Doctoral theses at NTNU, 2022:198

Abdolreza Sabzi Shahrebabaki

Articulatory Inversion for Speech Technology Applications

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abdolreza Sabzi Shahrebabaki

Articulatory Inversion for Speech Technology Applications

Thesis for the Degree of Philosophiae Doctor

Trondheim, June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

ISBN 978-82-326-6629-4 (printed ver.)
ISBN 978-82-326-6324-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:198

Printed by NTNU Grafisk senter

Abstract

Within the past decades advances in neural networks have improved the performance of a vast area of speech processing applications including the articulatory inversion problem which is concerned with estimating the vocal tract shape in the form of articulators' position based on the uttered speech. In spite of these improvements the articulatory inversion problem still needs improvements in order to be further utilized for other speech application as a complementary source of information. Articulatory measurements have been employed in various applications such as speech synthesis, computer aided pronunciation training and automatic speech recognition. Measuring articulator movements requires complex procedures and systems, which makes it impossible to perform measurements outside of labs. There are databases containing a limited number of speakers which have synchronously recorded articulator movements and uttered speech.

This thesis explores the articulatory inversion problem within different scenarios where there are mismatches between training data and test data. These mismatches include speaker mismatches within a database or across databases, mismatches in the speaking rate of speakers, and mismatches in the environment where the data are synthetically created by incorporating various noises.

The first part of the thesis focus on incorporating linguistic information such as forced aligned phonemic features, attribute features based on manner and place of articulation, and their combination with the acoustic features. Furthermore, new architectures are developed based on the acoustic landmarks theory which tells that abrupt changes in the speech spectrum are the results of changes in the articulators' configuration. Later on, transfer learning of articulatory information based on phonemic features is utilized to generate articulatory trajectories for the TIMIT database. Phone recognition experiments provide evidence of the effectiveness of the proposed transfer learning approach. Furthermore, a novel architecture is proposed to estimate articulatory

trajectories directly from the time domain speech signal by utilizing 1D convolutional filters. The 1D convolutional layers extract features and the decimation operators match the sampling rate of acoustic signal with the articulatory measurements' sampling rate. The data driven features extracted by 1D convolutional layers are better able to capture and compensate the variability resulted by mismatch in the speaking rates.

In the second part of the thesis the focus is on articulatory inversion performance in noisy conditions. Synthetically produced noisy acoustic data are used for this experiment evaluation. Speech enhancement based on deep neural networks prior to the articulatory inversion trained on clean data, slightly outperforms the articulatory inversion system trained on multi-condition noisy data. We propose a joint network which performs both speech enhancement and articulatory inversion. The articulatory inversion part of the joint model outperforms the trained model on multi-condition noisy data in the low signal to noise ratio range, namely 0, 5 and 10 dB. The estimated articulatory data are further used to train a word recognition system trained on clean acoustic and articulatory features for the WSJ dataset. For the noisy condition, the word error rate of the recognition system trained on both acoustic and articulatory data is significantly less than the model trained only on the clean acoustic data.

Preface

The thesis is submitted to the Norwegian University of Science and Technology (NTNU) for the partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The most of the doctoral work has been performed at the Department of Electronic Systems, NTNU, Trondheim, Norway. The work has been conducted under the supervision of Professor Torbjørn Svendsen from January 2016 to April 2020.

Professor Sabato Marco Siniscalchi was supervising me during my two months visiting from October 2019 to December 2019 at the Kore university of Enna, Enna, Sicily, Italy.

Six peer-reviewed scientific papers are the core of the dissertation which I have collaborated as the first author. I hope this work inspire others to investigate the applicability of articulatory data for other speech processing technologies, as the articulatory system is the common tool among human to produce speech and communicate.

Trondheim, March 2022
Abdolreza Sabzi Shahrebabaki

Acknowledgment

It feels like a few days ago when I started my PhD at Norwegian University of Science and Technology (NTNU) on the 18th of January 2016, in a different land far away from my home country, but meeting kind and supportive people made it an enjoyable environment.

First of all, I would like to thank my supervisor, Professor Torbjørn Karl Svendsen. His knowledge, trust and constant encouragement helped me a lot to reach this state for my PhD research. I would like to thank him for his valuable time, feedback and comments which he spent and provided on my research activities. Apart from his professional support, his personality and kindness made the PhD duration very joyful and memorable.

Besides my supervisor, I would like to thank Associate Professor Magne Hallstein Johnsen for his constant support, care and livening up the social life in the department.

I would also like to extend my thanks to Professor Sabato Marco Sinischalci for his knowledge, support and friendship during my visiting period at Kore univerity of Enna, Enna, Sicily.

Working at NTNU was very cheerful and vibrant, due to the nice friends, colleagues and faculty staffs. I would like to thank them all, specially Negar, Hamed, Ehsan, Jacob, Zala, Lahiru , Reza, Ali, Ashkan, Reinold, Giampiero, Stefan, Pierluigi, Randi, Kirsten, Nina and Erik.

I owe many thanks to Negar and Hamed, for assisting me to settle in Trondheim, and brightening up life outside the PhD through different social gatherings and activities. It would have been much harder without their care, kindness and consideration.

I owe my deepest appreciation to my parents, Maman and Baba, and my siblings Mohammadreza, Alireza, Farideh, Sedigheh and Gholamreza, for their priceless efforts and supports.

Lastly, I am grateful to my girlfriend, Shabnam, for her understanding, kindness, unlimited support and encouragement during my PhD study. The PhD journey came through with her help and support.

Trondheim, March 2022
Abdolreza Sabzi Shahrebabaki

Contents

Abstract **i**

Preface **iii**

Acknowledgment **v**

Contents **vii**

List of Figures **xi**

1 Introduction **1**
1.1 Objective of this study 3
1.2 Thesis Contributions 3
1.2.1 List of publications 5
1.2.2 Papers Not Included in the Thesis 5
1.3 Organization of the Thesis 6

2 Background **9**
2.1 Speech production 9
2.2 Articulatory phonology, manner and place of articulation 10
2.3 Articulatory parameters 12
2.3.1 Physical measurements 12
2.3.2 Types of articulatory features 14
2.4 Speech analysis, speech information representation . . . 15
2.4.1 Acoustic representation 16
2.4.2 Phonemic representation 16
2.5 Machine learning for AAI 16
2.5.1 Feed-forward deep neural network 19
2.5.2 Recurrent deep neural network 21

2.5.3 1D convolutional neural network 22
2.5.4 Temporal convolutional neural network 23
2.5.5 Transfer learning 24
2.6 Performance measurements 25
2.7 Application of AAI 26

3 Contributions of the thesis **27**
3.1 Deep architecture for acoustic/phonemic articulatory inversion 27
3.1.1 Exploring linguistic features together with acoustic features for the AI 28
3.1.2 1D-CNN feature extraction for the AAI 31
3.2 Transfer learning of AAI 33
3.2.1 Articulatory estimation for TIMIT by using the $f_{\text{AAI-base}}$ 35
3.2.2 Articulatory estimation for TIMIT by using the f_{PAI} 36
3.2.3 Teacher-student technique for training the $f_{\text{AAI-stud}}$ 36
3.2.4 Experimental results 37
3.3 AAI from speech waveforms 39
3.3.1 Articulatory estimation from time domain signal 40
3.3.2 Experiments and results 41
3.4 Robust AAI 44
3.4.1 Speech enhancement prior to the AAI 45
3.4.2 Joint training of DNN-SE and AAI system . . . 48
3.4.3 Experimental setups and results 50
3.4.4 ASR in noisy condition 53

4 Conclusion and Future work **55**
4.1 Conclusion 55
4.2 Future work 56

Bibliography **57**

Articles **71**

Paper A **73**

A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion 73

Paper B **79**

Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals 79

Paper C **85**

Transfer learning of articulatory information through phone information 85

Paper D **91**

A DNN Based Speech Enhancement Approach to Noise Robust Acoustic-to-Articulatory Inversion 91

Paper E **93**

On Robust Deep Learning Approaches for Joint Optimization of Speech Enhancement and Articulatory Inversion . . . 93

Paper F **107**

Raw Speech-to-Articulatory Inversion by Temporal Filtering and Decimation 107

List of Figures

2.1 Articulatory system of human from the lungs to the lips. . 10
2.2 Block diagram of a DNN-AAI system. 20
2.3 A long short term memory cell 21
2.4 A bidirectional RNN. 22
2.5 1D convolutional layer with kernel size 3. The arrows show which samples are used in the convolution. The solid arrows from input show samples for dilation rate equal to 1, and dashed arrows show the samples for the dilation rate equal to 2. 23
2.6 Stacking of 1D causal convolutional layers to build a TCN with kernel size 2, dilation rates [1, 2, 4, 8]. The lines with arrows show which samples are used in the convolution to estimate the output sample. The dashed lines are showing the estimation procedure for previous and future output samples. 24

3.1 Deep neural architectures which have mostly used in the AAI systems. 28
3.2 Average RMSE for manner of articulation from estimated trajectory by different input features. 29
3.3 Layer-wise feature representation by 1D convolutional layers for the AAI task. The 1×1, 1×3, and ..., are representing the convolution kernel shape. 32
3.4 FBE features for utterance "The birch canoe slid on the smooth planks."and the resulted convolutional feature maps for the 1st, 2nd and 3rd layers. 33
3.5 AF feature for fricative and PHN features for phoneme /ɜ/ (——) and channel output from the 1st 1D-CNN layer (- - - -) and 2nd 1D-CNN layer (—·—·). 34

3.6 Block diagram of the proposed transfer learning method from the HPRC to the TIMIT database, and knowledge distillations from phonemic features to acoustic features through articulatory space. Dashed arrows correspond to no training. 36
3.7 TV trajectories from f_{PAI}, $f_{\text{AAI-base}}$, and $f_{\text{AAI-stud}}$ for utterance "She slipped and sprained her ankle on the steep slope." 38
3.8 The sequence-to-sequence AAI systems, employing (top) handcrafted features, (middle) extracted features from speech frames by 1D-CNN, (bottom) extracted features from the whole speech sequence by 1D-CNN and decimation layers. 41
3.9 Frequency response of first layer of convolutional layers for extracting information from raw speech waveform. 43
3.10 DNN based SE system with 120 ms context of noisy LPSs, and clean LPSs and MFCCs as the output. 48
3.11 Network structure of joint training of SE and AAI systems 49
3.12 Average PCC for multi-condition data with respect to different SNR levels. The box plots represent the minimum, first quartile, median, third quartile, and the maximum of average PCC values. 51
3.13 Average PCC for multi-condition data on AAI-C and AAI-MC models, with respect to different noise types. 51

Abbreviations

AAI	acoustic-to-articulatory inversion.
AI	articulatory inversion.
ANN	artificial neural network.
ASR	automatic speech recognition.
BLSTM	bidirectional long short-term memory.
CALL	computer-aided language learning.
CAPT	computer-assisted pronunciation training.
CNN	convolutional neural network.
DBLSTM	deep bidirectional long short-term memory.
DCT	discrete cosine transform.
DNN	deep neural network.
DNN-AAI	deep neural network based acoustic-to-articulatory inversion.
DRMDN	deep recurrent mixture density network.
EMA	electromagnetic articulography.
FBE	Log scaled Mel filterbank energies.
GMR	Gaussian mixture model regression.
LPS	Log power spectra.
LSF	line spectral frequencies.
LSTM	long short-term memory.
MDN	mixture density network.
MFCC	Mel frequency cepstral coefficient.

MSE	minimum mean square error.
MSE	mean square error.
RBM	restricted Boltzmann machine.
ReLU	rectified linear unit.
RNN	recurrent neural network.
rt-MRI	real-time magnetic resonance imaging.
SD	speaker dependent.
SE	speech enhancement.
SI	speaker independent.
SR	speaker rate.
STFT	short-time Fourier transform.
TCN	temporal convolutional network.
TTS	text to speech.
WER	word error rate.
XRMB	X-Ray microbeam.

CHAPTER 1

Introduction

The human speech production mechanism starts from the lungs, which pushes the air through the human vocal tract to the acoustic environment. The human vocal tract has two main paths for the flow of air, namely, the oral cavity and the nasal cavity, and several parts, namely, vocal folds, which handle controlling the air flow, the palate, the tongue, teeth and lips that constrict the air flow. These parts are known as articulators. The different activation and constriction levels of articulators are the actual cause of the different sounds that are made by humans. For a major part of human sounds, air is conducted through the oral cavity, except for a few of them known as the nasals, where the air flows through the nasal cavity. Human speech is the result of a sequence of articulators' gestures that are smoothly varying over time. This sequence produces information in terms of a sequence of different sounds which carry different information. The smallest linguistic units in speech are known as phonemes. Therefore, the speech signal has acoustic information and phonemic information, which are the results of the complex human speech production system.

The problem of going back from the uttered speech to the articulatory movements is known as speech inversion or articulatory inversion (AI). As discussed earlier, the speech signal has acoustic and phonemic information, which can be used for the inversion problem. In applications where only acoustic information is available, e.g., automatic speech recognition (ASR), the inversion problem is referred as acoustic-to-articulatory inversion (AAI), and in applications where only textual information or phonemic information is available, e.g. text to speech (TTS), and in applications when both phonemic and acoustic infor-

mation are available, e.g. computer-aided language learning (CALL) and computer-assisted pronunciation training (CAPT), the inversion problem is referred as articulatory inversion (AI).

AAI has been an active area in the speech processing field for the past few decades. It is a highly nonlinear mapping function or regression [56, 57], and it suffers from non-uniqueness [36, 50, 56] that means the same acoustic sound can be produced with more than one unique articulator configuration.

There are several issues with the current research for the AAI or AI problem. The conducted research in the literature mainly focuses on speaker dependent (SD) AAI scenarios, where the system is trained for one specific speaker, whose articulatory measurements are available, and evaluated on the same speaker. Speaker independent (SI) scenarios have been investigated in the matched speakers condition, where training is done for several speakers and tested for one of the speakers in the training set. The available measured articulatory data are mostly for normal speaking rate and to the best of the author's knowledge there is only one work where they have investigated articulatory inversion for different speaking rates (SR) [28]. SR variation is one of the challenges in the speech processing application, and mismatched SR between training and evaluation data degrades the performance of systems. Except for the work in [70], all the researches in the AAI problem are conducted in clean conditions which is a shortcoming in terms of applicability of these systems for real-world applications. In real-world applications, there are conditions which affect the performance of the system, e.g., environmental noises, far-field or near-field microphone recordings, differences in microphone frequency response, etc.

In this thesis, we have tried to address these issues by suggesting new architectures and strategies. We used deep neural networks with novel architectures to improve the AI problem for SI scenarios. For mismatched SR, we performed the AAI from the time domain signal with novel fully convolutional layers which outperformed the performance of the state-of-the-art methods. At the end, we evaluated the AAI in presence of noise, and exploited deep neural network-based speech enhancement prior to the AAI, which significantly improved the performance for low signal to noise ratios (SNR).

1.1 Objective of this study

The aim of this study is to identify issues that hinders the applicability of articulatory inversion generated information and to propose techniques for removing or reducing the influence of these issues. Furthermore, to explore and analyze the applicability of estimated articulatory information in other speech technology applications. As it is mentioned earlier in this chapter, there are challenges in AAI towards real-world applications. (i) The first challenge is due to the limited number of speakers with recorded articulatory data. The mismatch between test and training speakers reduces the accuracy of the estimated articulatory trajectories. For coping with this shortcoming, new regression models need to be developed for the speaker independent AAI systems. (ii) Another source of mismatch is due to the different recording setups, e.g., different datasets, which causes degradation of AAI system performance. Tackling this issue can be done with some information which is not directly related to the acoustic signal, e.g., phonemic information. In this way, the speaker and dataset mismatches can be compensated. (iii) Speaking rate mismatch has a significant effect on the AAI system performance. The articulators' movements will be different for different speaking rate, which can explain the drop in performance. For tackling this issue, a new regression model with data driven features would be helpful. (iv) Another challenge in the AAI problem is mismatch in the acoustic environment. In real-world scenarios, the environment contains various sources of noise which degrades the performance of an AAI system trained on clean data. To compensate for the effect of noise, enhancement of noisy data would reduce the performance degradation of AAI system. The proposed approaches to cope with the possible issues (except SR variability) were evaluated by conducting several automatic speech recognition (ASR) experiments.

1.2 Thesis Contributions

In section 3.1, we propose approaches to deal with the issues identified in 1.1 using linguistic information along with the acoustic features in Paper A[75] to deal with issue (i). Furthermore, we use the 1D convolutional layers to extract related features with linguistic information

from the acoustic signal, for use in scenarios where only acoustic information is available , Paper B [78]. Both proposed architectures improve the AI system performance for SD and SI scenarios.

In section 3.2, the second issue (ii) is improved by transfer learning of articulatory information in the source domain through phonemic features, and then using a knowledge distillation-based teacher-student method to learn the articulatory information from acoustic information in the target domain, Paper C [76]. This method is evaluated by using the estimated articulatory features in an ASR task, which show learning articulatory features by the proposed method is more informative than estimating them by AAI system trained on the source domain.

In section 3.3, the mismatch between speaker rates (iii), is improved by utilizing 1D convolutional layers to extract the features from time domain speech signal, Paper F [79]. Instead of framing of speech signal to get the same sampling rate of articulatory data, decimation is used by strided convolution and pooling layers. The pooling is performed with overlap which results in non-uniform sampling of extracted features. The proposed method performs like the baseline in the matched speaking rate scenarios and outperforms the baseline in the mismatched scenarios.

In section 3.4, to deal with the last issue (iv) which is AAI in presence of noise, a deep speech enhancement network is employed in Paper D [73]. Earlier research has shown there is no gain by using speech enhancement based on signal processing methods, as a preprocessing module to the AAI trained on clean data, and a multi-condition trained AAI is needed. In our work we justify their claim, and then show that deep speech enhancement is helpful to improve the performance of AAI trained on clean data for low signal-to-noise ratios (SNRs). In the next step, we propose a model which jointly optimize the network parameters to perform both enhancement and inversion tasks , in Paper E[77]. The performance increases significantly for all the SNRs. We evaluate the performance of estimated articulatory trajectories by conducting several ASR experiments. The ASR systems trained on both clean acoustic and articulatory data are performing better in noisy scenarios compared to the systems which are trained on only acoustic data. The estimated articulatory trajectories from the proposed joint model improves the WER of ASR experiments compared to the available

baselines.

1.2.1 List of publications

All the papers listed below are outcomes of the research work carried out by the author of this dissertation. This includes 6 published papers.

Paper A: [75] **A. S. Shahrebabaki**, N. Olfati, A. S. Imran, S. M. Siniscalchi and T. Svendsen. (2019) "A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion." Proc. Interspeech 2019, 3775-3779.

Paper B: [78] **A. S. Shahrebabaki**, S. M. Siniscalchi, G. Salvi and T. Svendsen. (2020) "Sequence-to-Sequence Articulatory Inversion Through Time Convolution of Sub-Band Frequency Signals." Proc. Interspeech 2020, 2882-2886.

Paper C: [76] **A. S. Shahrebabaki**, N. Olfati, S. M. Siniscalchi, G. Salvi and T. Svendsen. (2020) "Transfer Learning of Articulatory Information Through Phone Information." Proc. Interspeech 2020, 2877-2881.

Paper D: [73] **A. S. Shahrebabaki**, S. M. Siniscalchi, G. Salvi and T. Svendsen. A DNN Based Speech Enhancement Approach to Noise Robust Acoustic-to-Articulatory Inversion. 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5.

Paper E: [77] **A. S. Shahrebabaki**, G. Salvi , T. Svendsen and S. M. Siniscalchi. "Acoustic-to-Articulatory Mapping with Joint Optimization of Deep Speech Enhancement and Articulatory Inversion Models." in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 135-147, 2022.

Paper F: [79] **A. S. Shahrebabaki**, S. M. Siniscalchi and T. Svendsen. Raw Speech-to-Articulatory Inversion by Temporal Filtering and Decimation. Proc. Interspeech 2021, pp.1184-1188.

1.2.2 Papers Not Included in the Thesis

Paper 1: [71] **A. S. Shahrebabaki**, N. Olfati, A. S. Imran and T. Svendsen. (2018) "Acoustic Feature Comparison for Differ-

ent Speaking Rates." In: Kurosu M. (eds) Human-Computer Interaction. Interaction Technologies. HCI 2018. Lecture Notes in Computer Science, vol 10903. Springer, Cham.

Paper 2: [72] **A. S. Shahrebabaki**, N. Olfati, A. S. Imran and T. Svendsen. "A Comparative Study of Deep Learning Techniques on Frame-Level Speech Data Classification." Circuits, Systems, and Signal Processing 38, 3501–3520 (2019).

Paper 3: [29] A. S. Imran, V. Haflan, **A. S. Shahrebabaki**, N. Olfati and T. Svendsen. "Evaluating Acoustic Feature Maps in 2D-CNN for Speaker Identification." In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, pp. 211-216. 2019.

Paper 4: [30] A. S. Imran, **A. S. Shahrebabaki**, N. Olfati and T. Svendsen. "A Study on the Performance Evaluation of Machine Learning Models for Phoneme Classification." In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, pp. 52-58. 2019.

Paper 5: [74] **A. S. Shahrebabaki**, N. Olfati, A. S. Imran, M. H. Johnsen, S. M. Siniscalchi and T. Svendsen, "A Two-Stage Deep Modeling Approach to Articulatory Inversion," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6453-6457, doi: 10.1109/ICASSP39728.2021.9413742.

1.3 Organization of the Thesis

The dissertation is in a paper collection format, which consists of six technical articles.

In chapter 2, background of articulatory inversion is presented. A brief description of speech production mechanism is presented, followed by articulatory phonology. In addition, the measuring techniques for articulators' movement are described, and available datasets with the articulatory measurements are presented. At the end of chapter 2, the regression techniques for AAI problem are mentioned, and the mostly used deep learning techniques are presented in more details. Chapter 3, is the collection of papers which are the outcome of this dissertation. In section 3.1, the papers which have tried to deal with the

mismatch in speakers (issue (i)), are presented with the main results and conclusions. In section 3.2, the paper for dealing with issues (i) and (ii) in a cross-dataset scenarios is presented. The baseline system, and proposed transfer learning and knowledge distillation approach are described, and results are evaluated based on ASR system performance using articulatory data. In section 3.3, the proposed method for dealing with issue (iii) is presented. The architecture for extracting features from time domain signal is described, and the results for various scenarios are presented. In section 3.4, the papers which explored AAI in presence of noise, are presented. The data preparation, preprocessing, and training steps are described in detail. The proposed method performance is evaluated based on objective metric and ASR performance in the form of WER. The last chapter concludes and suggests potential future directions. Finally, in the second part of the thesis, the research articles which are the scientific contribution of the dissertation are presented.

CHAPTER 2

Background

2.1 Speech production

The human speech production mechanism is quiet complex. The whole mechanism of speech production is controlled by human's brain. The production related muscles get constricted, with neural signals from the brain. Air flows through the glottis by means of lungs and further through the oral cavity or nasal cavity. The vocal folds affect the flow of air by vibration when making the voiced sounds, or by being relaxed having no effect on the airflow when making the unvoiced sounds. The velum, movements of tongue, teeth and lips filter the air stream and produce different sounds. Figure 2.1 visualizes the speech production from the lungs to the vocal tract. From now on, we consider the vocal tract part of the entire system as the intended articulators to explore. The speech waveform contains both acoustic and linguistic information. Different combinations of articulator gestures result in different sounds, called phonemes. In the production of each phoneme, articulators play critical, dependent and redundant roles [31]. The critical articulator plays a vital role in the production of a phone by significantly moving from its natural state. The dependent articulator follows changes that are imposed by movement of the critical articulator, due to the bio-mechanical correlation between them. The redundant articulator movement does not affect the phone's production.

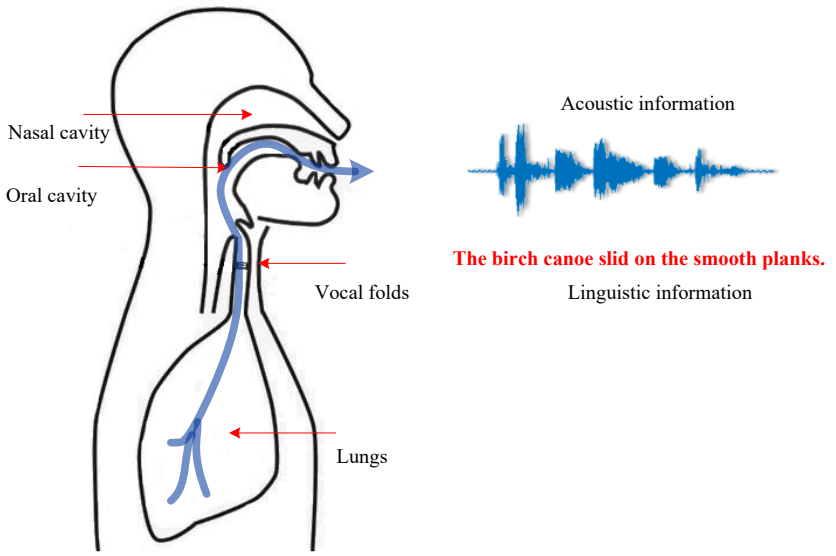


Figure 2.1: Articulatory system of human from the lungs to the lips.

2.2 Articulatory phonology, manner and place of articulation

The articulatory phonetics or articulatory attributes describe the process of articulation to make the speech sounds. It describes the speech sounds in terms of the articulators involved in their production. The distinctive English speech sounds can be described by the manner of articulation, place of articulation together with the voicing [33].

Manner of articulation describes the configuration adopted by the articulators in articulating a sound. For the consonant sounds, there are five main types [13]:

- Plosives: sounds in whose articulation the airstream is stopped by a brief closure of two speech organs and then released in a quick burst.
- Fricatives: sounds in whose articulation two speech organs narrow the airstream, causing friction to occur as it passes through.
- Affricates: sounds in whose articulation the airstream is stopped as for a plosive and then released slowly and partially with friction.

- Nasals: sounds in whose articulation the airstream is diverted through the nasal cavity as a consequence of the passage through the oral cavity being blocked by the lowering of the soft palate, or velum
- Approximants: sounds in whose articulation two speech organs approach each other and air flows continuously between them without friction.

For the vowel sounds, there are two main descriptors:

- Tongue height: The height of the closest part of the tongue to the palate describes the manner of articulation for vowel sounds as, *open* when the tongue is low in the mouth, *close* when the tongue is high, and when the tongue height is between the low and high points of the tongue are referred by *half-close* and *half-open*.
- Lip posture: The lips can be rounded or spread in various degrees to produce different vowels.

Place of articulation describes the consonants sounds in terms of where the constriction is happened in the vocal tract to restrict the air stream flow. The main types of place of articulation are as follows [13]:

- Alveolar: sounds made by the contact of tongue to the alveolar margin right behind the upper front teeth.
- Bilabial: sounds made with both lips by stopping the air stream.
- Dental: sounds made by the tongue tip contact against the upper front teeth.
- Glottal: sounds made by air stream passing the glottis when the vocal cords are closed or narrowed.
- Palatal: sounds made by tongue blade contact to the hard palate.
- Palato-alveolar: sounds made by tongue contact to the hard palate and alveolar margin.
- Post-alveolar: sounds made by tongue contact behind the alveolar margin.
- Velar: sounds made by tongue back contact to the soft palate.

For the vowel sounds, place of articulation is not very precise as the vocal tract does not constrict as much as consonant sounds. Due to that, the place of articulation for vowels describes which part of tongue is closer to the palate, as follows:

- Front: sounds made by the tongue tip rising towards the palate.
- Middle: sounds made by the tongue blade rising towards the palate.
- Back: sounds made by the tongue rare rising towards the palate.

2.3 Articulatory parameters

2.3.1 Physical measurements

There are various methods to measure the articulator movements, X-ray microbeam (XRMB) [100], electromagnetic articulography (EMA) [67], and real-time magnetic resonance imaging (rt-MRI).

- In the XRMB method, several gold pellets are placed at different articulators in the vocal tract and their movements are photographed by X-ray to obtain the articulators' trajectories. In this method, the pellets are on the midsagittal plane to track the significant movements of articulators, which are along the midsagittal and vertical axes. Audio is recorded simultaneously during the measurements of articulators' movements.
- EMA is a commonly used technique for measuring the articulators' movements. In this method, the electromagnetic coils are placed along the vocal tract in the midsagittal place to measure movements of the articulators. In addition, there are a few sensors as reference points to correct the head movements. The audio data is recorded simultaneously with the sensors' movements.
- Rt-MRI technique was employed in [55] to record high resolution videos with a low frame rate which results in a low temporal resolution. The audio signal is recorded during the MRI imaging which results in noisy speech recordings.

The XRMB and EMA methods has a higher temporal resolution for the articulators' movements compared to the rt-MRI method, and the recorded audio are less noisy in contrast with the rt-MRI.

2.3.1.1 Available EMA databases

There are several available speech corpora with EMA measurements. They will briefly be described in the following.

MOCHA-TIMIT: The Multi-channel Articulatory (MOCHA) database [102] consists of speech data and EMA data recorded simultaneously for one male and one female subject speaking British English. The EMA sampling rate is 500 Hz, and the speech sampling frequency is 16 kHz. The speakers were asked to utter 460 English sentences which cover a wide range of phonological and prosodic contexts.

MNGU0: The MNGU0 [62] database contains 1,263 utterances spoken by a single British speaker. The database contains parallel EMA data and acoustic data. Each EMA data frame is a 12-dimensional vector. Each dimension corresponds to an x- or y-coordinate of a coil attached in the midsagittal plane of the speaker's articulator and there are 6 coils in total.

USC-TIMIT: The USC-TIMIT database [54] consist of 460 sentences which were used in the MOCHA-TIMIT database. There are four speakers available, two female and two male native American speakers. Three sensors were attached to the tongue tip, midline and rear. Three other sensors were placed at the lower lip, upper lip and surface of the lower incisor. Moreover, three reference sensors were placed to the nasal bridge and behind right and left ears. The EMA sensors' sampling rate is 100 Hz and the recorded trajectories were smoothed by a low-pass filter with bandwidth 20 Hz. The audio signals were recorded with the sampling frequency of 44.1 kHz and were downsampled to 16 kHz.

HPRC: Haskins production rate comparison (HPRC) database [90] is also known as IEEE-EMA database. It contains recordings for eight native American English speakers, four female (F01-F04) and four male (M01-M04) speakers. There are 720 spoken utterances available in the dataset with both normal and fast speaking rate where the sentences are taken from the IEEE sentences [63]. For some of the normal speaking rate utterances, there are a few repetitions available. Speech waveforms are sampled at the rate of 44.1 kHz, and synchronously EMA recordings are available at a sampling rate of 100 Hz. EMA recordings are obtained from eight sensors, which record position of tongue rear (dorsum) (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and

jaw left (JAWL). The articulatory measurements are corrected for head movements and aligned to the occlusal plane in X, Y and Z directions, corresponding to movements from posterior to anterior, right to left and inferior to superior, respectively. The movements along the Y axis carry limited information and we thus only employed the measured data along X and Z axis.

2.3.2 Types of articulatory features

The EMA data consist of measured movements of various articulators recorded in the midsagittal plane. These EMA measurements are the articulators' movements in three-dimensional space. The movements range depend on the speaker's anatomy.

There are other types of articulatory features that are less dependent on speaker anatomy, which can be obtained from the EMA measurements. The features proposed in [32, 82] and are obtained by applying several geometrical transformations to the EMA measurements. These features are called tract variables (TV). TVs are relative measures and suffer less from non-uniqueness [48]. We use nine TVs, including lip aperture (LA), lip protrusion (LP), jaw angle (JA), tongue rear constriction degree (TRCD), tongue rear constriction location (TRCL). For TB and TT, we similarly calculate TBCD, TBCL, TTCD and TTCL, as explained below. The geometrical transformations are defined as follows.

$$\mathrm{LA}[n] = \sqrt{\Big(\mathrm{LL}_x[n] - \mathrm{UL}_x[n]\Big)^2 + \Big(\mathrm{LL}_z[n] - \mathrm{UL}_z[n]\Big)^2}, \tag{2.1}$$

$$\mathrm{LP}[n] = \mathrm{LL}_x[n] - \underset{m \in \text{allutterances}}{\text{median}} \mathrm{LL}_x[m]. \tag{2.2}$$

LA represents the distance between LL and UL sensors. LP is defined as the movement of LL from its median position in the X direction,

$$\mathrm{JA}[n] = \sqrt{\Big(\mathrm{JAW}_x[n] - \mathrm{UL}_x[n]\Big)^2 + \Big(\mathrm{JAW}_z[n] - \mathrm{UL}_z[n]\Big)^2}, \tag{2.3}$$

is defined as the distance between the JAW and UL sensors.

For each of the tongue sensors TR, TB and TT, two TVs are defined. Those TV features represent constriction locations (CL), which are the deviations from median of the corresponding sensor along the X axis, and the constriction degree (CD), which is the minimum distance between the corresponding tongue sensors position and the palate trace. TRCL and TRCD are defined as follows

$$\mathrm{TRCL}[n] = \underset{m \in allutterances}{\mathrm{median}} \mathrm{TR}_x[m] - \mathrm{TR}_x[n], \tag{2.4}$$

$$\mathrm{TRCD}[n] = \min_{x \text{ palate}} \Big\{ \sqrt{\Big(\mathrm{TR}_x[n] - x\Big)^2 + \Big(\mathrm{TR}_z[n] - z\Big)^2} \Big\}, \tag{2.5}$$

The remaining four variables TBCL, TBCD, TTCL and TTCD can be obtained in a similar way:

$$\mathrm{TBCL}[n] = \underset{m \in allutterances}{\mathrm{median}} \mathrm{TB}_x[m] - \mathrm{TB}_x[n], \tag{2.6}$$

$$\mathrm{TBCD}[n] = \min_{x \text{ palate}} \Big\{ \sqrt{\Big(\mathrm{TB}_x[n] - x\Big)^2 + \Big(\mathrm{TB}_z[n] - z\Big)^2} \Big\}, \tag{2.7}$$

$$\mathrm{TTCL}[n] = \underset{m \in allutterances}{\mathrm{median}} \mathrm{TT}_x[m] - \mathrm{TT}_x[n], \tag{2.8}$$

$$\mathrm{TTCD}[n] = \min_{x \text{ palate}} \Big\{ \sqrt{\Big(\mathrm{TT}_x[n] - x\Big)^2 + \Big(\mathrm{TT}_z[n] - z\Big)^2} \Big\}. \tag{2.9}$$

2.4 Speech analysis, speech information representation

As it is mentioned in Section 2.1, for the AI task input data, there are two sources of information available, 1) acoustic information and 2) phonemic information. Different acoustic representations, such as line spectral frequencies (LSFs) [39], perceptual linear predictive coding (PLP) [60], Mel-frequency cepstral coefficients (MFCCs)[18] and filter bank energies (FBEs) from STRAIGHT spectrum [35] have been employed as the input of the AAI system [75]. Among these features, MFCCs are reported to perform better compared to other features for SI-AAI [17, 82]. In this thesis we utilized MFCCs and FBEs as the

acoustic information. In addition, the phoneme sequence is used as phonemic information. In the following, we describe briefly both types of features.

2.4.1 Acoustic representation

The acoustic features are extracted from the windowed time domain signal. The window length is chosen to satisfy quasi-stationary assumption for using the Fourier transform, and the window shift is chosen based on the sampling rate of the articulatory measurements. The acoustic features can be calculated from the smoothed magnitude spectrum by the STRAIGHT method [35], or directly from the magnitude spectrum. The average energy of speech in selected frequency bands is calculated by employing 40 triangular filters which are linearly spaced on Mel-scale frequency axis. The Log-scaled energies in the overlapping frequency bands are called filter bank energy (FBE) features. For obtaining cepstral features a discrete cosine transform (DCT) is used. The low order DCT coefficients (13 coefficients including energy) are kept as the spectral envelope information. These coefficients are called Mel frequency cepstral coefficients (MFCCs).

2.4.2 Phonemic representation

In scenarios where transcription of waveforms is available, phonemic information can be employed. The phonemic information of spoken utterances is used to force align them with the acoustic features. We used the Penn phonetics lab forced aligner [108]. The TIMIT database [14] uses 61 phonemic categories for English, and we folded them onto 39 categories (PHN) based on [43]. Afterwards, each phone is represented as a one-hot 39-dimensional vector [5]. In this way, the speech information is represented in form of phones and their duration, which contains information for articulatory inversion task [5].

2.5 Machine learning for AAI

In the literature, various techniques are applied to the AAI problem. Codebook search-based method [2] is one of the earliest works for the AAI problem. They used five articulatory parameters to represent the articulation of vowels and vowel-like sounds. These five parameters are

as follows: the maximum constriction place distance to the glottis, the cross-sectional area of the maximum constriction place, the area of the mouth opening, the lip protrusion, and the vocal tract length. In their model, they reduced the articulatory parameters to four by defining the lip protrusion in terms of an arbitrary function of vocal tract length. The acoustic space was parameterized by the five formant frequencies, and their bandwidth and their amplitudes. The joint codebook of acoustic and articulatory parameters was stored in the computer and the inversion process was done by searching the codebook given the acoustic information to find the corresponding articulatory parameters. The quality of this inversion method is highly dependent on how good the articulatory space is covered by the codebook. The codebook method is also used in the [24], where they used synchronous speech and EMA measurements for vowels, vowel-to-vowel transition and closure /g/, to make the codebook. In [57], they made the codebook in a hierarchical procedure to represent the codebook in terms of hierarchy of hypercubes. They ensured that the inversion mapping function in each hypercube can be approximated by linear functions.

