, Yingming Gao, and Jinsong Zhang (2017). "Improving pronunciation erroneous tendency detection with convolutional long short-term memory". In: *Proceedings of the International Conference on Asian Language Processing (IALP)*. IEEE, pp. 52–56

Bibliography

1. Afshan, Amber and Prasanta Kumar Ghosh (2015). "Improved subject-independent acoustic-to-articulatory inversion". In: *Speech Communication* 66, pp. 1–16.
2. Aryal, Sandesh and Ricardo Gutierrez-Osuna (2013). "Articulatory inversion and synthesis: Towards articulatory-based modification of speech". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 7952–7956.
3. Atal, Bishnu S. and Lawrence Rabiner (1976). "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.3, pp. 201–212.
4. Bachu, Rajesh G. et al. (2010). "Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy". In: *Advanced techniques in computing sciences and software engineering*. Springer, pp. 279–282.
5. Back, Thomas, Ulrich Hammel, and H-P Schwefel (1997). "Evolutionary computation: Comments on the history and current state". In: *IEEE transactions on Evolutionary Computation* 1.1, pp. 3–17.
6. Bäck, Thomas and Hans-Paul Schwefel (1993). "An overview of evolutionary algorithms for parameter optimization". In: *Evolutionary computation* 1.1, pp. 1–23.
7. Badin, Pierre et al. (2010a). "Can you 'read' tongue movements? Evaluation of the contribution of tongue display to speech understanding". In: *Speech Communication* 52.6, pp. 493–503.
8. Badin, Pierre et al. (2010b). "Visual articulatory feedback for phonetic correction in second language learning". In: *L2SW, Workshop on Second Language Studies: Acquisition, Learning, Education and Technology*, pp. 1–10.
9. Ben-Youssef, Atef, Hiroshi Shimodaira, and David A Braude (2014). "Speech driven talking head from estimated articulatory features". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4573–4577.
10. Bernhardt, Barbara et al. (2003). "Speech habilitation of hard of hearing adolescents using electropalatography and ultrasound as evaluated by trained listeners". In: *Clinical Linguistics & Phonetics* 17.3, pp. 199–216.

11. Beyer, Hans-Georg and Hans-Paul Schwefel (2002). "Evolution strategies: A comprehensive introduction". In: *Natural Computing* 1.1, pp. 3–52.

12. Birkholz, Peter (2005). "3D-Artikulatorische Sprachsynthese". PhD thesis. Universität Rostock.

13. — (2013). "Modeling consonant-vowel coarticulation for articulatory speech synthesis". In: *PloS One* 8.4, e60603.

14. Birkholz, Peter, Susanne Drechsel, and Simon Stone (2019). "Perceptual optimization of an enhanced geometric vocal fold model for articulatory speech synthesis." In: *Proceedings of the Interspeech 2019*, pp. 3765–3769.

15. Birkholz, Peter, Bernd J Kröger, and Christiane Neuschaefer-Rube (2011). "Model-based reproduction of articulatory trajectories for consonant-vowel sequences". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.5, pp. 1422–1433.

16. Birkholz, Peter, Ingmar Steiner, and Stefan Breuer (2007). "Control concepts for articulatory speech synthesis". In: *Proceedings of the 6th ISCA Workshop on Speech Synthesis, Bonn, Germany*, pp. 5–10.

17. Birkholz, Peter et al. (2017). "Manipulation of the prosodic features of vocal tract length, nasality and articulatory precision using articulatory synthesis". In: *Computer Speech & Language* 41, pp. 116–127.

18. Boersma, Paul and David Weenink (2019). "Praat: Doing phonetics by computer (version 6.1.16) [Computer program]". In: http://www.praat.org/ (Last viewed 20 July 2020).

19. Browman, Catherine P. and Louis Goldstein (1992). "Articulatory phonology: An overview". In: *Phonetica* 49.3-4, pp. 155–180.

20. Bulla, Lukas (2017). *Automatische Erstellung Gestischer Partituren Für Einen Artikulatorischen Sprachsynthesizer (in English: Automatic generation of gestural scores for an articulatory speech synthesizer)*. Studienarbeit, TU Dresden.