Furthermore, statistical methods are employed for the AAI problem. In [93], a support vector regression (SVR) method was used to estimate the mapping between contextualized MFCC vectors and EMA measurements for MOCHA database. They used clustering for reduction of training size to deal with the training time which increases by $\mathcal{O}(3)$ of the training data amounts. A nearest neighbor algorithm is used for finding the samples with the minimum distance to the clusters representatives and use these data samples for training the SVR. This work was only conducted for one speaker. Furthermore, quantization of the acoustic space independently from the articulatory space may result in deficient articulatory space representative due to one-to-many mapping in AAI problem.

Gaussian mixture models (GMMs) were used in [92], to model the distribution of joint acoustic and articulatory space by using expectation-maximization (EM) algorithm [53] for the MOCHA database. For the regression function, two Gaussian mixture regression (GMR) was used based on different cost function optimization, the first one is based on minimum mean squared error (MMSE) estimation [34], and the second method was based on maximum-likelihood estimation (MLE) using the dynamic information for having a smooth estimated tra-

jectory. The MLE-GMR method improved the performance compared to the MMSE-GMR method. For acoustic space representation Mel-cepstral coefficients (MCCs) [38] were used. The concatenated MCC vectors were directly used or compressed by principal component analysis (PCA) method [101] when the context size was big, to prevent further difficulties in training the GMMs.

In 2012, in [26] a hidden Markov model (HMM) was used for cross-speaker AAI. The HMM estimated the articulatory trajectories for a reference speaker, and it was employed for another speaker by adapting to the speaker by the voice conversion method from [91].

Moreover, artificial neural network (ANN) based techniques are widely applied to the AAI problem. In [37], they employed a feed-forward neural network with four layers to estimate the articulatory motions from speech waveforms. A year after, [58] employed ANN for inferring articulatory gestures measured by X-ray microbeam data from acoustic information. They only used data containing six English stop consonants and observed that the critical articulators for production of consonants are showing higher correlation coefficient compared to the non-critical articulators.

A mixture density network (MDN) was utilized in [61] to estimate the articulatory space distribution by using a simple mixture model distribution on top of a neural network [6]. They used data from one of the speakers from MOCHA TIMIT dataset to train their MDN system. The MDN, estimates the conditional probability density function of articulators, and revealed the similar concept as they observed in [58], where the critical articulators have very small variance in contrast with the non-critical articulators. Furthermore, utilizing MDN for the AAI problem supplied a better performance than the ANN based systems.

By advancing in deep neural networks (DNN), [96] employed restricted Boltzmann machine (RBM) to train a deep belief network by stacking RBMs, from the acoustic features. They used the DBN as the pre-trained model and fine-tuned it to the articulatory data by adding output layer to back propagate the error between the measured and estimated articulators' positions. They improved the performance of the AAI task for MNGU0 dataset, by using the DNN compared to the earlier work where MDN was used.

Later, [47] utilized a deep recurrent neural network (RNN) which can learn the required context information by itself in contrast with the

fixed context window in DNN-based AAI models. They implemented a deep bidirectional long short-term memory (DBLSTM) and a deep recurrent mixture density network (DRMDN) to tackle the AAI problem. Their results on the MNGU0 dataset showed a significant improvement over the DNN-based baseline system, for both proposed architectures.

Later, RNN-based AAI system was implemented by [104] as a hierarchical estimation of phoneme sequence and articulatory parameters on a Mandarin Chinese AAI dataset. In the hierarchy, the first network was performing a monophone based phoneme recognition, and bottleneck features from this phone recognizer network were utilized in the second network for estimation of articulatory measurements. They found systems using phone sequence information in a hierarchical structure provide better estimation of articulatory trajectories.

In the following, we focus on the different deep neural architecture which are employed for AAI, in the literature and this work.

2.5.1 Feed-forward deep neural network

In this section a feed-forward DNN based AAI approach (DNN-AAI) will be described. This approach showed a significant improvement in AAI performance compared to the earlier regression methods [40, 50, 103]. DNNs approximate a mapping function between the input and output data. Considering a non-linear activation function e.g., Sigmoid function, rectified linear unit (ReLU) [22], the relationship between input and output data, will be a non-linear estimator. This makes the DNN a powerful tool for the AAI mapping which is highly non-linear as we mentioned in chapter 1. Another advantage of employing DNN is the ability of functioning with high dimensional data as input and mapping them to a different dimension as the output, e.g., in contrast with the Gaussian mixture model regression (GMR) which needs a preprocessing dimension reduction for large input vector sizes [92].

Figure 2.2 shows a DDN-AAI system where the input is a temporal context of the acoustic data and the output is articulatory data.

Considering a wide temporal context when estimating the articulatory movements is useful, as the co-articulation effect often extends beyond the phoneme level. By considering the acoustic feature for the n^{th} frame as $X[n]$, the corresponding augmented vector X_{aai} containing

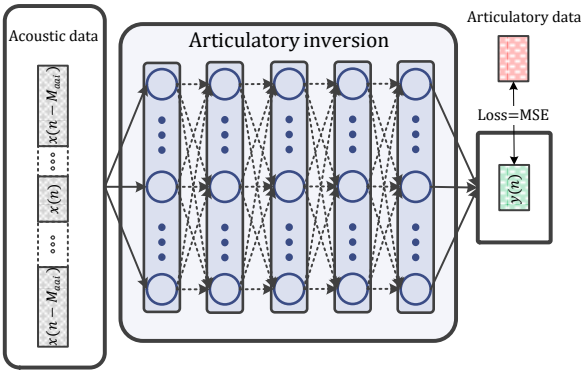


Figure 2.2: Block diagram of a DNN-AAI system.

the $X[n]$ and its context is given as:

$$X_{\text{aai}}[n] = \Big[X[n - 2 \times M_{\text{aai}}]^T, \ldots, X[n-2]^T, X[n]^T, \\ X[n+2]^T, \ldots, X[n + 2 \times M_{\text{aai}}]^T\Big]^T, \quad (2.10)$$

where M_{aai} denotes the number of left and right context frames which are added to $X[n]$. Let $Y[n]$ be the n^{th} vector of the articulatory estimates, then the regression for a DNN with L hidden layer can be written as:

$$\hat{Y}[n] = f_{L+1}\big({W_{L+1}^1}^\top f_L\big({W_L^1}^\top \ldots \big(f_1\big({W_1^1}^\top X_{\text{aai}}\big)\big)\big)\big), \quad (2.11)$$

where $(f_i, W_i^1), (i = 1, \ldots, L+1)$ are respectively the activation function of i^{th} layer and the matrix of weights between $(i-1)^{\text{th}}$ and i^{th} layer by considering the input layer as 0^{th} layer. For the task of regression because of having both positive and negative values, g_L should be the linear activation function or tanh if the absolute values of normalized articulatory data is less than one. All weight matrices are optimized during training by gradient based techniques with the back propagation algorithm [64] to minimize mean square error (MSE) between the estimated value $\hat{Y}[n]$ and the ground-truth value $Y[n]$. The estimated articulatory parameters $\hat{Y}[n]$ in this way are noisy and not smooth due to the one-to-one mapping of the DNN. For having a smooth estimation, the $\hat{Y}[n]$s need to be low-pass filtered for which we chose a

second order Butterworth filter to have a smooth estimation which is the physical nature of the articulators.

2.5.2 Recurrent deep neural network

In this section a recurrent neural network (RNN) approach will be presented, as RNN has demonstrated better results compared to DNNs [47, 104] because the temporal dynamic behavior is better captured through the memory elements of those recurrent architectures. Recurrent neural networks RNN have been utilized in many speech technology areas including speech recognition [21], language modeling [49], and articulatory inversion [47, 104, 109]. They are able to estimate any output samples from dynamical systems [66], conditioned on their previous samples. Having a non-causal condition by access to both past and future input samples, we can employ a bidirectional RNN to use the past samples within the forward layer and the future samples within the backward layer as shown in Figure. 2.4. Diamonds show the merge strategy of forward and backward layers output which can be summation and concatenation. Long short-term memory (LSTM) is a variant of RNN with a specific memory cell architecture for updating the hidden layers. In Figure. 2.3 a single LSTM memory cell is depicted where x_t and h_t are input and hidden vector, i_t, f_t, c_t and o_t are the input gate, forget gate, cell vector and output gate, respectively. The operation of this memory cell is formulated as follows:

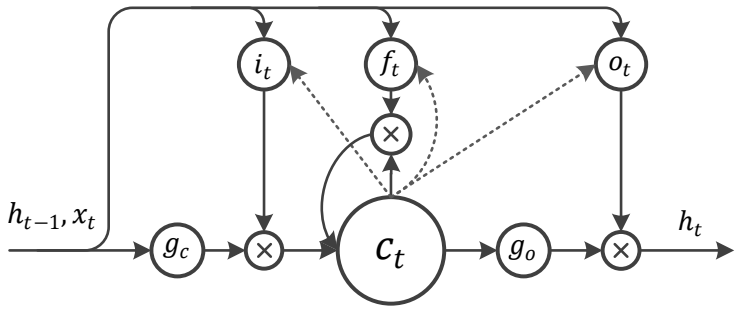


Figure 2.3: A long short term memory cell

$$i_t = \sigma\big(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\big) \tag{2.12}$$

$$f_t = \sigma\big(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\big) \tag{2.13}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_c\big(W_{cx}x_t + W_{ch}h_{t-1} + b_c\big) \tag{2.14}$$

$$o_t = \sigma\big(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o\big) \tag{2.15}$$

$$h_t = o_t \circ g_o(c_t) \tag{2.16}$$

The σ denotes the sigmoid function, g_c and g_o are the activation functions which are usually chosen as $tanh$, b is the bias vector for each gate (b_f is the forget gate bias vector). W denotes weight matrices where different subscripts show the connection between input/output and gates, for example, W_{ix} is the weight matrix between input vector and input gate. The operator $\circ$ indicates element-wise multiplication. A bidirectional long short-term memory (BLSTM) is realizable by using the LSTM memory cells (dotted ovals) in the forward and backward layer as shown in Figure. 2.4.

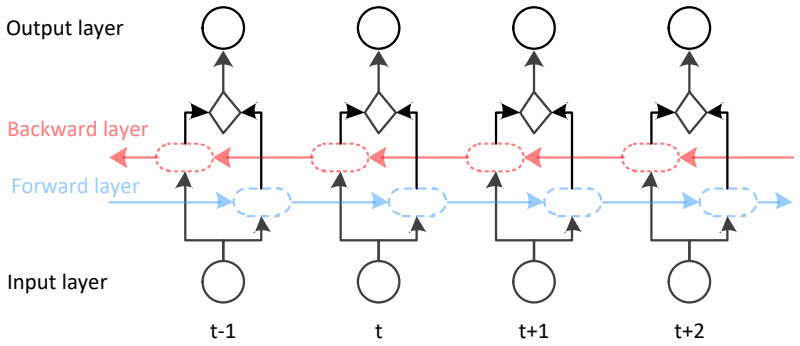


Figure 2.4: A bidirectional RNN.

2.5.3 1D convolutional neural network

The convolutional neural networks (CNNs) have been widely utilized in various domains, for one dimensional signals, as well as multi-dimensional signals. The kernel (filter) shape is defining the spatial dimension that convolution is performing on, e.g. a convolutional kernel with shape

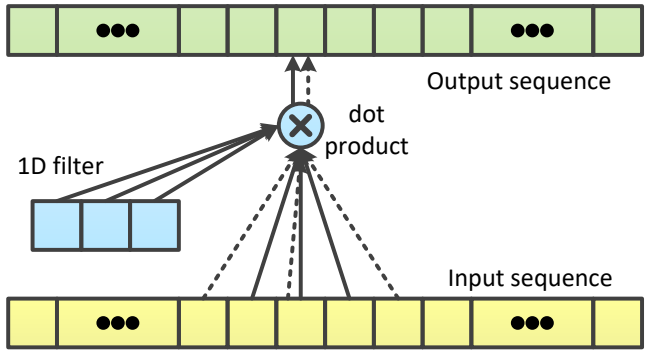


Figure 2.5: 1D convolutional layer with kernel size 3. The arrows show which samples are used in the convolution. The solid arrows from input show samples for dilation rate equal to 1, and dashed arrows show the samples for the dilation rate equal to 2.

M × 1 (M is filter length) is performing convolution along one dimension (1D-CNN), and a convolutional kernel with shape M × N (M and N are filter length in each axis) is performing a convolution in two dimensional (2D-CNN) space. The 1D convolutional filters are employed mostly for sequences, and they perceive the local features to obtain the global information of the whole input sequence. The convolutional filters can be applied to their input sequence with different strides and dilation rates. The stride value is describing the shift in filter over the input sequence, and it maximum value is the filter length to not miss any input samples in the convolution operation. Strides bigger than one result on down-sampled output sequence. The dilation rate defines the steps between input sequence samples with which filter coefficients are multiplied to form the convolution operation. Figure 2.5 demonstrate a 1D convolutional layer.

2.5.4 Temporal convolutional neural network

Temporal convolutional network (TCN) [4] is an specific form of CNN for sequential data. It utilizes the causal convolutional layers over the input sequence which means only information from past is used and the convolution at time t uses only samples from time t or earlier. The kernel size and dilation rate are very important to choose based on the input sequence length. The receptive field of TCN should be designed to cover the required length of input sequence. The receptive field de-

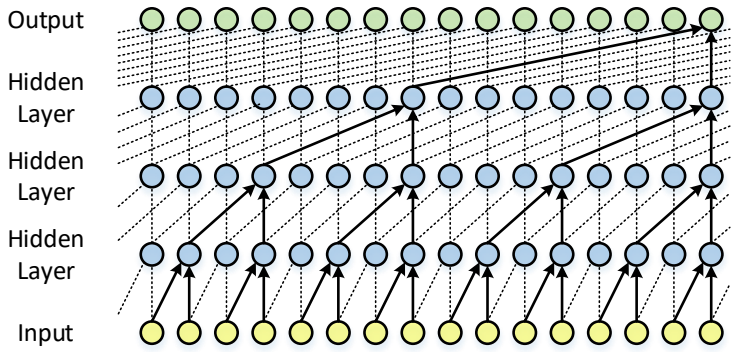


Figure 2.6: Stacking of 1D causal convolutional layers to build a TCN with kernel size 2, dilation rates [1, 2, 4, 8]. The lines with arrows show which samples are used in the convolution to estimate the output sample. The dashed lines are showing the estimation procedure for previous and future output samples.

pends on the kernel length, dilation rates and stacking of TCN layers on top of each other. Figure 2.6 demonstrate one layer of TCN with kernel size of 2, dilation rates [1, 2, 4, 8] which has the receptive field of 16 samples. The dilation rate is chosen to increase exponentially which enables network to have large receptive field with only few hidden layers. In this way, the network is computationally efficient while preserving the input resolution.

2.5.5 Transfer learning

Transfer learning is a machine learning technique to reuse a model trained on one task, on a second related task [19]. It tries to transfer the knowledge from the source domain to the target domain where the latter domain suffers from insufficient data or lack of some information. As deep learning approaches become dominant learning methods, transfer learning is extensively utilized in deep learning context by reusing the trained network in the target domain. This technique tends to work if the network input features are general which means features are suitable to both source and target tasks or domains. It is an important mechanism in deep learning to deal with insufficient data or lack of some information in the available data, and common steps to perform are as follows:

- **Source model selection:** Selecting a pretrained model from the source domain or task. The source domain mostly contains lots of training samples which makes the model suitable for general use.
- **Reusing model:** The selected pretrained model can be used directly as it is, or some parts of the model are used for the target task.
- **Model tuning:** In case of data scarcity and limited data, it is possible to fine tune the transferred model with the data in the target domain or task.

2.6 Performance measurements

To measure the performance of the AAI methods, the root mean squared error (RMSE) and the Pearson's correlation coefficient (PCC) metrics are used. The RMSE calculates the deviation between the estimated and the ground truth articulatory features as formulated in 2.17, and the lower RMSE shows a better performing inversion system.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i}\big(y(i) - \hat{y}(i)\big)^2}, \tag{2.17}$$

where $y(i)$ and $\hat{y}(i)$ are the ground truth and estimated articulatory features of the i^{th} frame, respectively.

The PCC measure is the normalised cross-correlation between the estimated and ground truth trajectories, and reports the similarity between these trajectories. The PCC value is in range [-1, 1], and the higher PCC shows a better inversion system. The PCC measure is defined as:

$$\text{PCC} = \frac{\sum_i (y(i) - \bar{y})(\hat{y}(i) - \bar{\hat{y}})}{\sqrt{\sum_i \big(y(i) - \bar{y}\big)^2 \sum_i \big(\hat{y}(i) - \bar{\hat{y}}\big)^2}}, \tag{2.18}$$

where $\bar{y}$, and $\bar{\hat{y}}$ are mean values of $y(i)$, and $\hat{y}(i)$.

The range of articulatory measurements can be different among different speakers, therefore, the PCC is better measure than the RMSE, to evaluate the speaker independent inversion system performance.

2.7 Application of AAI

Acoustic-to-articulatory inversion could be useful to understand the speech production mechanism. Also, the estimated articulatory features can be integrated and utilized in many important speech processing applications. For example, utilizing the estimated articulatory features together with the acoustic features improved the performance of automatic speech recognition (ASR) systems [51, 52, 87]. Augmenting the articulatory features with acoustic features improved the performance of dysarthric speech recognition systems [105]. Employing the estimated articulatory features improved the classification accuracy of depression severity level estimation [69]. Articulatory features can be employed to improve the quality of speech synthesis models [45, 46]. Utilizing the AAI in speech therapy systems, computer aided pronunciation training (CAPT) systems [26] and computer aided language learning (CALL) systems [3] would be useful by providing visual feedback of the articulators' positions from the acoustic signal.

CHAPTER 3

Contributions of the thesis

In this chapter, the contributions of thesis for the AAI problem are described briefly. Section 3.1 describes the new architecture and utilized features, for the AAI problem, which are proposed in Paper A [75] and Paper B [78]. In section 3.2, a new transfer learning approach for AAI task is developed and its performance was evaluated in an ASR task for the TIMIT database, which is taken from Paper C [76]. The effect of speaking rate variability on AAI problem is evaluated on section 3.3, and the proposed time domain architecture based on work from Paper F [79] is described. In section 3.4, AAI in noisy condition is explored and results for our proposed method from Paper D [73] and Paper E [77] are presented.

3.1 Deep architecture for acoustic/phonemic articulatory inversion

The AAI task has been explored for several decades from different perspective, e.g., by using various features and regression techniques, as mentioned in chapter 2, to predict articulatory information. In this section, first, we utilize acoustic and linguistic information for performing AI. Then, a new architecture is proposed to extract information from acoustic features based on acoustic landmark theory [84–86], which explains that significant changes in the articulatory configuration result in the abrupt changes in the speech spectrum.

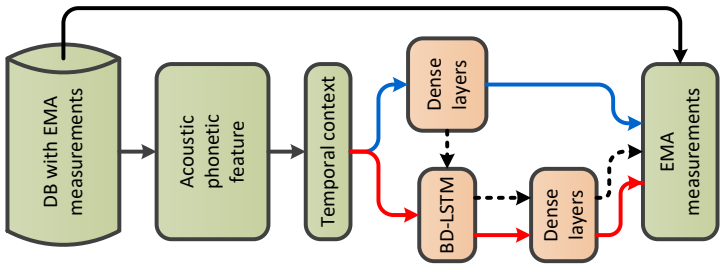


Figure 3.1: Deep neural architectures which have mostly used in the AAI systems.

3.1.1 Exploring linguistic features together with acoustic features for the AI

In the following, the effect of employing linguistic information in the form of PHNs (see Section 2.4.2) and attribute features AFs, which are PHN features projected to the manner and place of an articulation feature space (see Table 3.1), separately and together with the acoustic features, are analyzed. First, we describe the AFs for the TIMIT phoneme sets, and provide their activation in terms of manner and place of articulation. Then, explore where these features are performing better than the others, and what will be the performance of combining them.

3.1.1.1 Articulatory attribute representation

The articulatory phonetics or articulatory attribute features (AF) describe the process of articulation to make the speech sounds. It describes the speech sounds in terms of the articulators involved in their production, as we described them in Section 2.2. The distinctive English speech sounds can be described by the manner of articulation, place of articulation together with the voicing [33].

In Paper A, we used the TIMIT phone set for English data. This phone set was folded from 61 categories to 39 phones as in [44]. With the reduced phone set, a mapping was used to describe the phone according to their phonological features or articulatory attributes. The mapping is depicted in Table 3.1. The description considers 22 attributes, comprising manner and place of articulation for both vowel and consonant categories [80], and voicing. The attribute features are

binary, and more than one attribute feature is often active at the same time. These features are more language universal [41] compared to the phonetic representations. However, as mentioned in [89], this mapping is not theoretically accurate, both due to using binary features, and because of the mapping of vowels, and consonants into a common linguistic space, in spite of their differing definition of place of articulation.

We setup an experiment to investigate the performance of acoustic and linguistic features in estimation of articulatory features, in terms of the RMSE, for different manner of articulation groups. We trained inversion models with FBE, PHN, AF, and their pairwise combination to estimate the articulatory trajectories, and then calculate the RMSE error between the ground-truth and estimated articulatory trajectories within segments of speech based on their manner of articulation. Figure. 3.2 depicts the RMSE for different input features and their combination, for different manner of articulation groups. It can be observed that the acoustic features (FBEs) perform better for the vowel and approximant compared to the stand-alone PHNs and AFs. It can be interpreted as being the high dynamics in the vowels which cannot be modeled by one-hot encoded vectors in case of PHNs and several activated binary features for the case of AFs. The RMSE for the fricatives, nasals and stop sounds are better estimated by PHN and AF features in comparison with the FBEs. Combination of FBEs with either PHNs or AFs improves the performance of the inversion systems in all cases.

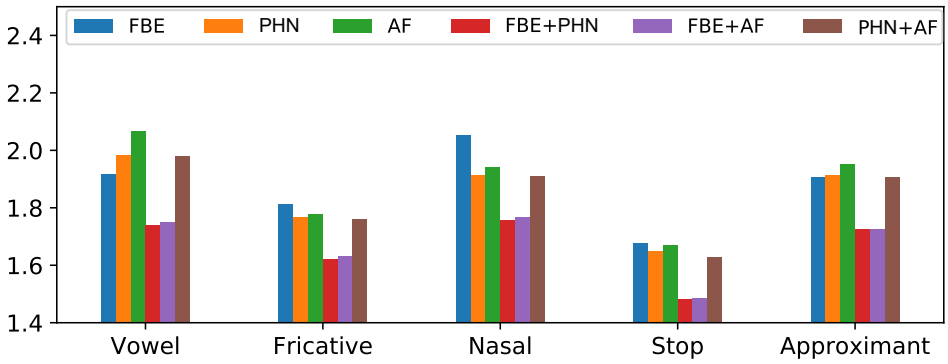


Figure 3.2: Average RMSE for manner of articulation from estimated trajectory by different input features.

Table 3.1: TIMIT phoneme list in terms of attribute features. The mapping is adapted from [44].

	Attribute	Phonemes
Manner	Vowel	iy ih eh ey ae aa aw ay ah ao oy ow uh uw er
	Fricative	jh ch s sh z zh f th v dh hh
	Nasal	m n ng
	Stop	b d g p t k dx
	Approximant	w y l r
Place	Coronal	d l n s t z
	High	ch ih iy jh sh uh uw y ow g k ng
	Dental	dh th
	Glottal	hh
	Labial	b f m p v w
	Low	aa ae aw ay oy
	Mid	ah eh ey ow
	Retroflex	er r
	Velar	g k ng
Others	Anterior	b d dh f l m n p s t th v z way aa ah ao aw ow oy uh uw g k
	Back	aa ae ah ao aw ow oy uh uw g k
	Continuant	aa ae ah ao aw ay dh eh er r ey l f ih iy oy ow s sh th uh uw v w y z
	Round	aw ow uw ao uh v y oy r w
	Tense	aa ae ao aw ay ey iy ow oy uw ch s sh f th p t k hh
	Voiced	aa ae ah aw ay ao b d dh eh er ey g ih iy jh l m n ng ow oy r uh uw v w y z
	Silence	sil

3.1.2 1D-CNN feature extraction for the AAI

In the previous section, we employed the one-hot encoded vector for the phonemic features, and binary valued feature vectors for the attribute features for the AI task. The AI models trained by phonemic and attribute features were performing similar to the AAI models trained with FBEs. These binary feature vectors provide the required information for the estimation of articulators' position, with nearly similar performance to the AAI system utilizing FBEs. By inspecting the binary feature sequences, one can infer that they contain information with respect to the phoneme's left and right context, change of phonemes and duration of the activated phoneme. The change of phonemes in the sequence is very important as it is stated in the [84–86] and is known by the acoustic landmark theory. The acoustic landmark theory has discovered that major changes in the articulators' gestures will lead to abrupt changes in the speech spectrum. Considering the observations and the acoustic landmark theory motivated us to find a solution for sensing the changes in energy in speech spectrum. We employed 1-D convolutional layers which are mostly known as feature extraction layers from sequences and widely used in many speech applications, e.g., ASR [1, 59], speech synthesis [97], and machine translation [42]. This is the first time, to the best of the authors' knowledge, that 1-D convolutional layers on the features are employed in the AAI task. Here we employ convolutional layers along the time axis: we consider the output of the filter-bank in each of the frequency bands as a one-dimensional data stream and apply the filters on it. These filters' outputs are then linearly combined and represent new feature maps. The computations are formulated as:

$$\boldsymbol{y}_{i,j}^{\text{cnn}} = b_j + \sum_{k=1}^{L_{i-1}} \mathbf{F}_i * \boldsymbol{y}_{i-1,k}^{\text{cnn}}, \tag{3.1}$$

where, $*$ shows the convolution operation of weights $\mathbf{F}_i$ in convolutional layer i with the feature maps $\boldsymbol{y}_{i-1,k}^{\text{cnn}}$ from the previous layer $i-1$. A bias b_j is added to the result of the convolution, to calculate the new feature map $\boldsymbol{y}_{i,j}^{\text{cnn}}$ for the j^{th} channel feature map. Zero padding is used to guarantee that the input sequence (acoustic space) and output sequence (articulatory space) have the same length. The 1D-CNN layers are used and concatenated along the channel axis as depicted in Figure. 3.3. The filter length is different in each of the CNN

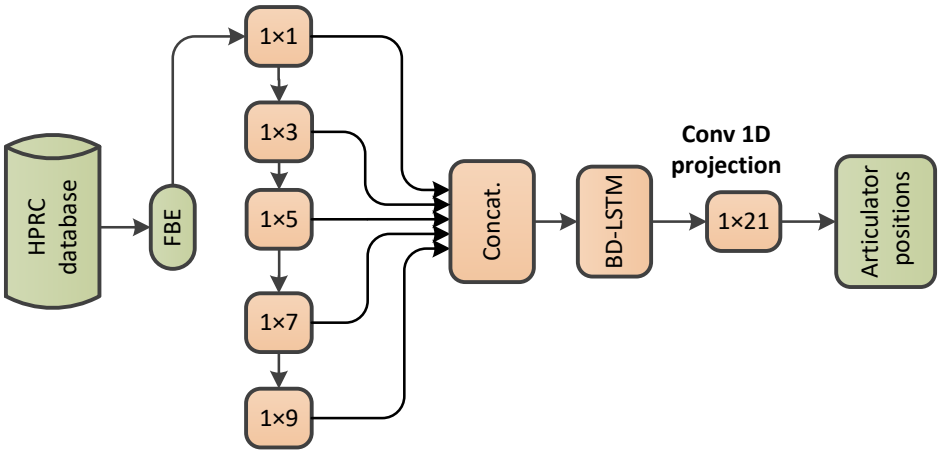


Figure 3.3: Layer-wise feature representation by 1D convolutional layers for the AAI task. The 1×1, 1×3, and ..., are representing the convolution kernel shape.

layers which provides more information about adjacent frames with different resolutions along the time axis. The convolutional layers play a key role by high-passing or low-passing different frequency bands. Different filters are sensing significant energy changes in different frequency bands of the speech spectrum, which may indicate a phone transition. The convolutional layer with longer kernels tries to capture more temporal information and filter out undesired temporal variabilities. After the convolutional layers, two BLSTM layers are employed to capture dynamical information and estimate smoothly varying articulator trajectories. As we described previously, 1D-CNN layers extract new features from the FBEs. These feature maps are weighted sums of sub-band signals which have been processed by filters with different frequency responses. Figure. 3.4 shows an example of FBEs, and network activations through the 1D-CNN model. We can see some channel activations match phonemic segments in the first layer. Going to the next layers, the filter outputs become sparser, and activations become more intense within the phoneme boundaries. For justifying our claim about channel output activations during the phonemic segments, we picked some channels output from the first layer by using correlation analysis with PHN and AF features as the reference patterns. This analysis provided a better insight for choosing the corresponding filter outputs

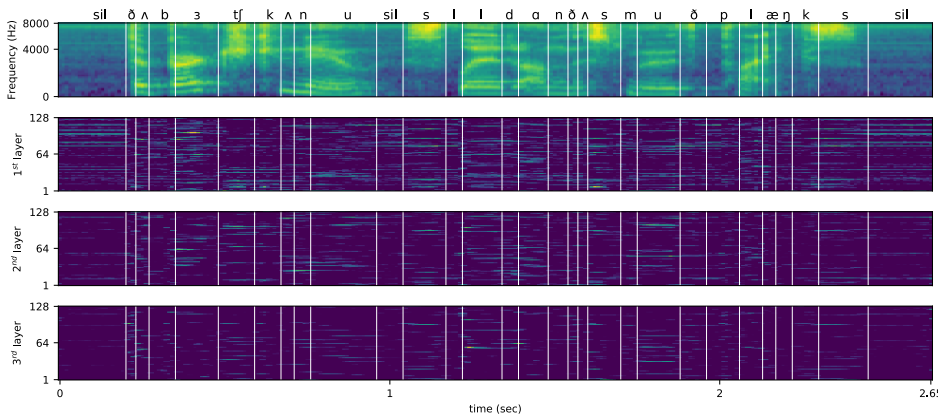


Figure 3.4: FBE features for utterance "The birch canoe slid on the smooth planks."and the resulted convolutional feature maps for the 1st, 2nd and 3rd layers.

with regards to PHN and AF features with higher correlation. As an example, we have chosen attribute fricative and the phoneme /ɜ/.The corresponding filters' output which are chosen after doing correlation analysis are depicted in Figure. 3.5. We can see that these filters outputs have high energies when the chosen attribute and phoneme are active. Therefore, we can claim these 1D-CNN layers are extracting linguistic information from FBEs. This is in line with our expectation of sensing the significant energy changes at the phone transition. Furthermore, we can see for the second CNN layer compared to the first CNN layer, we have less activation outside the ground truth activation times of the chosen attribute and phoneme.

3.2 Transfer learning of AAI

AI has been studied for a long time and has developed by employing different regression models to improve its performance in terms of RMSE and PCC as objective measures. Unfortunately, speech databases with simultaneous articulatory measurements are few, limited in size, and with a small number of speakers. The small number of speakers implies that the available data can only give sparse representations of the articulatory and acoustic spaces. Thus, the possibility of employing trained AI models in other applications will be limited. For better understanding of the problem, the different features in the AI task are defined as follows: the acoustic features, $\mathbf{x} \in \mathbb{R}^n$, the articulatory fea-

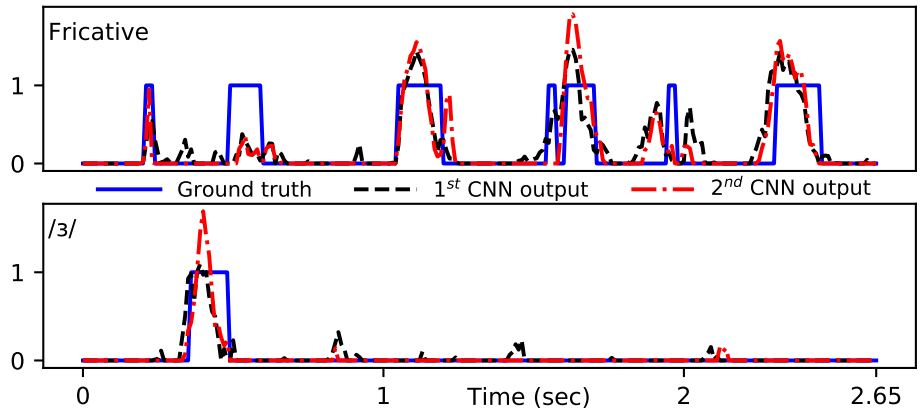


Figure 3.5: AF feature for fricative and PHN features for phoneme /ɜ/ (——) and channel output from the 1st 1D-CNN layer (- - - -) and 2nd 1D-CNN layer (—·—·—).

tures, $\mathbf{y} \in \mathbb{R}^m$, and the phone features, $\mathbf{p} \in \mathbb{B}^l$, where $\mathbb{R}$ is the field of real numbers, and $\mathbb{B}$ is the Boolean field. As the acoustic space is continuous, there should be enough data to cover the whole acoustic space and speaker variabilities, otherwise, the trained acoustic space is not generalized well enough to be used with a new set of speakers. By having access to the transcription information, it is possible to use the linguistic information for transfer of the articulatory information which are a limited number of vectors taken from a discrete space. In this way, the different speaker's variability reduces to the phone duration. This approach is only applicable for the tasks where the text is available. In cases where the transcription is not available, a possible suggestion is to transfer the knowledge through the linguistic features and then employing a teacher-student approach for distillation of articulatory data to the models which have only acoustic information and do not have the articulatory measurements. In this way the articulatory data are transferred in a less variable space and can then be employed for the other speech related tasks. Several experiments are done to evaluate the estimated articulatory trajectories for the baseline and the proposed teacher-student model. For performing transfer learning of articulatory information, the HPRC database is used as the source for articulatory information, and the TIMIT database is

employed to check the performance of transferred articulatory knowledge. The TIMIT database is well known in the speech processing field, and it contains manually labeled acoustic data. We have studied three ways to estimate articulatory features for the TIMIT dataset:

- using an AAI system trained on HPRC data ($f_{\text{AAI-base}}$),
- using a phonemic-to-articulatory (PAI) system trained on HPRC data (f_{PAI}),
- teacher-student approach to train an AAI model ($f_{\text{AAI-stud}}$) for the TIMIT data by using the provided features from the f_{PAI} model.

$f_{\text{AAI-base}}$ is considered as the baseline system, and the proposed system $f_{\text{AAI-stud}}$ is referred to as student system. In the following sections these three approaches are described briefly, and afterwards, the performance of the baseline and student systems are evaluated from two perspectives: (i) Speech production mechanism, and (ii) performance in ASR task by comparing phone error rate (PER).

3.2.1 Articulatory estimation for TIMIT by using the $f_{\text{AAI-base}}$

For the baseline system, an AAI model is trained based on the HPRC corpus with MFCCs as the input, and tract variables (TVs) as the output. The MFCC feature vectors are extracted from the windowed signals with frame length of 25ms and frame shift of 10ms (corresponds to the 100 Hz frequency rate of articulatory measurements). Five stacked 1-D convolutional layers of kernel size [1,3,5,7,9] are employed to extract the features from the MFCC features' sequence. Afterwards, the extracted features are passed through two BLSTM layers with 128 memory cells on each forward and backward directions. Finally, the recurrent layers output is fed to a 1D convolutional layer to project the extracted information from temporal dynamics of utterances to the output and estimate the TVs. The sequence-to-sequence based mapping is employed to get smooth varying trajectories, obviating the need for lowpass filtering the estimates. After the training of the AAI model is done, the trained AAI model is employed to estimate the TVs for the TIMIT utterances.

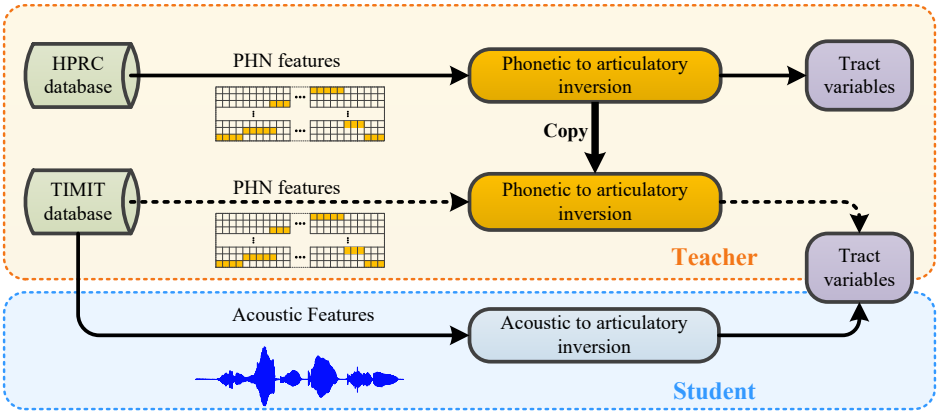


Figure 3.6: Block diagram of the proposed transfer learning method from the HPRC to the TIMIT database, and knowledge distillations from phonemic features to acoustic features through articulatory space. Dashed arrows correspond to no training.

3.2.2 Articulatory estimation for TIMIT by using the f_{PAI}

For training the PAI, the time aligned phonemic features are used as the input to the model, and the output targets are the TVs. The input is directly fed to two BLSTM layers with 128 memory cells in each forward and backward direction. The output of the second BLSTM layer is then passed to a 1-D convolutional layer to project the extracted information to the output.

3.2.3 Teacher-student technique for training the $f_{\text{AAI-stud}}$

In sections 3.2.1 and 3.2.2, we have obtained two models to estimate the articulatory features. As it is described in Section 3.2, the PAI model will have a better performance compared to the AAI model in the speaker mismatched conditions. The drawback of the PAI model is its applicability where no transcription is available. A teacher-student model is employed to deal with this shortcoming, as it is depicted in Figure. 3.6. A f_{PAI} model is trained based on data from HPRC dataset. The f_{PAI} model is used as the teacher model to estimate the TVs for the TIMIT dataset from the time-aligned phones. The estimated TVs from the f_{PAI} are used as the target for $f_{\text{AAI-stud}}$ model which uses the acoustic features as the input.

3.2.4 Experimental results

In the previous Sections, three ways for estimating the articulatory features for the TIMIT data are suggested. TIMIT does not have simultaneous articulatory measurements, and accordingly a comparison of the trajectory estimates with the ground truth is not possible. We resort to inspection of the trajectories and to assessment of the effectiveness of the estimated TVs in phone recognition experiments to examine the quality of the trajectory estimates. An example of estimated trajectories is shown in Figure 3.7. It can be observed (inside the solid ellipses) in Figure 3.7, that for production of the stop sound /p/, the LA is decreasing and LP is increasing, vowel /æ/ has wider LA or JA than vowels /eɪ/ or /oʊ/, which is in line with dropping of the jaw in production of vowel /æ/ while the jaw is slightly open in /eɪ/ or closed in /oʊ/. Furthermore, it can be observed that at the end of the utterance (inside the dashed ellipses), the values of the $f_{\text{AAI-base}}$ estimation do not decrease or increase for lip separation or protrusion, respectively, when the stop sound /p/ is present, and it is expected to have lowest values for the LA compared to the other phones in this sequence of phones. We can see the $f_{\text{AAI-base}}$ estimation of the LA for /l/ is less than the estimated value for /p/ which is wrong because for production of /p/ lips are closed and for production of /l/ lips are separated. That implies the $f_{\text{AAI-base}}$ model does not provide correct information with respect to speech production constraints.

For comparing the performance of the $f_{\text{AAI-base}}$ and $f_{\text{AAI-stud}}$ models, an ASR system has been utilized. The ASR model is an end-to-end phone recognizer [99] from the ESPnet toolkit [98]. The phone recognizer is an RNN encoder-decoder realized by BLSTM layers, combined with hybrid connectionist temporal classification (CTC) [20] and attention mechanism [7] for the end-to-end training and decoding steps in ASR. The phone recognizer architecture is as follows: the encoder has four layers of BLSTM with 320 cells, the decoder has one layer of LSTM with 300 cells, location-aware attention mechanism with 10 convolution filters of length 100, and the CTC and attention losses are equally weighted by 0.5. We used a predefined portion of the TIMIT for training which consists of all the SX and SI sentences from 462 speakers. The sentences from the remaining 168 speakers are meant for development and testing purposes.

Several experiments were conducted to gain insights on the role

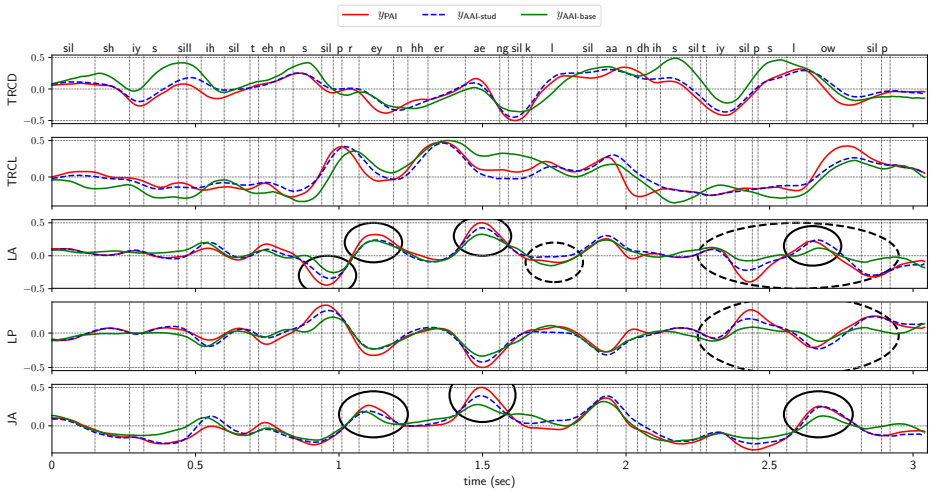


Figure 3.7: TV trajectories from f_{PAI}, $f_{\text{AAI-base}}$, and $f_{\text{AAI-stud}}$ for utterance "She slipped and sprained her ankle on the steep slope."

of the TV estimates in speech recognition. In the initial experiment, the phone recognizer was trained on static 23-dimensional FBEs, (x), only. In the second experiment, dynamic features were added to x and denoted as $(x, \Delta x, \Delta^2 x)$. The phone recognizers based on acoustic features only serve as baseline systems. The PER for different input features is reported in Table 3.2. $y_{\text{AAI-stud}}$ combined with x, significantly improves the recognition accuracy, and reduce the PER by 6.7% absolute on the test set. Interestingly, a slightly better PER, +0.2%, is obtained by replacing the 52-dimensional dynamic acoustic features $(\Delta x, \Delta^2 x)$ with the 9-dimensional $y_{\text{AAI-stud}}$. Moreover, employing the $y_{\text{AAI-stud}}$ obtains better performance than the $y_{\text{AAI-base}}$. The combination of $y_{\text{AAI-stud}}$ with $x, \Delta x, \Delta^2 x$ reduces the PER by 0.6%.

In conclusion, the proposed teacher-student approach for training $f_{\text{AAI-stud}}$, is performing better compared to the $f_{\text{AAI-base}}$ in both evaluations. The reason can be interpreted as the acoustic space representation for the $f_{\text{AAI-stud}}$ being more generalizable in contrast to the $f_{\text{AAI-base}}$, due to the fact that for the latter model training there are only eight speakers available while in the training step of $f_{\text{AAI-stud}}$ 462 speakers are employed.

Table 3.2: PER for acoustic features and their combinations with the estimated TVs from $f_{\text{AAI-stud}}$ and f_{PAI}. D denotes feature dimensionality.

feature type	D	Dev PER	Test PER
x	26	25.6%	27.9%
x, $y_{\text{AAI-base}}$	35	20.9%	23.3%
x, $y_{\text{AAI-stud}}$	35	**19.6%**	**21.2%**
x, Δx, $\Delta^2 x$	78	19.8%	21.4%
x, Δx, $\Delta^2 x$, $y_{\text{AAI-base}}$	87	19.8%	22.8%
x, Δx, $\Delta^2 x$, $y_{\text{AAI-stud}}$	87	**19.1%**	**20.8%**

3.3 AAI from speech waveforms

The feature representation of the speech signal has a critical effect on the performance of related applications. In the AAI task, hand-crafted features like, line spectral frequencies (LSF) [62, 96], log Mel filter bank energies (FBE) [75, 78], Mel frequency cepstral coefficients (MFCC) [27, 77, 104], etc. are commonly employed, to represent the acoustic space. The LSF features represent the parametric modelling of the speech spectrum, FBEs and MFCCs are representing speech signal energy in frequency sub-bands inspired by properties of the human auditory system. We will use the term "hand-crafted features" to contrast features where significant parameters for their computation are based on scientific knowledge and experimental best practice from features whose definition and computation are purely data driven. These hand-crafted features are extracted from the windowed signal (frame) which is necessary for applying the Fourier transform. Furthermore, the frame rate in AAI experimental research is constrained by the articulatory sampling rate. Choosing a fixed frame length and frame shift is not the optimal choice for semi-periodic and non-periodic parts of the speech signal [88]. In addition, using fixed filterbanks for feature extraction is not the optimal choice, as argued in various speech applications [12, 65, 68].

The hand-crafted features have been utilized in different scenarios, like speaker dependent or speaker independent AAI. MFCC features have shown better performance in the case of speaker independent systems [17], which could be explained by the knowledge of how they are extracted. The discrete cosine transform (DCT) is compressing the in-

formation of FBEs to the low order cepstral features and by liftering of higher order cepstral features, the detailed information with respect to the speakers are removed from these features, which makes it works better for speaker independent AAI systems. However, the AAI system performance degrades significantly for mismatched speaking rate, i.e. the test speaking rate is faster or slower than the training speaking rate [81]. The reason behind the degradation in performance could be described by more variability, both acoustically and articulatory, in fast speaking rate compared to the normal speaking rate. The convolutional neural networks have shown their capability of dealing with the variability in the input data when they are utilized as feature extractor layers. This property of convolutional layers motivates us to employ them for extracting features from raw speech signal.

In this section, we will investigate the AAI system performance for mismatched speaking rate and utilize the 1D convolutional layers as feature extractor from the time domain speech (or raw speech) signal.

3.3.1 Articulatory estimation from time domain signal

In previous work in the AAI field, the frame rate was chosen to make the feature vectors rate match the articulatory sampling rate, e.g., a frame rate of 10 ms will result in feature vectors rate of 100 Hz to match the sampling rate of articulatory measurements. After framing the speech signal either hand-crafted features, e.g., MFCC or data driven features were extracted and utilized for AAI problem. For extracting the data driven features, 1D-convolutional and pooling layers were utilized in [27] to extract features from the speech frames. Figure 3.8 demonstrates three different architectures for the AAI problem. The top green part is presenting the AAI systems which utilize BLSTM layers on top of hand-crafted features to estimate the articulatory features. The yellow rectangle in the middle of Figure 3.8 is presenting the 1D-convolutional architecture which is applied to the windowed raw speech signal. The bottom blue rectangle is our proposed method which utilizes 1D-convolutional layers on the time domain speech sequence and decimates the signal to the articulatory space sampling rate. Afterwards, a temporal convolutional network (TCN) is applied to the extracted features by the 1D-convolutional and decimation layers, to estimate the required dynamic information from speech features and use them for articulatory feature estimation. In the proposed archi-

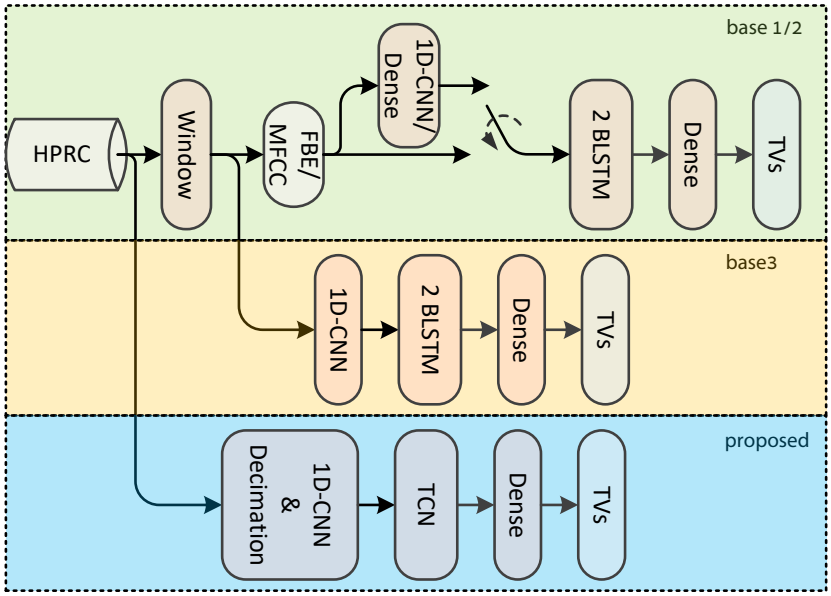


Figure 3.8: The sequence-to-sequence AAI systems, employing (top) hand-crafted features, (middle) extracted features from speech frames by 1D-CNN, (bottom) extracted features from the whole speech sequence by 1D-CNN and decimation layers.

tecture, the pooling layers have overlaps which provide a non-uniform sampling and preserve the required information for estimation of articulatory trajectories. The TCN extract the dynamic information of extracted features through dilated convolutions which cover the whole sequence. The output of the TCN module is fed to a time distributed dense layer for estimation of articulatory trajectories.

3.3.2 Experiments and results

For assessing the proposed architecture performance, three baselines were chosen from state-of-the-art methods. The first and second baseline systems uses hand-crafted features MFCCs and FBEs, respectively. We will refer to them by **base1** and **base2**. The system which applied the 1D convolutional layers on the windowed speech signal will be referred by **base3**.

All the experiments are done in SI scenarios, where test speakers are in both matched and mismatched conditions. In the matched

speaker condition, the AAI system is trained with training data from all of speakers and evaluated based on the same speakers' test data. In the mismatched speaker condition, the leave-one-speaker-out cross validation (LOSO) is used, where each of the available speakers are in turn kept as the test speaker, and the rest of speakers are used for training the AAI system. Furthermore, the matched and mismatched speaking rate conditions are evaluated for the baselines and the proposed method. The AAI systems performance is evaluated by the PCC measure. The results are reported in Table 3.3.

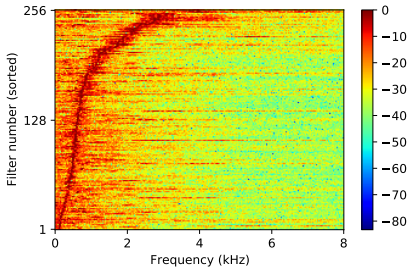
3.3.2.1 Matched speaking rate

As mentioned earlier in this chapter, the AAI systems are trained and evaluated for the matched speaking rate. In this way, we evaluate the systems' performance with respect to the speaker variability for each of the normal and fast SRs. Table 3.3 shows the proposed method achieves the best performance among all systems. For normal SR and matched speakers, the proposed method performs better compared to the baselines. In the mismatched speaker condition, the proposed method has reached an average PCC=0.72, better than any of the baselines. For the fast SR, the proposed and **base1** systems have the best performance in matched speaker condition with average PCC=0.79, and in the mismatched speaker condition, the proposed system is outperforming the baseline systems. It is worth mentioning, that the **base1** system which utilize the MFCC features outperforms significantly the **base2** system with the FBE features in the mismatched speaker condition. This superior performance of MFCCs over the FBEs can be explained by the DCT operation followed by liftering of higher order cepstral coefficients which contain spectral details of speech spectrum. The first convolutional layer filters in the proposed method and in the **base3** system are acting as filterbanks for extracting information from the raw waveform. The frequency response of the convolutional filters in the **base3** system is depicted in Figure 3.9(a). The center frequencies of the filters are sorted along the frequency axis. The filters' center frequencies are spanned linearly up to 1 kHz, and for frequencies higher than 1 kHz up to 4 kHz, the filters' center frequencies are spanned non-linearly. The frequencies higher than 4 kHz are attenuated in the **base3** system. The center frequencies of the first layer of 1D convolutional filters in our proposed method are depicted in Figure 3.9(b). The center frequencies

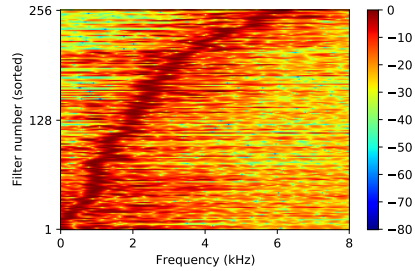
Table 3.3: The average PCC for different systems in the matched speaking rate condition. Spk cond indicates whether the speakers in the training and testing sets are matched or mismatched.

Spk cond	train-SR	Proposed	base1	base2	base3
matched	N	0.84	0.83	0.80	0.81
mismatched	N	0.72	0.7	0.66	0.7
matched	F	0.79	0.79	0.73	0.78
mismatched	F	0.66	0.64	0.58	0.62
NO. Parameters		377,827	544,009	1,585,033	873,481

are linearly spanned up to 3 kHz, and from 3 kHz up to 6 kHz the center frequencies are spanned non-linearly along frequency axis. The preservation of frequency components higher than 4 kHz is useful in the estimation of high frequency sounds, e.g., fricatives.



((a)) The magnitude response of learned filters sorted by center frequency for **base3** system.



((b)) The magnitude response of learned filters sorted by center frequency for the proposed method.

Figure 3.9: Frequency response of first layer of convolutional layers for extracting information from raw speech waveform.

3.3.2.2 Mismatched speaking rate

Different speaking rates affect the articulators' movements and characteristics of the produced speech signal. We have assessed the AAI system in the mismatched SR conditions. The proposed method performs better for the scenario where systems are trained on normal SR and tested on the fast SR, in both the matched and mismatched

Table 3.4: The average PCC for different systems in the mismatched speaking rate condition. Spk cond indicates whether the speakers in the training and testing sets are matched or mismatched.

Spk cond	train-SR	Proposed	base1	base2	base3
matched	N	0.76	0.71	0.70	0.73
mismatched	N	0.65	0.52	0.56	0.61
matched	F	0.78	0.78	0.73	0.78
mismatched	F	0.68	0.67	0.64	0.66

speaker condition. **base1** has the poorest performance compared to the other systems trained on normal SR, even compared to **base2**. It could be explained by the conclusion of [72], where FBEs compared to MFCCs, have shown a better phoneme recognition rate for scenarios with mismatch in SR and speakers. It should be recalled from the investigations in sections 3.1 and 3.2 that phonemic information is relevant for prediction of articulators' movements.

In the scenarios where systems are trained on fast SR, and tested on normal SR, the proposed method, **base1** and **base3** systems perform the same. The results are reported in Table 3.4. The systems trained with the fast SR and tested on normal SR (PCC=0.78) have shown a better performance compared to its counterpart. The fast SR contains more co-articulation in comparison with normal SR, therefore in the mismatched SR, the systems trained with fast SR perform better than the systems trained on normal SR.

3.4 Robust AAI

Most AAI systems are trained and evaluated in clean conditions (AAI-C), but what will be the performance drop of the AAI-C in noisy conditions? This evaluation will provide some insight into the real-world applicability of the AAI models. To the best of our knowledge, AAI in the presence of noise is only investigated in [70], where they proposed to train AAI system with multi-condition noisy data for dealing with noisy conditions. In this section, the AAI system performance is investigated in the presence of noise. In order to evaluate the noise effect,

three main procedures are considered. The first procedure is training a multi-condition AAI model (AAI-MC), the second approach is to perform speech enhancement (SE) prior to the existing AAI-C model, and the third approach is joint training of AAI and DNN-based SE (DNN-SE). The feed-forward DNN architecture is utilized for both AAI and DNN-SE models. The reason behind choosing a feed-forward architecture is due to its satisfactory performance, ease of implementation, and considering that we want to assess the systems in noisy condition and do not want to propose a new architecture. The multi-condition noisy data for the experiments are created by adding noises from the AURORA 2 corpus which are commonly used for realistic noisy scenarios to clean speech files. In the first set of experiments, the performance of the AAI-C model in presence of noise is evaluated. Then we train an AAI-MC model to assess the performance gain over the AAI-C model in multi-condition noisy data. In the second set of experiments the SE is used as the preprocessing stage for suppressing noise from speech and then using the enhanced speech through the AAI-C model to observe if enhancement is useful and results in an improved inversion performance. Based on the results reported in [70], the MSE based SE [11] as a preprocessing step is not helpful for AAI task. We reevaluate their findings using a DSP-oriented MMSE approach [8] which we denote DSP-SE. Then, we use a DNN based SE method [106], denoted DNN-SE, as a preprocessing step prior to the AAI-C system. We utilize DNN-SE as it has shown its strength in comparison to DSP-SE methods [106]. In the last set of experiments, a joint DNN-SE and AAI system is proposed and evaluated to deal with multi-condition noisy data. The performance of the inversion systems is evaluated based on the PCC measure. Furthermore, the inverted articulatory features are used in a transfer learning framework for an end-to-end ASR system based on the WSJ dataset in multi-condition noisy environment to evaluating their contribution for ASR in terms of word error rate (WER).

3.4.1 Speech enhancement prior to the AAI

Speech enhancement (SE) has been widely used in different applications of speech processing, e.g., [10, 16, 95]. The SE could be employed as a preprocessing system on the noisy data prior to the main application which is trained with the clean data. As it is mentioned in 3.4,

we employed two well-known approaches from the literature. The first method is an improved MMSE based SE from [8] which is based on estimation of noise spectrum which is assumed to be slower varying than the speech spectrum. We refer to this digital signal processing method by DSP-SE. The DSP-SE can be applied to a noisy signal without having knowledge of the noise condition because it is working based on noise and speech statistics. The second approach is based on DNN regression between noisy and clean speech [107]. In the DNN-SE method, various noisy signals at different SNR levels are mapped to their clean counterpart based on trained nonlinear function. This method needs to be trained with several noise types and noise levels to make it a generally applicable tool to use. In the following we briefly describe the DSP-SE and DNN-SE methods.

3.4.1.1 DSP-SE method

For the DSP-SE method, a MMSE based SE is used which utilizes the improved minima-controlled recursive averaging (IMCRA) [8] for estimation of noise. The SE method uses the optimally-modified log spectral amplitude (OM-LSA) speech estimator, which utilizes a gain function based on the geometric mean of two gain functions related to speech presence and absence. The enhanced power spectrum of speech is estimated by applying the gain function to the noisy speech power spectrum. For estimation of speech presence and absence, this algorithm uses an improved noise estimator version of [9], which combines time-varying recursive averaging with minima-controlled estimation of the a priori speech absence probability. The time-varying recursive averaging is a technique for noise power spectrum estimation. It recursively averages the past power spectral values of noisy signal using a time-varying frequency dependent smoothing parameter that is adjusted by probability of speech presence in sub-bands. The speech presence probability estimation is controlled by minima values of smoothed power spectrum of noisy signal. The noisy speech power spectrum over its local minima within a fixed temporal window provides the speech presence probability. The probabilities are estimated for each frame and each sub-band. The IMCRA contains two iterations of smoothing and minima tracking. The first iteration is performing a rough voice activity detection, and in the second iteration, smoothing removes sig-

nificant speech components which makes the minimum tracking robust during the speech activity.

3.4.1.2 DNN-SE method

For performing speech enhancement based on DNN (DNN-SE), a feed-forward architecture is utilized based on the work in [106, 107]. The DNN-SE system contains ReLU activation function [22] as the non-linearity for hidden layers, and linear activation function for the output layer. The DNN-SE network finds a nonlinear regression between the input noisy signal, and the clean speech signal as the output, in the training step. The non-linear blocks allow the network to better handle the non-linear relation between the noisy and clean signal, as mentioned in [106]. Most SE algorithms perform short-time Fourier transform (STFT) analysis on the noisy signal and enhance only the magnitude spectrum and keep the phase spectrum unchanged. We follow that approach. The DNN-SE input features are Log power spectra (LPSs) which are normalized by global mean and variance.

As the dynamic information in speech signal is very important, we take it into account by considering contextual information of M_{se} previous and future frames around the current frame, as the input for DNN-SE:

$$S_{\text{se}}[n] = \Big[S[n - M_{\text{se}}]^T, \ldots, S[n]^T, \ldots, S[n + M_{\text{se}}]^T\Big], \tag{3.2}$$

where the S_{se} is the contextualized LPS of the noisy signal as the input vector. To deal with the non-stationary property of noises, the context information for the DNN-SE (M_{se}) is chosen to be shorter compared to the required context information for the DNN-AAI system (M_{aai}).

Several choices are possible for the output of DNN-SE system. In a single-task approach the clean LPSs can be used, or in a multi-task approach the clean MFCCs are used in addition to the clean LPSs. Fig. 3.10 shows a sketch of DNN-SE system with multiple output tasks. In the multi-task case, the back propagated loss from the MFCC output acts as a regularizer and would prevent the model from being overfitted to the training LPSs. Moreover, the MFCC output acts as a constraint for the enhanced LPSs to be better predicted [107]. Although MFCC features can be generated from the LPS output, utilizing the MFCCs output by the DNN as a second task in a multi-task case gives us the

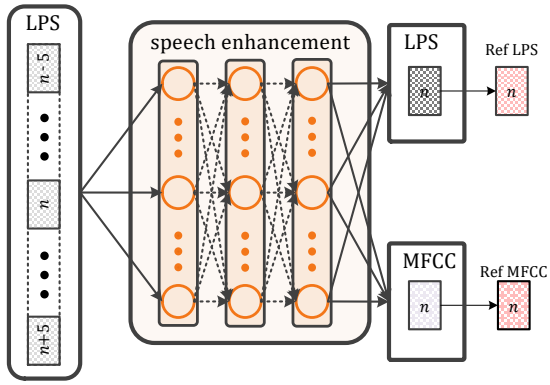


Figure 3.10: DNN based SE system with 120 ms context of noisy LPSs, and clean LPSs and MFCCs as the output.

opportunity of combining the SE and AAI systems to a system that is jointly trained, instead of two separately trained systems.

3.4.2 Joint training of DNN-SE and AAI system

In Sections 2.4.1 and 3.4.1.2, DNN-AAI and DNN-SE systems for inversion and enhancement tasks are described respectively, where the DNN-SE module is employed in a pre-processing step before the target AAI task to be accomplished with the DNN-AAI system. Since these two systems are separately trained and sequentially applied to the input noisy data, it is possible to connect them together and train a single network which performs both SE and AAI tasks jointly.

The joint system is trained with the optimization of MSE loss for LPSs, MFCCs, and TVs. However, the fusion of those two systems into one is challenging, because of different temporal contexts used to build the two systems independently. As it is mentioned in Section 3.4.1.2, the DNN-SE input context size M_{se} is smaller than the M_{aai}. In a joint architecture, the required frames for building the AAI input need to be fed to the DNN-SE module. This issue is being solved by using a sequence of S_{se}s according to the required frame time instances for performing the AAI task. The sequence S_{joint} is built as follows:

$$S_{\text{joint}}[n] = \Big[S_{\text{se}}[n - 2 \times M_{\text{aai}}]^T, \ldots, S_{\text{se}}[n-2]^T, S_{\text{se}}[n]^T, \\ S_{\text{se}}[n+2]^T, \ldots, S_{\text{se}}[n + 2 \times M_{\text{aai}}]^T\Big]. \quad (3.3)$$

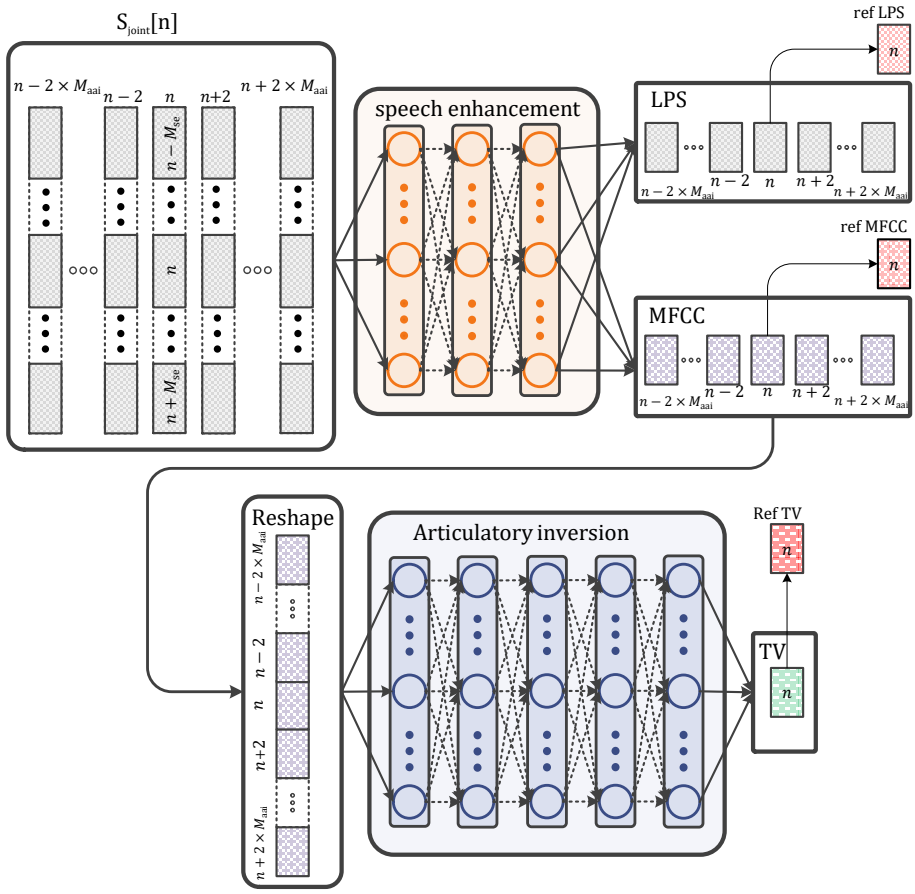


Figure 3.11: Network structure of joint training of SE and AAI systems

In this way, the DNN-SE module generates the required frames for making the vector $X_{\text{aai}}[n]$. During training step in the forward path, the input sequence is mapped to the output in the SE part of joint network, and the loss is only calculated for the S_{se}[n] and back propagated through the network to update the weights. This is illustrated in Fig. 3.11 where the LPS and MFCC tasks are considered for the current time n. The SE part of joint network is estimating the required MFCCs for the AAI task from the input S_{joint}. Further, the estimated sequence of MFCCs is concatenated and fed to the AAI part of network for estimating TVs. The joint architecture performs an AAI task from the enhanced MFCCs which could improve the performance compared to separately applying SE and performing inversion by the AAI-C sys-

tem.

3.4.3 Experimental setups and results

The goal of this work is to evaluate the AAI performance in multi-condition noisy data. The purpose of this evaluation is to utilize the AAI system for real world applications. In the following we describe the datasets we used in this work and various scenarios and experiments for the evaluation of AAI system in presence of noise.

3.4.3.1 Synthetic multi-condition data

For synthetically making noisy data, two datasets are employed. The data which is used for training the AAI systems are corrupted by noises from AURORA 2 dataset [23]. In one of the experiments for training the DNN-SE system, the noises from the Nonspeech dataset [25] are employed to have mismatched in noise scenarios. The simulated noise level is from 0 dB to 20 dB with 5 dB steps. In the following we briefly describe the noise datasets.

- AURORA 2 - AURORA 2 [23] is a speech corpus covering eight different noise types that are recorded in various places, namely, airport, crowd of people (babble), car, exhibition hall, restaurant, street, subway, and train station. Audio recordings contain stationary and non-stationary segments and are sampled at a rate of 8 kHz.
- Nonspeech dataset - The Nonspeech dataset [25], which contains 100 different environmental noises, is recorded with a 20 kHz sampling rate.

3.4.3.2 AAI in presence of noise

After preparation of multi-condition data, we evaluate the performance of AAI-C and AAI-MC models. The AAI-C model is trained based on clean data from HPRC dataset, and multi-condition noisy data is used to train an AAI-MC model. All the reported PCC scores are from performing leave-one-speaker-out cross-validation (LOSO) to incorporate each of the eight available speakers of the HPRC dataset in the test phase. Finally, the PCC is averaged among all eight speakers.

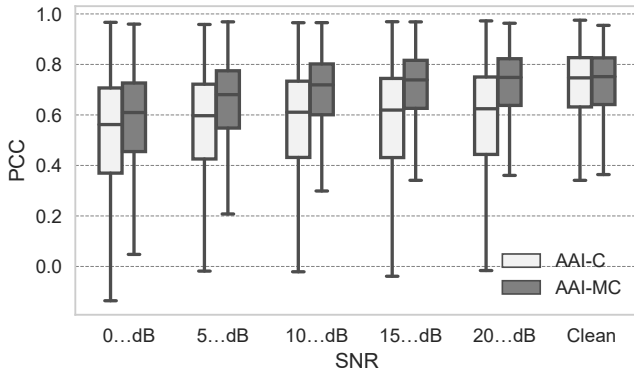


Figure 3.12: Average PCC for multi-condition data with respect to different SNR levels. The box plots represent the minimum, first quartile, median, third quartile, and the maximum of average PCC values.

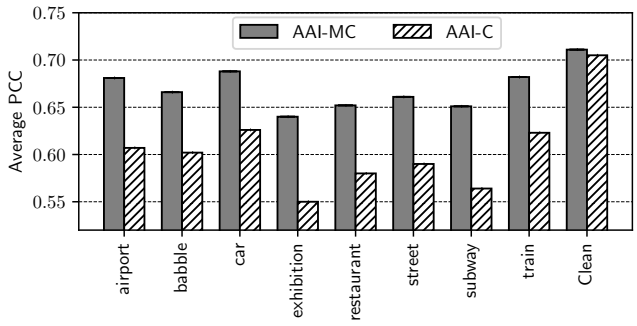


Figure 3.13: Average PCC for multi-condition data on AAI-C and AAI-MC models, with respect to different noise types.

In the first set of experiments the performance of the AAI system in presence of noise is evaluated. The averaged PCC scores for different SNR levels are depicted in Figure 3.12. The AAI-MC system is performing better than the AAI-C model in all of the SNR levels. AAI-MC performs slightly better than the AAI-C for the clean data, which can be explained by having more data and better extracted representations of acoustic space by the network. As is expected, both systems obtain lower PCC scores for the lower SNR levels. Also, the AAI-MC system performs almost the same for SNR $\geq$ 15 dB, and clean data.

The averaged PCC scores for different noise types are depicted in Figure 3.13 for AAI-C and AAI-MC systems. It can be observed that 'exhibition' and 'subway' noises have the greatest negative effects on AAI accuracy and cause a significant performance drop. in contrast, 'car' and 'train' noises have minor negative effects on the final AAI accuracy.

3.4.3.3 AAI performance for enhanced speech

The DNN-SE architecture is the same as [106], three hidden layers with 1024 hidden nodes, ReLU activation function, and the output layer activation is linear. A drop-out [83] rate of 10% was used in each hidden layer to contrast over-fitting. LPSs and MFCCs are extracted for 20 ms frame length with 10 ms frame shift. The LPSs of M_{se}=5 previous and future frames are concatenated and fed to the DNN-SE system. The MFCCs of M_{aai}=8 Three different scenarios have been utilized for training the DNN-SE module. The first scenario (DNN-SE1) is for matched speakers, noise types and SNRs. In the second scenario (DNN-SE2) noise types and SNRs are in matched condition, and speakers are in mismatched condition. In the first and second scenarios, for training data, the HPRC downsampled to 8 kHz dataset is used for speech material and AURORA 2 for noises. In the third scenario (DNN-SE3) speakers, noise types and SNRs are in mismatched condition. Here we use an 8 kHz version of the TIMIT and Nonspeech datasets are used for training data speech material and noises, respectively. The test data in all of the scenarios are the same: the speech data is from an 8 kHz version of the HPRC dataset and noises are from AURORA 2. For the AAI task we have utilized the AAI-C, AAI-MC and the joint model. The enhanced speech is used as input to the AAI-C system and multi-condition speech data is directly fed to AAI-MC and the joint model. The results are reported in Table 3.5. First, we notice from Table 3.5 that AAI-C tested on data enhanced by DNN-SE performs better than AAI-MC tested on multi-condition data. It should be mentioned that the AAI-C and DNN-SE3 systems are independently trained on different data. Therefore, having a DNN-SE as a preprocessing module allows to use an off-the-shelf AAI-C system avoiding training a new system from scratch. The DSP-SE coupled with AAI-C has a contrary behaviour: instead of improving performance, it degrades the performance of inversion system compared to using the noisy data directly.

Table 3.5: Performance of SI-AAI systems trained on clean and multi-condition data and tested on clean, multi-condition and enhanced data.

Test data	Enhancement	AAI-C	AAI-MC	Joint
Clean	None	0.705	0.710	0.707
Multi-Cond	None	0.595	0.665	**0.698**
Multi-Cond	DSP-SE	0.568	—	—
Multi-Cond	DNN-SE1-MT	0.699	—	—
Multi-Cond	DNN-SE1-ST	0.689	—	—
Multi-Cond	DNN-SE2-MT	0.670	—	—
Multi-Cond	DNN-SE2-ST	0.662	—	—
Multi-Cond	DNN-SE3-MT	0.678	—	—

The result is in line with [70]. Signal distortions introduced by DSP-SE could be the explanation, e.g. musical noise [94]. From Table 3.5, we notice the joint model performance is the best for multi-condition noisy data, as expected. It is due to the joint training of SE and AAI modules where the AAI part of network is being trained for the enhanced data.

3.4.4 ASR in noisy condition

For evaluating the proposed AAI model in presence of noise, we conducted several experiments on automatic speech recognition (ASR) for a continuous word recognition task. The WSJ0 [15] database is used in this task. The "dev93" part of WSJ0 corpus is used for training the ASR system and evaluation is carried out on the "eval92" part. Two ASR systems are trained based on an end-to-end architecture, the first one uses only acoustic features as the input (System 1), the second one utilizes both acoustic and articulatory features from the AAI system, as the input for ASR system (System 2). The end-to-end ASR system is based on the end-to-end ESPnet recognizer [98], which leverages both connectionist temporal classification (CTC) loss function, and an attention mechanism [99]. The word error rate (WER) is calculated as the evaluation metric for comparing these two ASR systems. The 8 kHz noisy version of WSJ0 data is synthetically made by adding noises from AURORA 2 at 0 and 10 dB SNRs. FBEs are utilized as the acoustic features, and TVs are used as the articulatory features. TVs are estimated by utilizing AAI-MC, joint and DNN-SE3 together with AAI-C systems. System 1 provides lower and upper bounds for

Table 3.6: WER for the "eval92" part of WSJ database for the two mentioned ASR systems.

Test Condition	System 1	System 2
Clean FBEs	5.3	—-
Clean FBEs + TV (AAI-MC)	—-	5.5
Clean FBEs + TV (DNN-SE+AAI-C)	—-	5.4
Clean FBEs + TV (Joint)	—-	5.4
Enh Clean FBEs	6.1	—-
10 dB FBEs	49.4	—-
10 dB FBEs + TV (AAI-MC)	—-	22.6
10 dB FBEs + TV (DNN-SE+AAI-C)	—-	19.8
10 dB FBEs + TV (Joint)	—-	19.1
Enh 10 dB FBEs	42.3	—-
0 dB FBEs	78.2	—-
0 dB FBEs + TV (AAI-MC)	—-	57.8
0 dB FBEs + TV (DNN-SE+AAI-C)	—-	51.4
0 dB FBEs + TV (Joint)	—-	49.8
Enh 0 dB FBEs	68.4	—-

WER, when we are evaluating on the clean and multi-condition data, respectively. System 2 determines the effect of articulatory information in the ASR task. The results of the ASR experiments are reported in Table 3.6. We can observe that the performance of System 1 is significantly degraded in the presence of noise, while System 2 suffers less from noise compared to System 1. The results allow us to argue the key role of articulatory information in the ASR task for noisy conditions.