21. Cao, Wen and Jinsong Zhang (2009). "The construction of a CAPL Chinese interlanguage corpus and its annotation". In: *Applied Linguistics* 4, pp. 122–131.

22. Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, pp. 1724–1734.

23. Ciregan, Dan, Ueli Meier, and Jürgen Schmidhuber (2012). "Multi-column deep neural networks for image classification". In: *IEEE conference on computer vision and pattern recognition*. IEEE, pp. 3642–3649.

24. Dai, Chengkai et al. (2020a). "Planning jerk-optimized trajectory with discrete time constraints for redundant robots". In: *IEEE Transactions on Automation Science and Engineering* 17.4, pp. 1711–1724.

25. Dai, Wang et al. (2020b). "Formant tracking using dilated convolutional networks through dense connection with gating mechanism". In: *Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*. Ed. by Helen Meng, Bo Xu, and Thomas Fang Zheng. ISCA, pp. 150–154.

26. Dang, Jianwu and Kiyoshi Honda (2001). "A physiological articulatory model for simulating speech production process". In: *Acoustical Science and Technology* 22.6, pp. 415–425.

27. — (2002). "Estimation of vocal tract shapes from speech sounds with a physiological articulatory model". In: *Journal of Phonetics* 30.3, pp. 511–532.

28. Dave, Namrata (2013). "Feature extraction methods LPC, PLP and MFCC in speech recognition". In: *International Journal for Advance Research in Engineering and Technology* 1.6, pp. 1–4.

29. Dekel, Ofer et al. (2012). "Optimal distributed online prediction using minibatches." In: *Journal of Machine Learning Research* 13.1, 165–202.

30. Drechsel, Susanne et al. (2019). "Modell einer Frauenstimme für die artikulatorische Sprachsynthese mit VocalTractLab". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 239–246.

31. Dusan, Sorin and Li Deng (2000). "Acoustic-to-articulatory inversion using dynamical and phonological constraints". In: *Proceedings of the 5th Seminar on Speech Production*. Citeseer, pp. 237–240.

32. Elie, Benjamin and Yves Laprie (2016). "Copy-synthesis of phrase-level utterances". In: *Proceedings of the EUSIPCO 2016, 24th European Signal Processing Conference*. IEEE, pp. 868–872.

33. Ellbogen, Tania, Florian Schiel, and Alexander Steffen (2004). "The BITS speech synthesis corpus for German". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Vol. 47. 45. Lisbon, Portugal: European Language Resources Association (ELRA), p. 40.

34. Fairee, Suthida, Booncharoen Sirinaovakul, and Santitham Prom-on (2015). "Acoustic-to-articulatory inversion using Particle Swarm Optimization". In: *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, pp. 1–6.

35. Fang, Qiang (2020). "Improving the performance of acoustic-to-articulatory inversion by removing the training loss of noncritical portions of articulatory channels dynamically". In: *Proceedings of the Interspeech 2020*, pp. 1371–1375.

36. Felps, Daniel, Christian Geng, and Ricardo Gutierrez-Osuna (2012). "Foreign accent conversion through concatenative synthesis in the articulatory domain".

In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.8, pp. 2301–2312.

37. Frankel, Joe and Simon King (2001). "ASR - articulatory speech recognition". In: *Proceedings of the EUROSPEECH 2001, 7th European Conference on Speech Communication and Technology, 2nd Interspeech Event, Aalborg, Denmark, September 3-7, 2001*. Ed. by Paul Dalsgaard et al. ISCA, pp. 599–602.

38. Fu, Szu-Wei et al. (2019). "Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement". In: *Proceedings of the International Conference on Machine Learning*. PMLR, pp. 2031–2041.

39. Fukushima, Kunihiko and Sei Miyake (1982). "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". In: *Competition and Cooperation in Neural Nets*. Springer, pp. 267–285.

40. Galantucci, Bruno, Carol A. Fowler, and Michael T. Turvey (2006). "The motor theory of speech perception reviewed". In: *Psychonomic Bulletin & Review* 13.3, pp. 361–377.

41. Ganchev, Todor, Nikos Fakotakis, and George Kokkinakis (2005). "Comparative evaluation of various MFCC implementations on the speaker verification task". In: *Proceedings of the SPECOM*. Vol. 1. 2005, pp. 191–194.