CHAPTER 4

Conclusion and Future work

4.1 Conclusion

The thesis aimed to research articulatory inversion (AI) and utilize the estimated articulatory information for the automatic speech recognition (ASR) problem. The primary work was to improve the performance of AI systems by using new architectures and linguistic information. In this way, the performance of AI system improved to better deal with speaker variabilities in mismatched speaker scenarios. The proposed architecture was designed to sense the significant temporal changes in the speech spectrum which result from the articulators' movements, based on acoustic landmarks theory. The linguistic information was employed together with acoustic features to improve the performance of AI where contextual information is available. Using the linguistic information in terms of attribute features (manner and place of articulation) could be useful for cross language AI. In addition, the linguistic features have less variability compared to the acoustic features, and we showed they are a better representation for transfer learning of the articulatory data to TIMIT dataset for performing ASR in clean condition.

We propose a novel architecture by utilizing the 1D convolutional filters in the time domain instead of using hand-crafted features. The proposed approach slightly outperformed hand-crafted features in the matched speaking rate (SR) condition, and significantly in the mismatched SR condition. As we know, SR is another variability in speech which causes significant drop in performance for many speech processing applications, including AI. In this way, the proposed approach

would be a better candidate for estimation of articulatory data that can be utilized in other applications.

At the last part, we explored the AI performance in the presence of noise which is another variability for speech due to the different environments in real world scenarios. We employed a feed-forward deep neural network to perform speech enhancement (SE) and utilized that as a preprocessing step for the AAI. The AI using enhanced speech performed slightly better than training a multi-condition AI. This minor improvement motivated us to perform joint training of SE and AI within a multi-task approach. Due to the difference in required temporal context of SE and AI modules, the architecture design was challenging. To overcome this challenge the flow of data to the network was modified to provide both tasks with the required temporal span of frames. The joint model performed as good as the AI network trained on clean data, even for low signal to noise ratios.

4.2 Future work

The articulatory inversion problem still needs more investigation due to the limited amount of available data compared to other speech tasks. By advancements in the deep learning techniques and approaches, utilizing generative models could be helpful to produce more data. In a more specific way, a conditional generative adversarial network could be a proper way to combine different datasets with articulatory data and generate more articulatory features. Utilizing attribute features in the form of manner and place of articulation would be an interesting approach for cross-language AI, and making the AI system universal to assess the speech production mechanism.

In the applications like pronunciation training where the transcription is available, training separate AI systems based on acoustic and forced aligned linguistic information could shed some lights on how to provide better feedback to second language learners to properly correct their mispronunciations.

Bibliography

[1] Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., Yu, D., 2014. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22, 1533–1545.

[2] Atal, B.S., Chang, J.J., Mathews, M.V., Tukey, J.W., 1978. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique. The Journal of the Acoustical Society of America 63, 1535–1555. doi:10.1121/1.381848.

[3] Badin, P., Youssef, A.B., Bailly, G., Elisei, F., Hueber, T., 2010. Visual articulatory feedback for phonetic correction in second language learning, in: Second Language Studies: Acquisition, Learning, Education and Technology.

[4] Bai, S., Kolter, J.Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 .

[5] Biasutto-Lervat, T., Ouni, S., 2018. Phoneme-to-articulatory mapping using bidirectional gated RNN, in: Interspeech, pp. 3112–3116.

[6] Bishop, C.M., 2006. Pattern recognition and machine learning. Information science and statistics, Springer, New York, NY. URL: https://cds.cern.ch/record/998831. softcover published in 2016.

[7] Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y., 2015. Attention-based models for speech recognition, in: Advances in neural information processing systems, pp. 577–585.

[8] Cohen, I., 2003. Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging. IEEE Transactions on Speech and Audio Processing 11, 466–475.

[9] Cohen, I., Berdugo, B., 2002. Noise estimation by minima controlled recursive averaging for robust speech enhancement. IEEE Signal Processing Letters 9, 12–15. doi:10.1109/97.988717.

[10] Donahue, C., Li, B., Prabhavalkar, R., 2018. Exploring speech enhancement with generative adversarial networks for robust speech recognition, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5024–5028.

[11] Ephraim, Y., Malah, D., 1985. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE transactions on acoustics, speech, and signal processing 33, 443–445.

[12] Fainberg, J., Klejch, O., Loweimi, E., Bell, P., Renals, S., 2019. Acoustic model adaptation from raw waveforms with sincnet, in: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE. pp. 897–904.

[13] Finch, G., 2016. Linguistic terms and concepts. Macmillan International Higher Education.

[14] Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1993. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n 93.

[15] Garofolo, John S., et al. CSR-I (WSJ0) Complete LDC93S6A. Web Download. Philadelphia: Linguistic Data Consortium, 1993. doi:10.35111/q7sb-vv12.

[16] Ghai, B., Ramanan, B., Müller, K., 2019. Does speech enhancement of publicly available data help build robust speech recognition systems? ArXiv abs/1910.13488.

[17] Ghosh, P.K., Narayanan, S., 2010. A generalized smoothness criterion for acoustic-to-articulatory inversion. The Journal of

the Acoustical Society of America 128, 2162–2172. doi:10.1121/1.3455847.

[18] Ghosh, P.K., Narayanan, S., 2011. Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion. The Journal of the Acoustical Society of America 130, EL251–EL257.

[19] Goodfellow, I., Bengio, Y., Courville, A., 2017. Deep learning (adaptive computation and machine learning series). Cambridge Massachusetts , 321–359.

[20] Graves, A., Jaitly, N., 2014. Towards end-to-end speech recognition with recurrent neural networks, in: International conference on machine learning, pp. 1764–1772.

[21] Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks 18, 602–610.

[22] Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S., 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405, 947.

[23] Hirsch, H.G., Pearce, D., 2000. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions, in: ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW).

[24] Hogden, J., Lofqvist, A., Gracco, V., Zlokarnik, I., Rubin, P., Saltzman, E., 1996. Accurate recovery of articulator positions from acoustics: New conclusions based on human data. The Journal of the Acoustical Society of America 100, 1819–1834. URL: https://doi.org/10.1121/1.416001, doi:10.1121/1.416001, arXiv:https://doi.org/10.1121/1.416001.

[25] Hu, G., Wang, D., 2010. A tandem algorithm for pitch estimation and voiced speech segregation. IEEE Transactions on Audio, Speech, and Language Processing 18, 2067–2079.

[26] Hueber, T., Ben Youssef, A., Bailly, G., Badin, P., Elisei, F., 2012. Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training, in: Interspeech, pp. 783–786.

[27] Illa, A., Ghosh, P.K., 2019. Representation learning using convolution neural network for acoustic-to-articulatory inversion, in: ICASSP, pp. 5931–5935.

[28] Illa, A., Ghosh, P.K., 2020. The impact of speaking rate on acoustic-to-articulatory inversion. Computer Speech & Language 59, 75–90.

[29] Imran, A.S., Haflan, V., Shahrebabaki, A.S., Olfati, N., Svendsen, T.K., 2019a. Evaluating acoustic feature maps in 2d-cnn for speaker identification, in: Proceedings of the 2019 11th International Conference on Machine Learning and Computing, pp. 211–216.

[30] Imran, A.S., Shahrebabaki, A.S., Olfati, N., Svendsen, T., 2019b. A study on the performance evaluation of machine learning models for phoneme classification, in: Proceedings of the 2019 11th International Conference on Machine Learning and Computing, pp. 52–58.

[31] Jackson, P.J., Singampalli, V.D., 2009. Statistical identification of articulation constraints in the production of speech. Speech Communication 51, 695 – 710. URL: http://www.sciencedirect.com/science/article/pii/S0167639309000466, doi:https://doi.org/10.1016/j.specom.2009.03.007.

[32] Ji, A., 2014. Speaker independent acoustic-to-articulatory inversion. Ph.D. thesis. University of Maryland, College Park, Maryland.

[33] Juneja, A., 2004. Speech recognition based on phonetic features and acoustic landmarks. Ph.D. thesis.

[34] Kain, A., Macon, M.W., 1998. Spectral voice conversion for text-to-speech synthesis, in: Proceedings of the 1998 IEEE In-

ternational Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181), pp. 285–288 vol.1. doi:10.1109/ICASSP.1998.674423.

[35] Kawahara, H., Masuda-Katsuse, I., De Cheveigne, A., 1999. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. Speech communication 27, 187–207.

[36] Kirchhoff, K., 1999. Robust Speech Recognition Using Articulatory Information. Ph.D. thesis. University of Bielefeld.

[37] Kobayashi, T., Yagyu, M., Shirai, K., 1991. Application of neural networks to articulatory motion estimation, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on, IEEE Computer Society. pp. 489–492.

[38] Koishida, K., Tokuda, K., Kobayashi, T., Imai, S., 1995. Celp coding based on mel-cepstral analysis, in: 1995 International Conference on Acoustics, Speech, and Signal Processing, pp. 33–36 vol.1. doi:10.1109/ICASSP.1995.479266.

[39] Korin, R., Hoole, P., King, S., 2011. Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus, in: Interspeech, Florence, Italy. pp. 1505–1508.

[40] Le, Z., Renals, S., 2008. Acoustic-articulatory modeling with the trajectory HMM. IEEE Signal Processing Letters 15, 245–248.

[41] Lee, C.H., Siniscalchi, S.M., 2013. An information-extraction approach to speech processing: Analysis, detection, verification, and recognition. Proceedings of the IEEE 101(5), 1089–1115.

[42] Lee, J., Cho, K., Hofmann, T., 2017. Fully character-level neural machine translation without explicit segmentation. Transactions of the Association for Computational Linguistics 5, 365–378.

[43] Lee, K.., Hon, H.., 1989a. Speaker-independent phone recognition using hidden markov models. IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 1641–1648.

[44] Lee, K.., Hon, H.., 1989b. Speaker-independent phone recognition using hidden markov models. IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 1641–1648. doi:10.1109/29.46546.

[45] Ling, Z.H., Richmond, K., Yamagishi, J., 2013. Articulatory control of hmm-based parametric speech synthesis using feature-space-switched multiple regression. IEEE Transactions on Audio, Speech, and Language Processing 21, 207–219. doi:10.1109/TASL.2012.2215600.

[46] Ling, Z.H., Richmond, K., Yamagishi, J., Wang, R.H., 2009. Integrating articulatory features into hmm-based parametric speech synthesis. IEEE Transactions on Audio, Speech, and Language Processing 17, 1171–1185. doi:10.1109/TASL.2009.2014796.

[47] Liu, P., Yu, Q., Wu, Z., Kang, S., Meng, H., Cai, L., 2015. A deep recurrent approach for acoustic-to-articulatory inversion, in: ICASSP, pp. 4450–4454.

[48] McGowan, R.S., 1994. Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests. Speech Communication 14, 19 – 48. URL: http://www.sciencedirect.com/science/article/pii/0167639394900558, doi:https://doi.org/10.1016/0167-6393(94)90055-8.

[49] Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S., 2010. Recurrent neural network based language model., in: Interspeech, Makuhari. pp. 1045–1048.

[50] Mitra, V., Nam, H., Espy-Wilson, C.Y., Saltzman, E., Goldstein, L., 2010. Retrieving tract variables from acoustics: A comparison of different machine learning strategies. IEEE Journal of Selected Topics in Signal Processing 4, 1027–1045. doi:10.1109/JSTSP.2010.2076013.

[51] Mitra, V., Nam, H., Espy-Wilson, C.Y., Saltzman, E., Goldstein, L., 2011. Articulatory information for noise robust speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 19, 1913–1924.

[52] Mitra, V., Sivaraman, G., Nam, H., Espy-Wilson, C., Saltzman, E., Tiede, M., 2017. Hybrid convolutional neural networks for articulatory and acoustic information based speech recognition. Speech Communication 89, 103–112.

[53] Moon, T.K., 1996. The expectation-maximization algorithm. IEEE Signal Processing Magazine 13, 47–60. doi:10.1109/79.543975.

[54] Narayanan, S., et al, 2014. Real-time magnetic resonance imaging and electromagnetic articulography database for speech production research (tc). The Journal of the Acoustical Society of America 136, 1307–1311.

[55] Narayanan, S., Toutios, A., Ramanarayanan, V., Lammert, A., Kim, J., Lee, S., Nayak, K., Kim, Y.C., Zhu, Y., Goldstein, L., et al., . USCTIMIT: A database of multimodal speech production data. Technical Report. USC, Tech. Rep., 2013.[Online] http://sail. usc. edu/span/usc-timit ....

[56] Neiberg, D., Ananthakrishnan, G., Engwall, O., 2008. The acoustic to articulation mapping: Non-linear or non-unique?, in: Ninth Annual Conference of the International Speech Communication Association.

[57] Ouni, S., Laprie, Y., 2005. Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion. The Journal of the Acoustical Society of America 118, 444–460.

[58] Papcun, G., Hochberg, J., Thomas, T.R., Laroche, F., Zacks, J., Levy, S., 1992. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. The Journal of the Acoustical Society of America 92, 688–700. URL: https://doi.org/10.1121/1.403994, doi:10.1121/1.403994, arXiv:https://doi.org/10.1121/1.403994.

[59] von Platen, P., Zhang, C., Woodland, P., 2019. Multi-Span Acoustic Modelling Using Raw Waveform Signals , 1393–1397URL: http://dx.doi.org/10.21437/Interspeech.2019-2454, doi:10.21437/Interspeech.2019-2454.

[60] Qin, C., Carreira-Perpiñán, M.Á., 2007. A comparison of acoustic features for articulatory inversion, in: Eighth Annual Conference of the International Speech Communication Association, pp. 2469–2472.

[61] Richmond, K., 2006. A trajectory mixture density network for the acoustic-articulatory inversion mapping, in: Ninth International Conference on Spoken Language Processing, pp. 577–580.

[62] Richmond, K., Hoole, P., King, S., 2011. Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus, in: Twelfth Annual Conference of the International Speech Communication Association.

[63] Rothauser, E., 1969. Ieee recommended practice for speech quality measurements. IEEE Trans. on Audio and Electroacoustics 17, 225–246.

[64] Rumellhart, D., 1986. Learning internal representations by error propagation. Parallel distributed processing 1, 318–362.

[65] Sainath, T.N., Weiss, R.J., Senior, A., Wilson, K.W., Vinyals, O., 2015. Learning the speech front-end with raw waveform CLDNNs, in: Proc. Interspeech 2015, pp. 1–5. doi:10.21437/Interspeech.2015-1.

[66] Schäfer, A.M., Zimmermann, H.G., 2007. Recurrent neural networks are universal approximators. International journal of neural systems 17, 253–263.

[67] Schönle, P.W., Gräbe, K., Wenig, P., Höhne, J., Schrader, J., Conrad, B., 1987. Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. Brain and Language 31, 26–35.

[68] Seki, H., Yamamoto, K., Nakagawa, S., 2017. A deep neural network integrated with filterbank learning for speech recognition, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 5480–5484.

[69] Seneviratne, N., Espy-Wilson, C., 2021. Speech based depression severity level classification using a multi-stage dilated cnn-lstm model. arXiv:2104.04195.

[70] Seneviratne, N., Sivaraman, G., Mitra, V., Espy-Wilson, C., 2018. Noise robust acoustic to articulatory speech inversion, in: Proc. Interspeech 2018, pp. 3137–3141. URL: http://dx.doi.org/10.21437/Interspeech.2018-1509, doi:10.21437/Interspeech.2018-1509.

[71] Shahrebabaki, A.S., Imran, A.S., Olfati, N., Svendsen, T., 2018. Acoustic feature comparison for different speaking rates, in: International Conference on Human-Computer Interaction, Springer, Cham. pp. 176–189.

[72] Shahrebabaki, A.S., Imran, A.S., Olfati, N., Svendsen, T., 2019a. A comparative study of deep learning techniques on frame-level speech data classification. Circuits, Systems, and Signal Processing 38, 3501–3520.

[73] Shahrebabaki, A.S., Marco Siniscalchi, S., Salvi, G., Svendsen, T., 2021a. A dnn based speech enhancement approach to noise robust acoustic-to-articulatory inversion, in: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. doi:10.1109/ISCAS51556.2021.9401290.

[74] Shahrebabaki, A.S., Olfati, N., Imran, A.S., Hallstein Johnsen, M., Siniscalchi, S.M., Svendsen, T., 2021b. A two-stage deep modeling approach to articulatory inversion, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6453–6457. doi:10.1109/ICASSP39728.2021.9413742.

[75] Shahrebabaki, A.S., Olfati, N., Imran, A.S., Siniscalchi, S.M., Svendsen, T., 2019b. A phonetic-level analysis of different input features for articulatory inversion., in: INTERSPEECH, pp. 3775–3779.

[76] Shahrebabaki, A.S., Olfati, N., Siniscalchi, S.M., Salvi, G., Svendsen, T., 2020a. Transfer learning of articulatory information through phone information. Proc. Interspeech 2020 , 2877–2881.

[77] Shahrebabaki, A.S., Salvi, G., Svendsen, T., Siniscalchi, S.M., 2022. Acoustic-to-articulatory mapping with joint optimization

of deep speech enhancement and articulatory inversion models. IEEE/ACM Transactions on Audio, Speech, and Language Processing 30, 135–147. doi:10.1109/TASLP.2021.3133218.

[78] Shahrebabaki, A.S., Siniscalchi, S.M., Salvi, G., Svendsen, T., 2020b. Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals. database 1, 5.

[79] Shahrebabaki, A.S., Siniscalchi, S.M., Svendsen, T., . Raw speech-to-articulatory inversion by temporal filtering and decimation. Submitted to Interspeech 2021 .

[80] Siniscalchi, S.M., Lee, C.H., 2009. A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition. Speech Communication 51, 1139–1153.

[81] Sivaraman, G., 2017. Articulatory representations to address acoustic variability in speech. Ph.D. thesis. University of Maryland, College Park.

[82] Sivaraman, G., Mitra, V., Nam, H., Tiede, M., Espy-Wilson, C., 2019. Unsupervised speaker adaptation for speaker independent acoustic to articulatory speech inversion. The Journal of the Acoustical Society of America 146, 316–329. URL: https://doi.org/10.1121/1.5116130, doi:10.1121/1.5116130, arXiv:https://doi.org/10.1121/1.5116130.

[83] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

[84] Stevens, K.N., 1981. Evidence for the role of acoustic boundaries in the perception of speech sounds. The Journal of the Acoustical Society of America 69, S116–S116.

[85] Stevens, K.N., 2000. Acoustic phonetics. volume 30. MIT press.

[86] Stevens, K.N., 2002. Toward a model for lexical access based on acoustic landmarks and distinctive features. The Journal of the Acoustical Society of America 111, 1872–1891.

[87] Sun, J., Deng, L., 2002. An overlapping-feature-based phonological model incorporating linguistic constraints: Applications to speech recognition. The Journal of the Acoustical Society of America 111, 1086–1101.

[88] Tan, Z.H., Kraljevski, I., 2014. Joint variable frame rate and length analysis for speech recognition under adverse conditions. Computers & Electrical Engineering 40, 2139–2149. URL: https://www.sciencedirect.com/science/article/pii/S0045790614002304, doi:https://doi.org/10.1016/j.compeleceng.2014.09.002.

[89] Tang, M., Seneff, S., Zue, V.W., 2003. Modeling linguistic features in speech recognition, in: Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003), pp. 2585–2588.

[90] Tiede, M., Espy-Wilson, C.Y., Mitra, D.G.V., Nam, H., Sivaraman, G., 2017. Quantifying kinematic aspects of reduction in a contrasting rate production task. The Journal of the Acoustical Society of America 141, 3580–3580.

[91] Toda, T., Black, A.W., Tokuda, K., 2007. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Transactions on Audio, Speech, and Language Processing 15, 2222–2235. doi:10.1109/TASL.2007.907344.

[92] Toda, T., Black, A.W., Tokuda, K., 2008. Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. Speech Communication 50, 215–227.

[93] Toutios, A., Margaritis, K., 2005. A support vector approach to the acoustic-to-articulatory mapping, in: Ninth European Conference on Speech Communication and Technology, pp. 3221–3224.

[94] Tran, T.D., Nguyen, Q.C., Nguyen, D.K., 2010. Speech enhancement using modified imcra and omlsa methods, in: International Conference on Communications and Electronics 2010, pp. 195–200.

[95] Triantafyllopoulos, A., Keren, G., Wagner, J., Steiner, I., Schuller, B.W., 2019. Towards Robust Speech Emotion

Recognition Using Deep Residual Networks for Speech Enhancement, in: Proc. Interspeech 2019, pp. 1691–1695. URL: http://dx.doi.org/10.21437/Interspeech.2019-1811, doi:10.21437/Interspeech.2019-1811.

[96] Uria, B., Renals, S., Richmond, K., 2011. A deep neural network for acoustic-articulatory speech inversion, in: NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning, pp. 1–9.

[97] Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., Saurous, R.A., 2017. Tacotron: Towards end-to-end speech synthesis, in: Proc. Interspeech 2017, pp. 4006–4010. URL: http://dx.doi.org/10.21437/Interspeech.2017-1452, doi:10.21437/Interspeech.2017-1452.

[98] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., Ochiai, T., 2018. Espnet: End-to-end speech processing toolkit, in: Proc. Interspeech 2018, pp. 2207–2211. URL: http://dx.doi.org/10.21437/Interspeech.2018-1456, doi:10.21437/Interspeech.2018-1456.

[99] Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T., 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing 11, 1240–1253. doi:10.1109/JSTSP.2017.2763455.

[100] Westbury, J.R., Turner, G., Dembowski, J., 1994. X-ray microbeam speech production database user's handbook. University of Wisconsin .

[101] Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometrics and intelligent laboratory systems 2, 37–52.

[102] Wrench, A., 1999. The MOCHA-TIMIT articulatory database. http://www.cstr.ed.ac.uk/artic/mocha.html.

[103] Wu, Z., Zhao, K., Wu, X., Lan, X., Meng, H., 2015. Acoustic to articulatory mapping with deep neural network. Multimedia Tools and Applications 74, 9889–9907.

[104] Xie, X., Liu, X., Wang, L., 2016. Deep neural network based acoustic-to-articulatory inversion using phone sequence information, in: Interspeech 2016, pp. 1497–1501.

[105] Xiong, F., Barker, J., Christensen, H., 2018. Deep learning of articulatory-based representations and applications for improving dysarthric speech recognition, in: Speech Communication; 13th ITG-Symposium, pp. 1–5.

[106] Xu, Y., Du, J., Dai, L., Lee, C., 2015. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing 23, 7–19.

[107] Xu, Y., Du, J., Huang, Z., Dai, L.R., Lee, C.H., 2017. Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement. `arXiv:1703.07172`.

[108] Yuan, J., Liberman, M., 2008. Speaker identification on the scotus corpus. Journal of the Acoustical Society of America 123, 3878.

[109] Zhu, P., Lei, X., Chen, Y., 2015. Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings, in: Interspeech, pp. 2192–2196.

Articles

Paper A

A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion

Abdolreza Sabzi Shahrebabaki and Negar Olfati, Ali Shariq Imran and Sabato Marco Siniscalchi and Torbjørn Svendsen

INTERSPEECH 2019

INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A phonetic-level analysis of different input features for articulatory inversion

Abdolreza Sabzi Shahrebabaki[1], *Negar Olfati*[1], *Ali Shariq Imran*[1], *Sabato Marco Siniscalchi*[2], *Torbjørn Svendsen*[1]

[1]Department of Electronic Systems, NTNU
[2]Department of Telematics, Kore University of Enna

{abdolreza.sabzi, negar.olfati, ali.imran, torbjorn.svendsen}@ntnu.no,
marco.siniscalchi@unikore.it

Abstract

The challenge of articulatory inversion is to determine the temporal movement of the articulators from the speech waveform, or from acoustic-phonetic knowledge, e.g. derived from information about the linguistic content of the utterance. The actual position of the articulators is typically obtained from measured data, in our case position measurements obtained using EMA (Electromagnetic articulography). In this paper, we investigate the impact on articulatory inversion problem by using features derived from the acoustic waveform relative to using linguistic features related to the time aligned phone sequence of the utterance. Filterbank energies (FBE) are used as acoustic features, while phoneme identities and (binary) phonetic attributes are used as linguistic features. Experiments are performed on a speech corpus with synchronously recorded EMA measurements and employing a bidirectional long short-term memory (BLSTM) that estimates the articulators' position. Acoustic FBE features performed better for vowel sounds. Phonetic features attained better results for nasal and fricative sounds except for /h/. Further improvements were obtained by combining FBE and linguistic features, which led to an average relative RMSE reduction of 9.8%, and a 3% relative improvement of the Pearson correlation coefficient.

Index Terms: Articulatory inversion, language learning, bidirectional long short term memory, Attributes, HPRC database

1. Introduction

Acoustic to articulatory inversion (AAI) is a challenging problem due to the many-to-one mapping in which different articulator positions may produce a similar sound. This many-to-one mapping makes AAI a highly non-linear problem. In AAI, the objective is to estimate the vocal tract shape, which is estimated by the articulator positions based on the uttered speech. AAI can be useful in many speech-based applications, in particular, speech synthesis [1], automatic speech recognition (ASR) [2, 3, 4] and second language learning [5, 6]. Over the years, researchers have addressed this problem employing various machine learning techniques including codebooks [7], Gaussian mixture models (GMM) [8], hidden Markov models (HMM) [9], mixture density networks [10], deep neural networks (DNNs) [11, 12, 13], and deep recurrent neural networks (RNNs) [14, 15, 16].

Exploiting RNNs for the AAI task has demonstrated better results compared to DNNs [14, 16] because the temporal dynamic behavior is better captured through the memory elements of those recurrent architectures. Acoustic features are commonly employed at the input of the AAI system [7, 8, 9, 10] , but linguistic features have been successfully used in recent years either as stand-along features [17], or together with acoustic features [15]. Moreover, representing the linguistic features in a bottleneck form extracted from a phone classifier has been used in [16]. Although leveraging knowledge from linguistic content together with acoustic features has proven to improve AAI systems, a deeper analysis explaining why redundant information makes the system perform better is missing. We think that gaining a better understanding about such a performance improvement would be helpful for some specific tasks, where the linguistic features are available from the text, e.g. language learning. This motivates us to compare state-of-the-art methods in [16, 17] and carry out additional analyses on the acoustic and linguistic features within phoneme boundaries which later can be employed in pronunciation scoring. That is, we focus on the evaluation in time intervals concerning a single phoneme instead of analyzing the whole EMA trajectory for the uttered speech. The rest of the paper is structured as follows. Section 2 presents Deep BLSTM recurrent neural networks. Section 3 describes the "Haskins Production Rate Comparison database"(HPRC) [18]. The database, feature representation, and the performance measurements undertaken in this study, followed by results in Section 4. Finally, Section 5 concludes the paper.

2. Deep BLSTM recurrent neural network

Recurrent neural networks (RNN) have been utilized in many speech technology areas including speech recognition [19], language modeling [20], and articulatory inversion [14, 15, 16]. They are able to estimate any output samples from dynamical systems [21], conditioned on their previous samples. Having a non-causal condition by access to both past and future input samples, we can employ a bidirectional RNN to use the past samples within the forward layer and the future samples within the backward layer as it is shown in Fig. 1. Diamonds show the merge strategy of forward and backward layers output which can be summation, concatenation, and etc. LSTM is a variant of RNN with a specific memory cell architecture for updating the hidden layers. This memory cell is formulated as follows

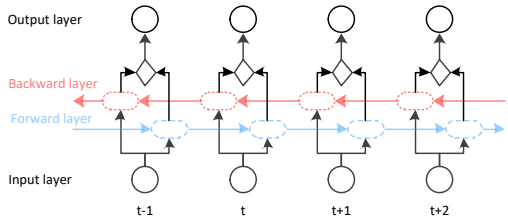


Figure 1: *A bidirectional RNN.*

http://dx.doi.org/10.21437/Interspeech.2019-2526

$$i_t = \sigma\big(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\big) \quad (1)$$

$$f_t = \sigma\big(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\big) \quad (2)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_c\big(W_{cx}x_t + W_{ch}h_{t-1} + b_c\big) \quad (3)$$

$$o_t = \sigma\big(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o\big) \quad (4)$$

$$h_t = o_t \circ g_o(c_t) \quad (5)$$

where x_t and h_t are input and hidden vector, i_t, f_t, c_t and o_t are the input gate, forget gate, cell vector and output gate, respectively. The σ is sigmoid function, g_c and g_o are the activation function which is usually chosen as $tanh$, b shows the bias vector for each gate (b_f is the forget gate bias vector) and weight matrices W with different subscripts which show the connection between input/output with gates, for example, W_{ix} is the input gate-input weight matrix. The operator $\circ$ shows the element-wise multiplication. A bidirectional LSTM (BLSTM) is realizable by using the LSTM memory cells (dotted ovals) in the forward and backward layer as shown in Fig. 1.

3. Experimental Setup

3.1. EMA database

There exists several techniques to measure the articulatory movements, e.g. MRI, microbeam x-ray, and electromagnetic articulography (EMA). Among them, EMA is the most widely used technique to simultaneously capture the speech and articulatory data. MOCHA-TIMIT [22], MNGU0 [23], and USC-TIMIT [24] are speech corpora which contain EMA data. Another such database is "Haskins Production Rate Comparison"(HPRC) [18]. This database contains EMA readings from eight native American English speakers, four male and four female. This database has 720 recorded sentences at normal and fast Speaking Rate (SR), respectively. Some of the sentences are uttered two times in the normal SR by each speaker. Table 2 shows the amount of data for different SR, where "N1", "N2", and "F1"represent the normal SR, repetition of some of the sentences with the normal SR, and fast SR respectively. The sampling rate of the recorded audio files is 44.1 KHz and the EMA recordings are sampled at 100 Hz. The EMA readings are obtained from the sensors placed in different locations of the mouth, tongue, and jaw. Precisely, eight sensors are used in this case which are placed on tongue rear/dorsum (TR), tongue blade (TB), tongue tip (TT), upper lip (UL), lower lip (LL), mouth left (ML), lower incisors/jaw (JAW), and jaw left (JAWL). The EMA readings of the articulatory movements from these carefully placed sensors is measured in the midsagittal plane in X, Y, and Z directions. The X-direction denotes the movement of the articulators from posterior to anterior, the Y denotes the right to left movement, while the Z denotes the inferior to superior articulatory movements. In this paper, we used six reading locations of X and Z direction, i.e., TR, TB, TT, UL, LL, and JAW. In other databases mentioned above these six locations are mostly used, while the Y direction is omitted as the contribution of right to left movement does not contribute much under normal continuous speech.

3.2. Input representation.

3.2.1. Acoustic representation

The acoustic features are extracted from audio downsampled to 16 KHz, using 25 ms frame length and 10 ms frame shift. The resulting features have a 100 Hz sampling rate, the same as the articulatory features. The acoustic features are calculated from smoothed spectrum by the STRAIGHT method [25] with 40 filters which are linearly spaced on Mel-scale frequency axis. The energies in the overlapping frequency bands are called filter bank energy (FBE) features. The extracted feature frame is concatenated with the M previous and future time for each frame as the network input.

3.2.2. Phonetic representation

Spoken utterances have been labeled with the Penn phonetics lab forced aligner [26]. There are 61 phonetic categories which are folded onto 39 categories [27] for TIMIT database [28] which are depicted in the first row of Table 1. Each phoneme is represented as a one-hot 39-dimensional vector [17].

3.2.3. Attribute representation

With the reduced phoneme set from 61 to 39, we use a mapping from phoneme to their phonological features known as attributes which are depicted in Table 1. We consider 22 attributes in this study, comprising manner and place of articulation for both vowel and consonant categories [29]. The attribute features are binary and more than one attribute feature is often active at the same time. These features are more language universal [30] compared to the phonetic representations.

3.3. Performance measurements

To measure the performance of the AAI methods, root mean squared error (RMSE) and Pearson's correlation coefficient (PCC) are chosen. The first criterion reports the deviation and the latter measures the similarity between estimated and the ground-truth trajectories. These measures are defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_i \big(y(i) - \hat{y}(i)\big)^2}, \quad (6)$$

$$\mathrm{PCC} = \frac{\sum_i (y(i) - \bar{y})(\hat{y}(i) - \bar{\hat{y}})}{\sqrt{\sum_i \big(y(i) - \bar{y}\big)^2 \sum_i \big(\hat{y}(i) - \bar{\hat{y}}\big)^2}}, \quad (7)$$

where $y(i)$ and $\hat{y}(i)$ are the ground-truth and estimated EMA value of the i^{th} frame respectively and $\bar{y}$ and $\bar{\hat{y}}$ are mean values of $y(i)$ and $\hat{y}(i)$. All results are based on training on the N1 subset and test on the N2 subset. 5% of the training data is used as the validation data which is used to stop training, to prevent the network from getting over-fitted to the training data.

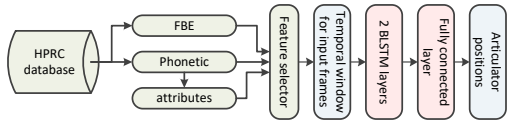


Figure 2: *Network structure for the articulatory inversion system*

3.4. Deep neural network architecture

The block diagram of the network architecture is shown in Fig. 2. It contains a feature selector module that selects among the input features to either output them individually or combine them two by two. The output of the feature selector goes to

Table 1: *American-English phonemes and associated attributes in terms of manner and place of articulations*

	ɑ	æ	ʌ	aʊ	aɪ	b	ʧ	d	ð	ɾ	ɛ	ɝ	eɪ	f	g	h	ɪ	i	ʤ	k	l	m	n	ŋ	oʊ	ɔɪ	p	ɹ	s	ʃ	t	θ	ʊ	u	v	w	j	z	sil
Vowel	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
Fricative	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0
Nasal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stop	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
Approx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0
High	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0
Coronal	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
Dental	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Glottal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Labial	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
Low	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Mid	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Retroflex	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Velar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Voiced	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0	1	1	1	1	1	1	0
Round	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0
Tense	1	1	0	1	1	0	1	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0	0	0
Anterior	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	0	1	0
Back	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
Continuant	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0
Vocalic	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0
Silence	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 2: *Available amount of data for different speaking rates for the HPRC database.*

Speaking rate	NO. utterances	Amount of data
N1	5756	$\sim$ 244 (minutes)
N2	1379	$\sim$ 55 (minutes)
F1	5735	$\sim$ 173 (minutes)

the next module, namely the temporal window for input frames. It takes M past and future frames and feeds this temporal context window to the BLSTM layers consisting of 128 cells in both forward and backward directions. $M = 15$ is used for this work which is resulted from some primary experiments. Sigmoid and tanh activation functions are used for the recurrent layers. At last, a fully connected network with the linear activation is used to map the BLSTM outputs to the articulator positions. The implementations used Keras [31] with TensorFlow backend [32].

4. Results

In this section, we evaluate the performance of different inputs to the AAI system. For having a fair comparison between different features, we used the same architecture for both state-of-the-art methods [16, 17] introduced in 3.4. However, feeding inputs directly to the BLSTM, instead of having several fully connected layers prior to BLSTM performs slightly better. We used the bottleneck features proposed in [16] but we got same performance as phonetic features. We argue that the reason is that the phonetic features are already a parsimonious representation of the input speech capturing information similar to that captured by bottlenecks. The experiments are done speaker dependently. The same context window of 15 past and future frames is used for all input features. Tables 4 and 5 show the PCC and RMSE results for each speaker, considering the acoustic (FBE), phonetic (Phn) and attribute (AF) features in the first three columns, and the combined features in the second three columns (FBE+Phn, FBE+AF, Phn+AF). Comparing the results, we observe that attribute features give worse results than the phonetic features for all speakers. The RMSE results of 4 show that combining features improve performance relative to stand-alone features. The RMSE improvement is in the range of 0.1 mm to 0.15 mm. Similarly, the PCC improvement for each speaker after combining the features is in the range of 0.01 to 0.03, as shown in table 5.

Table 3: *Average RMSE for phoneme*

	FBE	**Phn**	**AF**	**FBE+Phn**	**FBE+AF**	**Phn+AF**
/ɑ/	**2.05**	2.16	2.31	**1.87**	1.89	2.16
/æ/	**1.96**	2.21	2.22	**1.83**	1.88	2.17
/ʌ/	**1.94**	1.99	2.01	**1.77**	1.78	1.98
/aʊ/	**2.04**	2.21	2.32	1.80	1.86	2.21
/aɪ/	**2.00**	2.15	2.32	1.83	1.83	2.15
/ɛ/	**1.88**	1.94	1.95	1.74	1.74	1.96
/ɝ/	**1.87**	1.9	1.91	**1.71**	1.72	1.88
/eɪ/	**1.71**	1.82	1.83	**1.62**	1.63	1.83
/ɾ/	1.83	**1.81**	1.85	**1.65**	1.66	1.81
/i/	1.76	**1.71**	1.72	1.59	**1.58**	1.69
/oʊ/	2.06	**2.00**	2.05	1.79	**1.77**	2.02
/ɔɪ/	**1.99**	2.15	2.7	1.82	**1.81**	2.13
/ʊ/	1.89	**1.87**	1.91	**1.64**	1.70	1.94
/u/	1.84	**1.82**	1.84	**1.66**	**1.66**	1.79
/ʧ/	1.68	**1.66**	1.67	**1.56**	**1.56**	1.66
/ð/	1.96	**1.86**	**1.86**	**1.75**	1.76	1.86
/f/	1.84	**1.80**	1.82	**1.65**	1.66	1.77
/h/	**2.05**	2.26	2.27	**1.85**	1.92	2.28
/ʤ/	1.67	**1.64**	1.68	1.57	**1.55**	1.67
/s/	1.61	**1.57**	1.59	**1.48**	1.49	1.57
/ʃ/	**1.61**	1.63	1.60	1.49	**1.48**	1.59
/θ/	1.97	**1.75**	1.78	**1.63**	1.64	1.75
/v/	2.06	**1.89**	1.90	**1.72**	1.74	1.86
/z/	1.66	**1.61**	**1.61**	**1.51**	1.52	1.58
/m/	2.06	**1.95**	1.98	**1.80**	1.82	1.94
/n/	1.97	**1.89**	1.92	**1.73**	**1.73**	1.88
/ŋ/	2.13	**1.90**	1.92	**1.75**	**1.75**	1.9
/b/	**1.89**	1.90	1.90	1.75	**1.72**	1.87
/d/	**1.91**	1.92	1.93	**1.72**	1.74	1.89
/r/	–	–	–	–	–	–
/g/	2.04	**1.85**	1.88	**1.70**	1.72	1.84
/k/	**1.97**	**1.97**	1.99	1.76	**1.75**	1.94
/p/	2.01	**1.96**	2.03	**1.72**	1.74	1.94
/t/	**1.91**	1.94	1.96	**1.72**	1.73	1.91
/l/	1.87	**1.85**	1.87	1.69	**1.68**	1.82
/ɫ/	**1.95**	2.09	2.09	**1.82**	1.83	2.09
/w/	2.02	**1.98**	2.02	**1.79**	1.82	1.98
/j/	1.77	**1.74**	1.82	1.61	**1.58**	1.72

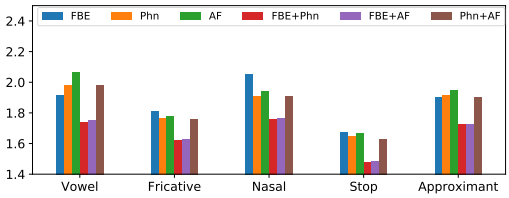


Figure 3: *Average RMSE for manner of articulation from estimated trajectory by different input features*

To gain a better understanding of the performance of each input feature, RMSE results for all phones averaged over all speakers and articulator positions is calculated for the input features investigated. This is depicted in table 3. A compact form of table 3 in terms of five phonetic classes is represented in Fig. 3. These phonetic classes are "vowel", "fricative", "nasal", "stop"and "approximant". We can conclude that FBE works better in case of vowels for stand-alone input features. By a deeper inspection on the results in Table 3, we can say FBE works better for phones where the place of articulation is low (/ɑ/, /æ/, /aʊ/, /aɪ/ and /ɔɪ/), whilst Phn works better in the case of high (/ɪ/, /i/, /ʊ/ and /u/). For all fricatives except /h/, Phn and AF perform better than FBE according to the Fig. 3 and Table 3. Phn and AF features are also better for nasals. In case of stops, all of the features are performing more or less the same except /g/ in which Phn is superior to FBE by 0.19 mm in RMSE.

Table 4: *RMSE for different input features.*

Spk.	FBE	Phn	AF	FBE+Phn	FBE+AF	Phn+AF
F1	**1.356**	1.365	1.405	**1.196**	1.221	1.352
F2	**1.601**	1.631	1.663	**1.449**	1.451	1.632
F3	1.308	**1.302**	1.320	**1.201**	1.202	1.293
F4	**1.469**	1.601	1.625	**1.308**	1.329	1.585
M1	1.208	**1.158**	1.173	1.074	**1.073**	1.151
M2	**1.667**	1.715	1.745	1.536	**1.530**	1.707
M3	1.565	**1.539**	1.566	**1.426**	1.441	1.523
M4	1.259	**1.235**	1.264	**1.124**	1.128	1.231
Avg.	**1.429**	1.443	1.470	**1.289**	1.296	1.434

Table 5: *PCC for different input features.*

Spk.	FBE	Phn	AF	FBE+Phn	FBE+AF	Phn+AF
F1	0.918	**0.921**	0.915	**0.937**	0.936	0.921
F2	0.848	**0.850**	0.845	**0.879**	0.878	0.852
F3	0.821	**0.830**	0.826	**0.850**	**0.850**	0.834
F4	**0.901**	0.894	0.890	**0.922**	0.920	0.895
M1	0.869	**0.883**	0.880	**0.897**	0.896	0.886
M2	**0.860**	0.855	0.850	**0.880**	**0.880**	0.856
M3	0.821	**0.831**	0.823	**0.850**	0.847	0.834
M4	0.832	**0.846**	0.839	0.863	**0.865**	0.845
Avg.	0.859	**0.864**	0.858	**0.885**	0.884	0.865

Moreover, RMSE for each of the articulator positions is calculated by different input features and shown in Table 6. By comparing different individual features we can see there are 0.01 to 0.05 mm differences in RMSE. In the combined input features, combination of FBE and phonetic features gives a better performance in most of cases. Moreover, there is not a big difference (less than 0.01 mm RMSE) between combining FBE with phonetic and attribute features, which are more universal among languages and the network architecture would not need any changes for using it in transfer learning for new languages.

Table 6: *Performance of AAI system in terms of RMSE.*

	FBE	Phn	AF	FBE+Phn	FBE+AF	Phn+AF
TD_x	**1.539**	1.568	1.596	1.426	**1.424**	1.555
TD_z	1.904	**1.868**	1.941	**1.667**	1.680	1.861
TB_x	**1.729**	1.759	1.788	**1.563**	1.564	1.742
TB_z	**1.864**	1.928	2.005	**1.690**	1.699	1.916
TT_x	**1.851**	1.878	1.896	1.665	**1.662**	1.857
TT_z	**1.922**	1.966	1.989	**1.711**	1.715	1.959
UL_x	**0.715**	0.722	0.727	**0.665**	0.668	0.718
UL_z	**1.214**	1.288	1.297	**1.129**	1.138	1.279
LL_x	0.863	**0.822**	0.844	**0.794**	0.802	0.821
LL_z	0.817	**0.750**	0.754	**0.726**	0.727	0.747
JAW_x	1.010	**0.999**	1.016	**0.910**	0.920	0.992
JAW_z	**1.725**	1.772	1.794	**1.527**	1.552	1.765

5. Conclusions

The problem of acoustic to articulatory inversion is addressed in this paper for different input feature types for a two-hidden layer BLSTM with 128 cells in each of its forward and backward layers. FBE features are chosen as the acoustic features and phonetic and attribute features are selected as the linguistic features. The experiments are conducted on a multi-speaker database which will be useful for further investigations on the speaker independent AAI systems. RMSE and PCC are computed for both stand-alone and combined features. Phonetic features have better capability of modelling vowels where the place of articulation is high whilst the vowels with the low place of articulation are better modelled by FBE features. Attribute features combined with acoustic features improve the articulatory inversion performance and will be helpful for transfer learning in case of new languages. Future works will focus on the jointly training of speakers and try building up a speaker independent framework by using linguistic features as the initial estimates.

6. Acknowledgements

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt.

7. References

[1] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of hmm-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 207–219, 2013.

[2] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[3] P. K. Ghosh and S. Narayanan, "Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. EL251–EL257, 2011.

[4] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "Integrating articulatory data in deep neural network-based acoustic modeling," *Computer Speech & Language*, vol. 36, pp. 173–195, 2016.

[5] P. Badin, A. B. Youssef, G. Bailly, F. Elisei, and T. Hueber, "Visual articulatory feedback for phonetic correction in second language learning," in *Second Language Studies: Acquisition, Learning, Education and Technology*, 2010.

[6] T. . B. P. . B. G. Youssef, Atef Ben / Hueber, "Toward a multispeaker visual articulatory feedback system," in *Proc. Interspeech 2011*, 2011, pp. 589–592.

[7] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatorytoacoustic transformation in the vocal tract by a computersorting technique," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1555, 1978.

[8] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[9] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[10] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[11] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[12] P. L. Tobing, H. Kameoka, and T. Toda, "Deep acoustic-to-articulatory inversion mapping with latent trajectory modeling," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec 2017, pp. 1274–1277.

[13] N. Seneviratne, G. Sivaraman, V. Mitra, and C. Espy-Wilson, "Noise robust acoustic to articulatory speech inversion," in *Proc. Interspeech 2018*, 2018, pp. 3137–3141. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1509

[14] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4450–4454.

[15] L. . C. Y. Zhu, Pengcheng / Xie, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Proc. Interspeech 2015*, 2015, pp. 2192–2196.

[16] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-659

[17] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-articulatory mapping using bidirectional gated rnn," in *Proc. Interspeech 2018*, 2018, pp. 3112–3116. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1202

[18] M. Tiede, C. Y. Espy-Wilson, D. Goldenberg, V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[19] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[20] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[21] A. M. Schäfer and H.-G. Zimmermann, "Recurrent neural networks are universal approximators," *International journal of neural systems*, vol. 17, no. 04, pp. 253–263, 2007.

[22] A. Wrench, "The MOCHA-TIMIT articulatory database," http://www.cstr.ed.ac.uk/artic/mocha.html, Queen Margaret Univ. College, Edinburgh, U.K., 1999.

[23] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Proc. Interspeech*, Florence, Italy, August 2011, pp. 1505–1508.

[24] S. Narayanan, A. Toutios, V. Ramanarayanan, A. Lammert, J. Kim, S. Lee, K. Nayak, Y.-C. Kim, Y. Zhu, L. Goldstein *et al.*, "Real-time magnetic resonance imaging and electromagnetic articulography database for speech production research (tc)," *The Journal of the Acoustical Society of America*, vol. 136, no. 3, pp. 1307–1311, 2014.

[25] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveign, "Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds1speech files available. see http://www.elsevier.nl/locate/specom1," *Speech Communication*, vol. 27, no. 3, pp. 187 – 207, 1999.

[26] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3878, 2008.

[27] K. . Lee and H. . Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, Nov 1989.

[28] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[29] S. M. Siniscalchi and C.-H. Lee, "A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition," vol. 51, 2009, pp. 1139–1153.

[30] C.-H. Lee and S. M. Siniscalchi, "An information-extraction approach to speech processing: Analysis, detection, verification, and recognition," *Proceedings of the IEEE*, vol. 101(5), pp. 1089–1115, 2013.

[31] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

Paper B

Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals

Abdolreza Sabzi Shahrebabaki and Sabato Marco Siniscalchi and Giampiero Salvi and Torbjørn Svendsen

INTERSPEECH 2020

Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals

Abdolreza Sabzi Shahrebabaki[1], *Sabato Marco Siniscalchi*[2], *Giampiero Salvi*[1,3], *Torbjørn Svendsen*[1]

[1]Department of Electronic Systems, NTNU
[2]Department of Computer Engineering, Kore University of Enna
[3] KTH Royal Institute of Technology, Dept. of Electrical Engineering and Computer Science
{abdolreza.sabzi, giampiero.salvi, torbjorn.svendsen}@ntnu.no,
marco.siniscalchi@unikore.it

Abstract

We propose a new acoustic-to-articulatory inversion (AAI) sequence-to-sequence neural architecture, where spectral sub-bands are independently processed in time by 1-dimensional (1-D) convolutional filters of different sizes. The learned features maps are then combined and processed by a recurrent block with bi-directional long short-term memory (BLSTM) gates for preserving the smoothly varying nature of the articulatory trajectories. Our experimental evidence shows that, on a speaker dependent AAI task, in spite of the reduced number of parameters, our model demonstrates better root mean squared error (RMSE) and Pearson's correlation coefficient (PCC) than a both a BLSTM model and an FC-BLSTM model where the first stages are fully connected layers. In particular, the average RMSE goes from 1.401 when feeding the filterbank features directly into the BLSTM, to 1.328 with the FC-BLSTM model, and to 1.216 with the proposed method. Similarly, the average PCC increases from 0.859 to 0.877, and 0.895, respectively. On a speaker independent AAI task, we show that our convolutional features outperform the original filterbank features, and can be combined with phonetic features bringing independent information to the solution of the problem. To the best of the authors' knowledge, we report the best results on the given task and data.
Index Terms: Acoustic-to-articulatory inversion, deep learning, sequence-to-sequence neural models, 1-D convolution.

1. Introduction

The acoustic to articulatory inversion (AAI) problem is about estimating the vocal tract shape in the form of articulator positions based on the uttered speech. The actual articulatory positions can be obtained from speakers through different techniques, such as MRI [1], X-ray microbeam [2], and electromagnetic articulography (EMA) [3]. In recent years, AAI has attracted increasing attention because of its suitability in different applications, namely speech synthesis [4, 5], second language learning [6, 7], and automatic speech recognition (ASR) [8]. In a companion paper submitted to this conference [9], we show that AAI is beneficial for the continuous phone recognition task. Unfortunately, this inversion problem is highly nonlinear and non-unique [10, 11], which means that different articulator configurations can produce the same sound. In addition *coarticulation* [12], i.e. the impact of adjacent phonemes on the articulators' movement, makes the AAI problem harder.

Different machine learning techniques and various input representations have been proposed to address the AAI task. For example, search of joint acoustic and articulatory space codebooks [13], Gaussian mixture models (GMMs) [14], hidden Markov models (HMMs) [7], mixture density networks (MDNs) [15], deep neural networks (DNNs) [16], and recurrent neural networks (RNNs) [17, 18]. It is reported to obtain better accuracy than the DNN-based solution proposed in [16] exploiting an RNN-based AAI approach [19]. This result was mainly due to the better capability at capturing temporal dynamics that the RNN has through its memory elements. Different acoustic representations, such as line spectral frequencies (LSF), Mel-frequency cepstral coefficients (MFCC) and filterbank energies (FBE) have also been employed as input of the AAI system [17, 18]. Linguistic features have also been proven useful when used as stand-alone input features [20], or together with acoustic features [21, 18]. Such linguistic features are for example: phonemic (PHN) and attribute (AF) features [18]. Those features can be estimated by using a phone recognizer [22] or, a forced phone aligner [18] whenever we have access to the transcription of the uttered speech, e.g. in language learning or speech synthesis applications.

Although LSTM-based RNNs are promising for tackling the AAI task, the AAI accuracy could be further improved by exploiting ad-hoc connectionist components that can help remove redundant information in the speech signal. In fact, there exist many sources of information in the speech acoustic signal, which are not all relevant for the target task. Deep learning methods can reduce the effects of that irrelevant information leveraging upon large amounts of training material and parameters; however, lack of ad-hoc corpora providing an appropriate amount of data is a peculiar curse of the AAI problem. Therefore, the use of connectionist blocks that can better exploit the intrinsic characteristic of the speech signal could be beneficial to improve AAI results. We know that the vocal tract movements encode the linguistic message, and the speech signal reflects these movements. Non-linguistic components in the speech signal have a rate of change that lies outside the typical rate of the change of the vocal tract. 1-D convolutional connectionist components can intrinsically be more robust to the speech variability by suppressing spectral components that change more slowly or quickly than the typical range of change of the speech signal. Furthermore, convolutional components offer the advantage to reduce the amount of connectionist parameters with respect to fully connected components, which implies that a smaller amount of data can be sufficient to learn the 1-D convolutional filters. Bi-directional recurrent components with LSTM gates can instead be used to capture temporal relationships and better estimate the articulatory parameters. In this work, we thus propose 1-D convolutional layers prior to the BLSTM-based recurrent blocks to project FBE features to a new space to deal with lack of data and temporal variability. Moreover, the scarcity of relevant speaker specific data makes build-

ing speaker dependent (SD) systems challenging, and the performance typically drops significantly when moving to speaker independent (SI) systems, where data from the test speaker is not used in the training stage of the neural architecture. To overcome the drop in performance caused by data scarcity in the SI configuration, we proposed to combine the feature maps from 1-D convolutions and phonetic features.

The rest of paper is organized as follows. We describe the proposed AAI approach in Section 2. The experimental setup is given in Section 3, where the "Haskins Production Rate Comparison database "(HPRC) [23], input features and output parameters, and network parameters are presented. The experimental results are discussed in Section 3. Section 5 concludes our work.

2. Proposed method

In this work, we propose a new AAI approach, where spectral sub-bands are independently processed in time by 1-D convolutional filters of different sizes. The learned features maps are then combined and processed by an RNN with BLSTM gates for preserving the smoothly varying nature of the articulatory trajectories. We use mel filterbank energies as features in the present work to have a higher resolution for low frequency bands.

1-D convolutional layers are mostly known as the feature extraction layers from sequences and widely used in many speech applications, e.g. ASR [24, 25], speech synthesis [26], and machine translation [27]. This is the first time, to the best of the authors' knowledge, that 1-D convolutional layers on the features are employed in the AAI task. Here we employ convolutional layers along the time axis: we consider the output of the filterbank in each of the frequency bands as a one dimensional data stream and apply the filters on it. These filters' outputs are then linearly combined and represent new feature maps. The computations are formulated as:

$$\boldsymbol{y}^{\text{cnn}}_{i,j} = b_j + \sum_{k=1}^{L_{i-1}} \mathbf{F}_i * \boldsymbol{y}^{\text{cnn}}_{i-1,k}, \tag{1}$$

where, $*$ shows the convolution operation of weights $\mathbf{F}_i$ in convolutional layer i with the feature maps $\boldsymbol{y}^{\text{cnn}}_{i-1,k}$ from the previous layer $i-1$. A bias b_j is added to the result of the convolution, to calculate the new feature map $\boldsymbol{y}^{\text{cnn}}_{i,j}$ for the j^{th} channel feature map. Zero padding is used to guarantee that the input sequence (acoustic space) and output sequence (articulatory space) have the same length. The 1D-CNN layers are used and concatenated along the channel axis as depicted in Fig. 1. The filter length is different in each of the CNN layers which provides more information about adjacent frames with different resolutions along the time axis. The first convolutional layer plays an important role by high-passing or low-passing different frequency bands. In our architecture, this layer has the goal of sensing significant energy changes in the speech spectrum, which may indicate a phone transition. It is built of first order FIR filters in the form $b_0 + m_0 z^{-1}$, where b_0 is a bias and m_0 a multiplicative factor. These can be either low-pass filters when b_0 and m_0 have the same sign, or high-pass, otherwise. The next convolutional layers tries to capture more temporal information and filter out undesired temporal variabilities. After those convolutional layers, two BLSTM layers are employed to capture dynamical information and estimate smoothly varying articulator trajectories. Further analysis with regards to the extracted feature maps and their representation is presented in Section 4.

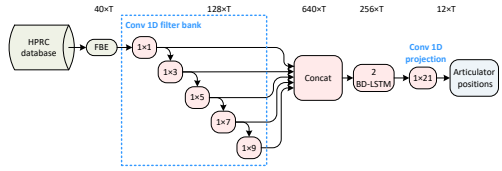


Figure 1: *Architecture of our proposed AAI method.*

3. Experimental Setup

3.1. Database

The EMA method is one of the most used techniques for recording of articulatory data which also allows for simultaneous recording of the speech. One of the available databases is the "Haskins Production Rate Comparison"(HPRC) [23], which covers material from eight native American English speakers, namely four female (F1-F4), and four male (M1-M4) speakers. There are 720 sentences available in this database with the normal and fast Speaking Rate (SR). For some of the normal speaking utterances, there are repetitions available.

Speech waveforms are sampled at rate of 44.1 KHz, and the synchronously recorded EMA data are sampled at 100 Hz. EMA data are measured from eight sensors capturing information about the tongue rear or dorsum (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The articulatory movements are measured in the midsagittal plane in X, Y and Z direction, which denote movements of articulators from posterior to anterior, right to left and inferior to superior, respectively. In this work, we used the X and Z directions of TR, TB, TT, UL, LL and JAW for the speaker dependent AAI. In case of SI modeling, we employed nine tract variables (TV) [28] which are obtained by geometric transformations on EMA measurements. Those TV are Lip Aperture (LA), Lip Protrusion (LP), Jaw Angle (JA), Tongue Rear Constriction Degree (TRCD), Tongue Rear Constriction Location (TRCL). In a similar way for TB and TT we have TBCD, TBCL, TTCD and TTCL, respectively.

3.2. Input representation

In our experiments, acoustic features are extracted from a downsampled waveform at 16 KHz using an analysis window of length $25ms$ with frame shift of $10ms$, yielding a frame rate that matches the EMA recordings. Acoustic features are calculated from 40 filters, which are linearly spaced on the Mel-scale frequency axis. Energies in the overlapping frequency bands are called filterbank energy (FBE) features. Phonetic (PHN) features are extracted by the Penn phonetics lab forced aligner [29]. Each PHN feature is represented as one-hot 39 dimensional vector [18], and the attribute features (AF) are directly mapped from PHN features as in [18].

3.3. Neural parameters & settings

We compare three different neural architectures. In the first and most simple configuration, referred to as BLSTM, the unprocessed filterbank energies are directly fed at the input of the neural architecture, which is BLSTM-based RNN. Two fully connected layers are introduced between the FBEs and the BLSTM-based RNN in the second configuration, referred to as FC-BLSTM. The third configuration, 1D-CNN-BLSTM, is our proposal, and 1-D convolutional filters are employed between

the FBEs and the BLSTM-based RNN. In all cases 2 BLSTM layers with 128 cells for each of the forward and backward layers are used. Sigmoid and tanh activation functions are used for the recurrent layers[18]. The output layer has 12 nodes, corresponding to the EMA dimension with linear activation function. In FC-BLSTM the first two layers are fully connected with 512 nodes with ReLU activation functions. In the 1D-CNN-BLSTM, 5 convolutional layers are used for feature extraction with the filter size of [1, 3, 5, 7, 9], respectively for each layer with ReLU non-linearity. The channel number for each of the convolutional layers are kept the same as $L_i = 128$. A batch size of 5 is used.

The experimental material is chosen from the subsets "N1" and "N2", which have the normal speaking rate. The training data consist of 576 sentences, validation and test data each contains 72 sentences. The data splitting for the HPRC database is as in [30]. Experiments were performed in an utterance by utterance fashion, which requires that all of the utterances are zero padded to 4 sec in the feature domain for ease of training implementation. The same strategy was applied to mean normalized EMA utterances in order to obtain 4 sec duration. The Adam optimizer [31] is chosen for training the network. Keras [32] with TensorFlow backend [33] were used to train all of the neural networks. An early stopping patience of 10 iterations has been employed by checking the validation loss function to prevent over-fitting to the training data.

3.4. Performance measures

To measure the accuracy of the AAI approach, root mean squared error (RMSE) and Pearson's correlation coefficient (PCC) are chosen. The first criterion reports the mean deviation between estimated and the ground-truth trajectories, and the latter measures the similarity of the two trajectories. The measures are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}\big(y(i) - \hat{y}(i)\big)^2}, \tag{2}$$

$$\text{PCC} = \frac{\sum_{i=1}^{N}(y(i) - \bar{y})(\hat{y}(i) - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}\big(y(i) - \bar{y}\big)^2 \sum_{i=1}^{N}\big(\hat{y}(i) - \bar{\hat{y}}\big)^2}}, \tag{3}$$

where $y(i)$ and $\hat{y}(i)$ are the ground-truth and estimated EMA values of the i^{th} frame, respectively; $\bar{y}$ and $\bar{\hat{y}}$ are mean values of $y(i)$ and $\hat{y}(i)$.

4. Experimental Results

In this section, we compare and contrast the three architectures described in Section 3.3. We also present additional analysis to gain a better understanding of the proposed approach.

4.1. Performance evaluation for acoustic features

For each of the three methods, we conduct 20 simulations in order to eliminate the effect of random initialization of the network parameters. Table 1 shows RMSE values, and we can observe that the proposed method outperforms both baseline approaches by almost 0.1 and 0.2 mm RMSE. A t-test shows that the reduction in RMSE with respect to both baselines is significant with p-values less than 0.05. The proposed method outperforms FC-BLSTM with lower number of parameters, as it can be observed by comparing the number of parameters reported in the table. Finally, PCC scores, related to the similarity between trajectories, are given in Table 2 and show a similar trend to that observed for RMSE.

Table 1: *RMSE for various baselines and proposed method for different speakers for AAI system.*

Speaker	**Neural Architecture & No.Parameters**		
	BLSTM	FC-BLSTM	1D-CNN-BLSTM
	571657	*1748233*	*1585033*
F1	1.363	1.226	**1.090**
F2	1.588	1.546	**1.380**
F3	1.296	1.231	**1.160**
F4	1.355	1.309	**1.200**
M1	1.211	1.133	**1.053**
M2	1.645	1.550	**1.435**
M3	1.523	1.479	**1.368**
M4	1.228	1.154	**1.048**
Avg.	1.401	1.328	**1.216**

Table 2: *PCC for various baselines and proposed method for different speakers for AAI system.*

Speaker	**Neural Architecture & No.Parameters**		
	BLSTM	FC-BLSTM	1D-CNN-BLSTM
	571657	*1748233*	*1585033*
F1	0.917	0.932	**0.945**
F2	0.852	0.858	**0.887**
F3	0.827	0.841	**0.861**
F4	0.916	0.921	**0.933**
M1	0.865	0.887	**0.902**
M2	0.861	0.880	**0.893**
M3	0.816	0.841	**0.860**
M4	0.825	0.856	**0.875**
Avg.	0.859	0.877	**0.895**

4.2. Feature extraction layers analysis

As we discussed in Section 2, 1D-CNN extract new features from FBEs. These feature maps are weighted sums of sub-band signals which have been processed by filters with different frequency responses. Fig. 2 shows an example of FBEs, and network activations through the 1D-CNN model. We can see some channel activations match phonemic segments in the first layer. Going to the next layers, the filter outputs become sparser and activations become more intense within the phoneme boundaries. For justifying our claim about channel output activations during the phonemic segments, we picked some channels output from the first layer by using correlation analysis with PHN and AF features as the reference patterns. This analysis provided a better insight for choosing the corresponding filter outputs with regards to PHN and AF features with higher correlation. As an example, we have chosen attribute fricative and phoneme /ɜ/ which are depicted in Fig. 3. The corresponding filters' output which are chosen after doing correlation analysis are depicted in Fig. 3. We can see that these filters outputs have high energies when the chosen attribute and phoneme are active. Therefore, we can say these 1D-CNN layers are extracting the linguistic information from FBEs. This is inline with our expectation of sensing the significant energy changes at the phone transition. Furthermore, we can see for the second CNN layer compared to the first CNN layer, we have less activation outside the ground truth activation times of the chosen attribute and phoneme. By comparing the results for the SD experiment using i) the proposed architecture, and (ii) the BLSTM model that uses PHN features along with FBEs, from Fig. 4, it can be observed, the proposed architecture's better performance could be explained by its inherent capability of 1D-CNN layers at ex-

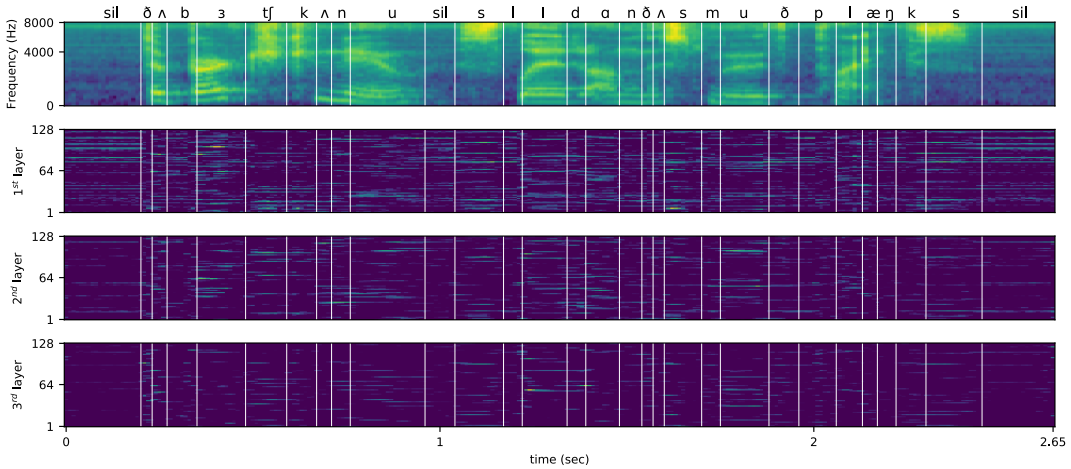


Figure 2: *FBE features for utterance "The birch canoe slid on the smooth planks." and the resulted convolutional feature maps for the* 1^{st}, 2^{nd} *and* 3^{rd} *layers.*

tracting speaker dependent information not available in one-hot encoded PHN features.

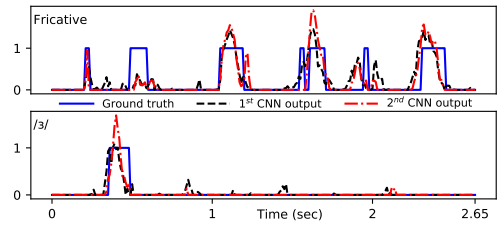


Figure 3: *AF feature for fricative and PHN features for phoneme* /ɜ/ (——) *and channel output from the* 1^{st} *1D-CNN layer (* - - - - *) and* 2^{nd} *1D-CNN layer (* -·-·- *).*

4.3. Speaker independent analysis

For evaluating the proposed method for SI training, we adopt a leave-one-out strategy, where each speaker is in turn considered as the testing speaker, and the rest of speakers are used in training. For articulatory data, we use TV trajectories as targets, and FBE and PHN features are fed at the input of the neural architectures. We used PCC as the performance measure, because of its intrinsic normalized nature that makes it less dependent on the differences between speakers' anatomy, and range of movements. We can observe from Fig. 4 that 1D-CNN improves the performance of both SD and SI configurations. Moreover, by comparing the performances of 1D-CNN with FBE, PHN and their combination, we can observe 1D-CNN has extracted more speaker dependent information while it is less speaker independent compared to FBEs. Using processed FBE features with 1D-CNN filters together with PHN features enhances the system performance in both SD and SI.

5. Conclusion

In this paper, we address the problem of articulatory inversion by employing 1D-CNNs as preprocessing layers to BLSTM layers. We show that this architecture improves the performance for the SD AAI task compared both to a BLSTM network alone, but also to BLSTM whose input is obtained with fully connected layers with a larger number of parameters. We also show that the representations obtained by the 1D-CNNs can be combined with phonetic features to improve performance both for SD and SI systems. The best result from only acoustic features for SD AAI of TV trajectories is PCC=0.895 and by considering phonetic features is PCC=0.901. As a comparison the SD results obtained by [30] on the same data set but with another architecture, is PCC=0.826. Our best results from only acoustic features for SI AAI of TV trajectories is PCC=0.755, by considering phonetic features with the proposed architecture, we reached averaged PCC equals to 0.810 for SI system. For the future works, we will focus on language learning and miss pronunciation detection by employing AAI systems while we have the transcription in this application.

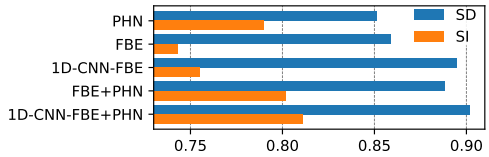


Figure 4: *PCC for the three different neural architecture tested in the present work in both speaker dependent, SD, (blue bar) and speaker independent, SI, (Orange bar) conditions.*

6. Acknowledgements

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt. The second author is supported by the PRIN 2007 project nr. JNKCYZ_002.

7. References

[1] S. Narayanan, K. N. S. Lee, A. Sethy, and D. Byrd, "An approach to real-time magnetic resonance imaging for speech production," *The Journal of the Acoustical Society of America*, vol. 115, no. 4, pp. 1771–1776, 2004.

[2] J. R. Westbury, G. Turner, and J. Dembowski, "X-ray microbeam speech production database user's handbook," *University of Wisconsin*, 1994.

[3] P. W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad, "Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract," *Brain and Language*, vol. 31, no. 1, pp. 26–35, 1987.

[4] K. Richmond and S. King, "Smooth talking: Articulatory join costs for unit selection," in *ICASSP*, 2016, pp. 5150–5154.

[5] R. Li and J. Yu, "An audio-visual 3d virtual articulation system for visual speech synthesis," in *2017 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, Oct 2017, pp. 1–6.

[6] A. B. Youssef, T. Hueber, P. Badin, and G. Bailly, "Toward a multi-speaker visual articulatory feedback system," in *Interspeech*, 2011, pp. 589–592.

[7] T. Hueber, A. Ben Youssef, G. Bailly, P. Badin, and F. Elisei, "Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training," in *Interspeech*, 2012, pp. 783–786.

[8] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1913–1924, Sep. 2011.

[9] A. S. Shahrebabaki, N. Olfati, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of subband frequency signals," in *submitted to Interspeech*, 2020.

[10] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. dissertation, University of Bielefeld, 1999.

[11] V. Mitra, "Articulatory information for robust speech recognition," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2010.

[12] S. Deena, S. Hou, and A. Galata, "Visual speech synthesis using a variable-order switching shared gaussian process dynamical model," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1755–1768, Dec 2013.

[13] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1555, 1978.

[14] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[15] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[16] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[17] A. Illa and P. K. Ghosh, "Representation learning using convolution neural network for acoustic-to-articulatory inversion," in *ICASSP*, 2019, pp. 5931–5935.

[18] A. S. Shahrebabaki, N. Olfati, A. S. Imran, S. M. Siniscalchi, and T. Svendsen, "A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion," in *Interspeech*, 2019, pp. 3775–3779.

[19] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *ICASSP*, 2015, pp. 4450–4454.

[20] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-articulatory mapping using bidirectional gated RNN," in *Interspeech*, 2018, pp. 3112–3116.

[21] P. Zhu, X. Lei, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Interspeech*, 2015, pp. 2192–2196.

[22] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501.

[23] M. Tiede, C. Y. Espy-Wilson, D. G. V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[24] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[25] P. von Platen, C. Zhang, and P. Woodland, "Multi-Span Acoustic Modelling Using Raw Waveform Signals," in *Proc. Interspeech 2019*, 2019, pp. 1393–1397. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2454

[26] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech 2017*, 2017, pp. 4006–4010. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1452

[27] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 365–378, 2017.

[28] A. Ji, "Speaker independent acoustic-to-articulatory inversion," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2014.

[29] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3878, 2008.

[30] N. Seneviratne, G. Sivaraman, and C. Espy-Wilson, "Multi-Corpus Acoustic-to-Articulatory Speech Inversion," in *Interspeech 2019*, 2019, pp. 859–863.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[32] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[33] M. Abadi and et al., "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

Paper C

Transfer learning of articulatory information through phone information

Abdolreza Sabzi Shahrebabaki and Negar Olfati and Sabato Marco Siniscalchi and Giampiero Salvi and Torbjørn Svendsen

INTERSPEECH 2020

Transfer learning of articulatory information through phone information

Abdolreza Sabzi Shahrebabaki[1], *Negar Olfati*[1], *Sabato Marco Siniscalchi*[2], *Giampiero Salvi*[1,3], *Torbjørn Svendsen*[1]

[1] Department of Electronic Systems, NTNU
[2] Department of Computer Engineering, Kore University of Enna
[3] KTH Royal Institute of Technology, Dept. of Electrical Engineering and Computer Science

{abdolreza.sabzi, olfati, giampiero.salvi, torbjorn.svendsen}@ntnu.no,
marco.siniscalchi@unikore.it

Abstract

Articulatory information has been argued to be useful for several speech tasks. However, in most practical scenarios this information is not readily available. We propose a novel transfer learning framework to obtain reliable articulatory information in such cases. We demonstrate its reliability both in terms of estimating parameters of speech production and its ability to enhance the accuracy of an end-to-end phone recognizer. Articulatory information is estimated from speaker independent phonemic features, using a small speech corpus, with electromagnetic articulography (EMA) measurements. Next, we employ a teacher-student model to learn estimation of articulatory features from acoustic features for the targeted phone recognition task. Phone recognition experiments demonstrate that the proposed transfer learning approach outperforms the baseline transfer learning system acquired directly from an acoustic-to-articulatory (AAI) model. The articulatory features estimated by the proposed method, in conjunction with acoustic features, improved the phone error rate (PER) by 6.7% and 6% on the TIMIT core test and development sets, respectively, compared to standalone static acoustic features. Interestingly, this improvement is slightly higher than what is obtained by static+dynamic acoustic features, but with a significantly less. Adding articulatory features on top of static+dynamic acoustic features yields a small but positive PER improvement.

Index Terms: Articulatory inversion, transfer learning, speech recognition, deep learning

1. Introduction

Parameters related to the position and movement of the articulators involved in speech production can be of use in numerous applications. Examples include automatic speech recognition (ASR) [1, 2], speech synthesis [3, 4], pronunciation training [5] and description of the speech production mechanism. The articulatory parameters can be derived by measuring the articulators' kinematics through different methods, such as magnetic resonance imaging (MRI) [6], X-ray microbeam [7], ultrasound [8] and electromagnetic articulography (EMA) [9, 10, 11]. Among these methods EMA is most frequently adopted as it allows using higher sampling rates and simple pre-processing is sufficient to extract the articulatory features from the measurements.

However, measuring the articulatory trajectories directly is not applicable in most real world applications since it requires instrumentation not available outside laboratories, and imposes heavy burdens on the subjects. Thus, in order to utilize articulatory parameters in speech processing applications, we need to estimate them from more accessible information. The most obvious information source is the speech acoustic waveform, and the task to be accomplished is acoustic-to-articulatory inversion (AAI). AAI is challenging from several aspects. The first problem is the one-to-many mapping problem because several articulator gestures may produce the same acoustic speech signal. A common approach to address this problem is to employ trajectory based deep neural networks [12, 13, 14, 15]. The next problem is insufficient amounts of data for adequate modeling of the acoustic space, leading to inferior performance for speaker independent (SI) scenarios compared to the speaker dependent (SD) scenarios, or matched speakers compared to mismatched speakers in SI scenarios. For the articulatory space, lack of data is also important, but the articulatory domain exhibits in general less variation compared to the acoustic space, which makes it less speaker dependent.