42. Gao, Yingming and Peter Birkholz (2018). "Speaking rate changes affect phone durations differently for neutral and emotional speech". In: *Proceedings of the EUSIPCO 2018, 26th European Signal Processing Conference*. IEEE, pp. 2070–2074.

43. Gao, Yingming, Hongwei Ding, and Peter Birkholz (2020). "An acoustic comparison of German tense and lax vowels produced by German native speakers and Mandarin Chinese learners". In: *The Journal of the Acoustical Society of America Express Letters* 148.1, EL112–EL118.

44. Gao, Yingming, Peter Steiner, and Peter Birkholz (2020). "Articulatory copy synthesis using long-short term memory Networks". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*. Ed. by Ronald Böck, Ingo Siegert, and Andreas Wendemuth. TUDpress, Dresden, pp. 52–59.

45. Gao, Yingming, Simon Stone, and Peter Birkholz (2019). "Articulatory copy synthesis based on a genetic algorithm". In: *Proceedings of the Interspeech 2019*, pp. 3770–3774.

46. Gao, Yingming et al. (2019). "Perception of German tense and lax vowel contrast by Chinese learners". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 25–32.

47. Gao, Yingming et al. (2020). "An investigation of the target approximation model for tone modeling and recognition in continuous Mandarin speech". In: *Proceedings of the Interspeech 2020*, pp. 1913–1917.

48. Gao, Yingming et al. (2021). “Comparing fundamental frequency of German vowels produced by German native speakers and Mandarin Chinese learners”. In: *The Journal of the Acoustical Society of America Express Letters* 1.7, p. 075203.

49. Gerazov, Branislav et al. (2021). *Evaluating features and metrics for high-quality simulation of early vocal learning of vowels*. arXiv: 2005.09986 [eess.AS].

50. Ghahremani, Pegah et al. (2014). “A pitch extraction algorithm tuned for automatic speech recognition”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 2494–2498.

51. Ghosh, Prasanta Kumar and Shrikanth S Narayanan (2011). “A subject-independent acoustic-to-articulatory inversion”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4624–4627.

52. Hill, David R., Craig R. Taube-Schock, and Leonard Manzara (2017). “Low-level articulatory synthesis: A working text-to-speech solution and a linguistic tool”. In: *The Canadian Journal of Linguistics/La revue canadienne de linguistique* 62.3, pp. 371–410.

53. Hinton, Geoffrey E et al. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv: 1207.0580 [cs.NE].

54. Hiroya, Sadao and Masaaki Honda (2004). “Estimation of articulatory movements from speech acoustics using an HMM-based speech production model”. In: *IEEE Transactions on Speech and Audio Processing* 12.2, pp. 175–185.

55. Hirst, Daniel (2011). “The analysis by synthesis of speech melody: From data to models”. In: *Journal of Speech Sciences* 1.1, pp. 55–83.

56. Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural Computation* 9.8, pp. 1735–1780.

57. Hogan, Neville (1984). “Adaptive control of mechanical impedance by coactivation of antagonist muscles”. In: *IEEE Transactions on Automatic Control* 29.8, pp. 681–690.

58. Hopfield, John J. (1982). “Neural networks and physical systems with emergent collective computational abilities”. In: *Proceedings of the national academy of sciences* 79.8, pp. 2554–2558.

59. Howard, Ian S. and Peter Birkholz (2018). “Using state feedback to control an articulatory synthesizer”. In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2018*, pp. 351–358.

60. — (2019). “Modelling vowel acquisition using the Birkholz synthesizer”. In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pp. 304–311.

61. Howard, Ian S. and Mark A. Huckvale (2005). “Training a vocal tract synthesiser to imitate speech using distal supervised learning”. In: *Proceedings of the SpeCom: 10th International Conference on Speech and Computer*. Vol. 2. University of Patras, Wire Communications Laboratory, pp. 159–162.

62. Howard, Ian S. and Piers Messum (2011). "Modeling the development of pronunciation in infant speech acquisition". In: *Motor Control* 15.1, pp. 85–117.

63. Howard, Ian Spencer and Piers Ruston Messum (2007). "A computational model of infant speech development". In: *XII International Conference "Speech and Computer" (SPECOM'2007)*, pp. 756–765.