In scenarios where the textual content of the spoken utterance is known, linguistic information, e.g. the predicted phone sequence for that utterance, can be used. Indeed, to cope with scarcity of input data for modeling the acoustic space in the AAI task, augmenting the acoustic features with linguistic information has been shown to improve the performance for SD scenarios [16, 13, 15]. Systems utilizing the linguistic information alone have also been reported to work quite well [17, 15] even when using binary features, e.g. one-hot encoded phonemic features (PHN, phone identity) or binary articulatory feature vectors, where multiple features can be active simultaneously [15]. The performance of linguistic information based articulatory inversion (AI) is in line with the reported results in [18], which confirms that front articulators in the vocal tract are related to the linguistic content and the back cavity articulators are more speaker specific. We report in [19] that utilizing linguistic features improves both SD and SI cases significantly. That performance boost is due to less variation between speakers in the linguistic space that is built from a limited set of discrete binary value vectors, in contrast with the acoustic space that is a continuous valued space. In fact, the speaker variability in the linguistic space is limited to the phone duration in the uttered speech sequence.

The advancement in deep neural networks for the task of AI and the positive effect of exploiting PHN features in this task motivate us to propose a new transfer learning approach for AI. We extract articulatory knowledge from a speech corpus providing articulatory measurements, e.g., the "Haskins production rate comparison" (HPRC), and use transfer learning to convey the knowledge to a scenario where articulatory measurements are not available, e.g., the TIMIT [20] phone recognition task. To this end, a teacher model is trained to perform phone-to-articulatory inversion (PAI) on HPRC. The trained teacher provides articulatory targets needed to build a student model that performs acoustic-to-articulatory inversion

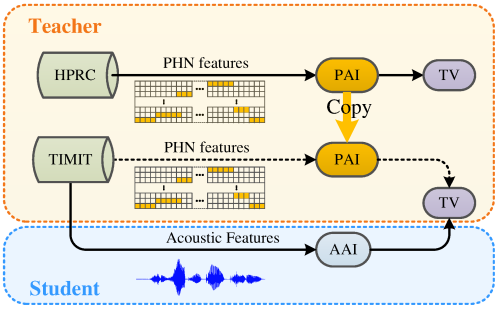


Figure 1: *Block diagram of the proposed transfer learning method from the HPRC to the TIMIT database, and knowledge distillations from phonemic features to acoustic features through articulatory space. Dashed arrows correspond to no training.*

(AAI) on TIMIT. Finally, we use the articulatory information that we estimate on TIMIT through AAI, as input features to perform phone recognition, demonstrating that articulatory features boost phone recognition accuracy.

The rest of paper is organized as follows. The proposed transfer learning method is described in Section 2. Corpora and evaluation methods are in Sections 3 and 5, respectively. Experiments and results are described in Section 5 followed by Section 6 to conclude our work.

2. Teacher-student approach to articulatory information transfer

The proposed approach is motivated by the following observation: Articulatory information can be useful for various speech processing tasks, such as ASR. However, such information is not usually available in corpora for speech recognition. Moreover, it may not be possible to estimate articulatory parameters from the speech signal (AAI) with a satisfactory level of accuracy, and speaker adaptive AAI suitable for typical ASR scenarios is a challenging task. To overcome this, we propose to use phonemic to articulatory inversion (PAI), which is speaker independent by design, as a bridge between scenarios where AAI can be estimated, and speech technology applications where this is usually not the case.

To put forth our solution, we define the following feature sets, and models. The acoustic features, $\mathbf{x} \in \mathbb{R}^n$, the articulatory features, $\mathbf{y} \in \mathbb{R}^m$, and the phone features, $\mathbf{p} \in \mathbb{B}^l$, where $\mathbb{R}$ is the field of real numbers, and $\mathbb{B}$ is the Boolean field. A teacher neural architecture is built on HPRC data to perform the mapping $f_{\text{PAI}} : \mathbb{B}^l \rightarrow \mathbb{R}^m$, from phonemic to articulatory features. This mapping is shown in the upper part in Figure 1. The teacher model not only performs PAI for the HPRC task, but it also provides the articulatory targets for performing PAI with TIMIT data. This process is shown in the middle part in Figure 1, where the PAI architecture is copied to be used with TIMIT phone features at its input and generates articulatory feature estimates at its output. Finally, a student neural architecture is built to perform the mapping $f_{\text{AAI}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ on the TIMIT task. The inputs are acoustic features extracted from the TIMIT waveforms; the outputs are articulatory targets, provided by the teacher neural networks. This step is shown in the bottom part in Figure 1.

With the above feature sets and models, we are ready to use f_{AAI} in order to recover the articulatory features directly from the speech signal without using any annotations. Those articulatory features can be used e.g. as supplemental information in an ASR task, with the goal of improving the overall system performance. In sum, we have built a framework to transfer the knowledge embedded into the articulatory parameters available in the HPRC task to the TIMIT task by using f_{PAI} and f_{AAI} systems, avoiding to address the mismatch between different recording settings and speaker characteristics through a adaptation stage, which is the conventional solution.

The two neural architectures used for articulatory estimation and shown in Figure 1 were trained by minimizing the mean square error (MSE) between estimated values and the ground truth. Those two neural architectures accomplish the following tasks:

Phone-to-articulatory inversion - PAI: This model is trained to estimate the output articulatory features, $\mathbf{y}$, from the input PHN features, $\mathbf{p}$. The PAI neural architecture consists of two bi-directional long short-term memory (BLSTM) layers having 128 cells for each forward and backward directions.

Acoustic-to-articulatory inversion - AAI: The AAI neural structure is a combination of five stacked 1-D convolutional layers of kernel size [1,3,5,7,9], followed by two BLSTM layers with 128 cells in each direction. The convolutional layers extract features from the input acoustic features, $\mathbf{x}$, and the BLSTM layers model temporal dynamics in the system and estimate the articulatory features, $\mathbf{y}$.

3. Corpora

3.1. HPRC

The "Haskins Production Rate Comparison"(HPRC) [11], is a multi-speaker EMA corpus with data from four female and four male native American English speakers. Sampling rates for the speech signal and the EMA recordings are 44.1kHz and 100Hz, respectively. Eight sensors were used to measure the articulators' trajectories. Those eight sensors are placed at the tongue rear (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The sensors movements are measured in the midsagittal plane in X, Y and Z direction, which denote movements of articulators from posterior to anterior, right to left and inferior to superior, respectively. In the HPRC corpus, sensors do not record significant movements in Y direction; we therefore generate information related to the articulatory movements by employing the geometrical transformations defined in [21] on the X and Z directions. Nine tract variables (TVs) are obtained, namely: Lip Aperture (LA), Lip Protrusion (LP), Jaw Angle (JA), in addition to Constriction Degree and Location for Tongue Rear (TRCD, TRCL), Tongue Blade (TBCD, TBCL) and Tongue Tip (TTCD, TTCL). The sampling rate of the articulatory features was maintained. The HPRC speech signals were resampled to 16kHz to match the TIMIT sampling rate.

3.2. TIMIT

The TIMIT database [22] consists of 6300 sentences spoken by 630 speakers from 8 major dialect regions of the United States. There is a predefined portion for training consisting of all the SX and SI sentences from 462 speakers with a total of 3696 sentences. The sentences from the remaining 168 speakers are meant for development and testing purposes. We will follow

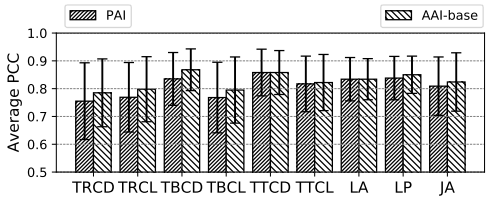


Figure 2: *Averaged PCC and standard deviation for different tract variables of the HPRC test set.*

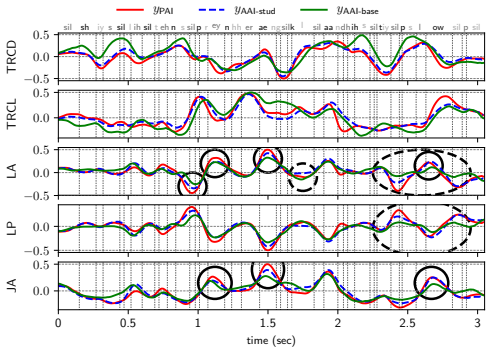


Figure 3: *TV trajectories from f_{PAI}, $f_{AAI\text{-}base}$, and $f_{AAI\text{-}stud}$ for utterance "She slipped and sprained her ankle on the steep slope."*

[23] and use the core test set spoken by 24 speakers for testing and the development set spoken by 50 speakers for validation. The core test set consists of 192 utterances and the development set consists of 400 utterances.

4. Evaluation methods

We used two evaluation methods to assess the proposed technique. The first method computes the Pearson's correlation coefficient explicitly on the target articulatory parameters. The second method is implicit and aims at demonstrating the effectiveness of our approach by inspecting the effects of using estimated articulatory features on the TIMIT phone recognition task.

4.1. Pearson's correlation coefficient

To measure the performance of the articulatory inversion methods, the Pearson's correlation coefficient (PCC) [24] is adopted. The PCC measures the similarity between estimated and ground truth trajectories and is defined as:

$$\text{PCC} = \frac{\sum_i (y(i) - \bar{y})(\hat{y}(i) - \bar{\hat{y}})}{\sqrt{\sum_i \left(y(i) - \bar{y}\right)^2 \sum_i \left(\hat{y}(i) - \bar{\hat{y}}\right)^2}}, \qquad (1)$$

where $y(i)$ and $\hat{y}(i)$ are the ground truth and estimated parameters value of the i^{th} frame respectively and $\bar{y}$ and $\bar{\hat{y}}$ are mean values of $y(i)$ and $\hat{y}(i)$.

4.2. End-to-end phone recognizer

There is no actual ground truth articulatory measurements for TIMIT; therefore, we verify the performance of the proposed approach through the phone error rate (PER) of a phone recognizer built on TIMIT data. In particular, the ESPnet recognizer [25] is used in this work. This phone recognizer is based on (i) an end-to-end encoder-decoder with hybrid connectionist temporal classification (CTC), and (ii) an attention mechanism [26]. The encoder part contains four layers of BLSTM with 320 cells, one layer of LSTM for the decoder with 300 cells, location-aware attention mechanism with 10 convolution filters of length 100, and the same weight, 0.5 for the CTC and attention losses. The interested reader is referred to [26] for more details.

5. Experiments & Results

We evaluate two different types of AI systems, namely PAI- and AAI-based systems. The PAI and AAI systems trained on HPRC material are referred to as f_{PAI} and $f_{\text{AAI-base}}$, respectively, and validated using the PCC measure. In order to assess the f_{PAI} accuracy for TIMIT data, the estimated TVs are visualized and discussed with regards to the speech production mechanism. The student model, which is referred to as $f_{\text{AAI-stud}}$, trained on the TIMIT acoustic data, is assessed from the inversion performance point of view, with the average PCC measure computed using the f_{PAI} as ground truth. An example of estimated TVs for $f_{\text{AAI-stud}}$ and $f_{\text{AAI-base}}$ are visualized. In addition, a comparative ASR performance test is carried out for the TIMIT corpus in terms of PER, to compare efficiency of the $f_{\text{AAI-base}}$ and $f_{\text{AAI-stud}}$ systems and their complementary information for ASR task. Implementations of AI systems are performed using Keras [27] with TensorFlow backend [28].

5.1. Articulatory, phonemic, & acoustic representations

The TVs are calculated for the HPRC data at a rate of 100Hz. In order to have the same 100Hz rate for the acoustic and phonemic feature, a 25ms sliding analysis window and 10ms frame shift are used for acoustic feature extraction. The spoken utterances in HPRC corpus were labeled with the Penn phonetics lab forced aligner [29]. There are 61 phone categories which are folded onto 39 categories [30] to match the conventional 39 phones used in TIMIT [20]. Each phone is represented as a one-hot 39-dimensional vector (PHN) [17]. For TIMIT, we use PHN features for estimating the TVs with the teacher network. For AAI accomplished through the student network, we use the feature vectors consisting of 13 Mel frequency cepstral coefficients (MFCCs). Finally, 23-dimensional Mel filter bank log energies (FBE) are employed along with 3 estimated pitch and voicing features as 26-dimensional static acoustic features in the ESPnet phone recognizer. We also consider first and second derivatives of the FBEs in the phone recognition task.

5.2. Phone-to-articulatory inversion on HPRC

The f_{PAI} input is a 39-dimensional phonemic feature vector, including silence. It should be noted that starting and ending silences have been removed with an energy based threshold speech activity detection (SAD) procedure. Moreover, the 9-dimensional TV features are utterance-based z-score normalized and scaled to be in range $(-0.5, +0.5)$. Training data from the all eight speakers is used to build the f_{PAI} system; whereas validation data is employed with the goal of preventing overfitting. In Fig. 2, we observe that the f_{PAI} is able to predict the articulators in the front vocal cavity akin to the $f_{\text{AAI-base}}$ system. This is inline with what reported in [18, 31], namely that the front articulators capture the linguistic content. The back cav-

Table 1: *PER for acoustic features and their combinations with the estimated TVs from $f_{AAI\text{-}stud}$ and f_{PAI}. D denotes feature dimensionality.*

feature type	D	Dev PER	Test PER
x	26	25.6%	27.9%
x, $y_{\text{AAI-base}}$	35	20.9%	23.3%
x, $y_{\text{AAI-stud}}$	35	**19.6%**	**21.2%**
x, Δx, $\Delta^2 x$	78	19.8%	21.4%
x, Δx, $\Delta^2 x$, $y_{\text{AAI-base}}$	87	19.8%	22.8%
x, Δx, $\Delta^2 x$, $y_{\text{AAI-stud}}$	87	**19.1%**	**20.8%**

Table 2: *Lower bound of PER for the estimated TVs from f_{PAI} combined with the FBEs.*

feature type	D	Dev PER	Test PER
y_{PAI}	9	12.3%	13.3%
x, y_{PAI}	35	8.8%	9.5%
$x, \Delta x, \Delta^2 x, y_{\text{PAI}}$	87	8.2%	9.1%

ity articulators relate closely to speaker specific properties as it is mentioned in [31], and this is reflected by the less precise prediction capability of the PAI system than the AAI system.

5.3. Acoustic-to-articulatory inversion on HPRC

The performance of $f_{\text{AAI-base}}$ system in terms of PCC is shown in Fig. 2. As discussed before, PCC values are comparable for f_{PAI} and $f_{\text{AAI-base}}$ systems for front vocal cavity. For the back cavity, the $f_{\text{AAI-base}}$ system performs better. We can attribute the better performance of the AAI in comparison with the PAI, to the matched speaker independent training style.

5.4. Teacher-student approach to AAI on TIMIT

In the proposed teacher-student approach to perform transfer learning and extract articulatory estimates from acoustic information, we use the f_{PAI} system previously trained on HPRC as the teacher. Articulatory parameters are estimated in terms of TV for TIMIT by feeding TIMIT phonemic transcriptions into the f_{PAI} system. In Fig. 3, we can observe (inside the solid ellipses) that for production of the stop sound /p/, the LA is decreasing and LP is increasing, vowel /æ/ has wider LA or JA than vowels /eɪ/ or /oʊ/, which is inline with dropping of the jaw in production of vowel /æ/ while the jaw is slightly open in /eɪ/ or closed in /oʊ/. Evaluation of the student model ($f_{\text{AAI-stud}}$) is carried out by the average PCC measure, which is 0.929 for the core test set of TIMIT. The PCC distribution is shown in Fig. 4 for each TVs. Estimations from $f_{\text{AAI-stud}}$ and $f_{\text{AAI-base}}$ are visualized in Fig. 3. We can observe that at the end of the utterance (inside the dashed ellipses), the values of the $f_{\text{AAI-base}}$ estimation do not decrease or increase for lip separation or protrusion, respectively, when the stop sound /p/ is present and it is expected to have lowest values for the LA compared to the other phones in this sequence of phones. We can see the $f_{\text{AAI-base}}$ estimation of the LA for /l/ is less than the estimated value for /p/ which is wrong because for production of /p/ lips are closed and for production of /l/ lips are separated. That implies the $f_{\text{AAI-base}}$ model does not provide correct information with respect to speech production constraints.

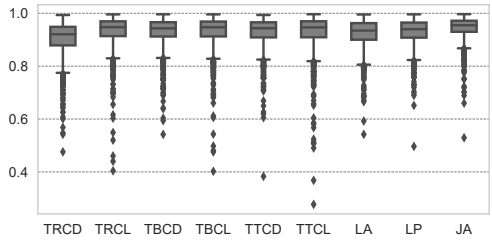


Figure 4: *Distribution of PCC between estimated TV trajectories from $f_{AAI\text{-}stud}$ and f_{PAI}.*

5.5. Exploiting TV estimates in phone recognition

We now explore the role of articulatory information in the task of phone recognition. The ESPnet recognizer in Section 4.2 is employed to build all of our phone recognizers. Several experiments are conducted in order to gain insights on the role of the TV estimates in speech recognition. In the initial experiment, we train the phone recognizer on static acoustic features, (x), only. In the second experiment, we include dynamic features to x and denote it as ($x, \Delta x, \Delta^2 x$). The phone recognizers based on acoustic features only serve as baseline systems. The PER for different input features is reported in Table 1. $y_{\text{AAI-stud}}$ combined with x, significantly improves the recognition accuracy, and reduce the PER by 6.7% on the test set. Interestingly, a slightly better PER, +0.2%, is obtained by replacing the 52-dimensional dynamic acoustic features ($\Delta x, \Delta^2 x$) with the 9-dimensional $y_{\text{AAI-stud}}$. Moreover, we can observe that employing the $y_{\text{AAI-stud}}$ obtains better performance than the $y_{\text{AAI-base}}$. The combination of $y_{\text{AAI-stud}}$ with $x, \Delta x, \Delta^2 x$ reduces the PER by 0.6%.

Finally, we used the TV features y_{PAI} (obtained from the phonemic transcriptions) alone and combined with $x, \Delta x, \Delta^2 x$ to calculate the lower bound of PER in this problem. The results are shown in table. 2.

6. Conclusions

This work proposes a new teacher-student method to transfer articulatory knowledge from the HPRC corpus through phonemic features onto the TIMIT corpus, which is purely acoustic. We exploit the transferred knowledge to build an acoustic to articulatory inversion (AAI) system for TIMIT with the goal of improving ASR performance. In this way, we obtained 0.6% improvements compared to the baseline system for PER when the mixed acoustic and estimated articulatory representations are used. Similarly we obtain better PER combining static acoustic and articulatory features (35 dim.) compared to dynamic acoustic features (78 dim.) proving that articulatory features are a more efficient representation of the dynamics of speech production. We also show that our method performs better than transferring AAI models trained on the HPRC corpus with acoustic adaptation. In the future, we will work on transfer learning of both acoustic and phonetic features to improve the performance of our AI system and getting closer to the PER lower bound.

7. Acknowledgements

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt. The third author is supported by the PRIN 2007 project nr. JNKCYZ_002.

8. References

[1] J. Frankel and S. King, "ASR-articulatory speech recognition," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[2] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1913–1924, Sep. 2011.

[3] K. Richmond and S. King, "Smooth talking: Articulatory join costs for unit selection," in *ICASSP*, 2016, pp. 5150–5154.

[4] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 207–219, 2013.

[5] T. Hueber, A. Ben Youssef, G. Bailly, P. Badin, and F. Elisei, "Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training," in *Interspeech*, 2012, pp. 783–786.

[6] S. Narayanan, K. N. S. Lee, A. Sethy, and D. Byrd, "An approach to real-time magnetic resonance imaging for speech production," *The Journal of the Acoustical Society of America*, vol. 115, no. 4, pp. 1771–1776, 2004.

[7] J. R. Westbury, G. Turner, and J. Dembowski, "X-ray microbeam speech production database user's handbook," *University of Wisconsin*, 1994.

[8] D. Porras, A. Sepúlveda-Sepúlveda, and T. G. Csapó, "Dnn-based acoustic-to-articulatory inversion using ultrasound tongue imaging," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[9] P. W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad, "Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract," *Brain and Language*, vol. 31, no. 1, pp. 26–35, 1987.

[10] R. Korin, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Interspeech*, Florence, Italy, August 2011, pp. 1505–1508.

[11] M. Tiede, C. Y. Espy-Wilson, D. G. V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[12] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[13] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501.

[14] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *ICASSP*, 2015, pp. 4450–4454.

[15] A. S. Shahrebabaki, N. Olfati, A. S. Imran, S. M. Siniscalchi, and T. Svendsen, "A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion," in *Interspeech*, 2019, pp. 3775–3779.

[16] P. Zhu, X. Lei, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Interspeech*, 2015, pp. 2192–2196.

[17] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-articulatory mapping using bidirectional gated RNN," in *Interspeech*, 2018, pp. 3112–3116.

[18] D. J. Broad and H. Hermansky, "The front-cavity/f2′ hypothesis tested by data on tongue movements," *The Journal of the Acoustical Society of America*, vol. 86, no. S1, pp. S113–S114, 1989. [Online]. Available: https://doi.org/10.1121/1.2027307

[19] A. S. Shahrebabaki, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals," in *press, Interspeech*, 2020.

[20] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[21] A. Ji, "Speaker independent acoustic-to-articulatory inversion," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2014.

[22] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recpgnition Workshop*, L. S. Baumann, Ed., Feb 1986, pp. 100–109.

[23] A. K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *EUROSPEECH*, 1997, pp. 401–404.

[24] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[25] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[26] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, Dec 2017.

[27] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[28] M. Abadi and et al., "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[29] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3878, 2008.

[30] K. . Lee and H. . Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, Nov 1989.

[31] A. Illa and P. K. Ghosh, "An Investigation on Speaker Specific Articulatory Synthesis with Speaker Independent Articulatory Inversion," in *Proc. Interspeech 2019*, 2019, pp. 121–125. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2664

Paper D

A DNN Based Speech Enhancement Approach to Noise Robust Acoustic-to-Articulatory Inversion

Abdolreza Sabzi Shahrebabaki and Sabato Marco Siniscalchi and Giampiero Salvi and Torbjørn Svendsen

ISCAS 2021).

Paper E

On Robust Deep Learning Approaches for Joint Optimization of Speech Enhancement and Articulatory Inversion

Abdolreza Sabzi Shahrebabaki and Sabato Marco Siniscalchi and Giampiero Salvi and Torbjørn Svendsen

IEEE/ACM TASLP 2022

Acoustic-to-Articulatory Mapping with Joint Optimization of Deep Speech Enhancement and Articulatory Inversion Models

Abdolreza Sabzi Shahrebabaki, Giampiero Salvi, Torbjørn Svendsen, *Senior Member, IEEE*, and Sabato Marco Siniscalchi, *Senior Member, IEEE*

***Abstract*—We investigate the problem of speaker independent acoustic-to-articulatory inversion (AAI) in noisy conditions within the deep neural network (DNN) framework. In contrast with recent results in the literature, we argue that a DNN vector-to-vector regression front-end for speech enhancement (DNN-SE) can play a key role in AAI when used to enhance spectral features prior to AAI back-end processing. We experimented with single- and multi-task training strategies for the DNN-SE block finding the latter to be beneficial to AAI. Furthermore, we show that coupling DNN-SE producing enhanced speech features with an AAI trained on clean speech outperforms a multi-condition AAI (AAI-MC) when tested on noisy speech. We observe a 15% relative improvement in the Pearson's correlation coefficient (PCC) between our system and AAI-MC at 0dB signal-to-noise ratio on the Haskins corpus. Our approach also compares favourably against using a conventional DSP approach to speech enhancement (MMSE with IMCRA) in the front-end. Finally, we demonstrate the utility of articulatory inversion in a downstream speech application. We report significant WER improvements on an automatic speech recognition task in mismatched conditions based on the Wall Street Journal corpus (WSJ) when leveraging articulatory information estimated by AAI-MC system over spectral-alone speech features.**

***Index Terms*—Deep neural network, Acoustic-to-articulatory inversion, Speech enhancement, Multi-task training, Speaker independent models.**

I. Introduction

THE human speech production system contains several organs, namely, lungs; trachea; larynx; throat; oral and nasal cavities. The oral cavity comprises several anatomical elements, such as velum, tongue, teeth, jaw and lips. Those elements are considered as the articulators. Articulator movements result in the production of various speech sounds. The problem of estimating the articulators' movements from the acoustic speech signal is referred to as acoustic-to-articulatory inversion (AAI). In recent years, AAI has attracted increasing attention because of its potential applications in speech processing. Examples include low bit rate coding [1], automatic speech recognition (ASR) [2]–[4], speech synthesis [5], [6], computer aided pronunciation training (CAPT) [7], [8], depression detection from speech [9], [10], and speech therapy [11], [12]. The articulators' movements can be measured and parameterized through various techniques, for instance real-time magnetic resonance imaging (rt-MRI) [13], X-ray microbeam [14], electromagnetic articulography (EMA) [15], and ultrasound [16]. Nevertheless, obtaining articulatory measurements is not practical in real world applications since it requires instrumentation not available outside laboratories, and imposes heavy burdens on the subjects. As a consequence, estimation of these parameters from the available source of information, which is the speech signal, must be achieved through an AAI system. Unfortunately, this inversion problem is highly non-linear and non-unique [3], [17], which means that different articulator configurations can produce the same sound. Moreover, coarticulation [18], i.e., the impact of adjacent phonemes on the articulators' movement, makes the AAI problem even harder. In addition, articulatory measurements are only available for a limited number of speakers. This limitation introduces an additional complexity to the AAI problem and urges building up speaker independent AAI systems (SI-AAI) that can be utilized for speech databases with no articulatory recordings.

The majority of available AAI works mainly focused on two different aspects: (i) the acoustic feature representation, and (ii) the solution to the AAI regression problem with different techniques. Different acoustic representations, such as Line Spectral Frequencies (LSFs) [19], Perceptual Linear Predictive coding (PLP) [20] and Mel-Frequency Cepstral Coefficients (MFCCs) [21] have been widely used for the AAI task. Filter-Bank Energies (FBEs) from STRAIGHT spectra [22] have also been employed as the input of the AAI system [23]. Among these features, MFCCs are reported to perform better compared to other features for SI-AAI [24], [25].

In the literature, various techniques are applied to the AAI problem, e.g. search-based algorithms in the joint codebook of the acoustic-articulatory space [26], [27], non-parametric and parametric statistical methods, such as support vector regression (SVR) [28], local regression approach based on K-nearest neighbour [29], joint acoustic-articulatory distribution by utilizing Gaussian mixture models (GMMs) [30], hidden Markov models (HMMs) [7], mixture density networks (MDNs) [31], deep neural networks (DNNs) [4], [32], and recurrent neural networks (RNNs) [23], [33]–[39]. Among those methods, the neural network based models outperform the rest by having the ability of dealing well with large context size and better modelling of acoustic and articulatory spaces.

It is also important to remark that most of the available AAI research is accomplished using clean data, with the goal of improving the AAI accuracy either for speaker dependent,

or speaker independent cases. Real world speech applications, however, suffer from the presence of environmental noise in the recordings, which in turn leads to a performance degradation of the AAI system. There are few works available for AAI in noisy conditions. Most of these works are in the field of robust ASR [4], [40], and use synthetically generated speech data obtained with an articulatory synthesizer and the Task Dynamics and Applications (TADA) system [41]. To the best of the authors' knowledge, there is only one work dealing with real articulatory measurements in noisy conditions [42], where the authors compared the accuracy of two AAI systems. One system was trained on clean data (AAI-C); the other system was built using multi-condition speech data (AAI-MC), including clean data. For the AAI-C system the noisy test data were optionally enhanced by minimum mean square error (MMSE) based speech enhancement (SE) [43]. The outcome of the study was twofold. First, AAI-MC seemed to be the best solution for dealing with noisy data. Second, MMSE-based SE on the noisy data led to a drop in the AAI-C performance on noisy data compared to both AAI-MC and to AAI-C with unprocessed speech. Such an outcome contrasts with the naïve expectation that enhanced speech should yield an improved performance. A possible explanation of the unexpected outcome could be that distortions and artifacts introduced by the MMSE-based method may have reduced the quality of the enhanced speech with respect to the AAI task.

Although SE based on the MMSE approach did not seem useful in AAI applications in noisy conditions, we observe that deep neural network (DNN) based approaches to SE have recently been shown to better overcome musical noise issues and introduce less distortion than traditional digital signal processing (DSP) methods [44]–[46]. Therefore, we argue that DNN-SE can play a key role in AAI too if used at a pre-processing stage before the downstream speech applications, as demonstrated for other tasks in [47]–[49], for instance. Our goal is therefore to clean up the input signal before sending it to the off-the-shelf AAI-C, avoiding the need to build an AAI-MC system leveraging multi-condition data. In addition, for speech recognition in noisy conditions which is more applicable in the daily usage, it would be helpful to apply enhancement as a pre-processor, and estimate the articulatory trajectories and subsequently utilize them in the recognition task.

We design our SE system using deep neural networks vector-to-vector regression with the goal to enhance the speech features. The deep model used for the AAI task is stacked on top of the SE model, allowing for joint optimization of the full model for further improved performance. In this way the overall neural model learns to enhance the noisy speech in a helpful way for the AAI goal. To better appreciate our experimental evidence, we compare and contrast our proposed approach with the MMSE-based speech enhancement with an improved noise estimation method, namely minima controlled recursive averaging (IMCRA) [50]. The IMCRA algorithm produces less distortion in the enhanced speech compared to the original MMSE based approaches, e.g. [43]. We will refer to IMCRA based system as DSP-SE. Moreover, we assessed the role that articulatory information, extracted with the proposed solution, could play in a downstream speech application using an ASR task in noisy condition, namely a hand-crafted noisy version of the Wall Street Journal (WSJ) task [51]. Experimental evidence clearly demonstrates the beneficial effect of combining articulatory information with standard spectral-based speech features when decoding noisy speech data using a character-based encoder-decoder end-to-end ASR system leveraging both a hybrid connectionist temporal classification (CTC) loss function, and the attention mechanism.

The rest of the paper is organized as follows. In section II different neural architectures are described. Section III introduce the corpora which are utilized in this research work, and in Section IV, different experiments are conducted and the results are discussed. Section V concludes our work and suggests future work.

II. AAI Systems

In this section, three different systems are described. The first system performs acoustic-to-articulatory inversion (AAI) directly on (noisy) speech using a deep model; the second systems consists of a feed-forward deep neural network based speech enhancement module (DNN-SE) and an AAI module based on a deep architecture; finally, the third system combines the DNN-SE and DNN-AAI module into a single deep architecture and joint training is used to fine-tune the overall AAI system. In the following, those three systems are discussed in detail.

A. DNN for Acoustic-to-articulatory inversion (DNN-AAI)

A speaker-independent (SI) design is used to deploy the DNN-AAI system, so that test speakers are removed from the developing material during the training phase. The input features are the standard MFCCs. These features have been shown to attain better performance than other speech features for the SI task [24] when higher order cepstral coefficients are removed. The smooth nature of articulatory trajectories and co-articulation effect suggest that the input temporal context should be long enough to capture the needed information with respect to to the output trajectories [3]. We select every other frame in a $2 \times M_{\text{aai}}$ window preceding and succeeding the current frame to construct the following extended input vector:

$$X_{\text{aai}}[n] = \Big[X[n-2\times M_{\text{aai}}]^T, \ldots, X[n-2]^T, X[n]^T, X[n+2]^T, \ldots, X[n+2\times M_{\text{aai}}]^T\Big]^T, \quad (1)$$

where X_{aai} is the contextualized MFCC vector for the AAI system and $[.]^T$ indicates the transpose operator. Employing every other frame gives us the benefit of longer temporal context with less parameters in the AAI model with no performance degradation.[1] Fig. 1 shows the structure of input data for a DNN-AAI system, where the output features are tract variables (TVs), which are described later in Section

[1]Experiments with different decimation factors, D, showed no PCC degradation for $D = 2$ and a moderate degradation for $D = 3$ and 4.

IV. The input features and output targets of the DNN-AAI system are mean and variance normalized at an utterance level. DNN-AAI systems trained on clean and multi-condition noisy speech data will in the following be denoted AAI-C and AAI-MC, respectively.

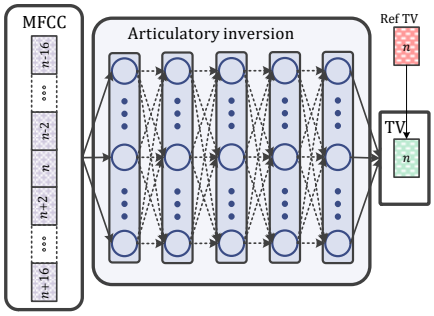


Fig. 1. DNN based AAI system with 340 ms input context of MFCCs, and tract variables (TV) as the output.

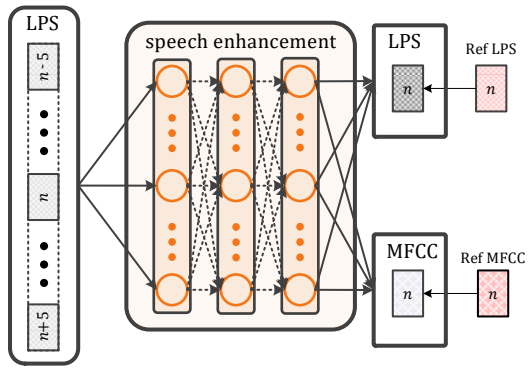


Fig. 2. DNN based SE system with 120 ms context of noisy LPSs, and clean LPSs and MFCCs as the output.

B. DNN-SE for AAI

In this solution, a DNN is first built to map noisy speech features into estimated clean features using a regression framework [46]. The AAI-C system is then used to estimate the articulatory trajectories. The DNN-SE system is based on a feed-forward layered structure of non-linear hidden layers and a linear output layer. The non-linear blocks allow the network to better handle the complex interactions between degraded noisy signal and its clean counterpart, as argued in [46]. The input features for the DNN-SE are globally mean and variance normalized Log Power Spectra (LPSs). LPSs have been obtained by taking the log of the squared magnitude of the signal's short-time Fourier transform (STFT). The DNN-SE enhances only the magnitude spectrum; therefore, the noisy phase is used in the reconstruction step (synthesis). In this work, we synthesise the enhanced speech waveform from enhanced magnitude and noisy phase spectrum using the the overlap-add method [52], which was also used in [46], to be able to assess the quality of the enhanced speech. To take into account context information, M_{se} previous and future frames around the current frame are used at the DNN-SE input:

$$S_{\text{se}}[n] = \left[S[n - M_{\text{se}}]^T, \ldots, S[n]^T, \ldots, S[n + M_{\text{se}}]^T\right], \quad (2)$$

where the S_{se} is the contextualized LPS of the noisy signal as the input vector.

It should be noted that M_{se} is shorter than M_{aai}, that is, less context is taken into account in the SE step. That is coherent with the non-stationary property of noises, which enables the network to have a better estimation of short-time noise spectrum to be suppressed. At a target level, there are several possible choices, namely, only clean LPS can be used in a single-task learning procedure, or both MFCC and LPS can be employed in a multi-task scenario. In the multi-task case, the back propagated loss from the MFCC output layer acts as a regularizer and would prevent the model to over-fit to the training data. Moreover, the MFCC-related output layer can be directly used as an input of the AAI-C system. Fig. 2 shows a sketch of DNN-SE system with multiple output tasks.

Although MFCCs can be derived from LPS through a transformation, joint estimation of enhanced LPS and MFCCs may impose additional constraints unavailable in the direct prediction of clean LPS. As discussed in [53], Mel-filtering is applied to make the acoustic features consistent with human auditory perception. However there is so far no prior auditory knowledge adopted in the LPS domain except for the log-compression, and clean LPS features could therefore be better predicted with a MFCC constraint imposed at the output layer. Furthermore, the correlation information among different channels can be incorporated in each MFCC coefficient due to the discrete cosine transformation (DCT) [54] operation. Therefore, we expect that correlated and consistent distortion across different frequency bins can be learned when predicting the clean LPS. Differently from [53] the DCT block in our pipeline also performs dimensionality reduction, since we use MFCCs for the AAI block.

C. Joint DNN-SE and DNN-AAI

In Sections II-A and II-B, we described the two independent DNN-based systems for AAI and SE task respectively, where the DNN-SE module could be employed in a pre-processing step before the target AAI task to be accomplished with the DNN-AAI system. Since the two independent systems are built within the connectionist framework, we can stack them back-to-front and obtain a single overall AAI system. The overall system can be further fine-tuned using the same loss employed to build the DNN-AAI system. However, the fusion of those two systems into a single one is challenging, because of the different temporal contexts used to build the two systems independently. As mentioned in Section II-B, the DNN-SE input context size (M_{se}) is smaller than AAI-C one (M_{aai}). The required frames for building the AAI input need to be provided at the input layer of the DNN-SE module. To this end, a

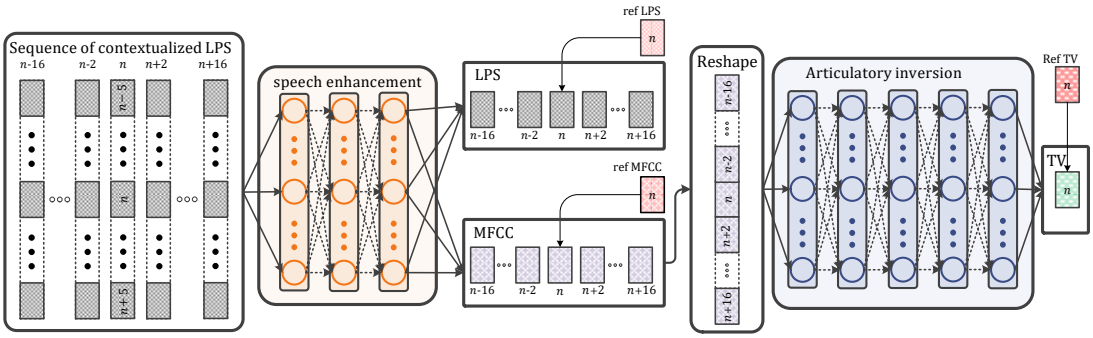


Fig. 3. Network structure of joint training of SE and AAI systems ($M_{se} = 5, M_{aai} = 8$)

sequence S_{joint} of contextualized speech vectors is presented at the input of the joint system, where the sequence is built as follows:

$$S_{\text{joint}}[n] = \Big[S_{\text{se}}[n - 2 \times M_{\text{aai}}]^T, \ldots, S_{\text{se}}[n-2]^T, S_{\text{se}}[n]^T, S_{\text{se}}[n+2]^T, \ldots, S_{\text{se}}[n + 2 \times M_{\text{aai}}]^T\Big], \quad (3)$$

The DNN-SE module thus generates all needed input frames for the AAI module, $X_{\text{aai}}[n]$. In the training stage, back propagated error for the enhancement part is limited to that referring to the middle contextualized vector in the input sequence. The proposed architecture is illustrated in Fig. 3 where the LPS and MFCC tasks are considered for the current time n. In this way the network parameters are trained on the current time n, while being able to deal with the different time-varying nature of the events to be handled in the two modules. For the AAI module, the output concerned with the MFCC task for each input sequence of contextualized LPSs is reshaped to build the AAI input vector. The overall system loss function based on mean squared error (MSE) is formulated as follows:

$$L_{\text{joint}} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{y}_i^{\text{LPS}} - \hat{\boldsymbol{y}}_i^{\text{LPS}}||^2 + ||\boldsymbol{y}_i^{\text{MFCC}} - \hat{\boldsymbol{y}}_i^{\text{MFCC}}||^2 + ||\boldsymbol{y}_i^{\text{TV}} - \hat{\boldsymbol{y}}_i^{\text{TV}}||^2, \quad (4)$$

where, $\boldsymbol{y_i}^{(\cdots)}$s are the reference output vectors, $\hat{\boldsymbol{y}_i}^{(\cdots)}$s are the estimated vectors for each output and N is the number of training samples.

III. Corpora and Data Representation

There are three tasks in this work, the main one is the AAI, the second one is speech enhancement, and the third task is automatic speech recognition. For the first task, two corpora are employed, the "Haskins Production Rate Comparison"database (HPRC) [55], which contains both acoustic and articulatory measurements, and the AURORA2 database [56] which contains eight noise types. For the second task, we additionally employ two datasets: TIMIT [57] with spoken American English; and Nonspeech [58] which contains 100 various noise types. For the third task, we use the WSJ dataset. In the following, the mentioned corpora are described in details. Furthermore, the representation of the acoustic and articulatory data is described.

A. Corpora

1) HPRC: The Haskins Production Rate Comparison (HPRC) database is selected as the main database for the AAI experiments. It contains recordings of eight native American English speakers, four female (F01-F04) and four male (M01-M04) speakers. There are 720 spoken utterances available in the dataset with both normal and fast speaking rate. For some of the normal speaking rate utterances, there are a few repetitions available. Speech waveforms are sampled at the rate of 44.1 kHz, and synchronous EMA recordings are available at a sampling rate of 100 Hz. EMA recordings are obtained from eight sensors, which record tongue rear or dorsum (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The articulatory measurements are aligned to the occlusal plane in X, Y and Z directions, corresponding to movements from posterior to anterior, right to left and inferior to superior, respectively. The movements along the Y axis carry limited information. In this work, we employed only the X and Z directions of TR, TB, TT, UL, LL and JAW. Furthermore, we used 80% of data for training, 10% for validation, and the remaining 10% for test.

2) TIMIT: TIMIT [57], [59] is a speech corpus consisting of 6300 sentences spoken by 630 speakers, covering 8 major dialect regions of the United States. The dataset includes two dialect sentences (SA), 1890 phonetically diverse sentences (SI), and 450 phonetically compact sentences (SX). The training set is predefined and consists of all the SX and SI sentences from 462 speakers with a total of 3696 sentences. The sentences from the remaining 168 speakers constitute the full test set. We use the core test set [59], covering speech material from 24 speakers, for testing purposes. A validation set spoken by 50 speakers is used to prevent over-fitting and performance tuning with respect to the validation data. The core test set consists of 192 utterances, and the development set consists of 400 utterances.

3) Wall Street Journal - WSJ: The WSJ [51] corpus is in two distinct parts: WSJ0 and WSJ1. The SI-84 training material from the WSJ0 covers 7,193 utterances (15 hours). The SI-284 (80 hours) data is formed by combining training data from both the WSJ0 and WSJ1 (26,515 utterances). For development and evaluation, 503 utterances (1.1 hour), and 333 utterances (0.7 hour) are used, respectively. Clean waveforms, sampled at 16kHz, and corresponding transcripts are provided for both WSJ0 and WSJ1. Waveforms were down-sampled to 8kHz to carry out our downstream ASR experiments. Moreover, testing waveforms were corrupted with noise in order to create mismatched conditions between training (clean) and testing (noisy) and better assess the effect of introducing articulatory information into an end-to-end ASR system. More details are given in Section IV-G.

4) AURORA 2: AURORA 2 [60] is a corpus of noisy speech created by adding noise of various types and levels to clean speech recordings. In this work we only employ the noise recordings consisting of eight different noise types that are recorded in different places, namely, airport, crowd of people (babble), car, exhibition hall, restaurant, street, subway, and train station. The recordings contain stationary and non-stationary noise segments, and are sampled at a rate of 8 kHz.

5) Nonspeech: The Nonspeech dataset [61], which contains 100 different environmental noises, is recorded with a 20 kHz sampling rate and was downsampled to 8 kHz for our experiments. The noise types available in the dataset are as follows, N1-N17: Crowd noise, N18-N29: Machine noise, N30-N43: Alarm and siren, N44-N46: Traffic and car noise, N47-N55: Animal sound, N56-N69: Water sound, N70-N78: Wind, N79-N82: Bell, N83-N85: Cough, N86: Clap, N87: Snore, N88: Click, N88-N90: Laugh, N91-N92: Yawn, N93: Cry, N94: Shower, N95: Tooth brushing, N96-N97: Footsteps, N98: Door moving, N99-N100: Phone dialing.

6) Simulated multi-condition dataset: Multi-condition waveforms are synthetically generated by randomly adding noise from AURORA2 and Nonspeech to the HPRC and TIMIT speech samples at different signal-to-noise ratios (SNR). The multi-condition data set also includes clean data. To match the 8 kHz sampling rate of the AURORA2 database the audio material from the other datasets is downsampled to 8 kHz. Another constraint is imposed by the 100 Hz sampling rate of the articulatory measurements, which leads to a frame shift of 10ms to match the 100 Hz sampling rate.

B. Articulatory data representation

As reported in [62], geometrical transformations can be applied to the EMA measurements in order to transform those measurements into tract variables (TVs). TVs have the property of being more speaker independent than the original measurements, because they are relative measures and suffer less from non-uniqueness [63]. We use nine TVs, including Lip Aperture (LA), Lip Protrusion (LP), Jaw Angle (JA), Tongue Rear Constriction Degree (TRCD), Tongue Rear Constriction Location (TRCL). For TB and TT, we also calculate TBCD, TBCL, TTCD and TTCL, as explained below. The aforementioned geometrical transformations are defined as follows:

$$\mathrm{LA}[n] = \sqrt{\Big(\mathrm{LL}_x[n] - \mathrm{UL}_x[n]\Big)^2 + \Big(\mathrm{LL}_z[n] - \mathrm{UL}_z[n]\Big)^2}, \tag{5}$$

$$\mathrm{LP}[n] = \mathrm{LL}_x[n] - \underset{m\in\text{all utterances}}{\text{median}}\ \mathrm{LL}_x[m]. \tag{6}$$

LA represents the Euclidean distance between LL and UL sensors. LP is defined as the movement of LL from its median position in the X direction,

$$\mathrm{JA}[n] = \sqrt{\Big(\mathrm{JAW}_x[n] - \mathrm{UL}_x[n]\Big)^2 + \Big(\mathrm{JAW}_z[n] - \mathrm{UL}_z[n]\Big)^2}, \tag{7}$$

is defined as the Euclidean distance between the JAW and UL sensors.

For each of the tongue sensors TR, TB and TT, two TVs are defined. Those TV features represent constriction locations, which are the deviations from median of the corresponding sensor along the X axis, and the constriction degree, which is the minimum distance between the corresponding tongue sensors position and the palate trace. TRCL and TRCD are defined as follows

$$\mathrm{TRCL}[n] = \underset{m\in\text{all utterances}}{\text{median}}\ \mathrm{TR}_x[m] - \mathrm{TR}_x[n], \tag{8}$$

$$\mathrm{TRCD}[n] = \min\Big\{\sqrt{\Big(\mathrm{TR}_x[n] - x_{pal}\Big)^2 + \Big(\mathrm{TR}_z[n] - z_{pal}\Big)^2}\Big\}, \tag{9}$$

where x_{pal} and z_{pal} are the palate coordinates on the occlusal plane.

The remaining four variables TBCL, TBCD, TTCL and TTCD can be obtained in a similar way:

$$\mathrm{TBCL}[n] = \underset{m\in\text{all utterances}}{\text{median}}\ \mathrm{TB}_x[m] - \mathrm{TB}_x[n], \tag{10}$$

$$\mathrm{TBCD}[n] = \min\Big\{\sqrt{\Big(\mathrm{TB}_x[n] - x_{pal}\Big)^2 + \Big(\mathrm{TB}_z[n] - z_{pal}\Big)^2}\Big\}, \tag{11}$$

$$\mathrm{TTCL}[n] = \underset{m\in allutterances}{\text{median}}\ \mathrm{TT}_x[m] - \mathrm{TT}_x[n], \tag{12}$$

$$\mathrm{TTCD}[n] = \min\Big\{\sqrt{\Big(\mathrm{TT}_x[n] - x_{pal}\Big)^2 + \Big(\mathrm{TT}_z[n] - z_{pal}\Big)^2}\Big\}. \tag{13}$$

C. Acoustic feature representations

As discussed in the previous sections, we study three tasks. The first task is the AAI, which is the main task in the present work; the SE is the second task. Both tasks are addressed under the DNN framework. AAI models are trained over MFCC feature vectors, which are extracted using a 20ms windowed signal with a frame shift of 10ms. 13-dimensional MFCC feature vectors are extracted from 23 Mel-scaled filter banks. For the AAI system we set $M_{\text{aai}} = 8$. This moderately long

temporal span covers 340ms of the input acoustic data. As already mentioned, the temporal context improves the AAI performance due to the smooth varying nature of the articulator trajectories. For the SE system, the log power spectra (LPS) (256 coefficients) are calculated for 20ms windowed signal with 10ms frame shift. The temporal context with $M_{\text{se}} = 5$ spans past and future frames around the target frame at time n, that is equivalent to 120ms of speech.

IV. Experiments and Results

The key experiments reported in this section are concerned with AAI, and the effect of speech enhancement on AAI. Speech enhancement quality is also reported for all of the DSP- and DNN-based systems investigated in this work. Finally, the role of AAI system in a downstream speech application is assessed using an ASR task in noisy condition.

Moreover, several experiments have been carried out to validate the proposed approach and fine tune all models. With respect to the optimization of network parameters, the AAI systems investigated in the present work have been built leveraging clean and multi-condition data, resulting in AAI-C and AAI-MC systems, respectively. The AAI-MC system is considered in the present study, because it was recently reported as the best AAI solution in noisy conditions [42]. For the evaluation of the optimized networks, clean, multi-condition and enhanced multi-condition data are used. Moreover, all of the AAI experiments are carried out in mismatched speaker conditions using the leave-one-speaker-out (LOSO) cross-validation scheme during the training phase. For speech enhancement, we compare and contrast the IMCRA-based DSP approach (DSP-SE), and the feature-based vector-to-vector regression with deep models for speech enhancement approach (DNN-SE) discussed in [46]. The DNN-SE is deployed using a deep feed-forward neural network with three hidden non-linear layers, each having 1024 nodes. ReLU activation functions [64] were employed in both AAI and SE neural modules; whereas, a linear activation function was used at the output layer. The PCC criterion was used to select the best performing network on the validation data. Moreover, early stopping prevents over-fitting to the training data, and training is halted either when the PCC on validation data does not improve for 10 consecutive epochs, or a total number of epochs equal to 100 has been reached. The ADAM optimizer [65] was employed to minimize the MSE between the ground-truth and estimated tract variables. All neural models implemented in our work were built using the Tensorflow library [66] with Keras API [67]. Drop-out [68] was used to contrast over-fitting, and a drop-out rate of 10% was used in each hidden layer. Different DNN-SE systems have been built using a different experimental setups, namely:

1) matched speakers, noise types, and SNRs between training and testing phases;
2) mismatched speakers but matched noise types and SNRs between training and testing phases;
3) mismatched speakers, noise types and SNRs between training and testing phases.

The purpose of latter experimental setup is to verify the applicability of DNN-SE in real-world conditions, where having similar speakers, noise types and SNRs is highly unlikely. In our experiments, we consider SNR levels in the range between -5 dB to 20 dB in incremental steps of 5 dB. In the following, experiments and results are presented and discussed in more detail, yet we first introduce the metrics used in this work to assess all systems.

A. Test data

Because we employ several corpora in this work, the data split needs to be clarified. In all simulations, the test set is from the HPRC database. In the case of multi-condition data, the test set is distorted by additive noises from AURORA2.

B. Performance metrics

The Pearson's correlation coefficient (PCC) was used as a measure of accuracy between the estimated and the reference TVs in the AAI systems. The reason for choosing PCC is that the PCC is a normalized measure and varies between -1 to 1, and it is independent from the difference in articulatory measurement's ranges which is related to speakers' anatomies. A higher value of the PCC shows better performance of inversion system.

Perceptual evaluation of speech quality (PESQ) was used to evaluate the quality of the enhanced speech [69]. For computing the PESQ, enhanced speech waveforms were synthesized from the enhanced LPS and the noisy phase spectra. The PESQ score ranges from -0.5 to 4.5, and the higher the PESQ score, the closer the enhanced speech is to the original clean speech. Indeed, PESQ has been proven to provide a high correlation to the quality scores rated by humans [70].

C. DNN-AAI results

Using LOSO cross-validation during training, each of the eight speakers, in turn, becomes a test speaker while the remaining seven speakers are used in the training phase. Reported results are thus averaged across all test speakers. Several experiments varying the number of hidden layers and nodes in the DNN were carried out. PCC is used to select the best AAI system using the validation data. In particular, the following configurations were investigated: $[100, 300, 500, 1000]$ nodes, and $[2, 3, 4, 5]$ hidden layers. The PCC value is reported in the upper panel in Figure 4 when clean data are used; PCC curves show that the best performing AAI system has 5 hidden layers with 100 nodes per layer. As the amount of available data is limited, it is reasonable that increasing the number of parameters would not lead to a performance improvement. The same set of experiments was executed using multi-condition data, and results are reported in the lower panel of Figure 4. We can see that either 4 or 5 hidden layers with 300 nodes can lead to the best PCC score. For our following experiments we have chosen the configuration with 4 hidden layers to save computational resources.

After tuning the neural parameters, the average PCC on the test set is calculated and reported with respect to two different aspects, namely: 1) SNR level, and 2) noise type. Experimental

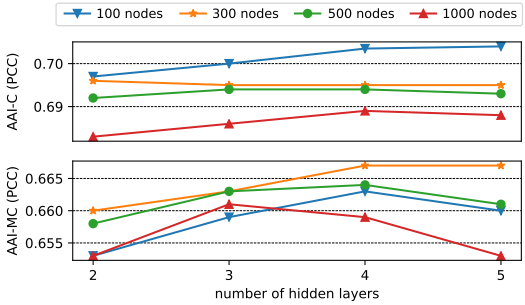


Fig. 4. Average PCC performance vs AAI DNN parameters with matched training and test data: clean data (top panel) and multi-condition data (bottom panel)

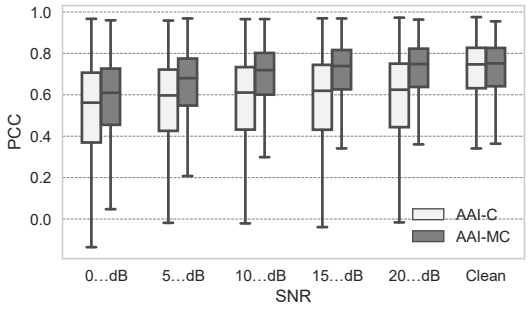


Fig. 5. Average PCC for multi-condition data with respect to different SNR levels. The box plots represent the minimum, first quartile, median, third quartile, and the maximum of average PCC values.

results for different SNRs are shown in Fig. 5 for both AAI-C and AAI-MC systems. It can be observed that AAI-MC attains almost similar PCC on clean data and noisy data at SNRs $\geq$ 15 dB. It can be concluded that the required speech information for the inversion are obtainable at these SNRs. The high standard deviation in the PCC distribution in Fig. 5 is due to several factors, e.g., different test speakers performance, different variation range for each of the TVs and the effect of various noise types.

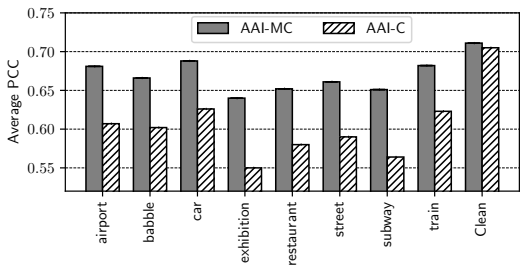


Fig. 6. Average PCC for multi-condition data on AAI-C and AAI-MC models, with respect to different noise types.

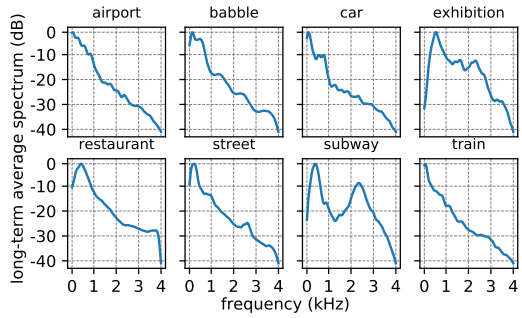


Fig. 7. Long-term average spectrum of different noise types in Aurora 2 database.

The effect of different noise types on the performance of AAI-C and AAI-MC systems is shown in Fig. 6 that shows the average PCC over different speakers and SNRs. It can be observed that 'exhibition' and 'subway' noises have the greatest negative effects on AAI accuracy and cause a significant performance drop; in contrast, 'car' and 'train' noises have a minor negative effects on the final AAI accuracy. Inspecting the long term averaged power spectrum of different noise types in Fig. 7, we can observe that a common feature of the noise types that cause the most severe degradation of the AAI performance is that they have considerable energy in frequency bands between 1 kHz and 3 kHz. For clean data, AAI-MC performs slightly better than the AAI-C, which can be explained thinking of the larger amount of training data used to build the system, i.e., a consequence of the data-augmentation effect, especially data at an SNR equal to 20 dB.

D. DNN-SE results

The DNN-SE system has been trained in three different scenarios. We briefly describe each scenario along with the corresponding training procedure in the following.

DNN-SE1 - Matched speakers, noise types, and SNRs: The HPRC dataset is used for the speech material, and the AURORA 2 noises are added to it in order to synthetically simulate noisy speech. All of the eight possible noises are added to the speech waveforms at different SNR levels. The same speakers, noise types and SNR levels are employed for creating training, validation, and test data. Furthermore, these settings are used in both the single and multi-task approaches (see Section II-B). SNR levels are [0, 5, 10, 15, and 20] dB.

DNN-SE2 - Mismatched speakers, matched noise types and SNRs: The speech material and noises are the same as those employed in the first experimental scenario. Mismatch between training and testing condition was inserted at a speaker level. For each speaker, a stand-alone network is built using the other seven speakers in the training phase, which

TABLE I
PESQ PERFORMANCE COMPARISON OF SINGLE-TASK (ST) AND MULTI-TASK (MT) SPEECH ENHANCEMENT SYSTEMS BASED ON DNN FOR THREE DIFFERENT SCENARIOS.

SNR	Noisy	DSP-SE	DNN-SE1		DNN-SE2		DNN-SE3
			ST	MT	ST	MT	MT
-5 dB	—	—	—	—	—	—	2.359
0 dB	1.51	1.700	2.554	2.653	2.365	2.528	2.580
5 dB	1.75	2.077	2.767	2.873	2.544	2.729	2.770
10 dB	2.06	2.533	2.955	3.069	2.702	2.907	2.937
15 dB	2.47	2.950	3.104	3.224	2.828	3.048	3.074
20 dB	2.97	3.316	3.205	3.333	2.919	3.148	3.180

TABLE II
PERFORMANCE OF SI-AAI SYSTEMS TRAINED ON CLEAN AND MULTI-CONDITION DATA AND TESTED ON CLEAN, MULTI-CONDITION AND ENHANCED DATA.

Test data	Enhancement	AAI-C	AAI-MC
Clean	None	0.705	0.710
Multi-Cond	None	0.595	0.665
Multi-Cond	DSP-SE	0.568	0.620
Multi-Cond	DNN-SE1-MT	0.699	0.711
Multi-Cond	DNN-SE1-ST	0.689	0.702
Multi-Cond	DNN-SE2-MT	0.670	0.701
Multi-Cond	DNN-SE2-ST	0.662	0.693
Multi-Cond	DNN-SE3-MT	0.678	0.697

is applied to speech from the given speaker in the testing phase. In doing so, the deep model is more realistic and better simulates real world applications compared to the previous scenario, which may be useful for a feasibility assessment. SNR levels are again [0, 5, 10, 15, and 20] dB.

DNN-SE3 - Mismatched speakers, noise types, and SNRs: In this third experimental scenario, the 8 kHz version of the TIMIT corpus is used for the speech material, and the challenging Nonspeech database is used for the noises. The validation set also comes from TIMIT and Nonspeech. The test set consists of material taken from the HPRC speakers and degraded by AURORA 2 noises. The SNR levels in training are [0, 5, 10, and 20] dB. Different SNRs, namely [-5, 0, 5, 10, 15, and 20] dB, are used in the test phase. These experimental conditions are closer to what one can expect in real production; moreover, our DNN-SE module is trained on independent data and noises with respect to the testing conditions, so it functions a general purpose SE tool.

Table I shows the average PESQ for models trained and tested as discussed above. A visual inspection of Table I shows that DSP-SE improves the average PESQ by 0.1 for 0 dB, 0.2 for 5 dB, 0.4 for 10 dB, 15 dB, and 20 dB. A main issue with the DSP-SE method is its poor performance at low SNRs, yet its strength is the inherent nature of the DSP solution that does not require training data and makes it a general SE tool for real-world applications. The best results are expected for DNN-SE1, which is trained in matched conditions for speakers, noise types, and SNR levels. Both single-task (DNN-SE1-ST) and multi-task (DNN-SE1-MT) configurations are evaluated. DNN-SE1-MT achieves a better performance than DNN-SE1-ST, as it can be observed comparing columns four and five in Table I. This confirms our intuition about the regularization effect of the multi-task configuration. Indeed, DNN-SE1-MT attains better PESQ compared to DSP-SE and DNN-SE1-ST in all tested SNRs.

Experiments in matched condition demonstrated the feasibility of our idea, and the positive effect of a multi-task configuration for a SE task. DNN-SE2 is built using a different training configuration, which takes into account a mild level of mismatch between training and testing phases. Therefore, a small drop in the SE performance is expected, and results reported in the sixth and seventh column in Table I confirm our expectation. Moreover, DNN-SE2-MT attains a performance comparable to DNN-SE1-ST in spite of the more challenging SE scenario. Given that multi-task is a viable way to boost SE performance in mismatched conditions, only DNN-SE3-MT is built in the third experimental scenario, the most realistic and challenging one. Since DNN-SE3-MT is trained as general purpose SE module, it is not a surprise that it shows PESQ values superior to those attained with DNN-SE2-MT. The key strength of DNN-SE3-MT compared to the DNN-SE2-MT is the larger number of speakers, and thereby speech material, used in the training phase along with the more challenging noises that the model had to deal with. In mismatched SNRs, very promising results are obtained; for example, at an SNR of 15 dB, DNN-SE3-MT slightly outperforms DNN-SE2-MT in terms of PESQ. For -5 dB the PESQ value is 2.359 which it is $\approx$0.2 less than 0 dB, and the PESQ value is 2.580 at 0 dB is $\approx$0.2 less than the PESQ value at 5 dB.

In general DNN-SE models have higher performance than DSP-SE at low SNR levels.

E. AAI on SE data

We investigated the effect of enhancing the speech data prior to AAI. Because it is unlikely that clean data are available in real production, SE modules are employed prior to the AAI-C system, as a pre-processing step. In doing so, we can use an off-the-shelf AAI-C model without exploiting MC training. We compare and contrast the effect of both DSP-SE and DNN-SE on AAI, and the AAI-MC performance is reported to ease the comparison.

First, we notice from Table II that AAI-C tested on data enhanced by DNN-SE performs better than AAI-MC tested on multi-condition data without enhancement. On the one hand, it can be argued that the improvement comes from an increase of the neural parameters obtained by coupling two deep models. On the other hand, it should be noted that the DNN-SE and the AAI-C deep model were independently trained on different data, and our solution allows us to use an off-the-shelf AAI-C system avoiding training a new system from scratch. This aspect should not be underestimated in a production pipeline of a real complex system. It should also be recalled that [42] reported DSP-SE to cause a drop in the AAI performance. We therefore further compare DSP-SE and DNN-SE effects on AAI-C. In Table II, we see that DSP-SE coupled with AAI-C indeed causes 0.11 drop in the PCC compared with our DNN-SE-MT3 coupled with AAI-C. Most importantly, for AAI-C, applying DSP-SE to the noisy data

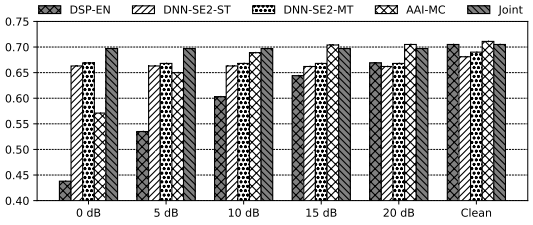


Fig. 8. PCC of the AAI-C on enhanced speech data, AAI-MC and joint systems on multi-condition data, at different SNRs.

reduces the PCC by 4.5% relative compared to using the noisy data directly. That result is in line with [42], and it could be explained by the signal distortions usually introduced by DSP-SE, such as musical noise [71]. In contrast, our DNN-SE method does not cause any drop in the AAI performance, and our findings open up a new path for DNN-based front-end approaches in speech applications. In Table II, we see that the DNN-SE system has a significant improvement over the DSP-SE, an increase of 0.11 in terms of average PCC, and a relative improvement of 19.36% is achieved using DNN-SE3-MT system over DSP-SE. For the sake of completeness, we report experimental results with multi-task (MT) and single-task (ST) training strategies in Table II, both in matched and mismatched speaker scenarios. The multi-task DNN-SE methods outperform single-task counterparts; whereas, a drop in PCC is observed when moving from matched to mismatched speakers. However, speech enhancement is performed to avoid building an AAI-MC system, we provided results using AAI-MC on clean, noisy and enhanced data for completeness. Interestingly, enhancing the noisy speech with the DNN-SE based systems improves the AAI-MC model's performance, in contrast with what is observed for DSP-SE. It should be noted the better performance of separate DNN-SE1-MT model with AAI-MC model in comparison with the joint model performance is due to the matched speaker condition of SE module.

A detailed comparison in terms of SNR values of the AAI-C on DSP-SE and DNN-SE systems, is shown in Fig. 8. Enhancement with DNN-SE2-MT always gives a better PCC in low SNR conditions. Moreover, DNN-SE2 and DSP-SE lead to similar PCC only in very high SNR. At 0 db, from Figures 6 and 8, we see that AAI-MC attains a PCC of 0.579, and AAI-C on DNN-SE enhanced data attains a PCC of 0.67, which accounts for a 15% relative improvement in favor of the proposed DNN-SE based AAI-C approach.

The DNN-SE methods cause degradation for the clean data performance compared to the performance of clean data on the AAI-C. The performance degradation of AAI-C, for enhanced clean data by DNN-SE system, can be explained by over-smoothing of enhanced speech compared to the natural ones or enhanced by DSP-SE method.

TABLE III
JOINT SPEECH ENHANCEMENT AND ARTICULATORY INVERSION PERFORMANCE IN TERMS OF PESQ AND PCC.

	0 dB	5 dB	10 dB	15 dB	20 dB
PESQ	2.655	2.864	3.050	3.197	3.301
PCC	0.697	0.697	0.697	0.697	0.697

F. Joint AAI and SE based on DNN

So far we have investigated the AAI system either using stand alone AAI systems or decoupled SE and AAI system. We now address both the SE and AAI tasks under a unified DNN framework, by coupling the two deep architectures into a single network and leveraging the availability of the MFCC output in the SE module. Then, the overall network can be jointly fine-tuned with the goal of accomplishing AAI. Training the joint model is challenging because back-propagation of different tasks affect each other and make the convergence slower compared to learning different models designed to accomplish different tasks. To improve convergence, there are two alternative procedures available:

1) First, the speech enhancement module is trained while keeping AAI parameters frozen, so that the gradient flows back through the network layers until the enhancement module converges. Next, the speech enhancement module weights are kept frozen, and the AAI parameters are updated till convergence. In this way, the training scheme will be similar to AAI training with enhanced multi-condition data.
2) Initializing each the connectionist parameters with the pre-trained DNN-SE3 and AAI-C weights, and then fine-tune the whole system with the goal of accomplishing AAI. In this way both modules start from a better initialization starting point.

We decided to use the second approach to carry out joint training of the SE and AAI blocks. The LOSO cross-validation approach is utilized for training of the joint model. The multi-condition data is kept the same as in the previous experiments, to have comparable results. Table III reports results with joint training. It is interesting to see that we can improve both SE and AAI tasks in terms of PESQ and PCC, respectively. It should be recalled that the DNN-SE has a primary task which corresponds to enhancing the LPS speech vector. By comparing PESQ values in Tables I and III, we can observe that the SE module in the joint model attains results close to the DNN-SE1-MT model which is the best performing enhancement model presented in this work. The AAI performance for different SNR levels are the same to the third decimal place. The AAI performance of the joint model on multi-condition data is PCC=0.697, and the AAI-C model performance on clean data is PCC=0.705. The joint model performance is closer to the AAI-C system on clean data than the performance of either the AAI-MC system on multi-condition data (PCC=0.665), or the AAI-C system on DNN-SE3-MT data (PCC=0.678). This performance is expected considering that the AAI part is tuned for the enhanced data in the joint training of the enhancement and inversion systems.

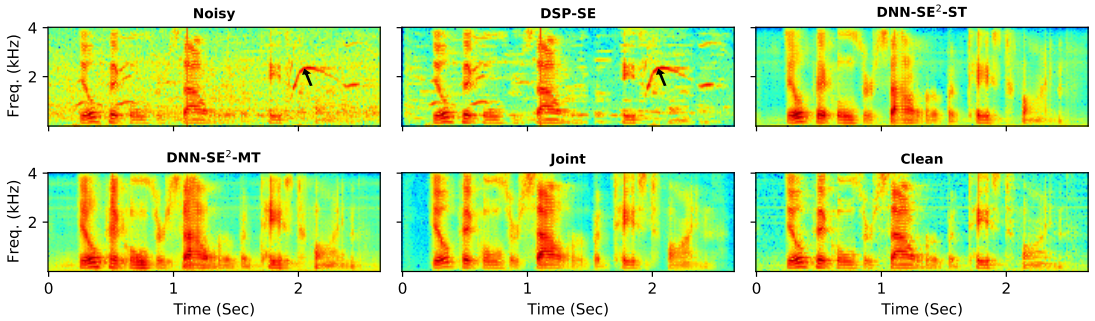


Fig. 9. Spectrogram of the utterance "Dill pickles are sour but taste fine.", corrupted by Exhibition noise at SNR=5 dB. (a) noisy speech with (PESQ=1.768), (b) enhanced by DSP-SE (PESQ=1.815), (c) single-task DNN based model (PESQ=2.204), (d) multi-task DNN based model (PESQ=2.55), multi-task DNN based model jointly with the articulatory inversion (PESQ=2.89), and (f) the clean speech signal. Black arrows indicate the high energy whistle sound.

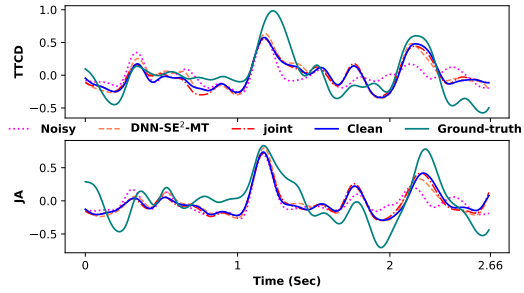


Fig. 10. TTCD and JA trajectories for utterance "Dill pickles are sour but taste fine.", distorted by exhibition noise at SNR=5 dB.

In addition, from Fig. 8, it can be observed AAI-MC system performs better than the AAI-C system with enhanced data by DNN-SE modules at SNR$\geq$10 dB. The joint model decrease this under performing.

Fig. 9 shows spectrograms for a testing utterance corrupted by exhibition noise at an SNR equal to 5 dB, clean, and enhanced with different SE methods. The DSP-SE method clearly introduces some distortions in the form of musical noise. Moreover, it could not remove high energy whistle sound (indicated by black arrows) starting at $\sim$ 1.93s. The DNN based methods are instead able to suppress different noise characteristics in the noisy signal but the over-smoothing affects the higher frequency components. However, the unwanted whistle sound is completely suppressed by all of DNN-SE methods.

TTCD and JA trajectories for the same selected utterance are depicted in Fig. 10. The "Noisy''one is estimated using AAI-MC model, and the other trajectories are enhanced and predicted by AAI-C model. From those trajectories in Fig. 10, we can argue that the enhanced speech by DNN-SE methods allows to obtain AAI accuracy like those obtained on the clean speech signal. The estimated trajectories by the AAI-MC with the noisy data as the input are very different with the estimated trajectories by the AAI-C model, e.g. the estimated JA at $\sim$ 2s which is due to the whistle distortion.

G. AAI for ASR

We now turn our attention on assessing the role of articulatory information on downstream speech tasks. To this end, a continuous word recognition task is considered, namely the WSJ0 [51], and several end-to-end automatic speech recognition (ASR) systems are built and contrasted to demonstrate the effect of TV information on the ASR performance in both clean, and noisy conditions. The word error rate (WER) is selected as the metric to compare the accuracy of all systems deployed in this section.

Clean data is already available with the WSJ0 corpus, and noisy data are synthetically generated by adding two noise types, namely exhibition, and subway. In the previous sections, the most adverse effects on AAI accuracy were caused by these noise types. Two SNR levels are used for training and testing, namely 0dB and 10 dB. WSJ waveforms are downsampled from 16kHz to 8 kHz. 60-dimensional log Mel filter bank energy (FBE) features were extracted using a 512-point short-time Fourier transform to compute the spectra of each overlapping windowed frame. A 32-ms Hamming window and a 16-ms window shift were adopted. The end-to-end ASR systems are all based on the end-to-end ESPnet recognizer [72], which is a character-based encoder-decoder model leveraging both a hybrid connectionist temporal classification (CTC) loss function, and an attention mechanism [73]. The encoder part contains 12 layers of BLSTM with 2048 cells, six layers of LSTM for the decoder with 2048 cells, and a location-aware attention mechanism with 10 convolution filters of length 100. The CTC loss and the attention loss were weighted by 0.2 and 0.8 respectively. Words are obtained from characters using an RNN language model, utilizing one LSTM layer with 1000 cells, which is trained on 65000 words from the WSJ1 corpus. In our experiments, the "dev93" part of WSJ0 corpus is used for parameter tuning. The actual evaluation is carried out on the for the "eval92" part.

We built two ASR systems using different data conditions, namely clean or noisy (0dB and 10dB), and different input

TABLE IV
WER FOR THE "EVAL92" PART OF WSJ DATABASE FOR THE TWO MENTIONED ASR SYSTEMS.

Test Condition	System 1	System 2
Clean FBEs	5.3	—-
Clean FBEs + TV (AAI-MC)	—-	5.5
Clean FBEs + TV (DNN-SE+AAI-C)	—-	5.4
Clean FBEs + TV (Joint)	—-	5.4
Enh Clean FBEs	6.1	—-
10 dB FBEs	49.4	—-
10 dB FBEs + TV (AAI-MC)	—-	22.6
10 dB FBEs + TV (DNN-SE+AAI-C)	—-	19.8
10 dB FBEs + TV (Joint)	—-	19.1
Enh 10 dB FBEs	42.3	—-
0 dB FBEs	78.2	—-
0 dB FBEs + TV (AAI-MC)	—-	57.8
0 dB FBEs + TV (DNN-SE+AAI-C)	—-	51.4
0 dB FBEs + TV (Joint)	—-	49.8
Enh 0 dB FBEs	68.4	—-

speech features, namely FBEs, or FBEs and TVs. The first system is trained on clean data and used FBE features; we refer to this system as **System 1**, and it sets a WER lower-bound when testing on clean data, and an upper-bound in noisy conditions. The second system, **System 2** is trained on clean data and leverages both FBE and TV features. System 2 allows us to assess the effect of articulatory information on the downstream ASR task.