64. Huang, Xuedong et al. (2001). *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR.

65. Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *Proceedings of the International Conference on Machine Learning*. PMLR, pp. 448–456.

66. Ittichaichareon, Chadawan, Siwat Suksri, and Thaweesak Yingthawornsuk (2012). "Speech recognition using MFCC". In: *International Conference on Computer Graphics, Simulation and Modeling*, pp. 135–138.

67. Jaitly, Navdeep and Geoffrey E Hinton (2013). "Vocal tract length perturbation (VTLP) improves speech recognition". In: *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language*. Vol. 117.

68. Janikow, Cezary Z. and Zbigniew Michalewicz (1991). "An experimental comparison of binary and floating point representations in genetic algorithms." In: *ICGA*, pp. 31–36.

69. Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar (2021). "A review on genetic algorithm: past, present, and future". In: *Multimedia Tools and Applications* 80.5, pp. 8091–8126.

70. Kawahara, Hideki and Masanori Morise (2011). "Technical foundations of TANDEM-STRAIGHT, a speech analysis, modification and synthesis framework". In: *Sadhana* 36.5, pp. 713–727.

71. Kawahara, Hideki et al. (2008). "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation". In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 3933–3936.

72. Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A method for stochastic optimization*. arXiv: 1412.6980 [cs.LG].

73. Kisler, Thomas, Uwe Reichel, and Florian Schiel (2017). "Multilingual processing of speech via web services". In: *Computer Speech & Language* 45, pp. 326–347.

74. Klatt, Dennis H. (1975). "Voice onset time, frication, and aspiration in word-initial consonant clusters". In: *Journal of speech and hearing research* 18.4, pp. 686–706.

75. Kohler, Klaus J, Benno Peters, and Michel Scheffers (2018). *The Kiel corpus of spoken German: Read and spontaneous speech.*

76. Krug, Paul Konstantin et al. (2021). "TargetOptimizer 2.0: Enhanced estimation of articulatory targets". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2021*. Ed. by Stefan Hillmann et al. TUDpress, Dresden, pp. 145–152.

77. Kuhl, Patricia K. (2004). "Early language acquisition: Cracking the speech code". In: *Nature Reviews Neuroscience* 5.11, pp. 831–843.

78. Ladefoged, Peter and Keith Johnson (2014). *A course in phonetics*. Cengage Learning.

79. Laprie, Yves et al. (2013). "Articulatory copy synthesis from cine X-ray films". In: *Proceedings of the Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, August 25-29, Lyon, France, Proceedings*, pp. 2024–2028.

80. Liberman, Alvin M and Ignatius G Mattingly (1985). "The motor theory of speech perception revised". In: *Cognition* 21.1, pp. 1–36.

81. Lin, Ju et al. (2018). "Improving Mandarin tone recognition based on DNN by combining acoustic and articulatory features using extended recognition networks". In: *Journal of Signal Processing Systems* 90.7, pp. 1077–1087.

82. Lin, Ju et al. (2020). "Improving pronunciation erroneous tendency detection with multi-model soft targets". In: *Journal of Signal Processing Systems* 92.8, pp. 793–803.

83. Ling, Zhen-Hua, Korin Richmond, and Junichi Yamagishi (2012). "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.1, pp. 207–219.

84. Ling, Zhen-Hua et al. (2009). "Integrating articulatory features into HMM-based parametric speech synthesis". In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6, pp. 1171–1185.

85. Liu, Peng et al. (2015). "A deep recurrent approach for acoustic-to-articulatory inversion". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4450–4454.

86. Ma, Xi et al. (2018). "Emotion recognition from variable-Length speech segments using deep learning on spectrograms." In: *Proceedings of the Interspeech 2018*, pp. 3683–3687.

87. Maeda, Shinji (1990). "Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model". In: *Speech Production and Speech Modelling*. Springer, pp. 131–149.

88. Mitra, Vikramjit et al. (2010). "Retrieving tract variables from acoustics: a comparison of different machine learning strategies". In: *IEEE Journal of Selected Topics in Signal Processing* 4.6, pp. 1027–1045.

89. Mitra, Vikramjit et al. (2013). "Articulatory trajectories for large-vocabulary speech recognition". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 7145–7149.

90. Mitra, Vikramjit et al. (2014). "Articulatory features from deep neural networks and their role in speech recognition". In: *2014 IEEE international Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3017–3021.

91. Mitra, Vikramjit et al. (2017). "Joint modeling of articulatory and acoustic spaces for continuous speech recognition tasks". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5205–5209.

92. Mittag, Gabriel and Sebastian Möller (2020). "Deep learning based assessment of synthetic speech naturalness". In: *Proceedings of the Interspeech 2020*, pp. 1748–1752.

93. Müller, Meinard (2007). *Information retrieval for music and motion*. Vol. 2. Springer.

94. Murakami, Max et al. (2015). "Seeing [u] aids vocal learning: Babbling and imitation of vowels using a 3D vocal tract model, reinforcement learning, and reservoir computing". In: *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, pp. 208–213.

95. Nam, Hosung et al. (2004). "TADA: An enhanced, portable Task Dynamics model in MATLAB". In: *The Journal of the Acoustical Society of America* 115.5, pp. 2430–2430.

96. Nam, Hosung et al. (2010). "A procedure for estimating gestural scores from natural speech". In: *Proceedings of the Interspeech 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. Ed. by Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura. ISCA, pp. 30–33.

97. Nam, Hosung et al. (2012). "A procedure for estimating gestural scores from speech acoustics". In: *The Journal of the Acoustical Society of America* 132.6, pp. 3980–3989.

98. Narayanan, Shrikanth et al. (2014). "Real-time magnetic resonance imaging and electromagnetic articulography database for speech production research (TC)". In: *The Journal of the Acoustical Society of America* 136.3, pp. 1307–1311.

99. Pagliarini, Silvia, Arthur Leblois, and Xavier Hinaut (2020). "Vocal imitation in sensorimotor learning models: A comparative review". In: *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1.

100. Panchapagesan, Sankaran and Abeer Alwan (2011). "A study of acoustic-to-articulatory inversion of speech by analysis-by-synthesis using chain matrices and the Maeda articulatory model". In: *The Journal of the Acoustical Society of America* 129.4, pp. 2144–2162.

101. Paszke, Adam et al. (2019). “PyTorch: An imperative style, high-performance deep learning library”. In: *Advances in Neural Information Processing Systems*, pp. 8024–8035.

102. Peng, Wenjie et al. (2021). “A practical way to improve automatic phonetic segmentation performance”. In: *Proceedings of the 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, pp. 1–5.

103. Perrier, Pascal and Susanne Fuchs (2015). “Motor equivalence in speech production”. In: *The handbook of speech production*, p. 225.

104. Philippsen, Anja (2021). “Goal-directed exploration for learning vowels and syllables: a computational model of speech acquisition”. In: *KI-Künstliche Intelligenz* 35.1, pp. 53–70.

105. Philippsen, Anja, Felix Reinhart, and Britta Wrede (2015). “Efficient bootstrapping of vocalization skills using active goal babbling”. In: *International Workshop on Speech Robotics at Interspeech 2015*.

106. Philippsen, Anja Kristina, René Felix Reinhart, and Britta Wrede (2014). “Learning how to speak: Imitation-based refinement of syllable production in an articulatory-acoustic model”. In: *Proceedings of the 4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, pp. 195–200.

107. — (2016). “Goal babbling of acoustic-articulatory models with adaptive exploration noise”. In: *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, pp. 72–78.

108. Picone, Joseph W. (1993). “Signal modeling techniques in speech recognition”. In: *Proceedings of the IEEE* 81.9, pp. 1215–1247.

109. Prom-on, Santitham, Peter Birkholz, and Yi Xu (2013). “Training an articulatory synthesizer with continuous acoustic data.” In: *Proceedings of the Interspeech 2013*, pp. 349–353.

110. — (2014). “Identifying underlying articulatory targets of Thai vowels from acoustic data based on an analysis-by-synthesis approach”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2014.1, pp. 1–11.

111. Prom-On, Santitham, Yi Xu, and Bundit Thipakorn (2009). “Modeling tone and intonation in Mandarin and English as a process of target approximation”. In: *The Journal of the Acoustical Society of America* 125.1, pp. 405–424.

112. Radeck-Arneth, Stephan et al. (2015). “Open source German distant speech recognition: Corpus and acoustic model”. In: *International Conference on Text, Speech, and Dialogue*. Springer, pp. 480–488.

113. Rao, K. Sreenivasa and Anil Kumar Vuppala (2014). *Speech processing in mobile environments*. Springer.

114. Reichel, Uwe D. (2012). “PermA and Balloon: Tools for string alignment and text processing”. In: *Proceedings of the Interspeech 2012, 13th Annual Conference of the International Speech Communication Association*, pp. 1874–1877.

115. Reichel, Uwe D. and Thomas Kisler (2014). "Language-independent grapheme-phoneme conversion and word stress assignment as a web service". In: *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2014*, pp. 42–49.

116. Richmond, Korin (2002). "Estimating articulatory parameters from the acoustic speech signal". PhD thesis. University of Edinburgh.

117. — (2006). "A trajectory mixture density network for the acoustic-articulatory inversion mapping". In: *Proceedings of the Interspeech 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*. ISCA, pp. 577–580.

118. — (2007). "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion". In: *Advances in Nonlinear Speech Processing, International Conference on Non-Linear Speech Processing, NOLISP 2007, Paris, France, May 22-25, 2007, Revised Selected Papers*. Ed. by Mohamed Chetouani et al. Vol. 4885. Lecture Notes in Computer Science. Springer, pp. 263–272.

119. Richmond, Korin, Phil Hoole, and Simon King (2011). "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus". In: *Proceedings of the Interspeech 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, pp. 1505–1508.

120. Richmond, Korin, Zhenhua Ling, and Junichi Yamagishi (2013). "On the evaluation of inversion mapping performance in the acoustic domain". In: *Proceedings of the Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. Ed. by Frédéric Bimbot et al. ISCA, pp. 1012–1016.

121. — (2015). "The use of articulatory movement data in speech synthesis applications: An overview—application of articulatory movements using machine learning algorithms—". In: *Acoustical Science and Technology* 36.6, pp. 467–477.

122. Rix, Antony W. et al. (2001). "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings*. IEEE, pp. 749–752.

123. Rudzicz, Frank, Aravind Kumar Namasivayam, and Talya Wolff (2012). "The TORGO database of acoustic and articulatory speech from speakers with dysarthria". In: *Language Resources and Evaluation* 46.4, pp. 523–541.

124. Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536.

125. Sering, Konstantin et al. (2019). "Resynthesizing the GECO speech corpus with VocalTractLab". In: *Studientexte zur Sprachkommunikation: Elektronische*

Sprachsignalverarbeitung 2019. Ed. by Peter Birkholz and Simon Stone. TUDpress, Dresden, pp. 95–102.

126. Shahrebabaki, Abdolreza Sabzi et al. (2019). "A phonetic-level analysis of different input features for articulatory inversion". In: *Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. Ed. by Gernot Kubin and Zdravko Kacic. ISCA, pp. 3775–3779.

127. Shi, Xingjian et al. (2015). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*, pp. 802–810.

128. Shitov, Denis, Elena Pirogova, and Margaret Lech (2018). "Assessment of features for neurocomputational modeling of speech acquisition". In: *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, pp. 1–5.

129. Shitov, Denis Dmitrievich (2020). "Computational speech acquisition for articulatory synthesis". PhD thesis. RMIT University.

130. Sivaraman, Ganesh et al. (2015). "Analysis of coarticulated speech using estimated articulatory trajectories". In: *Proceedings of the Interspeech 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, pp. 369–373.

131. Sivaraman, Ganesh et al. (2016). "Vocal tract length normalization for speaker independent acoustic-to-articulatory speech inversion." In: *Proceedings of the Interspeech 2016*, pp. 455–459.

132. Sorokin, Victor N. (2006). "Speech inversion: Problems and solutions". In: *Dynamics of Speech Production and Perception* 374, pp. 263–282.

133. Steiner, Ingmar and Korin Richmond (2009). "Towards unsupervised articulatory resynthesis of German utterances using EMA data". In: *Proceedings of the Interspeech 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. ISCA, pp. 2055–2058.

134. Steiner, Ingmar Michael Augustus (2010). "Observations on the dynamic control of an articulatory synthesizer using speech production data". PhD thesis. Saarland University.

135. Sun, Yifan and Xihong Wu (2020). *Embodied self-supervised learning by coordinated sampling and training*. arXiv: 2006.13350 [eess.AS].

136. Taal, Cees H. et al. (2011). "An algorithm for intelligibility prediction of time–frequency weighted noisy speech". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7, pp. 2125–2136.

137. Tan, Ke and DeLiang Wang (2018). "A convolutional recurrent neural network for real-time speech enhancement". In: *Proceedings of the Interspeech 2018, 19th*

Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018. Ed. by B. Yegnanarayana. ISCA, pp. 3229–3233.

138. Tang, Kit-Sang et al. (1996). "Genetic algorithms and their applications". In: *IEEE Signal Processing Magazine* 13.6, pp. 22–37.

139. Tiede, Mark et al. (2017). "Quantifying kinematic aspects of reduction in a contrasting rate production task". In: *The Journal of the Acoustical Society of America* 141.5, pp. 3580–3580.

140. Tiwari, Vibha (2010). "MFCC and its applications in speaker recognition". In: *International Journal on Emerging Technologies* 1.1, pp. 19–22.

141. Tourville, Jason A. and Frank H. Guenther (2011). "The DIVA model: A neural theory of speech acquisition and production". In: *Language and Cognitive Processes* 26.7, pp. 952–981.

142. Ulbrich, Christiane and Horst Ulbrich (2007). "Realisations and alternations in German /r/-realisation". In: *Proceedings of the Interspeech 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*. ISCA, pp. 2733–2736.

143. Uria, Benigno, Steve Renals, and Korin Richmond (2011). "A deep neural network for acoustic-articulatory speech inversion". In: *Proceedings of the NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*. Sierra Nevada, Spain.

144. Uria, Benigno et al. (2012). "Deep architectures for articulatory inversion". In: *Proceedings of the Interspeech 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, pp. 867–870.

145. Wang, Xiaoou, Thomas Hueber, and Pierre Badin (2014). "On the use of an articulatory talking head for second language pronunciation training: the case of Chinese learners of French". In: *10th International Seminar on Speech Production (ISSP 2014)*, pp. 449–452.

146. Westbury, John R., Greg Turner, and J Dembowski (1994). "X-ray microbeam speech production database user's handbook". In: *University of Wisconsin*.

147. Wrench, Alan (1999). *The MOCHA-TIMIT articulatory database*. URL: `https://www.cstr.ed.ac.uk/research/projects/artic/mocha.html`.

148. Wright, Alden H. (1991). "Genetic algorithms for real parameter optimization". In: *Foundations of Genetic Algorithms*. Vol. 1. Elsevier, pp. 205–218.

149. Wright, Stephen J. (2015). "Coordinate descent algorithms". In: *Mathematical Programming* 151.1, pp. 3–34.

150. Wu, Bin et al. (2016). "The construction of a Chinese interlanguage corpus". In: *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*. IEEE, pp. 183–187.

151. Xu, Anqi, Peter Birkholz, and Yi Xu (2019). "Coarticulation as synchronized dimension-specific sequential target approximation: An articulatory synthesis simulation". In: *Proceedings of The 19th International Congress of Phonetic Sciences (ICPhS)*, pp. 205–209.

152. Yang, Longfei et al. (2017). "Improving pronunciation erroneous tendency detection with convolutional long short-term memory". In: *Proceedings of the International Conference on Asian Language Processing (IALP)*. IEEE, pp. 52–56.

153. Zhang, Le and Steve Renals (2008). "Acoustic-articulatory modeling with the trajectory HMM". In: *IEEE Signal Processing Letters* 15, pp. 245–248.

154. Zhu, Pengcheng, Lei Xie, and Yunlin Chen (2015). "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings". In: *Proceedings of the Interspeech 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, pp. 2192–2196.

155. Zorila, Tudor-Catalin, Varvara Kandia, and Yannis Stylianou (2012). "Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression". In: *Proceedings of the Interspeech 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, pp. 635–638.