Table IV shows all results gathered in our experiments. System 1 is evaluated on three different conditions, namely clean, noisy, and enhanced FBE features obtained with DNN-SE3-MT. System 2 leverages TV features, which are obtained with the AAI-C model described in Section IV-C in the training phase. In the testing phase, however, TV features are obtained either using the AAI-MC model in Section IV-C, the DNN-SE3-MT+AAI-C model discussed in Section IV-E, or the joint model discussed in SectionIV-F. A visual inspection of Table IV reveals that System 1 attains the best results on clean FBE features with a WER equal to 5.3%, and attains the worst WER (6.13%) when tested in clean condition on enhanced data, as expected. The use of TV features along with clean FBE does not cause a significant increase of the WER. In noisy conditions, namely testing on FBE extracted on waveforms at 10dB and 0dB SNRs, we can see that System 1 attains the worst WERs as expected . Interestingly, the injection of TV features in System 2 boosts the ASR recognition performance significantly. Given that System 2 is also trained on clean FBE features as System 1, the latter results allow us to argue that articulatory information plays a key role in the selected downstream speech tasks. Moreover, the estimated TVs from the joint model have the most effect on the System 2 performance in terms of WER.

V. CONCLUSION

We have investigated into the speaker-independent AAI problem in noisy speech conditions. We have shown that DNN-based speech enhancement for input noisy signals can boost the performance of the AAI-C system trained on clean data. A good improvement was also observed for the AAI-MC system trained on multi-condition data. In the mismatched-speaker scenarios, enhancing multi-condition data with DNN-SE combined with the AAI-C model performed better than the straight AAI-MC system, which clearly demonstrates the effectiveness of the proposed speech enhancement pre-processing with deep models. Although the AAI-C system with speech enhanced by DNN-SE systems performs better than the AAI-MC system for noisy data, the performance at high SNR levels is degraded. To cope with this degradation, a joint model was proposed to perform both speech enhancement and articulatory inversion, which demonstrated its benefit over separate systems for each task. The joint system performance is close to the performance of clean data in AAI-C system. The key strength of applying DNN based enhancement methods prior to the AAI-C model, compared to the AAI-MC method are their better performance at low SNRs which is beneficial for ASR systems in presence of noise. Our experimental results also sheds new light on the AAI problem by contrasting what reported in the recent literature, namely speech enhancement does not bring any improvement when used in a pre-processing prior to AAI with noisy data [42]. Finally, we show that articulatory information can be useful in downstream speech applications, namely end-to-end ASR.

ACKNOWLEDGEMENTS

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt. The last author is supported by PRIN2017 JNKCYZ_002 project.

REFERENCES

[1] J. Schroeter and M. M. Sondhi, "Speech coding based on physiological models of speech production," *Advances in Speech Signal Processing*, pp. 231–267, 1992.
[2] J. Frankel and S. King, "ASR-articulatory speech recognition," in *Seventh European Conference on Speech Communication and Technology*, 2001.
[3] V. Mitra, "Articulatory information for robust speech recognition," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2010.
[4] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1913–1924, Sep. 2011.
[5] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 207–219, 2013.
[6] K. Richmond and S. King, "Smooth talking: Articulatory join costs for unit selection," in *ICASSP*, 2016, pp. 5150–5154.
[7] T. Hueber, A. Ben Youssef, G. Bailly, P. Badin, and F. Elisei, "Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training," in *Interspeech*, 2012, pp. 783–786.
[8] O. Engwall, "Analysis of and feedback on phonetic features in pronunciation training with a virtual teacher," *Computer Assisted Language Learning*, vol. 25, no. 1, pp. 37–64, 2012.
[9] B. S. Helfer, T. F. Quatieri, J. R. Williamson, D. D. Mehta, R. Horwitz, and B. Yu, "Classification of depression state based on articulatory precision." in *INTERSPEECH*, 2013, pp. 2172–2176.
[10] S. Sahu and C. Y. Espy-Wilson, "Speech features for depression detection." in *INTERSPEECH*, 2016, pp. 1928–1932.
[11] D. W. Massaro, S. Bigler, T. Chen, M. Perlman, and S. Ouni, "Pronunciation training: the role of eye and ear," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[12] S. Fagel and K. Madany, "A 3-d virtual head as a tool for speech therapy for children," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[13] S. Narayanan, K. N. S. Lee, A. Sethy, and D. Byrd, "An approach to real-time magnetic resonance imaging for speech production," *The Journal of the Acoustical Society of America*, vol. 115, no. 4, pp. 1771–1776, 2004.

[14] J. R. Westbury, G. Turner, and J. Dembowski, "X-ray microbeam speech production database user's handbook," *University of Wisconsin*, 1994.

[15] P. W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad, "Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract," *Brain and Language*, vol. 31, no. 1, pp. 26–35, 1987.

[16] D. Porras, A. Sepúlveda-Sepúlveda, and T. G. Csapó, "DNN-based acoustic-to-articulatory inversion using ultrasound tongue imaging," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[17] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. dissertation, University of Bielefeld, 1999.

[18] S. Deena, S. Hou, and A. Galata, "Visual speech synthesis using a variable-order switching shared gaussian process dynamical model," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1755–1768, Dec 2013.

[19] R. Korin, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Interspeech*, Florence, Italy, August 2011, pp. 1505–1508.

[20] C. Qin and M. Á. Carreira-Perpiñán, "A comparison of acoustic features for articulatory inversion," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[21] P. K. Ghosh and S. Narayanan, "Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. EL251–EL257, 2011.

[22] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3-4, pp. 187–207.

[23] A. S. Shahrebabaki, N. Olfati, A. S. Imran, S. M. Siniscalchi, and T. Svendsen, "A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion," in *Interspeech*, 2019, pp. 3775–3779.

[24] P. K. Ghosh and S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2162–2172, 2010.

[25] G. Sivaraman, V. Mitra, H. Nam, M. Tiede, and C. Espy-Wilson, "Unsupervised speaker adaptation for speaker independent acoustic to articulatory speech inversion," *The Journal of the Acoustical Society of America*, vol. 146, no. 1, pp. 316–329, 2019. [Online]. Available: https://doi.org/10.1121/1.5116130

[26] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1555, 1978.

[27] S. Ouni and Y. Laprie, "Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 118, no. 1, pp. 444–460, 2005.

[28] A. Toutios and K. Margaritis, "A support vector approach to the acoustic-to-articulatory mapping," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[29] S. A. Moubayed and G. Ananthakrishnan, "Acoustic-to-articulatory inversion based on local regression," in *INTERSPEECH*, 2010.

[30] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[31] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[32] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[33] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *ICASSP*, 2015, pp. 4450–4454.

[34] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501.

[35] A. Illa and P. K. Ghosh, "Representation learning using convolution neural network for acoustic-to-articulatory inversion," in *ICASSP*, 2019, pp. 5931–5935.

[36] A. Illa and P. K. Ghosh, "Low resource acoustic-to-articulatory inversion using bi-directional long short term memory," in *Proc. Interspeech 2018*, 2018, pp. 3122–3126. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1843

[37] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-articulatory mapping using bidirectional gated rnn," in *Proc. Interspeech 2018*, 2018, pp. 3112–3116. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1202

[38] A. S. Shahrebabaki, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals," *Proc. Interspeech 2020*, pp. 2882–2886, 2020.

[39] A. S. Shahrebabaki, N. Olfati, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Transfer learning of articulatory information through phone information," *Proc. Interspeech 2020*, pp. 2877–2881, 2020.

[40] V. Mitra, G. Sivaraman, H. Nam, C. Espy-Wilson, E. Saltzman, and M. Tiede, "Hybrid convolutional neural networks for articulatory and acoustic information based speech recognition," *Speech Communication*, vol. 89, pp. 103–112, 2017.

[41] H. Nam, L. Goldstein, E. Saltzman, and D. Byrd, "TADA: An enhanced, portable task dynamics model in MATLAB," *The Journal of the Acoustical Society of America*, vol. 115, no. 5, pp. 2430–2430, 2004. [Online]. Available: https://doi.org/10.1121/1.4781490

[42] N. Seneviratne, G. Sivaraman, V. Mitra, and C. Espy-Wilson, "Noise robust acoustic to articulatory speech inversion," in *Proc. Interspeech 2018*, 2018, pp. 3137–3141. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1509

[43] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[44] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[45] D. S. Williamson, Y. Wang, and D. Wang, "Reconstruction techniques for improving the perceptual quality of binary masked speech," *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. 892–902, 2014. [Online]. Available: https://doi.org/10.1121/1.4884759

[46] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[47] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. W. Schuller, "Towards Robust Speech Emotion Recognition Using Deep Residual Networks for Speech Enhancement," in *Proc. Interspeech 2019*, 2019, pp. 1691–1695. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1811

[48] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5024–5028.

[49] B. Ghai, B. Ramanan, and K. Müller, "Does speech enhancement of publicly available data help build robust speech recognition systems?" *ArXiv*, vol. abs/1910.13488, 2019.

[50] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, 2003.

[51] Garofolo, John S., et al. CSR-I (WSJ0) Complete LDC93S6A. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[52] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[53] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," 2017.

[54] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Comput.*, vol. 100, pp. 90–93, 1974.

[55] M. Tiede, C. Y. Espy-Wilson, D. G. V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[56] E. Rothauser, "Ieee recommended practice for speech quality measurements," *IEEE Trans. on Audio and Electroacoustics*, vol. 17, pp. 225–246, 1969.

[57] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recognition Workshop*, L. S. Baumann, Ed., Feb 1986, pp. 100–109.
[58] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2067–2079, 2010.
[59] J. S. Garofolo, L. F. Lamel, W. M. Fischer, J. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," Tech. Rep. NISTIR 4930, 1993.
[60] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.
[61] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2067–2079, 2010.
[62] A. Ji, "Speaker independent acoustic-to-articulatory inversion," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2014.
[63] R. S. McGowan, "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests," *Speech Communication*, vol. 14, no. 1, pp. 19–48, 1994.
[64] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.
[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[66] M. Abadi and et al., "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
[67] F. Chollet *et al.*, "Keras," https://keras.io, 2015.
[68] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.
[69] I.-T. Recommendation, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.
[70] S. Fu, C. Liao, and Y. Tsao, "Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality," *IEEE Signal Processing Letters*, vol. 27, pp. 26–30, 2020.
[71] T. D. Tran, Q. C. Nguyen, and D. K. Nguyen, "Speech enhancement using modified IMCRA and OMLSA methods," in *International Conference on Communications and Electronics 2010*, 2010, pp. 195–200.
[72] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456
[73] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

Abdolreza Sabzi Shahrebabaki received the B.Sc. degree in electrical engineering from Khajeh Nasir Toosi university of technology (KNTU), Tehran, Iran, in 2009, and the M.Sc. degree in electrical engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2012. He is currently working toward the Ph.D. degree with the signal processing Group, Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. His research interests are on Articulatory inversion, voice conversion, analysis by synthesis of speech, speech enhancement, signal processing, and deep learning.

Giampiero Salvi is Professor at the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, and Associate Professor at KTH Royal Institute of Technology, Department of Electrical Engineering and Computer Science, Stockholm, Sweden. Prof. Salvi received the MSc degree in Electronic Engineering from Università la Sapienza, Rome, Italy and the PhD degree in Computer Science from KTH. He was a post-doctoral fellow at the Institute of Systems and Robotics, Lisbon, Portugal. He was a co-founder of the company SynFace AB, active between 2006 and 2016. His main interests are machine learning, speech technology, and cognitive systems.

Torbjørn Svendsen is a professor at the Department of Electronics and Telecommunications at the Norwegian University of Science and Technology(NTNU). Dr. Svendsen received the siv.ing (MSc)and dr.ing. degrees from the Norwegian Institute of Technology (NTH) in 1980 and 1985, respectively. Dr. Svendsen has been a research scientist at SINTEF before joining NTH as an associate professor in 1988. Since 1995 he has been professor of speech processing at NTNU. He has had extended research stays at AT&T Bell Laboratories, Murray Hill, NewJersey; AT&T Labs, Florham Park, New Jersey; Griffith University, Brisbane,Australia, Queensland University of Technology, Brisbane, Australia and MIT. His research interests include automatic speech recognition; speech synthesis;speech coding and speech analysis and modeling. He has authored and co-authored more than 90 papers in these areas. Prof. Svendsen is a member of IEEE Signal Processing Society (SPS) and the International Speech Communication Association (ISCA). He has been a member of the IEEE SPS Speech Processing Technical Committee.

Sabato Marco Siniscalchi is a Professor at the University of Enna, an Adjunct Professor at the NTNU, and an Affiliate Faculty with the Georgia Institute of Technology. He received Doctorate degrees in Computer Engineering from the University of Palermo in 2006. In 2006, he was a Post Doctoral Fellow at the Ga Tech. From 2007 to 2010, he joined NTNU, Norway, as a Research Scientist. From 2010 to 2015, he was an Assistant Professor, first, and an Associate Professor, after, at the Kore University. From 2017 to 2018, he was a Senior Speech Researcher at Siri Speech Group,Apple Inc., Cupertino CA, USA. He acted as an associate editor in the IEEE/ACM Transactions on Audio, Speech and Language Processing, from 2015 to 2019. Dr. Siniscalchi is an elected member of the IEEE SLT Committee (2019-2021).

Paper F

Raw Speech-to-Articulatory Inversion by Temporal Filtering and Decimation

Abdolreza Sabzi Shahrebabaki and Sabato Marco Siniscalchi and Torbjørn Svendsen

INTERSPEECH 2021

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Raw Speech-to-Articulatory Inversion by Temporal Filtering and Decimation

Abdolreza Sabzi Shahrebabaki[1], *Sabato Marco Siniscalchi*[1,2,3], *Torbjørn Svendsen*[1]

[1]Department of Electronic Systems, NTNU, Norway
[2]Department of Computer Engineering, Kore University of Enna, Italy
[3]Georgia Institute of Technology, USA

{abdolreza.sabzi,torbjorn.svendsen}@ntnu.no, marco.siniscalchi@unikore.it

Abstract

We propose a novel sequence-to-sequence acoustic-to-articulatory inversion (AAI) neural architecture in the temporal waveform domain. In contrast to traditional AAI approaches that leverage hand-crafted short-time spectral features obtained from the windowed signal, such as LSFs, or MFCCs, our solution directly process the input speech signal in the time domain, avoiding any intermediate signal transformation, using a cascade of 1D convolutional filters in a deep model. The time-rate synchronization between raw speech signal and the articulatory signal is obtained through a decimation process that acts upon each convolution step. Decimation in time thus avoids degradation phenomena observed in the conventional AAI procedure, caused by the need of framing the speech signal to produce a feature sequence that perfectly matches the articulatory data rate. Experimental evidence on the "Haskins Production Rate Comparison" corpus demonstrates the effectiveness of the proposed solution, which outperforms a conventional state-of-the-art AAI system leveraging MFCCs with an 20% relative improvement in terms of Pearson correlation coefficient (PCC) in mismatched speaking rate conditions. Finally, the proposed approach attains the same accuracy as the conventional AAI solution in the typical matched speaking rate condition.

Index Terms: Acoustic-to-articulatory inversion, raw speech modelling, 1D-convolution, temporal convolutional network (TCN)

1. Introduction

Acoustic-to-articulatory inversion (AAI) refers to the problem of estimating the parameters that describe the movement of the articulators from the uttered speech. In recent years, AAI has attracted increasing attention because of its potential applications in speech processing. Examples include low bit rate coding [1], automatic speech recognition (ASR) [2, 3, 4], speech synthesis [5, 6], computer aided pronunciation training (CAPT) [7, 8], depression detection from speech [9, 10], and speech therapy [11, 12]. Several regression-based methods were devised to to deal with the AAI problem before the deep learning breakthrough. For example, non-parametric and parametric statistical methods, such as support vector regression (SVR) [13], joint acoustic-articulatory distribution by utilizing Gaussian mixture models (GMMs) [14], hidden Markov models (HMMs) [7], mixture density networks (MDNs) [15]. State-of-the-art approaches leverage sequence-to-sequence deep models, for example, recurrent neural networks (RNNs) in [16, 17, 18, 4, 19].

Interestingly, deep and non-deep methods focused mainly on properly tackling the high non-linearity and non-uniqueness issues in the AAI task. The speech representation commonly adopted was in the short-time frequency domain, e.g., Line Spectral Frequencies (LSFs) [20], Perceptual Linear Predictive coding (PLP) [21] and Mel-Frequency Cepstral Coefficients (MFCCs)[22]. Filter-Bank Energies (FBEs) from STRAIGHT spectra [23] have also been employed as the input of the AAI system [18], which uses a parametric modelling of the speech spectrum, and the human auditory system. Those hand-crafted speech features have been adopted due to their success in different speech processing areas, for instance, LSFs was useful in speech coding [24], and voice conversion [25], FBEs and MFCCs were widely adopted with success in speech recognition, speaker recognition [26], and voice conversion [27]. The first required step in extracting those features is the windowing of the speech signal in the time domain in the hope of satisfying the requirements of stationarity posed by the Fourier transform. However, the windowing typically has a fixed window duration and shift. A fixed analysis window and consequent constant frame rate are not optimal settings for modeling the different characteristics of different parts of speech signal [28]. In fact, non-stationary parts, such as plosives and transient speech, have shorter duration compared to the stationary parts (e.g., vowels). Such a deficiency causes a performance degradation in the final speech application, especially when the speaking rate (SR) becomes slower or faster [29] compared to a normal speaking rate: Changes in SR affect both dynamic and static properties of speech. The former are related to the duration of phonemes and their transient phase. The latter are related to the distortion in the spectrum: "This distortion may be caused by the unusual movement of articulators particularly when dealing with co-articulations" [29]. In addition, there are several works in speech applications [30, 31, 32] argueing that for particular tasks using fixed filterbanks is not the optimal choice.

The above mentioned issues motivated us to leverage 1D convolutional layers and decimation to extract suitable features for the AAI task directly in the tempora domain. The proposed solution will be presented in Section 2, where the key features to avoid the degradation phenomena are discussed. In Section 4, the experimental evidence will be reported, which clearly demonstrates the advantages of the proposed approach over conventional approaches. In particular, we demonstrate comparable results with state-of-the-art conventional AAI solution when speech features and articulatory features are synchronous. Moreover, our solution based on the raw speech waveform for articulatory inversion outperforms the conventional state-of-the-art AAI system leveraging MFCCs by an 20% relative improvement in terms of Pearson correlation coefficient (PCC) in mismatched speaking rate.

http://dx.doi.org/10.21437/Interspeech.2021-1429

2. Proposed Method

In the proposed method, the raw waveform is directly utilized to accomplish the AAI task. To deal with the mismatch in sampling rate between the acoustic speech signal and articulatory measurements, - the sampling rate of speech signal is much higher than that of the articulatory signal, a multi-stages decimation procedure is employed. Decimation can be accomplished by pooling layers - we use max-pooling layers, or leveraging the stride operation in the convolutional layers - samples are skipped while sliding the convolutional filters over the input. In this work, we employ both max-pooling layers and strides to decimate the input signal and reduce its rate to that of the articulatory signal, namely 100 Hz. The decimation is done gradually in several stages, which allows to cover a much bigger temporal span compared to that of hand-crafted features, which is limited to the frame length. Furthermore, using the max-pooling operation with overlaps provides a non-uniform downsampling of the signal that preserves the required information for the AAI task from the relevant region of speech. This is in contrast with the fixed and uniform downsampling factor needed to match the articulatory rate when extracting handcrafted speech features.

After having the decimated the input to match the target articulatory rate, a temporal convolutional network (TCN) [33, 34] is employed to captures the dynamics in the speech signal, which are beneficial for the estimation of articulators' movements. TCNs use hierarchy of temporal causal convolutions to capture short and long range patterns from the input signal leveraging upon dilated convolutions. The filter size k and dilation factor d affect the receptive field of a TCN. The receptive field of the TCN can be increased by choosing larger filter size, and augmenting the dilation factor so that the receptive field can cover the temporal length of $(k-1)d$. One of the TCN's key strengths is the possibility of parallelizing the operations in contrast to RNNs. Finally, the TCN output is fed into a 1D convolutional layer followed by a time distributed fully connected layer to estimate the articulatory information.

3. Experimental Setup

3.1. Database

The EMA method is one of the most used techniques for the recording of articulatory data, which also allows for simultaneous recording of the speech signal. One of the available databases with EMA recording is the "Haskins Production Rate Comparison"(HPRC) [35], which covers material from eight native American English speakers, namely four female (F1-F4), and four male (M1-M4) speakers. There are 720 sentences available in this database with the normal and fast Speaking Rate (SR). For some of the normal speaking utterances, there are repetitions available. The amount of data for each speaking rate (SR) is shown in Table 1, where ''N1'', ''N2'' and ''F1'' represent the normal SR; repetition of some of the sentences with the normal SR; and fast SR, respectively.

Speech waveforms are sampled at rate of 44.1 kHz, and the synchronously recorded EMA data are sampled at 100 Hz. EMA data is measured from eight sensors capturing information about the tongue rear or dorsum (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The articulatory movements are measured in the midsagittal plane in X, Y and Z direction, which denote movements of articulators from posterior to anterior, right to left and inferior to superior, respectively. In this work, we used the X and Z directions of TR, TB, TT, UL, LL and JAW for the speaker dependent AAI.

Table 1: *Available amount of data in HPRC database.*

SR	NO. utterances	Amount of data (minutes)
N1	5756	~ 244
N2	1379	~ 55
F1	5735	~ 173

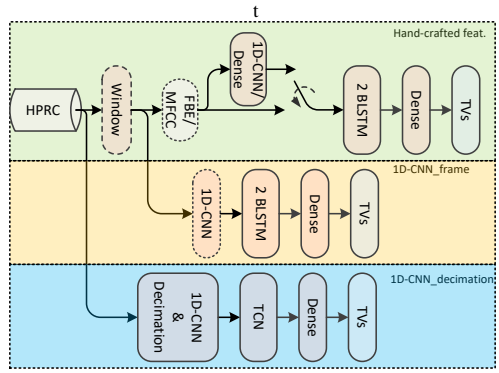


Figure 1: *The state-of-the-art S2S-AAI systems, employing (top) hand-crafted features, (middle) extracted features from speech frames by 1D-CNN, (bottom) extracted features from the whole speech sequence by 1D-CNN and decimation layers.*

The speech waveforms are downsampled to 16 kHz for performing AAI. For each of the fast and normal speaking rates, 80% of utterances are kept for training, 10% for validation data, and 10% for the test, with no overlap among them.

3.2. Input representation

In our experiments, acoustic features for the conventional AAI systems are extracted from a down-sampled waveform at 16 kHz using an analysis window of length $25ms$ with frame shift of $10ms$, yielding a frame rate to match rate of the EMA recordings. Acoustic features are calculated from 40 filters, which are linearly spaced on the Mel-scale frequency axis. Log energies in the overlapping frequency bands are called filterbank energy (FBE) features. By taking the discrete cosine transform from FBEs, MFCCs can be extracted. The first 13th cepstral features, including energy, are kept and higher cepstral features are liftered to remove the fine details of the spectral envelop.

3.3. Output representation

For the articulatory space representation, instead of using EMA measurements, tract variables (TVs) [36] are employed. TVs are relative measures and suffer less from non-uniqueness [37]. We employed nine TVs, which are obtained by geometric transformations on EMA measurements. Those TVs are Lip Aperture (LA), Lip Protrusion (LP), Jaw Angle (JA), Tongue Rear Constriction Degree (TRCD), Tongue Rear Constriction Location (TRCL). In a similar way for TB and TT we have TBCD, TBCL, TTCD and TTCL, respectively.

3.4. Neural Architectures

To better assess the proposed solution, three baseline systems are built following state-of-the-art guidelines, as shown in the top two panels in Figure 1. The first and second baseline systems employ hand-crafted features, MFCCs and FBEs. The baseline with MFCC features, **base1**, consists of two BLSTM layers with 128 cells in both forward and backward directions. The baseline with FBE features, **base2**, uses a cascade of 1D convolutional layers to extract high-level features from FBEs, and two BLSTM layers with 128 cells are used to provide dynamic information to the full connected layer to predict TVs [19]. The third baseline, **base3**, is inspired from [38], which is similar to our proposed method, but it utilizes a 1D convolutional layer to extract features over a *windowed* speech signal. In that 1D convolutional layer, 256 filters with size spanning 320 samples (20ms) are used for feature extraction; next, two BLSTM layers with 128 cells in each layer followed by a dense layer are used to predict TVs. It should be noted that a batch-normalization layer was employed after the 1D convolutional layer, following [38], to prevent vanishing gradient.

In our solution, which is showed in the bottom panel in Figure 1, the filter size of the convolutional layers can be very small due to multi-stage filtering. The first layer filter size thus spans 40 samples, which is around 2.5 milliseconds (ms): the following convolutional layer has filters with a size spanning 20 samples and with decimation through the max-pooling operator, the temporal span of second convolutional layer filters are 10ms. Filtering and decimation are carried out till features at 100Hz rate, which is equal to the TVs rate, are obtained. The time span for each of the feature vectors with rate 100 Hz is equal to $70ms$ considering all of the filtering and decimation layers. In our approach, the batch-normalization layer resulted to be useless, since there were not vanishing gradient issues. The TCN contains 64 filters with length 3 and dilation rates of power two up to 256, which is bigger than the maximum input sequence length (400 samples or 4 seconds). The TCN output are passed through a 1D convolutional layers followed by time distributed fully connected layer to predict TVs.

3.5. Performance metric

To measure the accuracy of the AAI approach, Pearson's correlation coefficient (PCC) is chosen. The PCC measures the similarity of the two trajectories, and it is a normalized score which is independent of different range of speakers' articulatory movements. The PCC measure is defined as follows:

$$\mathrm{PCC} = \frac{\sum_{i=1}^{N}(y(i)-\bar{y})(\hat{y}(i)-\bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}\left(y(i)-\bar{y}\right)^2 \sum_{i=1}^{N}\left(\hat{y}(i)-\bar{\hat{y}}\right)^2}}, \quad (1)$$

where $y(i)$ and $\hat{y}(i)$ are the ground-truth and estimated EMA values of the i^{th} frame, respectively; $\bar{y}$ and $\bar{\hat{y}}$ are mean values of $y(i)$ and $\hat{y}(i)$.

4. Experimental Results

In the first set of experiments, the goal is to compare and contrast the use of 1D convolutional filters to extract features directly in the temporal domain from either a windowed speech signal, i.e., **base3**, or without the windowing operation, i.e., our solution. Next, we compare the proposed method against all the three baseline systems in different experimental scenarios in terms of matching and mismatching SR conditions. All the AAI systems are speaker independent, and are evaluated both with matched and mismatched speakers for the training and testing. In the mismatched speaker scenario, the leave-one-speaker-out cross validation (LOSO) strategy is employed to carry out the assessment.

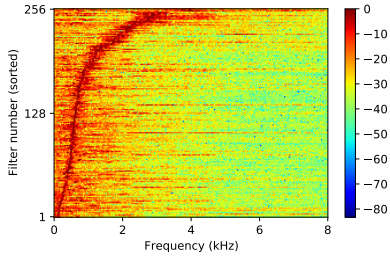


Figure 2: *The magnitude response of learned filters sorted by center frequency for **base3** system.*

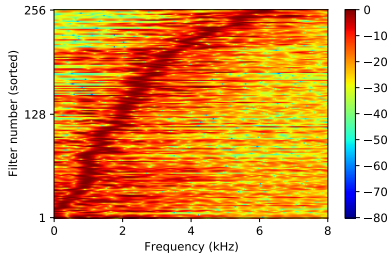


Figure 3: *The magnitude response of learned filters sorted by center frequency for the proposed method.*

4.1. 1D-CNN feature extractor

In the proposed and **base3** solutions, the first convolutional layer is extracting the features from the raw speech signal; however, a windowing pre-processing step is employed in **base3**. To better appreciate the effect of the windowing process, the characteristics of the learnt filters can be compared. To this end, the frequency response of filters is computed, and the magnitude responses are sorted by the center frequency along the frequency axis and displayed in Figure 2 for **base3**, and Figure 3 for **base3** and proposed methods, respectively. From Figure 2, it can be observed that ≈60% learnt filters' center frequency are linearly spread below 1000Hz and are non-linear above it. The highest center frequency of filters in **base3** system is less than 4000 Hz. The narrow-band magnitude response of filters can be described by the filters size which is 320 samples (20 ms). In Figure 3, due to the short filter size (2.5ms), the learnt band-pass filters have a bigger bandwidth compared to that of the **base3** system in Figure 2. Moreover, 75% of filters' center frequencies are non-linearly spread up to 3000 Hz. The center frequencies are up to 6000 Hz, which is due to short duration of the filters and therefore high frequency components of sounds do not filter-out through the filtering of first layer. The preservation of detailed information at high frequency is very useful in the estimation of TVs for high frequency sounds, such as fricatives.

Table 2: *The average PCC for different systems in the matched speaking rate condition. Spk cond indicates whether the speakers in the training and testing sets are matched or mismatched.*

Spk cond	test-SR	Proposed	base1	base2	base3
matched	N	0.84	0.83	0.80	0.81
mismatched	N	0.72	0.7	0.66	0.7
matched	F	0.79	0.79	0.73	0.78
mismatched	F	0.66	0.64	0.58	0.62
NO. Parameters		377,827	544,009	1,585,033	873,481

Table 3: *The average PCC for different systems in the mismatched speaking rate condition. Spk cond indicates whether the speakers in the training and testing sets are matched or mismatched.*

Spk cond	test-SR	Proposed	base1	base2	base3
matched	N	0.76	0.71	0.70	0.73
mismatched	N	0.65	0.52	0.56	0.61
matched	F	0.78	0.78	0.73	0.78
mismatched	F	0.68	0.67	0.64	0.66

4.2. Matched speaking rate

We now assess the effectiveness of the proposed solution in matched SR conditions. The training and test datasets have the same SR, as described Section 4, but the speaker condition, *Spk cond*, can be either matched of mismatched, as mentioned in the end of Section 4. Table 2 shows the average PCC results for different systems, where "N" and "F" stand for normal and fast SR respectively. PCC for all systems in normal SR is higher that that in fast SR. The latter is inline with what expected, since coarticulation effects are more severe in fast SR compared to those in normal SR, so capturing and tracking them is more challenging. Interesting, MFCCs allow better performance than FBEs, as observable by comparing **base1** and **base2** in Table 2. In the mismatched speaker condition, it can be observed that the system performs worse than the matched speaker condition by $\approx$0.12 in PCC for both normal and fast SR, which is expected (first and second row of Table 2). In matched speaker conditions, the proposed system attains the best results in terms of PCC and is competitive with the state-of-the-art **base1** system in fast SR. Interestingly, **base3** attains comparable or lower PCC compared to **base1** although the input features are in the temporal domain. The latter supports the discussion laid out in Section 4.1. Finally, the last row in Table 2 reports the number of parameters used in each system. A visual inspection of Table 2 allows us to argue that proposed solution attains, overall, the best results with significantly less network parameters.

4.3. Mismatched speaking rate

We now turn to the problem of performing AAI in mismatched SR. To this end, we use the systems in Section 4.2 but tested in mismatched speaking rate condition. To clear ideas: systems trained on normal SR data are evaluated on fast SR conditions, and vice versa. Table 3 summarizes the experimental evidence, in terms of average PCC, in both matched and mismatched speaker conditions. Tested on fast SR, the performance of systems trained on normal SR drops significantly compared to that obtained on normal speaking rate in Table 2. That is expected, since fast SR causes an increase in the overlap among articulators (increased coarticulation); therefore, AAI systems trained on normal SR can not model fast coarticulation movements in a proper way. However, the proposed method performance tested on fast SR achieve PCC=0.65 while the **base1** has PCC=0.52, which is a relative 20% improvement. There is no appreciable drop in PCC when systems trained on fast SR are tested on normal SR, and that is due to the fact that required information to model normal coarticulation is also available in fast SR data. By looking at the last row in Table 3, it can be observed that the results in fast SR trained model, is better when predicting the normal SR, which is another confirmation of easier prediction of TVs in normal SR which has less coarticulation.

5. Conclusion

In this work, we addressed the acoustic-to-articulatory problem is addressed in the temporal domain. Compared to conventional state-of-the-art AAI solutions based on hand-crafted short-term frequency features or windowed speech signal, 1D convolutional filters are used to extract features meaningful for the AAI task. Moreover, to match the articulatory rate, we avoid windowing, which reduce precision in capturing details at high frequency, and leverage instead decimation techniques. Moreover, a temporal convolutional network (TCN) followed by a dense layer is employed to map learned features to the TVs. Experiments are conducted on HPRC database, which provides synchronously recorded speech and EMA measurements for eight speakers. Experimental evidence demonstrates that our solution is feasible and attains top performance in mismatched speaking rate conditions, and competitive performance in matched speaking rate using however a significantly smaller amount of neural parameters.

6. Acknowledgements

This work has been supported by PRIN 2007 project nr. JNKCYZ_002.

7. References

[1] J. Schroeter and M. M. Sondhi, "Speech coding based on physiological models of speech production," *Advances in Speech Signal Processing*, pp. 231–267, 1992.

[2] J. Frankel and S. King, "ASR-articulatory speech recognition," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[3] V. Mitra, "Articulatory information for robust speech recognition," Ph.D. dissertation, University of Maryland, College Park, Maryland, 2010.

[4] A. S. Shahrebabaki, N. Olfati, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Transfer Learning of Articulatory Information Through Phone Information," in *Proc. Interspeech 2020*, 2020, pp. 2877–2881. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1139

[5] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 207–219, 2013.

[6] K. Richmond and S. King, "Smooth talking: Articulatory join costs for unit selection," in *ICASSP*, 2016, pp. 5150–5154.

[7] T. Hueber, A. Ben Youssef, G. Bailly, P. Badin, and F. Elisei, "Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training," in *Interspeech*, 2012, pp. 783–786.

[8] O. Engwall, "Analysis of and feedback on phonetic features in pronunciation training with a virtual teacher," *Computer Assisted Language Learning*, vol. 25, no. 1, pp. 37–64, 2012.

[9] B. S. Helfer, T. F. Quatieri, J. R. Williamson, D. D. Mehta, R. Horwitz, and B. Yu, "Classification of depression state based on articulatory precision." in *INTERSPEECH*, 2013, pp. 2172–2176.

[10] S. Sahu and C. Y. Espy-Wilson, "Speech features for depression detection." in *INTERSPEECH*, 2016, pp. 1928–1932.

[11] D. W. Massaro, S. Bigler, T. Chen, M. Perlman, and S. Ouni, "Pronunciation training: the role of eye and ear," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[12] S. Fagel and K. Madany, "A 3-d virtual head as a tool for speech therapy for children," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[13] A. Toutios and K. Margaritis, "A support vector approach to the acoustic-to-articulatory mapping," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[14] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[15] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[16] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *ICASSP*, 2015, pp. 4450–4454.

[17] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501.

[18] A. S. Shahrebabaki, N. Olfati, A. S. Imran, S. M. Siniscalchi, and T. Svendsen, "A Phonetic-Level Analysis of Different Input Features for Articulatory Inversion," in *Interspeech*, 2019, pp. 3775–3779.

[19] A. S. Shahrebabaki, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-Sequence Articulatory Inversion Through Time Convolution of Sub-Band Frequency Signals," in *Proc. Interspeech 2020*, 2020, pp. 2882–2886. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1140

[20] R. Korin, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Interspeech*, Florence, Italy, August 2011, pp. 1505–1508.

[21] C. Qin and M. Á. Carreira-Perpiñán, "A comparison of acoustic features for articulatory inversion," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[22] P. K. Ghosh and S. Narayanan, "Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. EL251–EL257, 2011.

[23] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds1speech files available. see http://www.elsevier.nl/locate/specom1," *Speech Communication*, vol. 27, no. 3, pp. 187 – 207, 1999.

[24] C. O. Mawalim, S. Wang, and M. Unoki, "Speech information hiding by modification of lsf quantization index in celp codec," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 1321–1330.

[25] M. Ghorbandoost, A. Sayadiyan, M. Ahangar, H. Sheikhzadeh, A. S. Shahrebabaki, and J. Amini, "Voice conversion based on feature combination with limited training data," *Speech Communication*, vol. 67, pp. 113–128, 2015.

[26] X. Liu, M. Sahidullah, and T. Kinnunen, "Learnable MFCCs for Speaker Verification," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, South Korea, May 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03139532

[27] A. S. Shahrebabaki, J. Amini, H. Sheikhzadeh, M. Ghorbandoost, and N. Faraji, "Reduced search space frame alignment based on kullback-leibler divergence for voice conversion," in *Advances in Nonlinear Speech Processing*, T. Drugman and T. Dutoit, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 83–88.

[28] Z.-H. Tan and I. Kraljevski, "Joint variable frame rate and length analysis for speech recognition under adverse conditions," *Computers & Electrical Engineering*, vol. 40, no. 7, pp. 2139–2149, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790614002304

[29] X. Zeng, S. Yin, and D. Wang, "Learning speech rate in speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[30] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[31] H. Seki, K. Yamamoto, and S. Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5480–5484.

[32] J. Fainberg, O. Klejch, E. Loweimi, P. Bell, and S. Renals, "Acoustic model adaptation from raw waveforms with sincnet," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 897–904.

[33] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.

[35] M. Tiede, C. Y. Espy-Wilson, D. G. V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[36] G. Sivaraman, V. Mitra, H. Nam, M. Tiede, and C. Espy-Wilson, "Unsupervised speaker adaptation for speaker independent acoustic to articulatory speech inversion," *The Journal of the Acoustical Society of America*, vol. 146, no. 1, pp. 316–329, 2019. [Online]. Available: https://doi.org/10.1121/1.5116130

[37] R. S. McGowan, "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests," *Speech Communication*, vol. 14, no. 1, pp. 19 – 48, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167639394900558

[38] A. Illa and P. K. Ghosh, "Representation learning using convolution neural network for acoustic-to-articulatory inversion," in *ICASSP*, 2019, pp. 5931–5935.

ISBN 978-82-326-6629-4 (printed ver.)
ISBN 978-82-326-6324-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology