# The Impact of Audible Feedback on EMG-to-Speech Conversion

Zur Erlangung des akademischen Grades eines
**Doktors der Ingenieurwissenschaften**
des Fachbereich 3
der Universität Bremen

vorgelegte

Dissertation

von

Lorenz Diener

Erste Gutachterin: Prof. Dr.-Ing. Tanja Schultz
Zweiter Gutachter: Thomas Hueber, Ph.D., HDR

# Abstract

Research interest in speech interfaces that can function even when an audible acoustic signal is not present – so-called *Silent Speech Interfaces* – has grown dramatically in recent years, as the field presents many barely exploded avenues for research and huge potential for applications in user interfaces and prosthetics.

EMG-to-Speech conversion is a type of silent speech interface, based on electromyography: It is the direct conversion of a facial electrical speech muscle activity signal to audible speech without an intermediate textual representation. Such a direct conversion approach is well suited to speech prosthesis and silent telephony applications and could be used as a pre-processing step to enable a user to use a regular acoustic speech interface silently. To enable these applications in practice, one requirement is that EMG-to-Speech conversion systems must be capable of producing output in real time and with low latency, and work on EMG signals recorded during silently produced speech.

The overall objective of this dissertation is to move EMG-to-Speech conversion further towards practical usability by building a real-time low-latency capable EMG-to-Speech conversion system and then use it to evaluate the effect of audible feedback, provided in real-time, on silent speech production.

Specific issues we address in this dissertation include:

**Enabling real-time low-latency EMG-to-Speech conversion:** We build a low-latency EMG-to-Speech conversion system capable of operating with close to no delay. We introduce new EMG preprocessing and feature extraction methods to achieve this goal, and implement them in a flexible framework that enables pipelined, multi-threaded real-time processing of biosignals. We also investigate the potential of different approaches for mapping EMG features to audio features, and different vocoding techniques used to convert audio features to an audio waveform.

**Practical corpora and evaluation:** We record several new data corpora designed to evaluate specific facets of EMG-to-Speech conversion. Specifically, we record data for comparing the conversion of isolated and continuous speech, data to investigate the potential of speak-along recordings for training and evaluating EMG-to-Speech conversion, and data to investigate how to compensate for time-correlated changes in the EMG signal. We also introduce and verify the TLAcc a method to better compare $F_0$ trajectories of two pieces of speech for similarity.

**The effect of feedback:** We perform a study in which participants can hear either simplified or complex (full-speech) system feedback output while they are speaking and investigate whether this changes user speech – either while feedback is present, or even afterwards as the participant learns to use the system. We find limited evidence for the former for sessions in which the system was able to produce reasonably accurate feedback.

# Zusammenfassung

Das Forschungsinteresse an Sprachschnittstellen, die auch dann noch funktionieren, wenn ein hörbares akustisches Signal gar nicht vorhanden ist – sogenannte *Silent Speech Interfaces* – ist in den letzten Jahren stark angestiegen, da das Forschungsgebiet viele bisher kaum untersuchte Forschungsmöglichkeiten bietet und ein großes Potential für praktische Anwendungen im Bereich der Benutzerschnittstellen und Prothesen hat.

EMG-to-Speech-Konversion ist eine Art Silent Speech Interface, basierend auf Elektromyographie. Es handelt sich hierbei um die direkte Umsetzung von Sprach-Gesichtsmuskelaktivität in hörbare Sprache ohne eine textuelle Zwischenstufe. Ein solcher Ansatz ist gut zum bauen von Sprachprothesen und für stille Telefonie geeignet und kann zudem als Vorstufe verwendet werden um die stille Benutzung von herkömmlichen Sprachschnittstellen zu ermöglichen. Damit dies gelingen kann ist es notwendig, dass das EMG-to-Speech-Konversionssystem seine Ausgabe in Echtzeit und mit niedriger Latenz sowie mit EMG-Signalen die während dem stillen Sprechen aufgezeichnet werden funktioniert.

Das übergeordnete Ziel dieser Dissertation ist es, EMG-to-Speech-Konversion durch die Entwicklung eines Online-fähigen EMG-to-Speech-Systems näher an die praktische Verwendbarkeit zu bringen und mit diesem den Einfluss von hörbarem Feedback auf EMG-to-Speech-Konversion zu untersuchen.

Im speziellen beschäftigen wir uns in dieser Dissertation mit folgenden Problemkomplexen:

**EMG-to-Speech-Konversion in Echtzeit und mit niedriger Latenz:** Wir entwickeln ein onlinefähiges EMG-to-Speech-Konversionssystem, das nahezu ohne Latenz arbeitet. Um dieses Ziel zu erreichen führen wir neue EMG-Features und Vorverarbeitungsmethoden ein und implementieren sie als Teil eines flexiblen Frameworks, das uns ermöglicht, Biosignale effizient und in mehrere Stufen aufgeteilt sowie parallelisiert zu verarbeiten. Wir untersuchen das Potential verschiedener neuartiger Ansätze für die Konversion von EMG- zu Audiofeatures und evaluieren verschiedene Vocoding-Ansätze, die die Audiofeatures zur Ausgabe in eine Audio-Waveform überführen.

**Praxisorientierte Korpora und Evaluationsansätze:** Wir nehmen mehrere neue Datensätze auf, um bestimmte Facetten der EMG-to-Speech-Konversion genauer zu Untersuchen. Insbesondere nehmen wir Daten auf, um isolierte Worte mit kontinuierlicher Sprache zu vergleichen, machen Aufnahmen bei denen eine vorhandene Audiodatei still mitgesprochen wird, um das Potential solcher Aufnahmen Für Training und Evaluation zu untersuchen, und präsentieren einen Korpus, mit dem Methoden zur Kompensation von Zeitkorrelierten Änderungen im EMG-Signal untersucht werden können. Wir präsentieren des weiteren eine neue Methode zum Vergleichen zweier $F_0$-Trajektorien – die TLAcc.

**Der Einfluss von hörbarem Feedback:** Schlussendlich führen wir eine Studie durch, im Rahmen derer Teilnehmer*innen beim stillen Sprechen entweder vereinfachtes (Sprachkorreliertes Summen) oder komplexes (Hörbare Sprache) Feedback hören. Hierbei untersuchen wir, ob und wie sich dies auf die Sprachproduktion auswirkt – entweder während Feedback vorhande ist, oder auch darüber hinaus. Wir können letztere Annahme nicht bestätigen, finden aber teilweise Belege für die erstere Annahme – für diejenigen Aufnahmen in denen das System gut kontrolliert werden konnte.

# Acknowledgements

A dissertation is written by one person, but not by one person working alone. Completing this work would not have been possible if not for the people who have helped and supported me along the way.

First of all, I would like to thank my primary supervisor, Prof. Dr.-Ing. Tanja Schultz, for her support throughout my dissertation. Without her, I would have neither had the opportunity to work on silent speech processing research, nor the support to continue this work. Her feedback, not only on scientific matters but also on matters of development as a scientist, was invaluable during the course of my PhD. She opened up opportunities for me by enabling me to travel to conferences and get into contact with other scientists, allowing me to expand both my network and my perspectives and knowledge of current research.

In the same vein, want to thank Thomas Hueber, Ph.D., HDR, not only for co-supervising my dissertation, but also for his valuable feedback and the interesting discussions we had at conferences and during his visit to the Cognitive Systems Lab in Bremen.

During my PhD, I received much support from the colleagues I worked with. The discussions we had and practical experience they shared were instrumental in guiding my work. I would like to especially thank Jochen Weiner, Felix Putze, Christian Herff and Miguel Angrick, who helped me much in this regard. I would also like to thank Matthias Janke, on whose work in EMG-based Silent Speech interfaces I build with this dissertation. I would also like to thank Elke Nakonetzki and Eric Schädler – their support in administrative and technical matters saved me many headaches over the years.

I also wish to thank the students which I was allowed to supervise during my time as a PhD student. Some of the work they did, as part of a thesis, internship or research assistant job, has resulted in them being co-authors on papers of mine, and supervising them has allowed me to gain valuable experience in mentoring others. I would like to especially thank Gabriel

Ivucic, Darius Ivucic, Gerrit Felsch, Kevin Scheck, Tejas Umesh and Varshini Prakash.

Finally, I wish to thank my friends and especially my family – my parents Albrecht and Marina Diener and my brother Vincent Diener, as well as my significant other Wendy Tran, for their encouragement, patience and support during the past 6 years, and for being there for me whenever I needed them.

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

*This chapter introduces and motivates the goals of this thesis. It first gives a brief introduction to Silent Speech Interfaces based on direct synthesis and explains the importance of real-time feedback. Thereafter, it summarizes the main contributions of this thesis. Finally, it presents a brief outline of the structure of this document.*

## 1.1 Speech technology

Audible speech – be it face to face or via telephony – is the primary way in which humans communicate with each other. With advancements in computing power and speech technology research, speech communication has become even more important as speech-based man-machine-interfaces have become ubiquitous. Where available, speech interfaces offer a natural, mobile, hands- and eyes-free alternative to touch-based means of interaction.

Most people, in most situations, do not have any trouble using these speech interfaces – however, there also many reasons why people may be unable or unwilling to use an audible acoustic speech interface:

***Acoustic noise sensitivity:*** In environments where background noise drowns out the audible acoustic speech signal – e.g. at a busy airport or in a car – the performance of audible speech interfaces degrades.

In extreme cases, such as on a factory floor loud enough to require hearing protection, they may stop working altogether.

***Disturbing bystanders:*** In places where silence is expected, such as in a library or on public transport, speech interfaces cannot be used. Even in places where silence is not expected, quiet may be preferable – such as in a call center, where reduced noise levels make for a better work environment. Finally, a user of a speech interface may be wary of transmitting confidential information such as private data, passwords or pin codes in case someone is listening in.

***Inability to produce a clean speech signal:*** Finally, people who cannot produce speech, e.g. laryngectomees, often cannot use speech interfaces at all or can only use such interfaces with degraded performance.

For speech technology, such people and situations pose an important challenge: The more pervasive such technology becomes, the more difficult it will be to justify speech based interfaces simply not being able to handle some situations, or the exclusion of people from services simply because they are unable to produce a clean speech signal.

## 1.2 Silent Speech Interfaces

In those situations, where regular speech interfaces fail to deliver, *Silent Speech Interfaces* (SSIs) – speech interfaces that do not rely on the presence of an audible acoustic signal to function – can continue to function, expanding the scope of situations in which speech can be used to communicate.

SSIs have been built using many different modalities [SWH$^+$17] – examples include ultrasound [GGT$^+$18, FHG$^+$17], permanent-magnetic articulography [GCG$^+$16], microwave radar [BSW$^+$18], *surface electromyography* (sEMG, with muscle movement [JD17] or sub-vocal [KKM18]), non-audible murmur recorded with a throat microphone [TS05] or even electrocorticography [HJD$^+$16] – for a more thorough overview of SSIs, see Section 3.2.

With such signals, it is possible to try to solve a variety of speech tasks, such as recognizing speech (automatic speech recognition) [WJH$^+$14], analysis of speech production to investigate how speech and language function on a physiological level, or, as is the focus of my research, generating an audible speech signal from a non-audible one (direct synthesis).

## 1.3 EMG-to-Speech Conversion

In my thesis, I focus on one specific type of SSI – the direct conversion of surface electromyographic signals to audible speech in real time and with low latency, or EMG-to-Speech for short, using a statistical mapping of the EMG signal to an audio representation which can be synthesized to generate immediate audible acoustic output. Such interfaces have various advantages over other approaches:

***Surface EMG is less-invasive:*** Unlike sensors such as needle EMG, electrocorticography or EMA/PMA, surface EMG does not require puncturing the skin, implanting devices or even the insertion of sensors into the oral cavity. This is a critical advantage for any sensing modality meant to drive practical user interfaces, especially interfaces for healthy users.

***No linguistic restrictions:*** As direct synthesis attempts to directly convert the input sensor signal to an audio signal without an intermediate text step, it does not require a vast amount of language-specific information (such as a pronunciation dictionary or phone set) – only a set of parallel signals is required for training.

***Direct synthesis can preserve paralingual information:*** The paralingual contents of acoustic speech (e.g. intonation, stress or pitch and speaker identity) are an important part of speech communication. They carry emotional context and can help disambiguate language (e.g. “I never said *she* stole my money” versus “I never said she *stole* my money.”. Direct-Synthesis SSIs can capture such information, whereas SSI approaches based on speech recognition require an intermediate text form that discards it.

***Direct synthesis enables user-in-the-loop approaches:*** Feedback is an integral part of the speech production process. While the primary feedback mechanism for human speech is somatosensory feedback [HN11], silent speech is not merely modal speech without the sound – it is not a natural way of speaking, and thus presents an adverse condition that humans have to consciously address during (silent) speech production. It is known that humans produce speech differently when speaking silently [WTJS09], and complex adaptation of speakers to their acoustic environment [Lom11] as well as the issues caused by distortion of audible feedback [SKRL02] are well documented phenomenona. Mode differences like this are a problem for Silent Speech

Interfaces, which are usually trained on audibly-spoken speech (because they require parallel audio and sensor data). A direct synthesis system is able to produce such feedback, which could allow users to adapt to the system, increasing performance.

***Conversational use:*** As direct synthesis systems are able to produce output right away rather than having to make a user wait until, at worst, a complete sentence has been spoken, they enable a natural conversational usage flow that would be hard to achieve with a recognition based system.

## 1.4 Problem Statement and Main Contributions

Previous approaches to EMG-to-Speech conversion have always been tested in an offline context – making several assumptions that must be overcome in moving towards a more practical system. With this thesis, we attempt to address several of those assumptions.

### 1.4.1 Mode Differences and Parallel Data

It is known that there are differences between the facial movements during audible speech production versus speaking silently in terms of the produced EMG signal [WTJS09]. These differences are neither simple hyper- nor hypoarticulation, but rather than that are complicated and difficult to quantify. Since they are differences in movement, they affect the EMG signal. This is a problem for an EMG-to-Speech system trained on audible data: Since the input is different, output quality degrades. In this thesis we explore and evaluate two approaches to addressing this problem:

- We obtain parallel silently-spoken EMG and audibly-spoken speech through a speak-along recording procedure.
- We evaluate reducing the impact of mode differences via audible feedback.

In our evaluation, our goal is to test whether audible feedback changes how people speak and if the presence of audible feedback improves the performance of silent speech conversion.

### 1.4.2 Building an Online EMG-to-Speech Conversion System

The engineering challenges of building a practically viable online EMG-to-Speech conversion system are numerous. Each component of such a system (both the hardware and software components) must be real-time capable and cannot introduce large processing latencies, and the entire system must be implemented in a way that minimizes delays, as feedback that is delayed beyond 50 ms is known to cause changes in speaking behaviour, including a higher rate of disfluencies [SKRL02]. Additionally, in practice, the system has to be designed to be robust towards changes in the signal and artifacts. One of the contributions of this thesis is building such a system:

- We build a system that uses a pipeline architecture to minimize processing delay and allow for the use of multi-processing in the conversion system.
- We evaluate the performance of EMG-to-Speech conversion over time within a session.
- We evaluate how to reduce the time taken to train a system and the potential of adaptation in improving performance.

In our evaluation, we aim to test whether adaptation within a session can improve the performance later in the session, and whether we can adapt a pre-trained system to a new speaker.

### 1.4.3 Improving synthesis quality

The improvements in speed and practicable usability come at the price of quality. Therefore, even though the main focus of this thesis is real-time capability, we nevertheless also evaluate ways of improving the output quality of our system:

- We evaluate improving the EMG-to-Speech feature mapping using different feature transformation approaches.
- We explore the use of different synthesis methods in EMG-to-Speech conversion, including neural vocoders.

In our evaluation, we test if more complex neural architectures can generate output with a better quality than a feed-forward neural network, and whether using neural vocoders can improve the output quality of an EMG-to-Speech conversion system.

### 1.4.4 Implications of the Low-Latency Condition

The latency requirements imposed by the requirement to produce output that, ideally, is concurrent with a users expectation of when speech production should take place mean that the state of the art TD-15 feature set (which requires future context at several stages for feature extraction) used in offline EMG-to-Speech conversion cannot be used in online EMG-to-Speech conversion systems, so an alternative is required:

- We introduce and evaluate a different set of features that are causal and can thus be computed with low latency and without future context.

In our evaluation, we test how much the output quality of EMG-to-Speech conversion is reduced by not being able to rely on future context.

### 1.4.5 Evaluation

Finally, when trying to evaluate direct synthesis Silent Speech Interfaces, there is always the question of how to produce performance metrics. Common audio quality and audio comparison metrics work well when the signals are not very distorted, but when distortion is high (as is the case in current generation direct synthesis SSIs), the metrics become less meaningful. Human listening tests are the gold standard, however, they are time-consuming and impractical when exploring a large space of potential parameters or methods. Additionally, for evaluating silent operation (where no reference is available) it is not possible to compute metrics which do require such a reference. We contribute to the problem of evaluating direct-synthesis SSIs in several ways:

- We evaluate methods for obtaining reference audio for silently spoken speech – alignment via DTW as well as evaluation using speak-along data.

- We introduce and evaluate a new metric for comparing fundamental frequency trajectories, the TLAcc.

In our evaluation, we test how these measures compare to other objective measures and to gold standard human evaluations of output quality.

## 1.5 Structure of the Thesis

The rest of this thesis is structured as follows: Chapter 2 introduces the physiological and physical basis of the EMG signal and its recording as well as the fundamental principles of voice conversion. Chapter 3 introduces related work to position this work in the space of SSI research. It also introduces the principles underlaying the baseline offline EMG-to-Speech conversion system used in comparisons between online and offline EMG-to-Speech conversion. Chapter 4 gives an overview of the recording setups, data corpora (previous work as well as corpora newly recorded as part of this thesis) used in this work. Chapter 5 introduces the signal processing and features used in both conversion as well as evaluation and studies performed to evaluate different aspects of EMG-to-Speech conversion. Chapter 6 describes the new low-latency EMG-to-Speech conversion system as a whole as well as its components. Chapter 7 presents the results of our user-in-the-loop study. Finally, Chapter 8 summarizes our results and provides an outlook on potential future avenues for EMG-to-Speech conversion research.

Chapter 2

# Background

*This chapter introduces the background necessary to understand the contributions made by this dissertation. It gives an overview of the speech EMG signal – the biophysiology of its generation and how it is measured – as well as the basics of audible speech. It additionally introduces voice conversion as the foundation on which EMG-to-Speech conversion is built. Finally, it provides some theoretical background on the statistical methods employed in this dissertation.*

## 2.1 The Speech EMG Signal

To understand the speech EMG signal, it is necessary to understand both the anatomy of human muscles as well as that of acoustic speech production in addition to how they relate to each other. The following sections give a brief overview of these topics and then introduce how the EMG signal is measured.

### 2.1.1 Muscle Anatomy

All human movement is caused by the contraction of muscles. Muscles are tissue which, when electrically stimulated, contract. This contraction is initiated by potentials entering the muscle via the nervous system and triggered by, in case of voluntary movement, the brain. There are different types of muscles: Smooth muscles, the cardiac muscle and skeletal muscles.

**Smooth muscle** s are not generally voluntarily innervated (though some can be). They are often controlled by parts of the nervous system intrinsic to the organ they are a part of itself, or from the autonomous nervous system. An example are the muscles of the gastrointestinal tract.

**The cardiac muscle** is a special case of non-voluntarily innervated muscle tissue, specific to the heart. Physiologically, it has properties similar to both skeletal and smooth muscles.

**Skeletal muscles** are the voluntarily innervated muscles that control movement of the skeletal system. Since speech is produced by voluntary movement, the rest of this work will focus entirely on such skeletal muscles

Skeletal muscles are connected to bones or other movable structures in the body (such as the eye) via tendons at a minimum of two points. They move these structures by contracting, pulling the attachments points towards each other. Since muscles can only contract, not expand, they often come in antagonistic pairs where the contraction of a muscle relaxes its antagonist and vice-versa – one muscle to move a bone in one direction, and another to move it in the opposite one. An example of such an antagonistic pair are the biceps and triceps.

Skeletal muscle movement is initiated through the activation of at least one *motor neuron* in the spinal cord by a nerve action potential conducted to the motor neuron from the brain via the nerves of the central nervous system. The synapses of such a motor neuron (called *neuromuscular junctions*, made up of an axon on one side of the synaptic cleft and muscle cell on the other) are connected to many *myocytes* (muscle fibers - the cells making up the tissue of a muscle). Together, a motor neuron and all the muscle fibers it innervates make up the smallest unit of a muscle that can be innervated (and thus, made to contract) on its own. Together, they are called a *motor unit.*

### Principle of Muscle Fiber Contraction

In rest, the membranes of a muscle fiber are negatively polarized with a potential difference of circa $-70mV$ due to a difference in the concentration of sodium ($Na^+$), potassium ($K^+$) and chloride ($Cl^-$) ions inside and outside

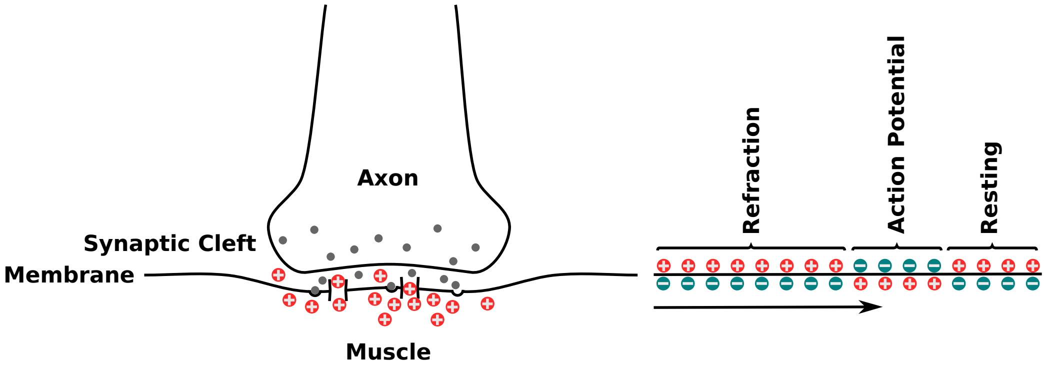


**Figure 2.1** – Generation (left) of action potentials and their conduction (right) along a muscle fiber.

of the muscle fiber cell. The membrane allows $K+$ ions to pass to the outside of the cell. This diffusion of ions continues until the electrical potential and diffusion pressure balance out.

Upon activation, the motor neuron releases the neurotransmitter *acetylcholine* into the synaptic cleft, which binds to post-synaptic receptors that cause the membrane potential to become slightly more positive. Once the membrane potential reaches a critical potential, called the threshold potential, the cells sodium/potassium channels open, allowing $Na+$ to rush into the cell, which causes the local membrane potential to rapidly become even more positive, resulting in an *action potential*. The local depolarization causes further opening of sodium channels along the muscle fiber, allowing the action potential to progress along it [SL11, p. 30ff]. The total resulting action potential of all muscle fibers innervated by one motor neuron simultaneously is called a *Motor Unit Action Potential* (MUAP). A sequence of such motor unit action potentials generated from a single motor unit is called a "MUAP train". Figure 2.1 illustrates this action potential generation and conduction.

Inside the cell, this causes (via the secretion of another ion, $Ca^{++}$ [MP13, p. 17f]) *myosin heads* inside the *myofibrils* to repeatedly bind to *actin filaments* and fold over, causing the myosin and actin filaments inside the muscle cell to slide past each other and shortening the cell [SL11, p. 26f].

After some time, the $Na+$ channels close again. Sodium is once again prevented from streaming into the cell, which allows the $K^+$ ions to return the cell membrane to its resting potential (superfluous $Na^+$ ions are eventually removed from the cell by the *sodium-potassium pump*) and the cell is ready to start over the process and contract again.

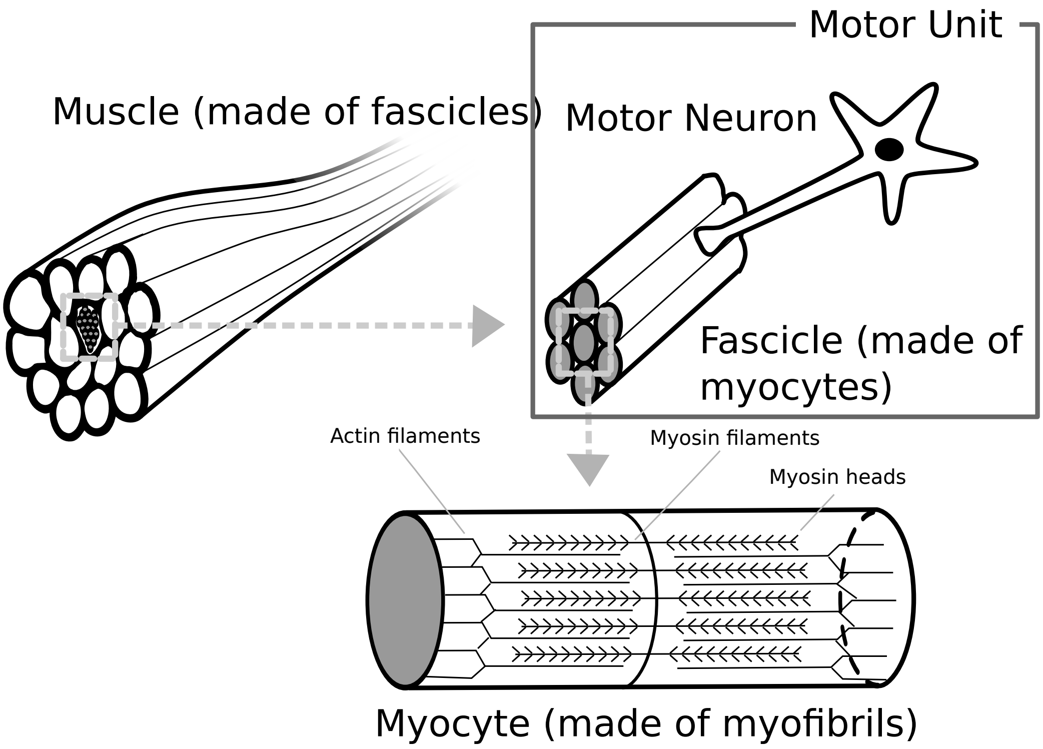

Motor Unit
Muscle (made of fascicles)
Motor Neuron
Fascicle (made of myocytes)
Actin filaments
Myosin filaments
Myosin heads
Myocyte (made of myofibrils)

**Figure 2.2** – Schematic depiction of the structure of a skeletal muscle.

### Muscle Structure

The previous section explained how a single muscle fiber is structured and how it contracts. A whole muscle is made of many such muscle fibers, connected to and innervated by one or more motor neurons. Figure 2.2 shows the structure of a skeletal muscle down to actin and myosin fibers.

The contraction of an entire muscle is achieved by the contraction of some or all of its muscle fibers. There are two basic mechanism for controlling the force with which a muscle contracts: Motor unit recruitment and rate coding.

Motor unit recruitment means the activation of an increasing number of motor units making up a muscle. Motor neurons are recruited by the central nervous system by size – i.e. by the number of muscle fibers they control, from smallest to largest. The more muscle fibers are part of a contraction, the greater the force with which a muscle can contract. How the sizes of motor units are distributed differs between muscles, depending on function: Muscles that need to contract with great precision may have motor units controlling very few muscle fibers (e.g. the ocular muscles, with units with an *innervation*

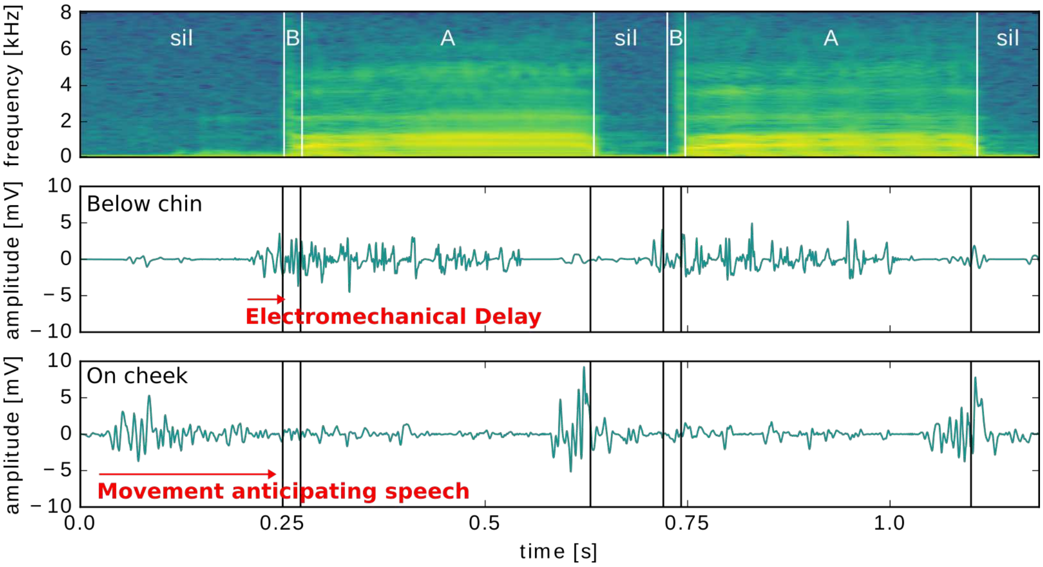


**Figure 2.3** – Illustration of two kinds of time offset between acoustic speech signal (spectrogram, above) and related EMG signals (below): A movement that anticipates the resulting acoustic speech, and electromechanical delay (EMG signals have been strongly band-pass filtered for illustration purposes, phone locations labeled manually).

*ratio* of as low as 3 [RG11, p. 183]), whereas muscles that do not need such precision will have motor units with sizes upwards of 1000 [KHJG01, p. 7].

Rate coding, on the other hand, refers to the repeated activation of motor units. The greater the rate at which a motor unit is repeatedly activated, the greater – up to the *maximum voluntary contraction* (MVC) – the contraction of the fibers that are part of the motor unit.

How much each of these methods contributes to the contraction of a muscle depends on the structure of the muscle. Generally, for most muscles, maximum motor unit recruitment is achieved at 50% MVC force, and at up to 80% for larger muscles [MP13, p. 6ff]. Additionally, the recruitment pattern may change as muscle fatigue increases.

### Electromechanical Delay

Muscles, when innervated, do not move instantly - there is a small delay between membrane depolarization and movement onset, called the *electromechanical delay* (EMD) [CK79] and, for speech, further delay between the onset of movement and resulting sound. Figure 2.3 illustrates these effects. The exact delay depends on the muscle and innervation speed and strength as well as the sound that is being produced. In most EMG-to-Speech conversion research, the treatment of EMD has been comparatively simple: The EMG signal was universally delayed by 50 milliseconds, a value that was empirically determined in [JSW+06]. As this work deals with real-time low-latency EMG-to-Speech conversion, this isn't possible here: Any delaying of the EMG signal relative to the audio signal results in a system being trained to generate delayed output. We will therefore compare the effect of compensating for EMD on different types of features later in this work.

### 2.1.2 The Surface EMG Signal

As explained above, the movement of skeletal muscles is controlled by electric membrane potentials. These potentials, through volume conduction in the tissue surrounding muscle fibers, can be measured via surface electrodes. This is the basis of surface EMG recording.

There are two basic electrode configurations for EMG recordings: Unipolar derivation and bipolar derivation. In either case, the EMG signal is measured as a potential difference between two electrodes and differentially amplified to maximize common mode rejection (in practice, with a common ground and driven right leg for noise suppression [VRPG90] – explained later in this section).

The actual interface between ion conduction inside the body and electron conduction inside an electrical cable is the *electrode*. Conversion between ionic and electron conduction is achieved by a chemical reduction/oxidation reaction (meaning that such electrodes eventually have to be replaced or re-coated) [OT 19]. The majority of electrodes used in EMG are typically made from silver and silver-chloride [MP13, p. 125f] (and are therefore usually referred to as "Ag/AcCl" electrodes). However, the electrodes with which the data presented as a part of this dissertation was recorded are instead made from gold plated copper [OT 19].

Recording is performed either in a *unipolar* configuration, between an electrode on an electrically active area and a reference electrode on an electrically inactive area, or in a *bipolar* configuration, between two (usually close by) electrodes both on the electrically active area (see Figure 2.4). The advantage of bipolar measurement is that due to a smaller distance between the electrodes, noise is more likely to affect both electrodes in the same way, making it effectively suppressible by differential amplification.

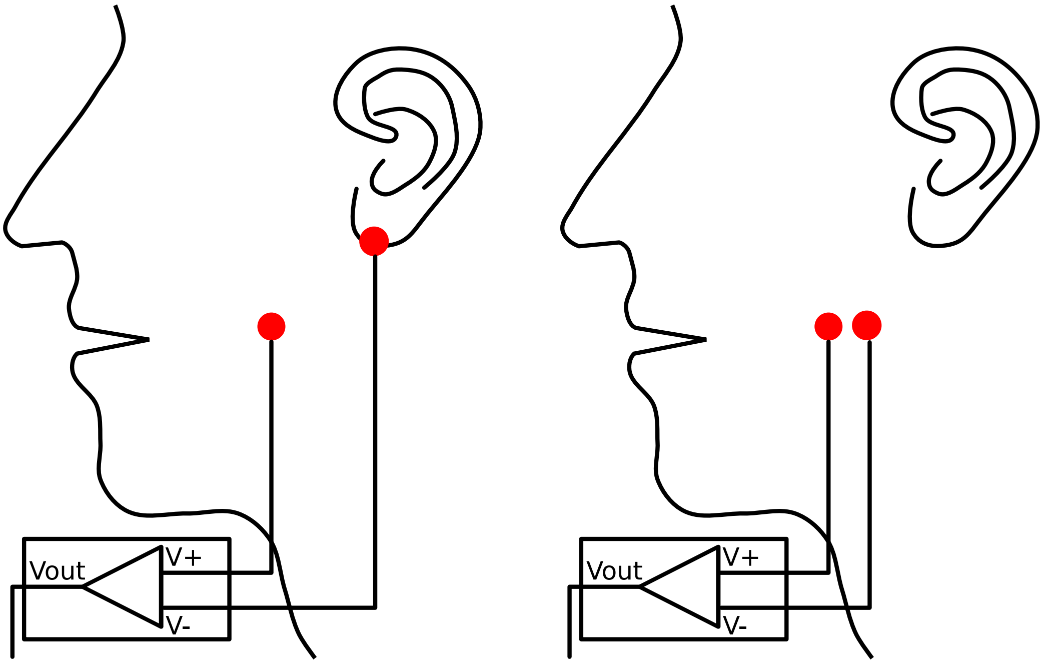


**Figure 2.4** – Unipolar (left, with the reference electrode on the electrically inactive ear lobe) versus Bipolar (right, deriving between two electrodes on electrically active territory) signal derivation.

Five factors determine what kind of signal arrives at an EMG electrode: The active motor units, the units firing rate, the position of the electrodes relative to the motor unit myocytes recorded, the conduction between myocytes and electrodes, and body-internal as well as external noise and artifacts. In the following, each will be briefly considered.

The *amount of active motor units* and the firing rate (also called *rate coding*) have, from a surface EMG perspective, a similar effect: Both result in more motor unit action potentials per second. Whether these MUAPs belong to the same MUAP train or not does not matter – while it is possible to record single fibers and differentiate single action potentials with invasive EMG recording, this is not possible with surface EMG, where they primarily result in an increase in signal amplitude.

The reason for this is the effect of *volume conduction* in tissue on the surface signal. These effects are two-fold: It leads to a spatial low-pass filtering (washing-out over space) of the signal [MP13, p. 89], as well as the summation of signals from multiple sources (either different fibers within the same muscle, or different muscles).

The main effect of *electrode position* relative to the signal source is attenuation. While the human body is not a perfectly homogeneous conductor, the signal energy of EMG potentials still broadly decreases with the inverse of the square of the distance between fiber and electrode [Lag02, p. 31ff]. The other effect is the position of an electrode pair with regard to muscle fiber direction – electrode pairs parallel to the muscle fiber direction will measure a stronger signal than electrode pairs orthogonal to it. Figure 2.1 shows why: If electrodes are placed orthogonal to a fiber, the action potential reaches them at the same time, and no potential difference can be measured.

Finally, there are many *artifacts* to consider in electrophysiological recordings. These can be split into two groups: Biological artifacts and technical artifacts.

Biological artifacts are artifacts that originate within the body. Examples include:

**Muscle cross-talk:** Due to the effects of volume conduction, it is not easily possible to record signals just from a specific muscle or group of muscles. Instead, the signal will often contain EMG signals originating from muscles we did not intend to record. This is called “muscle cross-talk”. A special case of muscle cross-talk is interference by the heart muscle, which (for facial sEMG) can occur when detachment of one electrode of a pair effectively causes derivation between the ground electrode attached to an extremity and the electrode still attached to the face.

**Other electrophysiological interference:** There are other sources of electrical fields in the body. An artifact that can sometimes be found in facial EMG recordings close to the eye is interference from the electric dipole of the eyes (the so-called electrooculogram).

**Movement artifacts:** Finally, muscle contraction changes the shape of the muscle and surrounding tissue unless counteracted, which changes the properties of the volume conductor and can therefore result in low-frequency artifacts. For this reason, medical EMG is often measured under isometric contraction (contraction of a muscle without changing its length, e.g. pushing against a solid obstacle without any actual motion). For user interface applications, we cannot in general restrict the users movement, so this is not an option, however, movement artifacts are a type of artifact that can be target-correlated: Since we are interested in generating speech signals that the movement would have resulted in, artifacts purely from speech-related movements are not detrimental.

Technical artifacts are artifacts resulting from sources external to the body. The most important technical artifacts are:

**Electrode impedance changes:** These can range from slow drift over time (due to electrode gel drying out or, conversely, sweat buildup) to sudden and total (electrode detachment). The former presents as a low frequency change in signal mean and range whereas the latter can result in various noise patterns. Both can be compensated for in digital signal preprocessing to some extent, and this dissertation presents techniques for compensating for them in the context of EMG-to-Speech conversion.

**Environmental interference** : Electromagnetic fields from power lines, radio transmissions and electrical components (e.g. a computer and monitor used for EMG recording) can easily be picked up by the human body acting as an antenna as well as cables connecting electrodes to the amplifier or components of the amplifier itself.

There are various techniques for suppressing artifacts in EMG recording. The most important one for technical artifact removal is differential amplification: In theory, since the electromagnetic field being picked up by the human body is approximately the same for two electrodes on the same muscle, the fact that bipolar EMG is recorded and amplified as a difference of two voltages measured by electrodes in close proximity should be enough to mostly eliminate external electromagnetic influence. In practice, however, there is another difference to consider: Cables and technical components of the amplifier also pick up noise from the environment (even when the amplifier is – as is common safety practice – isolated from mains power), resulting in a potential difference between the ground of the patient and the ground of the amplifier. The basic technique for this is to connect the amplifiers common ground to an electrically neutral area of the patient with an additional electrode. A more advanced technique (which is used by the amplifier that all new recordings in this work were performed with) is *driven right leg* circuitry: A driven right leg circuit measures the potential difference between amplifier and patient ground (via another additional electrode) and feeds it back into the patient with inverted phase (via a fifth electrode), creating an active noise cancellation feedback loop.

## 2.2 Principles of Speech Production

Acoustic speech, on a signal level, is a longitudinal waveform. Humans produce this waveform exhaling a stream of air, which is excited by the vocal chords and then modulated by different obstacles by passing through the

**0** Lower lip
**1** Upper lip
**2** Teeth
**3** Alveolar ridge
**4** Palate
**5** Velum
**6** Uvula
**7** Pharynx
**8** Tip (apex) of the tongue
**9** Blade (lamina) of the tongue)
**10** Back (dorsum) of the tongue
**11** Middle (radix) of the tongue
**12** Glottis (including vocal folds)
**13** Epiglottis
**14** Nasal cavity

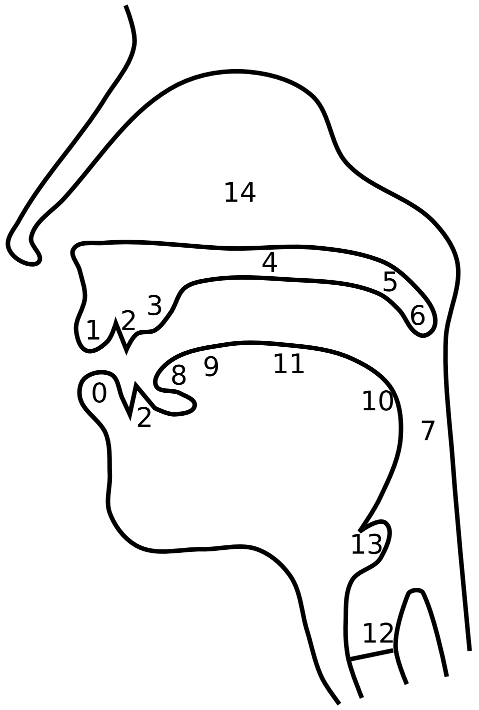


**Figure 2.5** – A cross-section of the human vocal tract, with articulators marked.

cavities and *articulators* of the *vocal tract*. A schematic view of the vocal tract with articulators marked can be seen in Figure 2.5.

When modeling speech, is common to treat these two parts separately. The lungs and vocal chords, as the source, introduce an excitation signal: An oscillation at a given *fundamental frequency* ($F_0$), defined as one opening/closing cycle, if the vocal chords are vibrating, or a white noise signal when they are not. The excitation signal is then passed through and modified by the vocal tract acting as a filter. This is called the source-filter model of speech production [Fan81], further illustrated in Figure 2.6. A weakness of this model is that it treats speech as always either fully unvoiced of fully voiced – in actual speech, mixed excitation (speech that is both to a degree voiced and unvoiced) is possible. Section 2.4.2 will provide further background on the implications of this for speech synthesis.

When considering the speech EMG signal, the part of this model that is of greater interest to us is the vocal tract. This is for three reasons:

***Primary location of articulation:*** While the excitation and fundamental frequency are by no means unimportant, the primary way by which

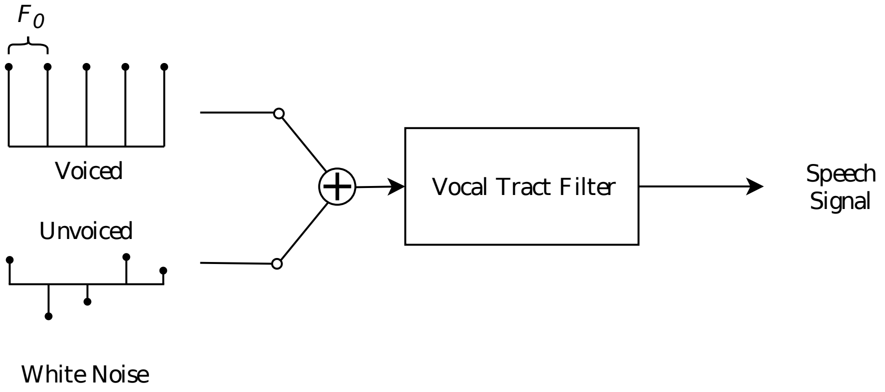


**Figure 2.6** – Illustration of the source-filter model of speech production: A fundamental frequency (voiced speech) or white noise (unvoiced speech) is used to generate an excitation signal, which is modulated by the vocal tract filter to generate the final speech waveform.

information is encoded into the speech signal is by the articulators. A speech signal with uniform white noise excitation – whispered speech – can still be understood, a speech signal with only excitation can not.

***Fully present in silent articulation:*** When articulating silently, the articulators still move, but the vocal chords do not. A speech interface wanting to operate on silent speech thus cannot rely on vocal chord movement.

***Measurable by facial surface EMG:*** Finally, only the muscles controlling the articulation apparatus are easily measurable using facial surface EMG. While breathing and vocal chord tension are, of course, also controlled by muscles, they are further spread through the body and not always measurable without invasive methods, which are not acceptable for user interface use.

### 2.2.1 Classification of speech sounds

In the following, we will describe how sounds are classified according to the system defined by the International Phonetic Association [Int99]. To classify a sound, we have to start with whether it is a sound created by an obstruction that causes turbulence inside the vocal tract, or a sound which does not. The former type, with many articulators involved, is called a *consonant*, the latter,

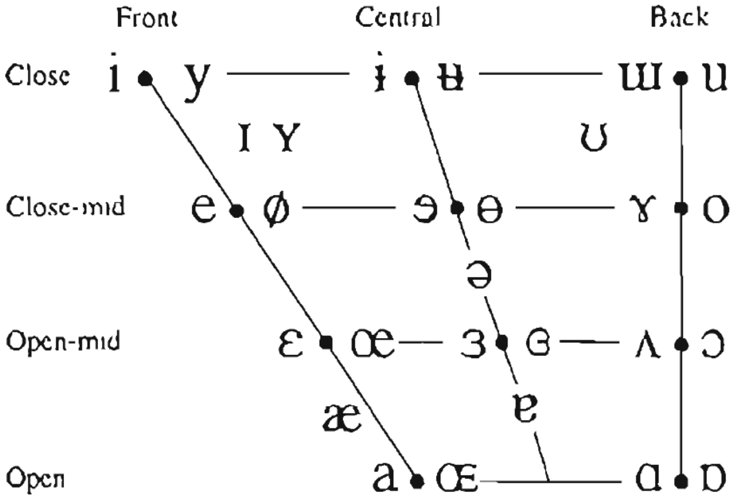


**Figure 2.7** – The International Phonetic Alphabet trapezoid for vowels [Int99]. When two symbols are present the one on the left represents a non-rounded vowel and the one on the right represents a rounded vowel.

with the involvement of only few articulators and no turbulent flow inside the vocal tract, is called a *vowel.* They are further broken down by different attributes.

Vowels are defined by their *frontness*, *height* and *roundedness.* They are always voiced.

***Frontness:*** The vowel frontness relates to the position of the tongue inside the mouth. Vowels where the highest point of the tongue is close to the lips are called *frontal* (e.g. the ee in free) vowels, vowels where the highest point of the tongue is towards the middle of the mouth are called *central* (e.g. the oo in goose) and those where it is towards the back of the mouth are called *back* (e.g. the o in go) vowels.

***Height:*** Similar to the frontness, we can classify vowels by how high up the highest point is inside the mouth: Such vowels are called *close* (high – e.g. again ee in “free” ) or *open* (low – e.g. the ough in thought). Vowels in between these are referred to as *close-mid* (mid-high) and *open-mid* (mid-low) vowels.

***Roundness:*** Finally, vowels can be produced with the lips either *rounded* (e.g. the oo in “goose”) or *unrounded* (e.g. the e in me).

**Table 2.1** – Consonant places of articulation.

| Name | Articulators | Example |
|---|---|---|
| Bilabial | Both lips | The m in "man" |
| Labio-dental | Lower lip / upper teeth | The f in "fan" |
| Linguo-labial | Upper lip / tongue | None in english |
| Dental | Upper teeth / tongue | The th in "this" |
| Alveolar | Alveolar ridge (front) / tongue | The n in "run" |
| Post-alveolar | Alveolar ridge (back) / tongue | The sh in "shin" |
| Retroflex | Palate (front) / tongue | The n in "run" (Indian English [BM14, p. 289]) |
| Palatal | Palate / tongue | The y in "yes" |
| Velar | Velum / tongue | The ng in "ring" |
| Uvular | Uvula / tongue | The c in "caught" (Australian english) |
| Pharyngeal | Pharynx / tongue back | None in English |
| Glottal | Glottis | The h in "hat" |

Figure 2.7 illustrates this classification in the form of the International Phonetic Alphabet vowel trapezoid.

Consonants, on the other hand, are defined by their *place of articulation*, *manner of articulation* and – since they can be voiced or unvoiced to various extents – *degree of phonation*.

***Place of articulation:*** The place of articulation characterizes which articulators the obstruction that primarily characterizes the consonant takes place, called the *place of articulation* [Can05]. Table 2.1 gives a list of the common places of articulation.

***Manner of articulation:*** The manner of articulation tells us in which way the aforementioned articulators are used to produce the consonant sound. Table 2.2 gives an overview of these *manners of articulation* [Can05].

***Degree of phonation:*** Finally, since unlike vowels, they are not always voiced, we can differentiate consonants by their degree of *phonation*: During speaking, air coming from the lungs first passes by the vocal folds. By vibrating, they can add *voicing* to the produced sound (an example would be the v in "van"), which would otherwise be *voiceless* (such as the f in "fan", which is otherwise identical to the aforementioned j).

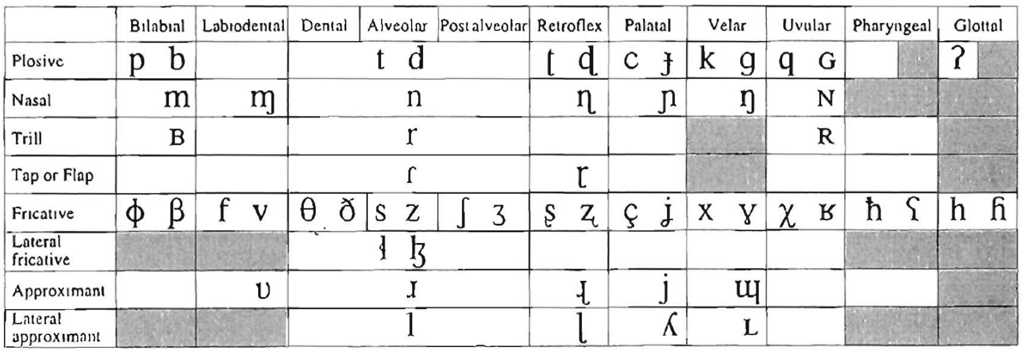
**Table 2.2** – Consonant manners of articulation.

| Name | Description | Example |
|---|---|---|
| Plosive | An occlusive sound, i.e. a sound where the airflow through the vocal tract stops completely before resuming again (also called a “Stop”) | The p in “pass” |
| Nasal | A sound where air flows primarily through the nasal cavity | |
| Fricative | A sound resulting from turbulent airflow at a place of articulation due to partial obstruction | The f in “fricative” |
| Affricate | A stop changing into a fricative after a short time | The j in “jam” |
| Approximant | A sound where there is some, but very little obstruction | The y in “yes” |
| Lateral | An approximant with airflow around the sides of the tongue | The l in “lateral” |
| Flap | A stop too brief to allow for buildup of air pressure | The t in “butter” (US English) |
| Trill | A sound resulting from the repeated opening and closing of the vocal tract | A “rolled r” |

Figure 2.8 shows the IPA table for the consonants described above. Note that, in addition to these *pulmonic* consonants – consonants produced by exhaling air from the lung – there are also *non-pulmonic* consonants which are produced without exhalation. As this work deals only with English language, these non-pulmonic phones will not be given any further consideration.

### 2.2.2 Fundamental Frequency

As voiced phones are generated by vibration of the vocal folds, they have a certain pitch, called the *fundamental frequency* or $F_0$. The variation of the fundamental frequency over the course of speech is called *intonation*. In some languages (e.g. Mandarin Chinese), called “tonal languages”, short-term $F_0$ differences do encode meaning – this variation in $F_0$ is called the *tone*. While the $F_0$ variation, in non-tonal languages, mostly carries *paralinguistic information* such as stress, mood and emotion, it is important to note that



| | Bilabial | Labiodental | Dental | Alveolar | Postalveolar | Retroflex | Palatal | Velar | Uvular | Pharyngeal | Glottal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Plosive | p b | | | t d | | ʈ ɖ | c ɟ | k g | q ɢ | ▨ | ʔ ▨ |
| Nasal | m | ɱ | | n | | ɳ | ɲ | ŋ | ɴ | ▨ | ▨ |
| Trill | ʙ | | | r | | | | ▨ | ʀ | | ▨ |
| Tap or Flap | | | | ɾ | | ɽ | | ▨ | | | ▨ |
| Fricative | ɸ β | f v | θ ð | s z | ʃ ʒ | ʂ ʐ | ç ʝ | x ɣ | χ ʁ | ħ ʕ | h ɦ |
| Lateral fricative | ▨ | ▨ | | ɬ ɮ | | | | | | ▨ | ▨ |
| Approximant | | ʋ | | ɹ | | ɻ | j | ɰ | | | ▨ |
| Lateral approximant | ▨ | ▨ | | l | | ɭ | ʎ | ʟ | | ▨ | ▨ |

**Figure 2.8** – The International Phonetic Alphabet chart for pulmonic consonants [Int99]. When two symbols are present the one on the left represents a unvoiced consonant and the one on the right represents a voiced consonant. Shaded areas denote phones considered to be impossible to produce.

this information can still be important for understanding the meaning of a spoken sentence.

### 2.2.3 Speaking Modes

Speech is normally produced audibly, however, this dissertation also deals with speech produced without an audible acoustic signal. We therefore consider two *speaking modes* in this dissertation:

**Modal Speech:** Modal speech, also called *audible speech* is acoustic audible speech, produced in the way that a healthy individual would produce speech if promoted to speak without any further instructions. Its characteristics have been described in Section 2.2.

**Silent Speech:** Silent speech is speech produced silently, without an accompanying acoustic signal, but with articulator movement – merely mouthing words without vibration of the vocal chords or exhalation without enough force to cause the turbulent airflow required for consonants. While this is ideally the only change between silent speech and modal speech, in practice, individuals prompted to speak silently articulate differently, resulting in a change in EMG signal that Silent Speech Interfaces need to compensate for during silent operation [WJS11].

There are several other speaking modes that are worth mentioning in this context, however, the focus of this dissertation is not on either of them, and they are mentioned here mainly for completeness sake:

**Whispered Speech:** Whispered speech is speech produced completely unvoiced. All voiced sounds are changed to unvoiced sounds, but the speech remains otherwise unchanged [Doe42]. A sub-type of whispered speech is *non-audible murmur* – whispered speech that is produced too quietly to be picked up via transmission in air, but still causes vibrations of the bone and throat that can be recorded using special microphones [OSH08].

**Sub-Vocal Speech:** Sub-vocal speech is speech produced without visible external movement. It occurs naturally during reading [FAE58], and there is also evidence that it can be produced voluntarily [KKM18].

**Imagined Speech:** Imagined speech is imagining the process of speaking, i.e. imagining moving the articulators to produce speech but not actually moving them [DSLD10].

**Inner Speech:** Inner speech is purely mental speech without any movement or intention of movement, real or imagined [Sok12].

### 2.2.4 Muscles of the Articulation Apparatus

Having explained the EMG signal as well as its acquisition and how speech is produced by air exhalation and moving articulators, what remains is the connection between those two – which facial muscles move the articulators, and how they are captured by surface EMG.

#### The Tongue

Of all the muscles related to speech production, the tongue is both the largest and most important one. It rests in and forms the lower surface of the middle part of the vocal tract. Most consonant production involves the tongue as an obstruction in some form (refer back to Table 2.1 for details), and its shape and position is the primary determinant of what a vowel sounds like. For this reason, we would like to capture the electromyogram of the tongue with great detail – however, due to the tongues position relative to the facial surface, it is not easy to capture tongue movement with surface electromyography.

The most promising electrode positions for recording the tongues electrical activity are below the chin, placed far enough towards the back of the head to not be directly over bone. In this work, we use an electrode strip placed in this area for the recording of tongue EMG (see Section 4.1 for more details).

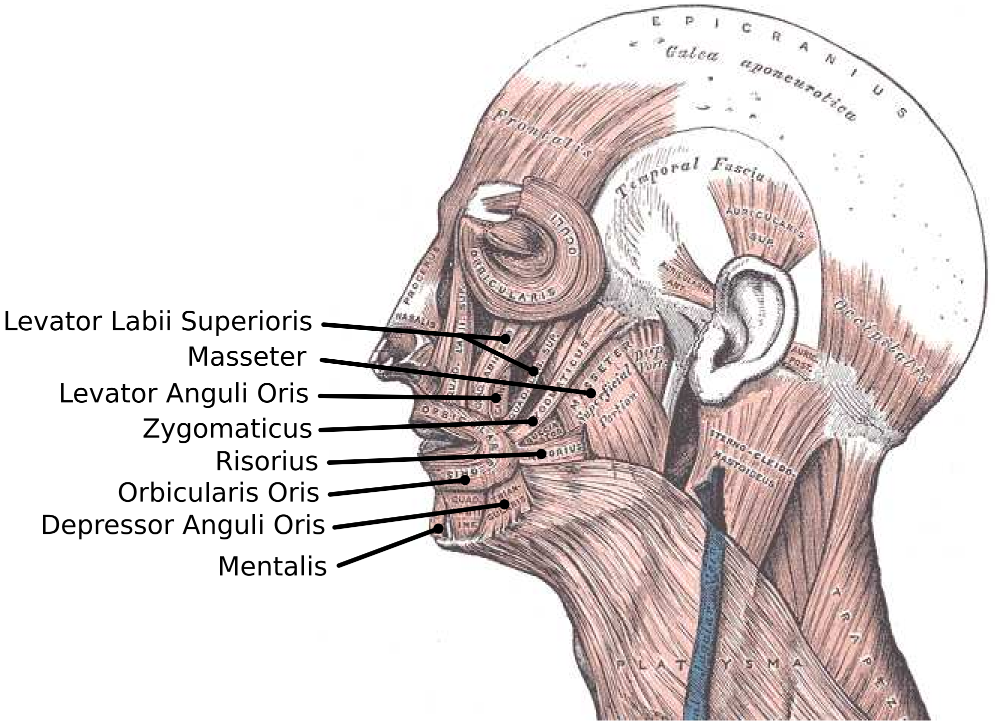


**Figure 2.9** – Anatomical sketch of the facial muscles, with notable speech-related muscles labeled (Adapted from [GL18]).

Another muscle which assists with tongue placement, the digastric, is also located in this area.

### Other facial muscles

Most of the remaining muscles controlling the articulation apparatus are located to the sides of the face. Notable muscles, shown with labels in Figure 2.9, include [UCL02]:

**Depressor/Levator Anguli Oris:** The levator and depressor anguli oris muscles raise (levator) or lower (depressor) the corners of the mouth by pulling them up or down, without pulling them outwards. They are arranged as an antagonistic pair, with fibers running straight up from the corner of the mouth for the levator muscle and straight down for the depressor.

**Levator Labii Superioris:** The levator labii superioris pulls the upper lip straight upwards. It is a sheet of muscle tissue that runs from the side of the nose to the upper end of the upper lip.

**Zygomaticus Major/Minor, Risorius:** The zygomaticus major and minor muscles as well as the musculus risorius act to pull the corners of the mouth outwards and upwards, resulting in a smile-like expression. They run from the corners of the mouth outwards towards the area between the ear and eyes.

**Orbicularis Oris** The musculus orbicularis oris surrounds the mouth, and by contracting, causes the mouth to close up and round (a "puckering" of the lips). Though its action is like that of a sphincter, it is not actually a sphincter muscle but is instead split into four different quadrants, with muscle fibers going from the center of the face towards the corners of the mouth below and above the lips.

**Masseter** The masseter, when innervated, contracts to pull the lower jaw and teeth up, closing the teeth. The musculus masseter is (compared to the rest of the muscles mentioned in this list) rather large and comparatively strong, as its primary function is to enable chewing. It runs from the lower end of the jaw up towards the cheek bones.

**Mentalis** The mentalis muscle pulls the skin of the lower chin and, consequently, the lower lip, upwards and forwards, resulting in a "pouting" expression. It is split into a left and right part (not separately innervated), running from the center of the chin to the lower lip.

Many of these muscles are located on the side of the head in the cheek area and run from the center of the head to towards its back. They can therefore be captured relatively well by electrodes on the cheek, as is done for the data recorded in this dissertation – the exact setup will be described in Section 4.1.

## 2.3 Relation between EMG and Speech Production

Given that the electrical activity of the muscles presented in the previous section can be recorded on the surface, and that these muscles move the articulation apparatus, generating the speech signal, it can be seen that it is possible to infer information about speech from facial EMG.

As an initial example of this relation between the EMG signal and the resulting speech signal, we present an analysis of the EMG signals of the muscles described above during the production of vowels, vowel-consonant-vowel sequences as well as in rest, first presented in the context of our work on facial muscle stimulation [SAD+19].

(1) *depressor anguli oris*
(2) *levator labii superioris*
(3) *zygomaticus major*
(4) *orbicularis oris inferior*
(5) *levator anguli oris*
(6) *masseter*
(7) *mentalis*

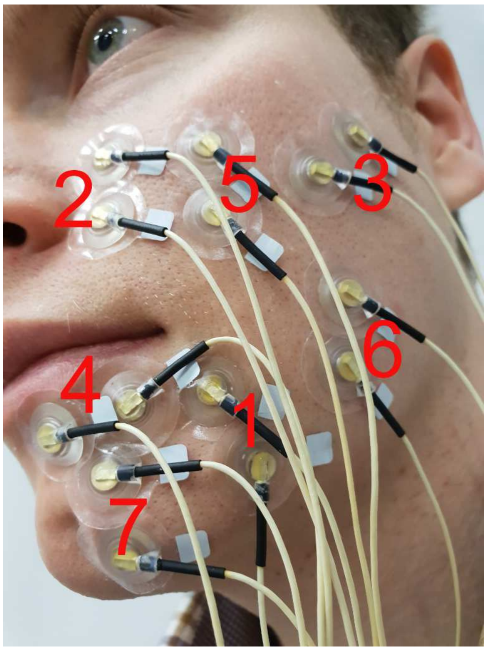


**Figure 2.10** – EMG electrode positioning for facial muscle activity analysis. Electrode pairs labeled to indicate which muscle is being targeted for recording by that electrode pair using bipolar derivation.

The electrode configuration used for this analysis can be seen in Figure 2.10. Here, a Ag/AgCl single electrode montage was used to try to isolate specific facial muscles as much as possible. For electrode placement for specific muscles, we followed recommendations from the Handbook of Psychophysiology [CTB07] and, where no such recommendations were available, estimated positions from known muscle locations from physiology literature [DB06, CP14]. Signals were recorded using bipolar derivation, with a high-pass filter at 10 Hz for DC offset removal and a 500 Hz low-pass filter for anti-aliasing, and digitally sampled at 2048 Hz for analysis. For further details on the hardware used to capture these signals, refer to Section 4.1.

For the following recording, the participant was asked to produce different sequences of speech sounds (given in the IPA phonetic alphabet, see Figure 2.7 and Figure 2.8):

- Vowels – [a], [e]
- Vowel-Consonant-Vowel sequences – [aʋa], [eʋe]
- Silence – muscles completely relaxed, no sound (labeled SIL)

The resulting recordings were analyzed by calculating the frame based power for 32ms windows extracted with 10ms overlap. A box plot of the results can be seen in Figure 2.11. It is immediately apparent that there is a clear difference between the EMG activity producing different speech sounds for many of the muscles measured.

The *depressor anguli oris* shows low activity for silence, but consistently high activity for the produced sounds. This is likely due to its role in assisting in opening the mouth. The *mentalis*, which we would expect to have stronger activation for consonant sounds than plain vowels appears to show similar activation patterns – this may be due to cross-talk of close-by muscles such as the depressor labii inferioris.

*Levator labii superioris* and *levator anguli oris* show higher activity for the consonant-vowel-consonant sequences. This coincides with expectations of what the activity should look like: To produce a [ʋ] sound, the lower lip makes contact with the upper teeth, and thus, the upper lip needs to be raised to allow for air to escape.

Similarly, we see slightly higher activity in the *orbicularis oris inferior* for sequences with consonants: This is due to the mouth rounding motion required to produce the [ʋ].

The *zygomaticus* is known to act together with the levator anguli oris to widen the mouth. However, while the levator anguli oris' activation showed a clear pattern, we barely registered any zygomaticus activity. This is likely because [ʋ] is an approximant rather than a full fricative, so a strong widening of the mouth was not needed and the action of levator labii superioris and levator anguli oris were sufficient.

The *masseter*, finally, shows similar activation for both measurements while producing speech and not producing speech. While this does not match our expectations, it is easily explained: The participant in the recording kept his mouth closed during recording of the silent data instead of letting his jaw hang down slack, which requires some amount of contraction of the masseter, resulting in the observed pattern of activation.

While the *tongue* EMG is not recorded in this study, as the focus was on the investigation of surface muscles, the results also illustrate the importance of

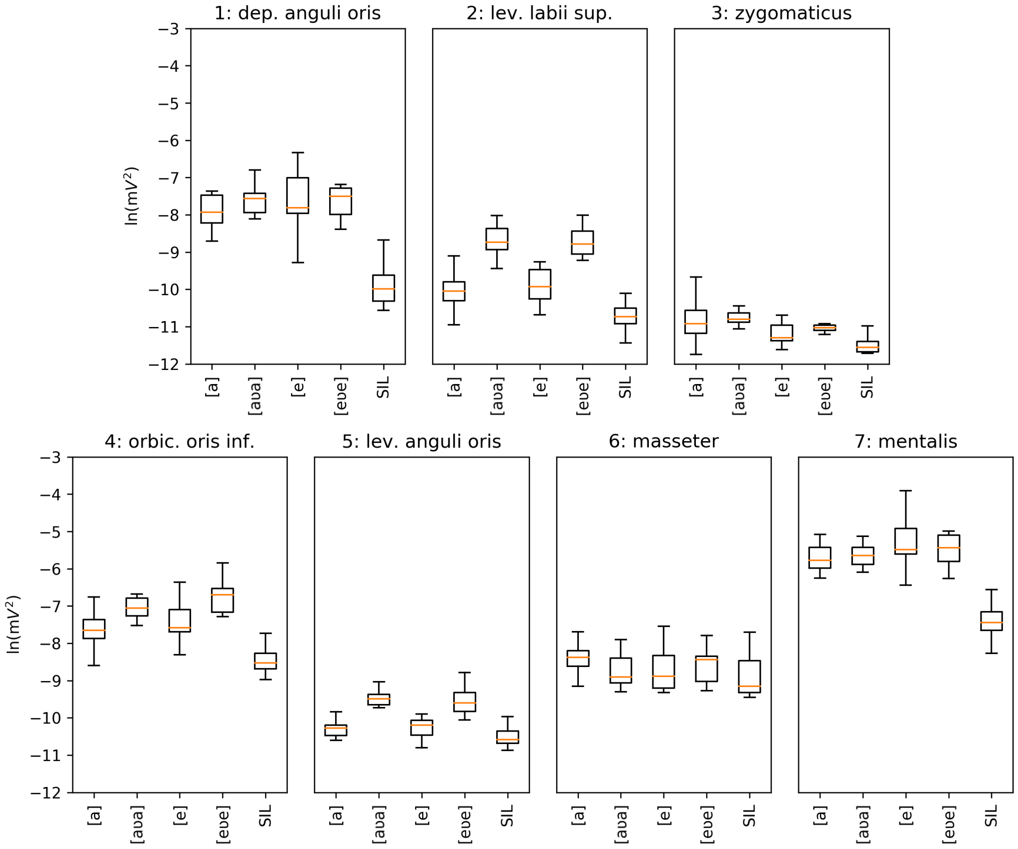


**Figure 2.11** – EMG activities of facial muscles during relaxation, vowel production and vowel-consonant-vowel production. Red bar shows median, boxes show interquartile range, whiskers indicate maximum and minimum values. Y-axis is logarithmic.

this muscle well: The productions of phone sequences with [a] and [e] do not differ significantly. This, too, is expected: While there are minor differences in mouth shape depending on how exactly they are realized, both are unrounded vowels, so the largest difference between them is the position and shape of the tongue in the mouth.

## 2.4 Acoustic Speech and Speech Synthesis

The previous section explained the physiological and technical basis for EMG recording and speech production. This section will provide background on speech synthesis driven by a speech signal (voice conversion) and how it relates to EMG-to-Speech conversion.

### 2.4.1 The Acoustic Speech Signal

The physiological background of acoustic speech production has already been explained in Section 2.2. This section will focus on how to record, encode and process this signal.

#### Recording Acoustic Speech

Audible acoustic speech, as produced by humans, is a longitudinal waveform traveling through air. To process this signal with computer algorithms, it has to be converted into digital form first.

The first step of this process is capturing the signal using a microphone, turning it from a wave in air into an analog voltage signal. A typical microphone of the type used in this work (a so-called condenser microphone) works as follows: The wave hits a diaphragm attached to a plate capacitor, which is kept at a fixed charge level. As the sound waves hit the diaphragm, it deforms. As a result, the distance between the plates of the capacitor changes, resulting in a change in capacitance. Since the capacitors charge is kept constant, and with capacitance being charge divided by voltage, the capacitance change results in a change in voltage, which can then be further processed [BB16, p. 82f].

#### Digitization

The second step in capturing an audio signal for digital processing is digitization. It is composed of two sub-steps, typically achieved simultaneously by an analog-digital converter – sampling and quantization.

The first sub-step is the conversion from a signal that has a value at any point in time to a series of values at given time steps – called *sampling*, illustrated in the middle row of Figure 2.12. A single value thusly obtained is called a sample, and the rate at which samples are created is called the sampling rate. When the sampling rate is strictly more than twice the highest frequency component that the signal contains, then this process is lossless and the continuous signal can be perfectly reconstructed from the sampled signal [Nyq28]. This is called

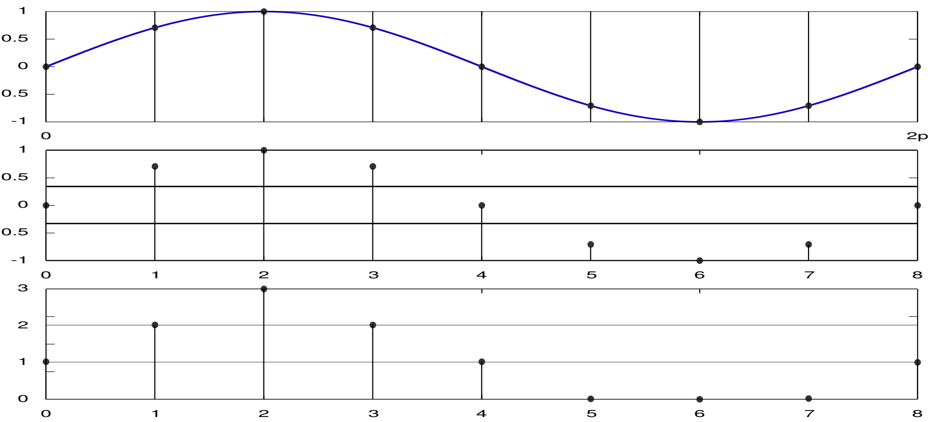


**Figure 2.12** – A continuous analog signal (top) is first sampled at a rate of $4\pi$Hz (middle) and then quantized with $2^2$ steps (bottom).

the Nyquist-Shannon sampling theorem, and the frequency that is half the sampling rate is called the Nyquist frequency. Frequency components above the Nyquist frequency cause spectral components of the sampled signal to end up mirrored at the Nyquist frequency looking like other frequencies when the signal is reconstructed. This effect is called *aliasing*. To avoid aliasing, it is necessary to remove frequency components above the Nyquist frequency using an analog filter. For an acoustic speech signal, relevant information is concentrated in the range below 8000 Hz. We therefore sample the signal at 16000 Hz.

The second sub-step is the division of each analog value into different discrete steps – called *quantization* [BB16, p. 82f]. An illustration of quantization can be seen in the bottom row of Figure 2.12. Unlike sampling, quantization is not lossless: There will generally be a small rounding error between the actual analog value and the resulting assigned digital value, called *quantization noise*. To keep this noise small, it is important to choose a sufficient number of sampling steps. This number is usually given in bits (where quantization with $n$ bits means $2^n$ steps) and called the *bit depth*. What bit depth is appropriate depends on the application – for audio speech processing, 16 bit is considered sufficient, and this is the value all data discussed in this dissertation uses.

### 2.4.2 Acoustic Speech Representation and Vocoding

The previous section has explained how to record the audio signal as a digital stream of quantized sample values called *pulse code modulation* (PCM) audio. This representation is very information-rich and can be played back easily.

However, it is not ideal for use in EMG-to-Speech conversion: It contains a great amount of information beyond speech that we do not require, and has a very high sample rate. We therefore need to convert it into a representation more suitable for further processing. The following sections introduce two such representations, both defined by a vocoder: Mel-Log Spectrum Approximation features and LPCNet features.

### Mel-Log Spectrum Approximation

The *Mel-Log Spectrum approximation* (MLSA) filter is a classic vocoding technique that makes use of the decomposition of the speech signal into an excitation source and vocal tract filter [Ima83] (see Figure 2.6). In the analysis step for MLSA, two types of features are extracted: A special type of invertible *Mel-Frequency Cepstral Coefficients* (MFCCs), and a fundamental frequency ($F_0$) values.

**MFCCs:** The calculation of MFCCs starts with the extraction of windows to calculate the MFCC features on – in this work, we use 512-sample Blackman windows and a 10 ms frame shift, a value that has proven to work well in previous work with EMG-based speech processing [JMHSW06]. For each window, a cepstrogram is calculated by calculating the Fourier transform, applying a logarithm and finally the inverse Fourier transform. The next step differs from classical MFCCs, as it needs to be invertible for the MLSA filter to work: Instead of a Mel filterbank, the cepstrogram is transformed to the Mel scale using a frequency warping approach, with the frequency warp being calculated using an algorithm introduced by Tokuda et.al. [TKI94]. Given an input cepstrum $a^{(t)}[f]$ for cepstrogram frame $t$, with quefrencies $f = 0...F$ and starting with a zero vector for the initial output warped cepstrogram $\tilde{a}_0[m] = 0$, to get a warped cepstrogram with $M$ coefficients, we evaluate iteratively for $f = F...0$ and $\alpha = 0.42$:

$$\tilde{a}_i[m] = \begin{cases} a[f] + \alpha * \tilde{a}_{i-1}[0] & m = 0 \\ (1 - \alpha^2) * \tilde{a}_{i-1}[0] + \alpha * \tilde{a}_{i-1}[1] & m = 1 \\ \tilde{a}_{i-1}[m-1] + \alpha * (\tilde{a}_{i-1}[m] - \tilde{a}_i[m-1]) & m = 2...M \end{cases} \tag{2.1}$$

The output warped cepstrogram frame is then $\tilde{c}^{(t)} = a_0$. This calculation is performed separately for each frame. It can be further improved by iteratively minimizing the error that was introduced in the approximation using Newton-Rhapson gradient descent [FT92].

$F_0$**:** The MLSA filter requires an excitation signal. We therefore need to extract a $F_0$ value with which we can then generate an excitation that the MLSA filter can be applied to. In this work, we use the Yin algorithm [DCK02], which is based on signal autocorrelation. First, the signal (windowed in the same way as it was for calculating MFCCs) is cross-correlated with itself, and a cumulative mean normalized difference is calculated from this autocorrelation and the original windowed signal. Then, the $F_0$ is extracted as the highest peak within a given pitch range (sensible values for human speech are between 85 Hz and 300 Hz) in this function. Finally, when the value of the peak is below a given threshold (i.e. the strongest detected periodic component is less strong than that threshold), a $F_0$ of 0, meaning no voicing (this is called the "discontinuous $F_0$), is the result.

To perform MLSA synthesis from MFCCs and a $F_0$s, the first step is to generate the excitation from the $F_0$ values. This is done by generating a waveform containing either excitation pulses with the given frequency, or white noise when the discontinuous $F_0$ is 0, of the desired output length. This excitation waveform is then filtered by the MLSA filter, given by the digital transfer function:

$$D(z) = exp(F(z)) = exp\left(\sum_{m=0}^{M} \tilde{c}^{(t)}[m] * z^{-m}\right) \tag{2.2}$$

Where $c^{(t)}$ are the coefficients for frame $t$ calculated according to 2.1. For efficiency, Equation 2.2 is usually calculated using an approximation of the exponential transfer function by a linear combination of four digital filters implementing the recursive sum $F(z)$ (this is called Padé approximation) [Ima83].

### LPCNet

Classical vocoders perform analysis and synthesis based on filter theory and make strong and rigid assumptions about the nature of the speech signal to extract features that generalize well. *Neural vocoders* are a more recent technique, where many of the assumptions are replaced by statistical models implemented using neural networks. Neural vocoders are, in effect, autoencoders for PCM speech waveforms.

A problem of many available neural vocoders, such as WaveNet [vdODZ+16] or WaveRNN [KES+18], is that inference is very slow (even if a highly efficient GPU implementation is available) or requires that the audio representation for an entire utterance is already known ahead of time. They are therefore not an option for low-latency systems. This section briefly describes LPCNet [VS19a, VS19b], a neural vocoder capable of low-latency real-time operation even on a CPU, which is ideally suited for use in a real-time EMG-to-Speech conversion system. For a brief introduction to neural network terminology in general, please refer to Section 2.5.1.

LPCNet is a neural vocoder that converts frames of 20 audio parameters, extracted with a length of 20 ms and a shift of 10 ms, directly into 16 bit PCM waveforms. The audio representation used by LPCNet is again split into excitation and filter parameters, in this case, the pitch period, pitch correlation and 18 Bark-scale [Zwi61] cepstral coefficients. Pitch period is estimated using a method based on normalized autocorrelation [VSJV13] and a transition penalty to avoid sudden jumps, optimized using dynamic programming over a four frame window. The pitch correlation can then be calculated from the obtained pitch period. The Bark scale cepstral coefficients are calculated by first taking the logarithm of the spectrum for the input samples of one frame multiplied by a Vorbis window [Mon04]. This log spectrum is then divided into 18 Bark-spaced bands and a DCT is applied to obtain the *Bark-Scale Cepstral Coefficients* (BFCCs).

The architecture of LPCNet is comprised of two sub-networks, one operating at the rate of the input data (the “frame rate network”) and one operating at the rate of the output data (the “sample rate network”). An overview of the entire network architecture can be seen in Figure 2.13.

The frame rate network is a neural network made of feed-forward (labeled “FC” in Figure 2.13) and convolutional (labeled “conv”) layers. It takes as its input the full set of 20 LPCNet features with one frame of context into the future as well as the past and calculates a set of conditioning parameters for the sample rate network that are then held constant for the duration of one input frame.

The sample rate network is a recurrent neural network. It takes several inputs:

- The conditioning parameters from the frame rate network.
- The sample value from the previous time step.
- A set of predictions for the next sample values, minus the excitation. These are calculated via linear prediction with the previous time steps sample values, using linear prediction coefficients calculated from the BFCCs by first converting them back into a full power spectrum, then calculating the signal autocorrelation via the inverse Fourier transform

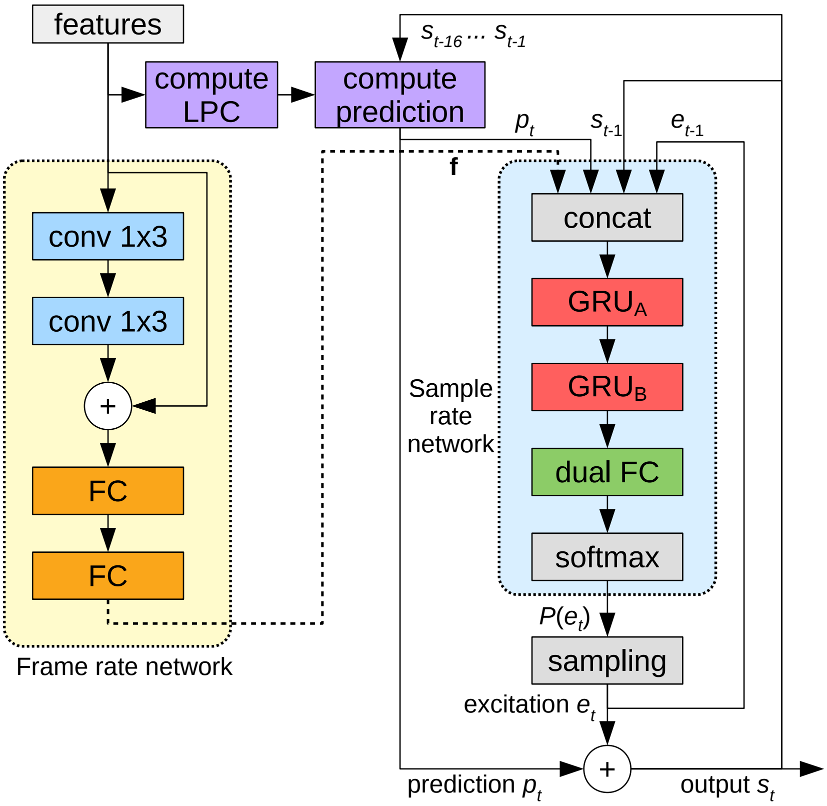


**Figure 2.13** – Structure of the LPCNet neural vocoder (reproduced from [VS19a] with permission).

and then finding the best linear least squares predictor corresponding to this autocorrelation.

- The output of the sample rate network from the previous time step.

The sample rate network consists of two layers of gated recurrent units (GRU in Figure 2.13), followed by a fully connected layer split into two independent halves and finally a softmax output layer. The network is trained to output the excitation signal, which is the residual of the linear prediction output. Sample and excitation values are encoded as 8 bit µ-law values, leading to an output dimensionality of 256. Finally, the output from the sample rate network is added to the linear prediction output, resulting in the final sample value output.

# 2.5 Algorithmic Background

This section will explain the algorithmic basics of two important tools used in this dissertation: Neural networks (used for most of the EMG feature to Audio feature conversion) and time alignment of two sequences using dynamic time warping.

## 2.5.1 Neural Networks

Artificial *neural networks* are a type of statistical model characterized by being connectionist, that is, made up of many small computational units which feed into each other to make up a larger model [GBC16, p. 164ff]. This dissertation employs different neural networks for several different tasks: For vocoding (in the case of neural vocoders) and as the primary means for EMG-to-Speech feature transformation.

Neural networks are generally categorized by the structure of their connections. The following sections will briefly introduce the *artificial neuron* as the basic building block of neural networks and then present three types of neural networks (basic *feed-forward deep neural networks* (DNNs), *convolutional neural networks* (CNNs) and *recurrent neural networks* (RNNs) that are used in this dissertation.

### The Artificial Neuron

An artificial neuron is defined by the following mathematical operation:

$$o(\vec{x}) = h\left(\left(\sum_{i=0}^{m} x_i * w_i\right) + b\right) \tag{2.3}$$

Here, $o$ is the output of the neuron and $\vec{x}$ is the $m$ input values being fed into its inputs (which can be vector valued). $\vec{w}$ and $b$ are the trainable parameters of the neuron, called the *weight* and the *bias*, which have to be determined by statistical estimation methods. Since the bias can easily be implicitly modeled by adding an additional input that is always 1 and the bias as an additional weight, it is often omitted from descriptions of neural networks, and we will also do so from this point on in this dissertation. $h(x)$ is the neurons *activation function*, typically non-linear to allow neural networks to learn non-linear relationships.

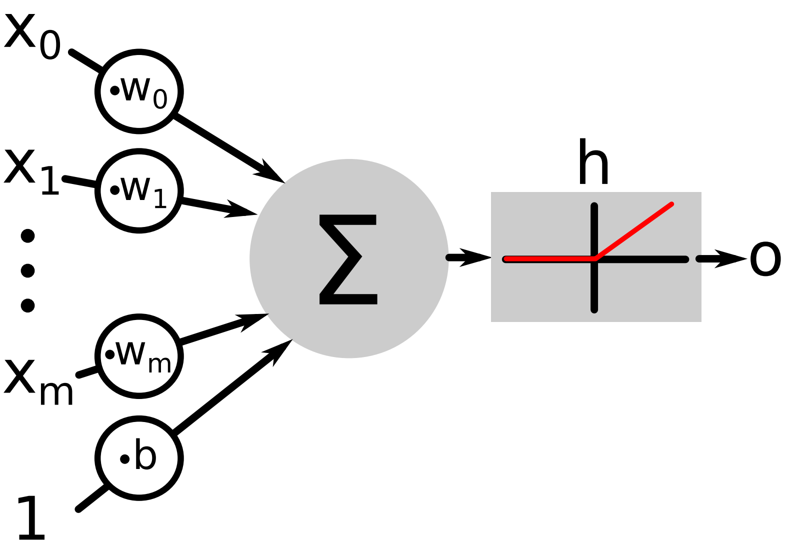


**Figure 2.14** – A single artificial neuron: The inputs $\vec{x}$ are multiplied with the weight vector $\vec{w}$ (with 1 added to the input vector and $b$ added to the weights to implicitly model the bias), then summed and passed through the activation function $h$ to generate the output.

The artificial neuron is easiest to understand graphically – Figure 2.14 provides a graphical explanation of a single artificial neuron, with the same labels as used in Equation 2.3.

To determine suitable parameters for the weights, we use gradient descent to optimize the prediction error of the neuron. For a training sample $\vec{x}$ for which we expect reference output $y$, and with a *loss function* $l(a, b)$ that, given two values returns a real value that indicates the error between those two values, we can calculate the error of the prediction as:

$$\epsilon = l(o(x), y) \tag{2.4}$$

We now want to adjust the weights of the neuron in a way that decreases $\epsilon$. To do this, we calculate the partial derivatives of $\epsilon$ for $\vec{w}$, using the chain rule:

$$
\begin{aligned}
\frac{\partial \epsilon}{\partial \vec{w}} &= \frac{\partial l(\vec{x}, y)}{\partial \vec{w}} \\
&= \frac{\partial l(o(\vec{x}), y)}{\partial \vec{w}} \\
&= l'(o(\vec{x}), y)\frac{\partial o(\vec{x})}{\partial \vec{w}} \\
&= l'(o(\vec{x}), y)\frac{\partial h\left(\sum_{i=0}^{m} x_i * w_i\right)}{\partial \vec{w}} \\
&= l'(o(\vec{x}), y)h'\left(\sum_{i=0}^{m} x_i w_i\right)\frac{\partial \sum_{i=0}^{m} x_i * w_i}{\partial \vec{w}} \\
&= l'(o(\vec{x}), y)h'\left(\sum_{i=0}^{m} x_i w_i\right)\vec{x}
\end{aligned} \tag{2.5}
$$

If the derivative of the loss and activation functions is known, we can now use the result of Equation 2.3 to iteratively update the weights of the neuron in a way that reduces the error, with a learning rate $r$:

$$
\begin{aligned}
\nabla\epsilon &= \left(\frac{\partial \epsilon}{\partial w_0}, ..., \frac{\partial \epsilon}{\partial w_m}\right) \\
\vec{w}_{\text{new}} &= \vec{w} - r\nabla\epsilon
\end{aligned} \tag{2.6}
$$

This process can be repeated (starting from random weights) until the training has converged to a satisfactory degree. Note that, while the training is guaranteed to converge (since the error will get smaller with every step), there is no guarantee that it will converge to a global minimum, or that the weights obtained in this way are weights for a statistical model that generalizes well. We will discuss ways to address this issue later in this section.

The choice of activation function is an important hyperparameter in neural networks. There are two requirements that an activation function must satisfy in practice:

**Nonlinearity:** The activation function must be nonlinear. If it was a linear function, any combination of neurons could only be used to represent linear functions. Once a nonlinearity is introduced, it becomes possible to connect neurons to represent more complex relationships.

**Differentiability:** To infer the weights for a neuron, we need to be able to calculate the partial derivatives of the loss with regard to the weights. This requires that the activation function has a well-defined derivative.

A popular choice for the activation function is the *rectifier* or *rectified linear* (ReLU) function. It is defined as follows:

$$\begin{aligned} h_{\text{ReLU}}(x) &= \max(0, x) \\ h'_{\text{ReLU}}(x) &= \begin{cases} 1 & \text{if x } > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.7}$$

This activation function has the benefit of being very efficient to compute, making it an attractive choice for large neural networks which have to evaluate this function potentially millions of times for training and inference. While it is not differentiable in 0, we can define the derivative at 0 to be 0 for practical use.

Another common activation is the logistic sigmoid (often called just the sigmoid function):

$$\begin{aligned} h_{\text{sigmoid}}(x) &= \frac{1}{1+e^{-x}} \\ h'_{\text{sigmoid}}(x) &= h_{\text{sigmoid}}(x) * (1 - h_{\text{sigmoid}}(x)) \end{aligned} \tag{2.8}$$

While the sigmoid and its derivative are not as efficient to compute as the ReLU function, it has the advantage of being bounded between 0 and 1 inclusive – this is a useful property when constructing cells for recurrent neural networks, as we will see later in this section. Finally, another popular choice for activation functions is the hyperbolic tangent ("tanh"), which is a rescaling of the logistic sigmoid to range -1 to 1 inclusive, centered around 0.

### Feed-Forward Neural Networks

We will now consider how to combine many artificial neurons into a powerful statistical model. The basic principle of feed-forward neural networks is that to arrange many neurons into a *layer* of neurons with a certain *layer width.* The complete network consists of a number of such layers, and the neurons from each layer feed into the neurons of the next one, i.e. the outputs of the

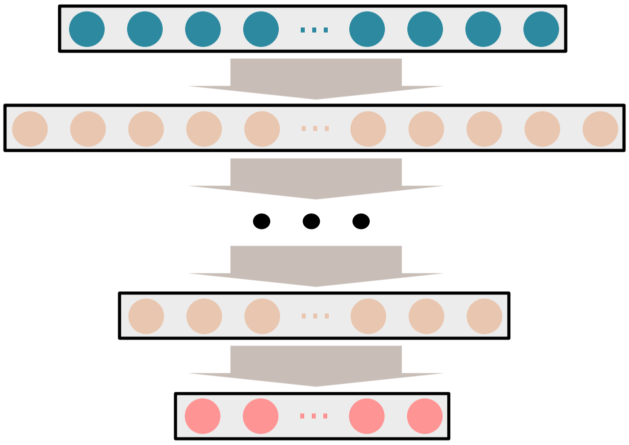


**Figure 2.15** – A feed-forward deep neural network: Neurons are arranged into layers, and all the outputs from the neurons of one layer are fed into each neuron of the following layer as inputs.

neurons in layer $j$ become the inputs of the neurons in layer $j+1$. The number of layers is called the *depth* of the network, and a feed-forward neural network with many such layers is called a *deep neural network* (DNN). Figure 2.15 illustrates the concept of a feed-forward DNN.

To calculate the weights for every neuron in this DNN, we simply apply the rules from Equations 2.5 and 2.6. For neurons in the output layer, they apply directly and without any change. For neurons in layers before the output layer (so-called *hidden layers*), we can write the partial derivative of $\epsilon$ in terms of the partial derivative of the error of the following layer. Let $w^{(j)}$ be the weights for a neuron in layer $j$ and $o^{(j)}(\vec{x})$ the function that computes that neurons output for input $\vec{x}$. Then:

$$\begin{aligned}
\frac{\partial \epsilon}{\partial \vec{w}^{(j)}} &= \frac{\partial l(o^{(j)}(o^{(j+1)}(\vec{x})), y)}{\partial \vec{w}^{(j)}} \\
&= \frac{\partial \epsilon}{\partial o^{(j)}(o^{(j+1)}(\vec{x}))} h' \left( \sum_{i=0}^{m} x_i * w_i^{(j)} \right) \vec{x} \\
&= w^j h'^{(j+1)} l'^{(j+1)} h' \left( \sum_{i=0}^{m} x_i * w_i^{(j)} \right) \vec{x}
\end{aligned} \tag{2.9}$$

where $h'^{(j+1)}$ and $l'^{(j+1)}$ are the derivatives of the activation and loss functions already evaluated for the following layer. Using this method, we can iteratively compute the partial derivatives of the error required to update the weights for each layer according to Equation 2.6, going backwards starting from the output layer. This is called *backpropagation*.

For training to be stable, the weight update has to be performed for the entire set of training data at the same time, however, this can cause very slow convergence and may not always be possible in practice when the amount of training data is larger than the amount of available memory. For this reason, the training data is usually split into small batches of a fixed size, so called $mini-batches$, and processed one mini-batch at a time, with a weight update after every mini-batch. This is called *Stochastic Gradient Descent* (SGD).

Usually, instead of using only the learning rate to control the convergence speed, a momentum term is added to the update: Instead of just updating the weights as described above, the weights are updated using an exponential moving average of all weight updates, usually with a decay factor of 0.9. The *Adaptive Momentum* (Adam) [KB14] optimizer combines this with adaptive per parameter learning rates: It keeps an exponential moving average of the mean (with a decay of 0.9, as before) and non-centered variance (with a decay of 0.999) of weight updates and then perform the weight update using the exponential moving average of the mean scaled by the learning rate divided by the square root of the exponential moving average of the non-centered variance. This allows training to progress more strongly in directions where larger changes are occurring, which helps guide optimization down "ridges" in the surface of the loss function.

### Convolutional Neural Networks

*Convolutional Neural Networks* (CNNs) [LB$^+$95] are a type of feed-forward neural network that includes special layers that, unlike the layers described in the previous section, are not fully connected (i.e. do not have every neuron from one layer connecting to every neuron from the next layer) but instead have neurons with *local connectivity* and *shared weights*.

**Local connectivity:** Every neuron only receives input from a spatially local slice of neurons from the previous layer.

**Shared weights:** Weights for connections with the same offset between input and output neuron are shared between all neurons, creating a so called *filter* or *kernel*.

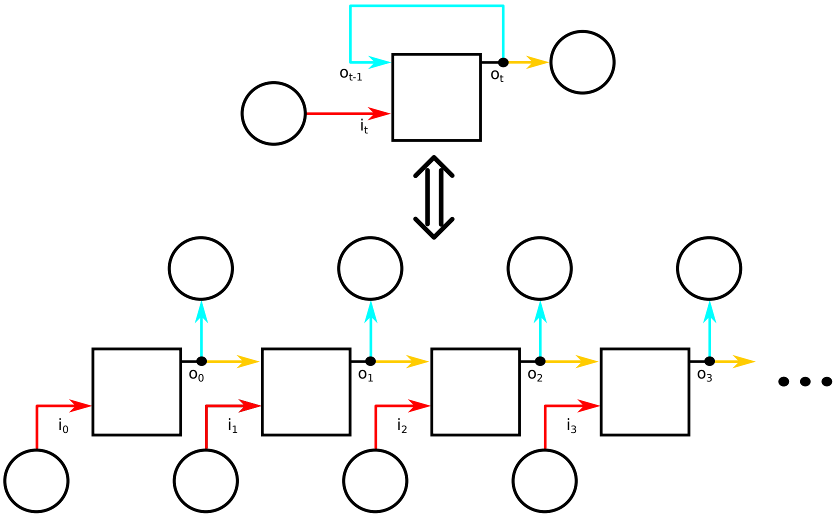


**Figure 2.16** – Back-propagation through time: A recurrent connection is turned into a feed-forward connection by unrolling the network. Colours indicate shared weights.

This, in effect, makes the layer perform an operation similar to the mathematical discrete convolution operation (hence “convolutional layer”). This convolution operation can also be performed with different step sizes (called “strides”), and there can be many different filter kernels of a given kernel sizes feeding into the next layer as different elements of the output vector. Convolutional layers with a stride larger than one reduce the sizes of the dimensions over which the kernels are applied (this is called “downsampling”, whereas fractional strides increase the dimensions size (this is called “upsampling” or “transposed convolution”).

### Recurrent Neural Networks

DNNs and CNNs only allow feed-forward connections – connections that go from a neuron closer to the input to a neuron farther from the input. This is not the case for Recurrent Neural Networks, where connections within the same layer or even backwards connections are allowed. Instead of taking single values as their input, RNNs take a series of values as input and process the values in this series in order, with backwards connections using the value from the previous step. This effectively allows the network to keep internal state and model series relationships directly.

This raises the question of how to train such a network – since there are now feedback loops, how can we back-propagate correctly? The answer is *back-propagation through time* [Moz95]: For training, the network is unrolled so that there is an instance of the network for each time step, with backwards connections becoming forward connections from the network for time step $t$ to the network for time step $t + 1$. Weights are shared between the instances. Having converted the RNN into a network with forward connections only, we can now train it with the standard back-propagation algorithm. The process of back-propagation through time and the sharing of weights is illustrated in Figure 2.16.

The strength of recurrent neural networks is their ability to keep state and directly model series relationships. For this reason, it is common to build cells of neurons – acting externally like neurons themselves – that have access to a memory cell which can be written to and read from. This dissertation uses two types of these units – *Long-Short Term Memory* (LSTM) [HS97] units and *Gated Recurrent Units* (GRUs) [CVMG$^+$14]. Both are similar in construction: They have sigmoid-activated neurons that decide how much weight should be assigned to the current input versus the last time steps output (called “gates”). The difference between the LSTM and the GRU is that the LSTM keeps the cell state separate from the output to the next layer, whereas the GRU does not. This makes the LSTM more powerful, whereas the GRU is more efficient.

### Dropout Regularization

Neural networks, like all machine learning models, must be able to generalize to unseen data: It is not sufficient if the model simply learns to reproduce training examples perfectly, since in practice, the model will have to process data it has never seen before (this is called “overfitting”). This is a problem especially in low data situations, where we may wish to train a neural network model with a large amount of parameters with only very few training samples per parameter, as is the case in EMG-to-Speech conversion. The technique used to avoid *overtraining* models in the work performed for this dissertation is called *dropout* regularization.

Dropout regularization [SHK$^+$14] is conceptually very simple: In each training iteration, a number of neurons of the layer to which we want to apply dropout regularization (usually, half of the units of that layer) is temporarily removed from the model, with their output being replaced by 0, and no weight updates being performed for them.

Dropout regularization is in effect an implicit form of *bagging*: It has the same effect as training many small sub-networks separately and then building a final model that is the average of the sub-networks. It increases the sparsity of neuron activations, meaning that neurons process the input from fewer of the previous layers neurons. This is desirable since it reduces the correlation between the activation of different neurons, which improves generalization ability.

#### Batch Normalization

Batch normalization [IS15] is another regularization and training acceleration technique used in the training of neural networks. It addresses the problem that, as a layers weights are trained, the output distribution of that layer (and therefore the input distribution of the following layer) changes, which slows convergence during training. Batch normalization works by normalizing the inputs of a layer to have zero mean and unit variance on a per-mini-batch basis during training. For inference, the population mean and variance are used instead. This both enables higher learning rates (by making the training procedure more resilient to exploding and or vanishing gradients) and regularizes the model by introducing random variation to layer inputs.

### 2.5.2 DTW Time Alignment

When working with silent speech data, one problem we are faced with is that an audible acoustic reference signal is simply not available. This means that for silent operation systems, we cannot easily calculate similarity scores to a reference, since none exists. A possible solution to this problem is to record an audible acoustic reference signal separately and then align it with the output of the system we wish to evaluate. A common technique for aligning two signals in time, which we use in this dissertation, is *Dynamic Time Warping* (DTW) [Ita75].

The DTW algorithm takes as its input two sequences of elements $A = (a_0, a_1, ..., a_m)$ and $B = (b_0, b_1, ..., b_n)$ and a function $d(a \in A, b \in B)$ that computes a distance metric between those elements. It begins by constructing a *distance matrix* $M$ so that $M_{i,j} = d(a_i, b_j)$. This matrix can be efficiently computed iteratively by re-using distance values already computed in the previous step (in literature, this is often called “dynamic programming”). It then finds the path from $M_{1,1}$ to $M_{m,n}$ that has the smallest total distance, under the constraint that in each step, $i$, $j$ or both must increase by exactly 1. This path now defines an alignment between the two sequences, which can be used for aligning the two input sequences to one another. The aligned sequences are now the same length, and we can calculate evaluation metrics between them.

Chapter 3

# Related work

*This chapter gives a brief overview of key works in the fields of voice conversion, biosignal based spoken communication and specifically EMG-based Silent Speech Interfaces to position this dissertation within these fields.*

## 3.1 Voice Conversion

Acoustic *Voice Conversion* (VC) is the conversion of one speakers acoustic speech to another speakers acoustic speech – i.e. given a source speaker A and target speaker B, the problem of VC is how to synthesize audio data that contains the speech and language content from speaker A's utterance, but with speaker B's speaker identity. While it is possible to perform this task using speech recognition followed by resynthesis, the approach most interesting in practice is the direct conversion of one audio signal to another without an intermediate textual representation. This is called Voice Conversion. EMG-to-Speech conversion is similar in that here, too, we wish to transform one type of speech signal directly to another. Early research into EMG-to-Speech conversion was based heavily on classical voice conversion methods such as unit selection and *Gauss-mixture mapping* (GMM) regression. We begin this section with a brief introduction into the history of these methods, followed by relevant work about the use of these methods in EMG-to-Speech conversion later in this chapter.

Stylianou et al. [SCM98] present a basic voice conversion system based on Gauss-mixture regression. To create a Gauss-mixture regression model, parallel speech feature vectors of the source and target speaker are required. These are obtained by first recording the source and target speaker reading the same text, then splitting the recordings into pitch-synchronous frames, calculating MFCC features for each frame and aligning these feature vector sequences using the DTW algorithm (see Section 2.5.2). A joint Gaussian mixture model is then trained from these parallel feature vectors. This model can be used to convert source speaker MFCCs of a frame to target speaker MFCCs by finding the target speaker MFCCs that, together with the source speaker MFCCs, maximize the overall likelihood of the joint Gauss-mixture model. These MFCCs can then be used to synthesize a speech waveform using the MLSA algorithm (see Section 2.4.2 of this Dissertation). Toda et al. [TBT07, TMB12] extend this conversion framework by introducing a technique to maximize not only the likelihood of a single feature vector, but the whole feature vector sequence at a time. Moriguchi et al. [MTS$^+$13] provide similar results for recorded electrolaryngeal speech.

Sundermann et al. [SHB$^+$06] present early work on performing voice conversion using a different technique, based on Unit Selection. Unit Selection is a concatenative approach to synthesizing speech – i.e. it works by concatenating snippets of audio (usually with overlap and smoothing between those snippets) to generate speech output. In commercial text-to-speech synthesis systems, unit selection is a common choice for generating very high quality output from a very large (20+ hours) speech database. The system presented by Sundermann et al. works as follows: A database of speech segments and corresponding MFCC features, called "units", is created from training data recordings of the target speakers speech. These segments each correspond to one frame as described in the previous section. To convert speech from the source speaker, it is first split into frames with the same parameters as were used for database creation. MFCC feature vectors are then calculated for each input frame, and a unit from the database is selected according to a weighted sum of two costs: The target cost, representing the difference between source MFCCs and database unit MFCCs, and the concatenation cost, representing the difference between the previously selected unit MFCCs and the candidate unit MFCCs (i.e. the difference between neighbouring units). The audio segments of the units that minimize the total cost over the entire input speech (calculated using Viterbi search [HB96]) are then concatenated to create the output speech waveform.

Current research into voice conversion is usually based on either feature transformation or end-to-end conversion (i.e. converting one waveform directly into another with one single model) using neural networks. Desai et al. [DRY$^+$09] present initial results for neural network based voice conversion. They use DTW-aligned fixed duration parallel source/target frames to train a four layer feed forward neural network. This network is capable of mapping source speaker MFCCs to target speaker MFCCs with improved performance compared to the then state of the art GMM-based VC methods. More recent work, such as AutoVC by Qian et al. [QZC$^+$19] focuses on building multi-speaker models that are able to realize the conversion of arbitrary input speech to a target speaker with only very little target speaker data ("zero-shot" voice conversion).

## 3.2 Biosignal-Based Speech Communication

In Chapter 1 of this dissertation, we introduced EMG-to-Speech conversion as a type of Silent Speech Interface. This section will introduce the wider field of biosignal based speech communication and introduce some modalities with which Silent Speech Interfaces have been implemented in the past.

One popular sensing technology used for investigating SSIs is ultrasound. Using high frequency (typically above 2 MHz) sound waves, boundaries between tissue can be imaged at large depths. For Silent Speech Interfaces, this allows the imaging of the tongue – ultrasound tongue imaging. This modality has been explored from various different angles. Early work in driving a vocoder from ultrasound data is presented by Hueber et al. [HCD$^+$08]. They present a system that can acquire frontal as well as lateral video of the mouth in tandem with ultrasound imaging of the tongue with an ultrasound transducer below the chin. The authors present evaluations using this system based on an audible speech and ultrasound recording of the CMU Arctic corpus. They use PCA-based feature extraction and a GMM-HMM system to build a monophone recognition system, which is then used to drive unit selection based speech synthesis. While this initial system is unable to consistently produce intelligible speech, it shows the feasibility of ultrasound-based speech synthesis. The authors extend their work in a later publication [HB16], where they build a direct conversion silent speech interface using a hidden Markov model that also takes phonetic information into account. Grósz, Gosztolya et al. [GGT$^+$18, GGT$^+$20] build on these results by using a DNN regressor to estimate F0 and MFCC parameters. They build a session-adaptive ultrasound based synthesis system that, in a

MUSHRA listening test in which listeners were asked to rate naturalness, was able to obtain a score of 22%. Xu et al. [XWG19] also build a neural network ultrasound SSI plus video SSI, employing an encoder-decoder approach with a convolutions plus LSTM architecture. They evaluate their approach on the 2010 silent speech challenge data, on which their approach outperforms all previous approaches.

Another example of an ultrasound-based SSI is real-time tongue visualization [Hue13, FHG$^+$17] for speech therapy after tongue surgery. Hueber et al. [GRHF$^+$20] present initial work in evaluating such a system in a clinical setting – though they do not find a statistically significant benefit.

Magnetic articulography is the recording of magnetic fields emitted by magnets attached to the articulators. There are two variants of this concept: *Electromagnetic articulography* (EMA), in which electromagnetic coils are used, and *Permanent-magnetic articulography*, which uses permanent magnets. An example of driving a speech synthesis system based on EMA data and a DNN feature mapping is presented by Bocquelet et al. [BHG$^+$16], who obtain highly intelligible results with an 8 coil setup. While EMA allows the determination of the exact absolute position of the articulators, and is commonly used in medical applications, PMA only allows for position estimates and is more suitable for SSI applications because it can operate without wires and does not require a large sensor setup [CSM$^+$19]. Early work in building a system based on the latter is presented by Fagan et al. [FEG$^+$08], who present a system using 7 magnets and 6 magnetic sensors attached to wearable glasses. Using data with this setup, they are able to recognize isolated words from a 9 word vocabulary with an accuracy of 97%. González et al. [GCG$^+$16] demonstrate a similar, more streamlined recording system. They present a wearable sensor system using 6 magnets: Two attached to the upper lip, two attached to the lower lip, and two attached to the tongue. The field generated by these magnets is measured by three magnetic sensors (and an additional sensor to compensate for environment noise) with three channels each. The authors record two datasets: A dataset containing only sequences of up to seven English digits, audibly produced by three speakers, and a larger dataset containing sequences of 958 consonants-vowel combinations audibly produced by a single speaker. Using these datasets, they train GMM-based PMA to audible speech conversion systems that can generate intelligible speech output. The authors later extend this work to continuous speech [GG18]. Here, they record phonetically balanced subsets of the CMU Arctic corpus for 6 subjects each. Based on this data, the authors train and evaluate PMA to audible speech conversion models, this time based on neural networks. They perform a transcription based intelligibility test and obtain ~75% word accuracy.

*Non-audible murmur* (NAM) is whispered speech that is spoken too quiet to be perceived by human listeners or, in fact, to be recorded by standard microphones. It can be measured with a throat microphone – a stethoscopic microphone that picks up vibrations directly from the skin. Toda [TS05] et al. present an initial system for the conversion of NAM speech to modal, audible speech. The system is based on GMM voice conversion, and is evaluated for one speaker. They present a transcription intelligibility evaluation based on utterance fragments, and demonstrate a very slight improvement over NAM in terms of word accuracy. With later improvements to the technique and data used [TMB12], they obtain a word accuracy of ~70% for NAM converted to modal speech, and ~76% for NAM converted to whispered speech.

Birkholz et al. [BSW$^+$18] introduce a novel sensing technique for articulation activity recording: Microwave radar imaging of the vocal tract. They propose a two flat foil antenna system, with one antenna attached below the chin and another attached on the cheek. They have each antenna, in turn, emit a 6 millisecond linear electromagnetic frequency sweep between 2 and 12 GHz and measure the complex spectrum of the signal transmitted to the other antenna as well as the signal reflected back to the transmitting antenna. The authors record such spectra for audible sustained productions of 23 different sounds by two subjects. They demonstrate that using these signals, it is possible to build speaker-dependent recognizers that can differentiate the sounds with an accuracy of up to 93%. A different less explored sensing technique is used by Stone et al. [SB20], who explore electro-optic stomatography – the measuring of lip shape, tongue-palate distance and tongue-palate contact patterns using a combination of electrical and optical sensors inserted into the mouth. The authors evaluate this sensing technology in a cross-speaker setting and obtain accuracies of, on average, ~62% on a 10 German digits corpus, and ~56% on a 30 frequent German words corpus.

The production of speech starts inside the human brain. Brain signal recording might therefore seem like an ideal choice for building SSIs. However, both the spatial and temporal resolution of non-invasive techniques is too low to infer anything but whether speech activity is taking place. For this reason, invasive EEG techniques such as *electrocorticography* (ECoG) or stereotactic electroencephalography with deep brain electrodes is used to investigate how information about speech might be decoded from brain activity. Herff et al. [HHdP$^+$15] present early work in this area. They record ECoG data from 7 subjects undergoing epilepsy treatment (who have had the electrodes implanted to perform mapping of seizure loci and eloquent cortex before surgery) during audible speech production. Using this data, they build a brain signal based speech recognition system able to reach error rates between

25% and 50% on a 10 word vocabulary – significantly better than chance level. They also show that the brain regions that provide maximum discriminability for a given time offset in the data correspond to regions associated with speech production, motion planning, and finally, audio perception. In follow-up work, Herff et al. [HJD$^+$16] build a first direct conversion SSI that is able to synthesize audible speech directly from ECoG data using a linear (LASSO) regression approach.

## 3.3 EMG-Based Silent Speech Interfaces

While the previous section focused on modalities other than EMG, this section is focused specifically on introducing prior and related work based on EMG.

### 3.3.1 Recognition Based Silent Speech Interfaces

The first steps towards SSIs based on EMG are presented by Chan et al. [CEHL01]. They present a first study in which a system is trained to discriminate words (ten word vocabulary, digits from “zero” to “nine”) based on 5 electromyographic channels (bipolar derivation, attached to the inside of a face mask). They extract features from the whole word EMG data (a 1024 ms window starting 500 ms before word onset) using a wavelet transform and train session-dependent linear discriminant analysis word classifiers for 2 speakers. Overall, they obtain a classification error of 6.5% on average, clearly demonstrating that it is possible to extract information about speech from facial EMG data. They also present results in reducing the past context available for classification by starting the window later than 500 ms before the word. Here, they demonstrate that the classification accuracy is significantly reduced when less past context is available, demonstrating that the EMG signal preceding audible speech contains important information about the audible speech signal being produced.

Jou et al. [JSW$^+$06] build on this work and extend it from isolated words to continuous speech. They introduce the TD-N features that the features presented in this work are based on, and present a first continuous speech recognition system based on these features. They train a context independent phone based HMM-GMM speech recognizer. Using data from a six channel EMG recording session of a single speaker, they train a system that achieves a word error rate of ~32% on a 108 word vocabulary.

More recent work in EMG-based speech recognition is presented by Proroković et al. [PWSS19]. Compared to the paper presented in the previous paragraph, the presented system has been improved in several ways. First, it trades the Gauss-mixture models for more powerful DNN-based estimators. These estimators do not estimate probabilities for phones, but instead use bundled phonetic features (first introduced in [SW10]), which relate more closely to the way speech is produced than phones do, and are therefore a better match for the EMG signal. The authors produce a session-adaptive EMG-based speech recognizer, which is adapted using model-agnostic meta learning. They use the UKA corpus (see Section 4.2.1) They produce a system that, with 40 utterances for adaptation, achieves a word error rate of ~4.9% on the same 108 word vocabulary as mentioned in the previous paragraph.

While the previous paragraphs introduced systems which are all based on audible EMG data, extending recognition to silent operation is also an important area of research. Kapur et al. [KKM18] present an interesting approach that takes non-audible recording to the extreme. They present an EMG-based isolated word recognition system based on sub-vocal surface EMG recording – i.e. EMG recording with no visible movement of the muscles. They record a data set of 750 productions of isolated digits for 10 users and train a CNN based word recognizer using the unusual approach of performing MFCC extraction on a 7-channel sub-vocal EMG signal. Using these recognizers, they obtain an error rate of on average ~8% on the 10 word vocabulary.

### 3.3.2 Direct-Synthesis Based Silent Speech Interfaces

In Janke et al. [JD17] we summarize works in EMG-to-Speech. Furthermore, we implement and compare EMG-to-Speech systems following four approaches for EMG-to-Speech conversion:

**GMM-based:** The GMM-based system presented is based on GMM-based voice conversion. This EMG-to-Speech conversion approach, which is the earliest such approach, was first introduced by Toth et al. [TWS09]. It differs from voice conversion in that the features used on the input side are EMG TD-15 features instead of audio features. Since GMM systems do not deal with high feature dimensionality well, this approach additionally requires input dimensionality reduction. The presented system implements this using force-phone labels obtained by forced alignment using an acoustic speech recognizer. As output features, the system uses MFCCs and F0s.

**Unit Selection:** The unit selection based system, first presented in [ZJWS14], works much like a unit selection voice conversion system, with EMG TD-15 input features. The system presented improves on a basic unit selection based EMG-to-Speech conversion system by employing *unit clustering* [DJS15a]: To improve both output quality and conversion speed, similar units are grouped using k-means clustering, and the mean EMG and audio signals of all clustered units are computed and used to replace the original units in the codebook.

**LSTM-based:** The LSTM-based system uses recurrent neural networks (bidirectional LSTMs) to convert non-stacked (TD-0) EMG features to MFCCs. This approach contrasts with the other approaches presented in that it does not use explicit context but instead lets the recurrent model learn context dependencies implicitly.

**DNN-based:** Finally, the DNN-based approach, first presented in [DJS15b], uses a 3-hidden-layer deep neural network to convert EMG TD-15 features to MFCCs and F0s. Unlike the GMM model, this model does not require LDA feature reduction.

We compare these systems using the A500+ (see Section 4.2.2) data, using the MCD score as well as a four way preference listening test. They find that, in terms of the MCD score, the DNN system performs best, while the GMM system performs worst. The LSTM and Unit Selection systems performing at about the same level with MCD scores that are approximately in the middle between the GMM and DNN systems. In the preference test, the DNN system once again performs best, followed by the GMM and LSTM systems, while the unit selection approach performs worst.

Chapter 4

# Recording and Corpora

*This chapter describes the recording setups and data corpora that were used to generate the results presented in this dissertation. This includes both previous work as well as corpora that were recorded as part of this dissertation.*

## 4.1 Recording Setups and Devices

As described in the introduction to surface EMG recording, there are many factors that affect the sEMG signal. In this section, we will briefly describe the setups used to obtain data for the different corpora presented later in this chapter and explain their advantages and disadvantages. Mainly, there are two different setups used: A *single-electrode setup* and an *array electrode setup*. All recordings are done in sessions, one session being a continuous recording without removing the electrodes.

### 4.1.1 Single-Electrode Setup

The single-electrode setup, introduced by Maier-hein et al. [WS11], uses a total of 10 electrodes (standard Ag/AgCl cup electrodes, 4mm electrode diameter) placed to record specific muscles. Four of the electrodes are used in a bipolar configuration, while four more are used in unipolar configuration

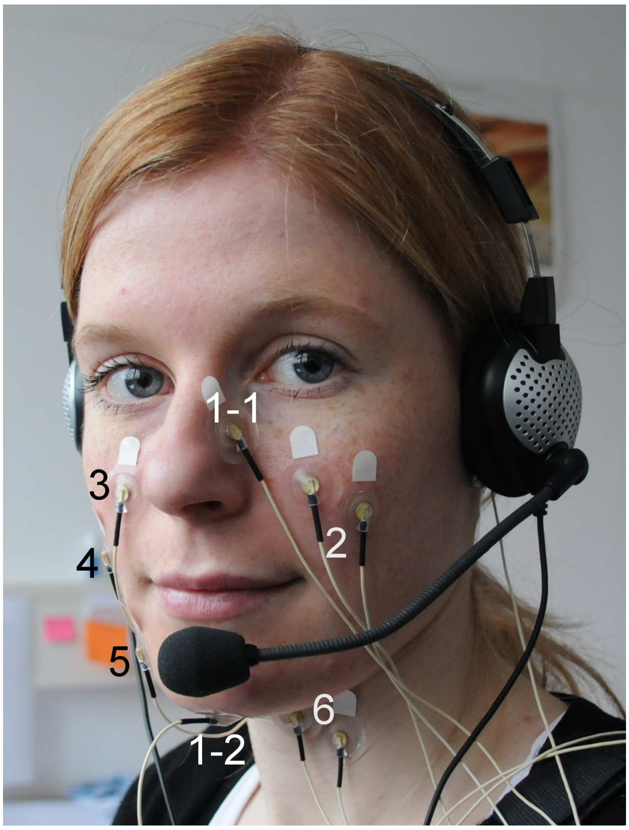


**Figure 4.1** – Single-Electrode electrode montage. Black numbers indicate derivation against a reference electrode placed behind the ear (not numbered).

and measured against two reference electrodes on electrically neutral territory, attached on the nose and behind the ear. The overall electrode placement can be seen in Figure 4.1. In total, this results in six recorded channels:

**Channel 1** is recorded between a reference electrode on the nose (1-1) and an electrode attached below the chin (1-2). It is intended to capture information about the tongue.

**Channel 2** is a bipolar channel, recorded between two electrodes (2) attached on the cheek, on a line going outwards from the nose, angled downwards. It records signals from the levator anguli oris and, to a lesser extent, the zygomaticus muscle.

**Channel 3** is a monopolar channel, derived between an electrode on the cheek (3, positioned similarly to the electrodes for channel 2) and a reference electrode behind the ear (not labeled). It is intended primarily for recording the zygomaticus major and, additionally, the levator anguli oris.

**Channel 4** is another monopolar channel with the (unlabeled) electrode behind the ear as reference electrode. The other electrode (4) is placed next to the corner of the mouth and is intended for capturing signals from the upper end of the platysma.

**Channel 5** uses an electrode (5) that is placed further down and forward compared to channel 4 and is, once again, derived against the (unlabeled) ear reference. It records signals from, again, the platysma, as well as the depressor anguli oris. This channel is usually omitted when building EMG-to-Speech conversion systems, as it tends to yield artifact-prone signals.

**Channel 6** is derived in bipolar configuration, with two electrodes (6) attached on the front part of the neck, just below the head. Like channel 1, it is intended to capture signals from the tongue.

These channels are recorded using a Becker Meditec Varioport biosignal recording system. They are filtered using an analog high-pass filter with a cutoff frequency of 60 Hz and sampled at 600 Hz.

### 4.1.2 Array-Based Setup

The array-based setup uses electrodes which are regularly arranged as part of an electrode grid or strip, with a fixed 10 mm inter-electrode distance. The advantages of such a setup are twofold. Firstly, the electrodes require less time and expertise to attach, since there is only one electrode grid and one electrode strip to attach instead of 10 single-electrode. This makes the setup substantially faster and thus more practical. Secondly, the high density and large number of electrodes allows for the extraction of more detailed spatial information than the single-electrode setup. The downside of the array setup, compared to the single-electrode setup, also relates to electrode count: Since there are more electrodes, spread over a large area, the likelihood of at least one electrode detaching during the recording (due to movement) is much higher.

A schematic of the montage used for recordings presented in this section can be seen in Figure 4.2. It uses a 4x8 grid electrode on the cheek, covering the cheek surface muscles of the speech apparatus. The positioning mimicks the position in which somebody would hold a cell phone, which could in the

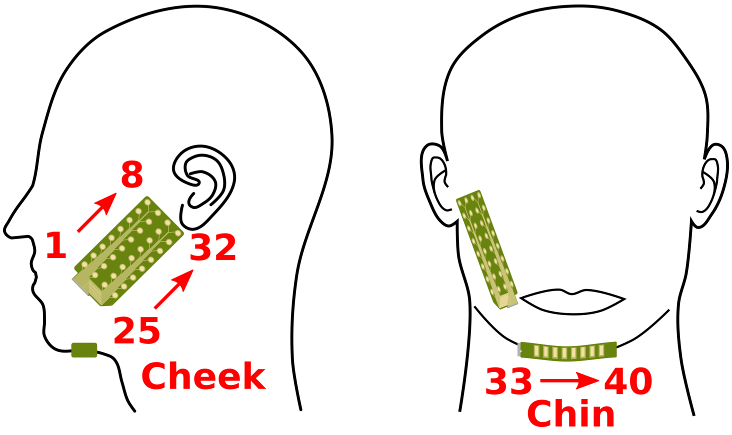


**Figure 4.2** – EMG array electrode positions and numbering for the array-based setup. Derivation is chained-differential, i.e. channel 1 is between electrode 1 and 2, channel 2 is between electrode 2 and 3, etc.

future integrate similar recording technology, and covers important speech apparatus muscles in the cheek. It additionally uses an 8 electrode strip attached below the chin, far enough back so that the electrodes are not placed directly on bone, to capture information from the tongue. The derivation is performed in a chained differential fashion: The first channel is measured between electrode 1 and 2, the second channel between electrode 2 and 3, et cetera. “Border” channels (e.g. between the last electrode of the first column and the first electrode in the second column, are recorded, but generally dropped in pre-processing in our systems. Data recorded for this dissertation was recorded with an OT Bioelletronica Quattrocento multi-channel EMG amplifier.

## 4.2 Existing Data Corpora

To evaluate the performance of EMG-to-Speech conversion and compare different systems, it is necessary to have fixed data sets with parallel EMG and audio data. These may be tailored in several ways to allow for different kinds of evaluations. This section describes available (prior-work) corpora that are used to produce evaluations in this dissertation, but that have not been recorded by the author.

**Table 4.1** – EMG-UKA Corpus: Speaker breakdown. (*) indicates session is part of the trial corpus, numbers in brackets indicate number of sessions / utterances that are part of the trial corpus.

| | #sessions | | | |
|---|---|---|---|---|
| Speaker | Total | Large | Multi-Mode | #utterances |
| 1 | 3 | 0 | 3 | 450 (0) |
| 2 (*) | 33 (3) | 1 (1) | 15 (2) | 3720 (820) |
| 3 (*) | 1 (1) | 0 | 1 (1) | 150 (0) |
| 4 | 2 | 0 | 2 | 300 (150) |
| 5 | 1 | 0 | 1 | 150 (0) |
| 6 (*) | 1 (1) | 0 | 1 (1) | 150 (150) |
| 7 | 2 | 0 | 2 | 300 (0) |
| 8 (*) | 20 (8) | 1 | 7 (2) | 2159 (600) |
| Total | 63 (13) | 2 (1) | 32 (6) | 7379 (1720) |

### 4.2.1 EMG-UKA

This corpus contains acoustic and EMG speech signals recorded in parallel, including a marker channel to compensate for different delays in the signal recording paths. The audio data was recorded at a sampling rate of 16 kHz, with a standard close-talking microphone, whereas the EMG signals were recorded using the single-electrode recording setup. The corpus includes a total of 63 sessions recorded from 8 speakers, featuring 3 different speaking modes (modal speech, silent speech, whispered speech) as part of 32 multi-mode sessions. The speakers were not native English speakers, however, the recording supervisors ensured that English words were pronounced correctly. Each session contains recordings of 50 (or, for the Large sessions, at least 500) utterances of read English speech per mode available in the session. The utterances come from a broadcast news domain. A breakdown of the sessions by type can be found in Table 4.1 and a summary of session durations can be seen in Table 4.2. While a modal speech signal is not available for the silent speaking mode and thus, not recorded, the EMG signal is always available.

### 4.2.2 EMG-ArraySingle-A-500+ Corpus

The EMG-ArraySingle-A500+ Corpus is a corpus recorded with the evaluation of offline EMG-to-Speech conversion in mind. It contains both sessions

**Table 4.2** – EMG-UKA Corpus: Subset Breakdown

| | | | duration ([h:]mm:ss) | |
|---|---|---|---|---|
| Subset | #Spk | #Sess | Average | Total |
| Audible (Small) | 8 | 61 | 03:08 | 3:11:34 |
| Whispered (Small) | 8 | 32 | 03:22 | 1:47:42 |
| Silent (Small) | 8 | 32 | 03:19 | 1:46:20 |
| Audible (Large) | 2 | 2 | 27:02 | 54:04 |
| Whole Corpus | 8 | 63 | | 7:32:00 |

recorded using the array-based as well as sessions recorded using the single-electrode setups described above. Data for the corpus was recorded using an OT Bioelettronica EMG-USB2 multichannel EMG amplifier.

The corpus contains 6 recording sessions total, recorded from three different speakers – 2 male (Spk1 and Spk2) and 1 female (Spk3). All speakers were German native speakers and thus read the sentences with German-accented English. Four of the recording sessions consist of sessions of 500 phonetically balanced English utterances, based on previous work [SW10]. The remaining two sessions additionally incorporate utterances from the Arctic [KB04] and TIMIT [GLF$^+$93] corpora, giving a total of 1103 utterances for the smaller and 1978 utterances for the bigger of these two sessions. The recorded utterances were manually checked for artifacts, and any utterances where such issues were found were removed. All of the utterances in the EMG-ArraySingle-A-500+ corpus were produced audibly.

Each session is split into pre-determined training, development and evaluation sets. Table 4.3 gives a detailed overview of the sessions and how they are split into the sets. There are two sessions using the single-electrode setup (marked "Single") and four using the array-based setup (marked "Array").

In addition to the data from the base EMG-ArraySingle-A500+ corpus we also use additional sessions recorded from speaker 1 for some evaluations. These sessions are listed in Table 4.4.

| | | Length [mm:ss] | | | # of utterances | | |
|---|---|---|---|---|---|---|---|
| Speaker/Session | Gender | Train | Dev | Eval | Train | Dev | Eval |
| S1-Single | m | 24:23 | 02:47 | 01:19 | 450 | 50 | 20 |
| S1-Array | m | 28:01 | 03:00 | 00:47 | 450 | 50 | 10 |
| S1-Array-Lrg | m | 68:56 | 07:41 | 00:48 | 984 | 109 | 10 |
| S2-Single | m | 24:12 | 02:42 | 00:49 | 447 | 49 | 13 |
| S2-Array | m | 22:14 | 02:25 | 01:10 | 450 | 50 | 20 |
| S3-Array-Lrg | f | 110:46 | 11:53 | 00:46 | 1,771 | 196 | 10 |
| **Total** | | 278:32 | 30:28 | 05:39 | 4,552 | 504 | 83 |

**Table 4.3** – Data corpus information for the EMG-ArraySingle-A-500+ corpus, including speaker/session breakdown. Speaker 1 and 2 are male, speaker 3 is female.

| | Length [mm:ss] | | | # of utterances | | |
|---|---|---|---|---|---|---|
| Speaker/Session | Train | Dev | Eval | Train | Dev | Eval |
| S1-Array-2 | 25:19 | 02:42 | 02:43 | 450 | 50 | 40 |
| S1-Array-3 | 26:23 | 02:47 | 02:28 | 450 | 50 | 40 |
| S1-Array-Small-1 | 05:23 | 01:16 | 01:13 | 140 | 30 | 30 |
| S1-Array-Small-2 | 06:39 | 01:22 | 01:20 | 140 | 30 | 30 |

**Table 4.4** – Additional sessions by speaker 1 from the EMG-ArraySingle-A500+ corpus.

## 4.3 Newly Proposed Data Corpora

As part of this dissertation, several new corpora were recorded and used to evaluate different aspects of EMG-to-Speech conversion. This section presents these new corpora and the evaluations performed using them.

### 4.3.1 CSL-EMG-Words-CVVC Corpus

For speech recognition, it is known that the task of recognizing isolated words is very different from the task of large-vocabulary continuous speech recognition due to co-articulation effects and lack of clearly defined word boundaries [JSW+06]. To evaluate how these differences affect EMG-to-

Speech conversion and if we should therefore focus on tasks involving single isolated word productions rather than continuous speech, we recorded a new corpus that contains several different types of utterances: Continuous speech, consonant-vowel and vowel-consonant sequences, isolated words and digits: The CSL-EMG-Words-CVVC corpus. This corpus was first presented at the 13th ITG Conference on Speech Communication [DBS18].

### Corpus Design

For continuous speech, we used a subset of the broadcast news domain sentences from the EMG-ArraySingle-A500+ corpus. Each session in the CSL-EMG-Words-CVVC corpus contains 390 of these sentences: 300 training utterances as well as the complete evaluation (40 sentences) and development (50 sentences) sets, for a total of 390 sentences.

To get a high coverage of consonant-vowel and vowel-consonant sequences (CVs and VCs, respectively) with regards to our continuous speech block, we statistically examine the continuous speech training set utterances. We calculate a frequency distribution of CV and VC sequences and choose the most frequent sequences (within the 90th percentile, rounding up to the nearest 5). This results in 85 CVs and 75 VCs total.

To allow for a more consistent pronunciation of these sequences, we added a context around the combinations (e.g. “T_AK_E” for AK or “_FE_DERAL” for FE) for prompting during recording — note, however, that participants were instructed to read only the CV or VC, not the surrounding context. In a few exceptions the resulting words were infeasible for use in the CSL-EMG-Words-CVVC corpus because of their structure, e.g. words with a dental fricative (”th” sound) or diphthongs like ”EO” or ”OU”, where the CV or VC goes across the boundary of the sound, or where one of the letters in the CV/VC was silent. Examples are T_HI_S, FO_UN_D or PE_OP_LE. In these specific cases, we used the next frequent word without these drawbacks.

As a step between continuous sentences and isolated CV/VCs, we use isolated words. The words we include in the CSL-EMG-Words-CVVC corpus were selected from a set of words used for intelligibility evaluations in telephony [HWHK63]. The original modified rhyme test corpus contains 300 words in total — 150 by variation of initial (phonetic) elements (labeled “IV”) and 150 by variation of final elements (labeled “FV”), in groups of six words. For this corpus, we selected 180 words (30 groups of six), 90 words with initial variation and 90 words with final variation. An example of a group with variations of initial elements is: *LED - SHED - RED - BED - FED - WED* and a group with final variation: *BAT - BAD - BACK - BASS - BAN - BATH*. This specific setup allows for multiple-choice intelligibility testing, with similar or dissimilar words.

Finally, the CSL-EMG-Words-CVVC corpus is complemented by digits from 0 to 9, which can act as a simple reference set that can be recorded in a short amount of time. In total, each session in the CSL-EMG-Words-CVVC corpus contains 740 utterances.

#### Recorded Data

Using the setup and corpus described in this section, we have recorded six sessions of parallel EMG and Audio data from different speakers. Our subjects (Four male, two female) were between ~20 and ~30 years old and are all non-native English speakers, speaking German-accented English. The recording supervisors ensured that English words and CV/VC combinations were pronounced correctly. All of the recorded subjects were healthy and reported never having had any speech disorders. Subjects were thoroughly informed about the recording procedure and experimental evaluations to be done with recorded data and informed consent of all subjects was obtained before recording. In total, we recorded ~4 hours of data. A detailed breakdown into the different parts of the corpus for all recorded speakers can be found in Table 4.5. Recordings were performed in a shielded chamber. Audio signals were recorded using a Behringer Xenyx 302 audio interface and a RODE NT-1 condenser microphone. EMG signals were recorded using an OT Bioelettronica Quattrocento EMG amplifier, using our array-based setup (see Section 4.1.2).

### 4.3.2 CSL-EMG-Speak-Along Corpus

One challenge in building SSIs for silent speech is that, since there is no reference audio data available for silent speech, it is not possible to train systems that operate on this type of speech directly. It is also not possible to use any objective evaluation methods that require an audible reference. A data-based approach to solving this issue is using a *speak-along* recording protocol: Speakers are first recorded while speaking audibly and then once again recorded while silently mouthing along with a played-back recording of their own voice. It is then possible to use the audible audio recording and the

| | | | Words | | | | Sentences | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Speaker | m/f | Digits | IV | FV | CV | VC | Train | Dev | Test | Total |
| Spk1 | f | 00:20 | 02:42 | 02:39 | 02:15 | 01:58 | 25:00 | 04:06 | 03:44 | 42:44 |
| Spk2 | m | 00:20 | 02:42 | 02:46 | 02:14 | 01:57 | 21:19 | 03:26 | 03:09 | 37:53 |
| Spk3 | f | 00:26 | 03:32 | 03:32 | 03:02 | 02:44 | 23:04 | 03:45 | 03:23 | 43:28 |
| Spk4 | m | 00:22 | 03:00 | 03:04 | 02:30 | 02:14 | 19:43 | 03:12 | 02:50 | 36:55 |
| Spk5 | m | 00:21 | 02:45 | 02:48 | 02:30 | 02:14 | 23:29 | 03:48 | 03:27 | 41:22 |
| Spk6 | m | 00:19 | 02:40 | 02:43 | 02:42 | 02:12 | 20:16 | 03:15 | 03:00 | 37:07 |
| Total (hh:mm:ss) | | | | | | | | | | 03:59:29 |

**Table 4.5** – Data corpus breakdown for recorded utterances of the CSL-EMG-Words-CVVC corpus (mm:ss)

| | Audible | | | Speak-Along | | | Silent | |
|---|---|---|---|---|---|---|---|---|
| Speaker | train | dev | eval | train | dev | eval | dev | eval |
| 921 | 21:27 | 4:10 | 4:09 | 21:60 | 4:14 | 4:16 | 4:39 | 4:01 |
| 922 | 20:12 | 3:52 | 3:52 | 20:31 | 3:54 | 3:54 | 4:16 | 3:49 |
| 923 | 18:39 | 3:40 | 3:36 | 20:45 | 3:60 | 3:57 | 3:59 | 3:34 |
| 925 | 22:49 | 4:05 | 4:14 | 24:02 | 4:22 | 4:27 | 3:39 | 3:21 |
| 926 | 19:46 | 3:43 | 3:46 | 21:11 | 4:08 | 4:06 | 4:41 | 4:06 |
| 927 | 22:13 | 4:04 | 4:11 | 23:34 | 4:19 | 4:25 | 4:24 | 3:52 |

**Table 4.6** – Data corpus breakdown for recorded utterances of the CSL-EMG-Speak-Along corpus (mm:ss)

silent EMG recording as a parallel data pair for training and evaluation. If training a system using such speak-along data is feasible, we could build EMG-to-Speech systems not only for people who can presently produce audible speech but also for people who have already lost their voice. For this group, we could then simply have them mouth along with any voice recordings that may be available.

To test our speak-along recording protocol we recorded a corpus of speak-along data, the CSL-EMG-Speak-Along corpus. The recording for each session was performed in two steps: First, we record set of 350 sentences – a subset of the EMG-ArraySingle-A500+ utterance list – 250 training sentences and 50 development and evaluation sentences each, read out audibly. We then record the entire set again using the speak-along protocol. Finally, we perform a non-speak-along silent recording of the development and evaluation sets to be able to compare the cross-mode performance of systems trained on audible as well as speak-along data. We recorded a total of 6 sessions with 6 different speakers. A breakdown of the recorded data can be found in Table 4.6.

Recordings were performed in an acoustically and electromagnetically shielded chamber. Audio signals (both from the microphone and the re-played speak-along audio) were recorded using a Behringer Xenyx 302 audio interface. The microphone used was a RODE NT-1 condenser microphone. EMG signals were recorded using an OT Bioelettronica Quattrocento EMG amplifier, using our array-based setup (see Section 4.1.2). An evaluation of the suitability of our speak-along recording procedure for EMG-to-Speech conversion based on the data presented here will be shown in Section 5.4.3.

### 4.3.3 CSL-EMG_Array corpus

One of the numerous challenges in building an online EMG-to-Speech conversion system, compared to building a system for offline evaluation, is that dealing with time-related signal variations becomes a must even for single session systems. As time passes during a session, the signal will change in several ways due to changes in skin condition (e.g. sweating), muscle condition (e.g. fatigue) and electrode-skin contact. For evaluating methods for use in online systems, it is therefore not valid to simply record a large block of data and randomly split it into training and testing sets. Instead, the time variance has to be taken into account explicitly. The CSL-EMG_Array corpus, recorded as part of this dissertation and first presented at INTERSPEECH 2020 [DRVS20] is a parallel EMG-Audio data corpus that is suitable for evaluating online EMG-to-Speech conversion systems and adaptation methods.

#### Design

The CSL-EMG_Array corpus consists of sessions recorded in a block-wise manner, with a total of 7 blocks recorded in a fixed sequence in numerical order (i.e. first block 1, then block 2, then block 3, etc.) and prompts within a block presented in randomized order (as opposed to previous corpora, which record all utterances in a randomized manner with no time structure, as one single block). This closely mirrors the real online EMG-to-Speech conversion scenario and therefore (unlike corpora where training and testing data do not have any temporal structure) allows for the development and testing of online EMG-to-Speech conversion systems with realistic estimates of online

**Table 4.7** – Amount of sentences for different recording blocks (amounts in parentheses include additional sentences only present for silent testing mode sessions).

| Subset | train | dev | eval |
|---|---|---|---|
| (Block0_Align) | - | (50) | (40) |
| Block1_Initial | 250 | 50 | 40 |
| Block2_Adapt1 | 20 | 20 | 20 |
| Block3_Eval1 | - | 30 | 20 |
| Block4_Adapt2 | 20 | - | - |
| Block5_Eval2 | - | 30 | 20 |
| Block6_Adapt3 | 20 | - | - |
| Block7_Eval3 | - | 30 | 20 |
| Total | 310 | 160 (210) | 120 (160) |

performance. The prompts are English sentences from the broadcast news domain, and are split into training, development and evaluation subsets. Each session contains a total of 590 (680 for silent-testing-mode sessions – see the explanation below) utterances. The number of utterances was chosen to fit within the maximum length of time after which speaker fatigue and changes in electrode condition become too large of a problem to obtain useful data. The sizes of different blocks was allocated to ensure that there is sufficient training data first, and the remainder split between adaptation and evaluation.

Block 1 includes recordings of the entire set of sentences available in the corpus (the full training, development and evaluation sets). It can be used to train and optimize EMG-to-Speech systems and to create a baseline for evaluation in a manner that is comparable to offline EMG-to-Speech conversion.

Block 2, 4 and 6 each contain 20 training sentences (identical in each case). These can be used as adaptation data for adapting a system within one session. Block 2 additionally contains 20 sentences each for development and evaluation. This data can be used for evaluating different training strategies on data that is recorded close to but not concurrently with the training data.

Block 3, 5 and 7 contain 30 development and 20 evaluation utterances to evaluate these strategies on data not recorded concurrently with the data that the system is being trained on. This matches the evaluation scenario of a real online EMG-to-Speech conversion system, where compensation for time-related artifacts is required. Table 4.7 presents an overview of the utterance counts in each block and subset.

There are two types of sessions in the corpus: Audible-testing-mode sessions, and silent-testing-mode sessions. For the audible-testing-mode sessions, subjects were prompted to simply read out the utterances as they normally would, and parallel EMG- and Audio signals are included for each utterance. For silent-testing-mode sessions, subjects were asked to silently mouth all sentences that are part of the development or evaluation subset in blocks 1 through 7 (i.e. mouthing without producing sound while reading along) – for these, only an EMG signal is included, as reference audio signal is not produced. Note that this means that for these sessions, it is not possible to directly compare the systems output with a reference signal since an acoustic signal does not exist when people speak silently.

The lack of audible acoustic reference data in silent-testing-mode sessions is a problem when trying to evaluate EMG-to-Speech systems built for this mode: Common measures such as the MCD score rely on such a reference signal and cannot be computed when it is not available. To still allow for objective evaluation, silent sessions include an additional Block 0 (marked with parentheses in Table 4.7 that contains an audible recording of the development and evaluation utterances (both EMG and Audio). This data can be used to evaluate EMG-to-Speech conversion output using *dynamic time warping* (DTW) alignment or similar techniques. In addition to the EMG- and audio data, metadata about the recordings (including transcripts) are also included.

### Recording setup

Recordings were performed in a recording chamber shielded against acoustic and electromagnetic interference. The audio signals included in the corpus were recorded using a RODE NT-1 condenser microphone and a Behringer Xenyx 302 audio interface. The EMG signals were recorded using our array based setup with an OT Bioelettronica Quattrocento EMG amplifier (see Section 4.1.2). Cross-row channels were not excluded and instead provided as-is. Finally, one channel was added to both the EMG- and audio signal, containing a marker that is pulled high by the EMG amplifier at the start of each utterance, allowing for easy synchronization of the EMG and audio signals by alignment of the markers. Audio data was sampled at 16000 Hz. The EMG signal was sampled at 2048 Hz with a 0.3 Hz DC offset removal and a 500 Hz anti-aliasing filter applied, and re-scaled to millivolt range (i.e. an EMG signal value of 1 for a channel means 1 mV of measured voltage difference).

**Table 4.8** – Session durations broken down by training, development and testing set as well as speaker gender and session mode.

| | | | Total (mm:ss) | | | Mean (mm:ss) | | |
|---|---|---|---|---|---|---|---|---|
| Session | mode | m/f | train | dev | eval | train | dev | eval |
| Spk1 | aud | m | 25:21 | 12:06 | 10:33 | 4.9 | 4.5 | 5.3 |
| Spk1-Sil | sil | m | 21:33 | 13:17 | 11:46 | 4.2 | 3.8 | 4.4 |
| Spk2 | aud | f | 26:14 | 11:50 | 10:09 | 5.1 | 4.4 | 5.1 |
| Spk3 | aud | f | 24:33 | 11:16 | 10:16 | 4.8 | 4.2 | 5.1 |
| Spk3-Sil | sil | f | 23:20 | 15:33 | 13:42 | 4.5 | 4.4 | 5.1 |
| Spk4 | aud | m | 31:31 | 14:04 | 12:26 | 6.1 | 5.3 | 6.2 |
| Spk5 | aud | m | 20:53 | 9:29 | 8:14 | 4.0 | 3.6 | 4.1 |
| Spk6 | aud | m | 28:42 | 13:09 | 11:30 | 5.6 | 4.9 | 5.8 |
| Spk6-Sil | sil | m | 28:40 | 16:25 | 14:21 | 5.5 | 4.7 | 5.4 |
| Spk7 | aud | f | 25:13 | 11:37 | 10:22 | 4.9 | 4.4 | 5.2 |
| Spk8 | aud | m | 20:50 | 10:02 | 8:30 | 4.0 | 3.8 | 4.2 |
| Spk8-Sil | sil | m | 20:44 | 12:51 | 10:59 | 4.0 | 3.7 | 4.1 |
| **All** | | | 297:35 | 151:38 | 132:49 | 4.8 | 4.3 | 5.0 |

### Recorded speakers and sessions

The corpus contains 12 sessions from a total of 8 speakers. 4 speakers (speakers 2, 4, 5 and 7) recorded audible sessions only, the other 4 (speakers 1, 3, 6 and 8) recorded both an audible and a silent session. The recorded speakers read English sentences but are not native English speakers. They were allowed to re-attempt recording as often as desired if they felt they needed to correct their pronunciation. Three of the speakers were female, and five speakers were male. Speakers ages ranged between 19 and 32 years old. A detailed breakdown of the sessions can be found in Table 4.8. In total, 9.5 hours of data are available. Informed written consent of all recorded speakers was acquired prior to the collection of data.

Chapter 5

# Evaluation and Signal Processing

*This chapter describes and motivates the evaluation and feature extraction methods that were used in this dissertation as well as newly developed for real-time EMG-to-Speech conversion. It also presents experiments performed to inform and evaluate the design of these methods as well as the baseline offline EMG-to-Speech conversion system used in this dissertation.*

## 5.1 EMG Signal Processing and Features

This section introduces the two different EMG feature sets used in this dissertation, the pre-existing TD15 features [JSW$^+$06] and the newly developed C-TD15 features, as well as the technique developed to ensure a consistent EMG signal range for more resilient online processing of the EMG signal.

### 5.1.1 Available Features: TD15

TD15 features, introduced by Jou et al. [JSW$^+$06], are the standard features in EMG-to-speech conversion. They are calculated from the raw EMG signal as follows:

First, the EMG signal is aligned with the audio signal, and then shifted 50 ms into the future, such that (assuming an EMD of 50 ms) EMG samples are aligned with the audio samples that they are maximally relevant for.

Each channel of the signal is then split into low-frequency and high-frequency components. A low frequency signal is obtained by applying a nine point double averaging filter to the EMG signal. A high-frequency signal is then calculated by subtracting the low-frequency signal from the raw EMG.

Both low- and high frequency signal are windowed using a rectangular window, both using the same window length and frame shift (the frame shift is fixed at 10 ms, while we evaluate multiple different window sizes in this dissertation). For each frame, we extract five different features that together make up one TD0 feature frame:

- the low-frequency signal power
- the low-frequency signal mean
- the high-frequency signal power
- the high-frequency signal rectified mean
- the high-frequency signal zero-crossing rate

The TD0 frames of all channels are then combined and stacked into the past as well as the future for 15 frames each to create the final TD15 feature frames.

The original TD15 feature set [JSW$^+$06] has proven to be a resilient and effective choice for EMG-to-Speech conversion (a baseline evaluation using these features can be found in Section 5.3 of this dissertation). However, there are several issues with TD15 features that prevent their use in a practical low-latency online EMG-to-Speech conversion system.

The main issue is that it requires substantial amounts of future context, both explicitly through stacking (150 ms into the future) as well as implicitly (to calculate the 9 point double average, requiring 9 frames of context – 15 ms when sampling with 600 Hz). The shifting of the EMG signal relative to the audio compensates for this to an extent: The latency is reduced by 50 ms because the feature transformation is effectively trained to produce the audio frame 50 ms into the future relative to the current EMG frame. In total, any system using TD15 features will have a best-case latency of 115 ms on top of any time taken for computations, which is not acceptable for a system meant for conversational use.

A smaller, but not insignificant issue is the 9 point double average itself: It defines a low-pass filter in terms of a fixed sample rate of 600 Hz, which is not a sample rate commonly supported by EMG recording equipment. In practice, this means that we either have to resample the EMG signal, or replace the double averaging with a different filter that approximates the triangular low-pass filter that it represents whenever we are using a signal with a sample rate other than 600 Hz, which is not guaranteed to give the same results (this dissertation takes the latter approach).

### 5.1.2 Proposed Features

As explained in Section 5.1.1, the TD15 feature set cannot be used when building an online EMG-to-Speech conversion system. For this reason, we introduce a new feature set: The *causal TD15* (C-TD15) features, which can be calculated with very little latency and no explicit future context. We also introduce a means of performing running normalization on EMG data to ensure consistent signal levels and suppress noise.

#### C-TD15 Features

To calculate C-TD15 features for a single EMG channel, the signal is again split into a high-frequency-band and a low-frequency-band part by application of third-order Butterworth high- and low-pass filters with a cutoff frequency of 134 Hz (resulting in a delay of approx. 12 samples). The high- and lowband signals are then each processed into frames with a fixed length (we evaluate different window lengths as part of this dissertation) and 10 ms shift. From the resulting frames, the lower-band power, lower-band mean, higher-band power, higher-band zero-crossing rate and higher-band absolute-value mean are calculated, resulting in one C-TD1 frame. The C-TD1 frame is stacked together with the 14 preceding C-TD1 frames to obtain the final C-TD15 feature vector for that channel. To calculate the C-TD15 features for a multi-channel EMG signal, the C-TD15 features for each channel are calculated separately and then concatenated to obtain the combined multi-channel EMG feature vector.

While TD15 features were designed for offline use, and therefore try to position the central EMG frame of a stacked feature vector in such a way that it is aligned with the audio frame that it is maximally relevant for, compensating for EMD, the situation is different for online use. Due to the lack of future

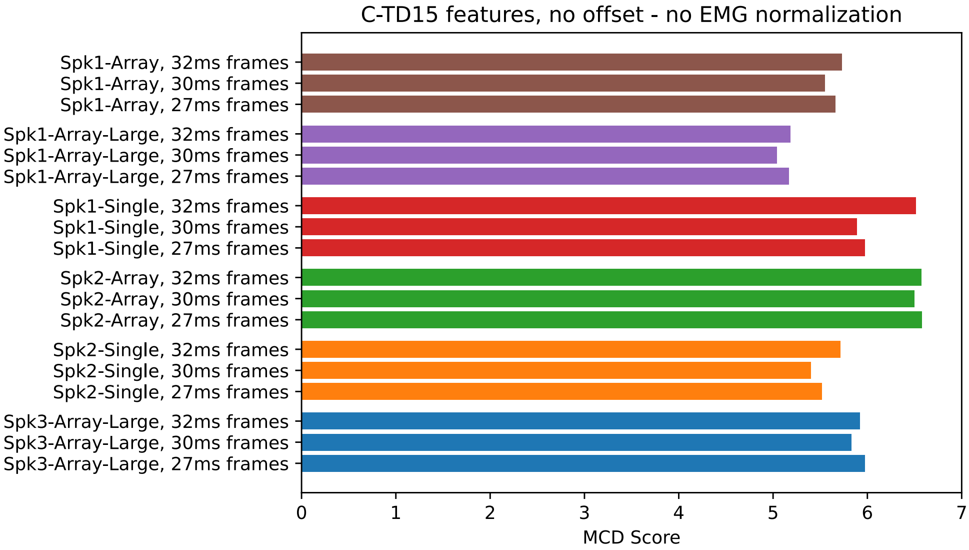


**Figure 5.1** – Results of performing EMG-to-Speech conversion using a neural network system with C-TD15 features calculated with 0 ms EMG-Audio shift.

stacking, shifting the signal may cause important information to be lost if the EMD for a specific muscle and motion is less than 50 ms.

We use our baseline system to evaluate two C-TD15 feature variants: One with 0 ms shift (Figure 5.1) and one with 50 ms shift (Figure 5.2), using different frame sizes. Interestingly, we observe that the features with 0 ms shift (overall mean MCD of ~ 5.76) perform significantly better than the features with 50 ms shift (mean MCD of ~ 5.99, one-tailed independent sample t-test, tested at a level of p ¡ 0.05). There are two effects that likely contribute to this. One is that while the EMD is 50 ms on average, different phones may have more or less delay. The other is that due to effects such as co-articulation, future context can be relevant for a specific phone. Therefore, universally shifting the signal without stacking into the future may sometimes cut off EMG data that would be important for a given phone that is being produced. It is therefore preferable to not shift the signal if the rest of the system is fast enough to produce output without perceptible delay.

### EMG Normalization

We next present an initial method for addressing these differences, enabling EMG-to-Speech conversion in a realistic real-time online scenario. We achieve

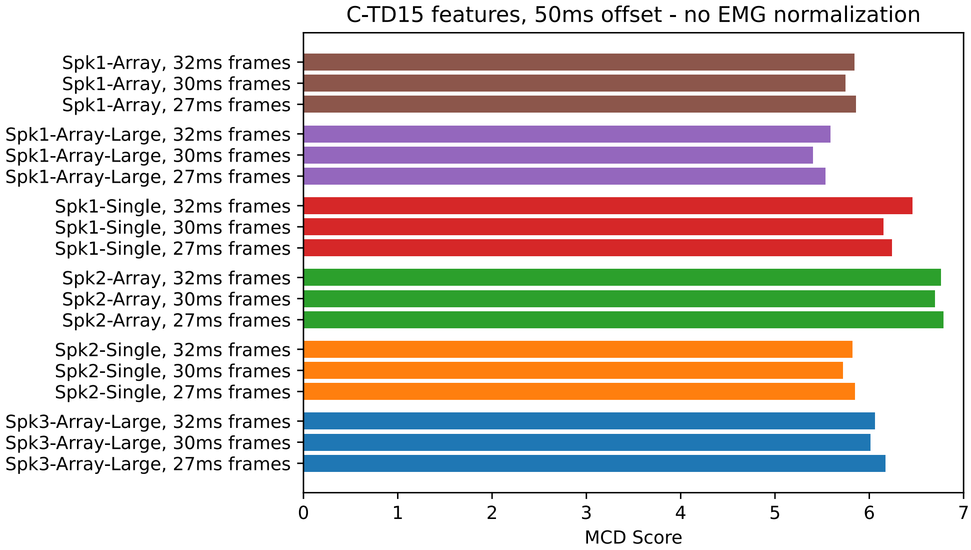


**Figure 5.2** – Results of performing EMG-to-Speech conversion using a neural network system with C-TD15 features calculated with 50 ms EMG-Audio shift.

this by performing running normalization of the EMG signal. We keep track of the 99th percentiles of the absolute value of EMG channels over 250 ms. We then normalize all samples using this 99th percentile value, unless such a normalization would result in an amplification greater than 100. This keeps the signal in a range of -1 to 1, compensating for drift and short artifacts while not amplifying noise from detached electrodes.

To test the effect of this normalization approach on output quality in general, we again use our baseline system, both using TD15 and C-TD15 features, and add EMG normalization before the features are calculated. We use this system to train and evaluate using the EMG-ArraySingle-A-500+, which allows us to compare the effects that this method has for single-electrode as well as array-electrode based sessions. The results can be seen in Figure 5.3 (TD15 features) and Figure 5.4 (C-TD15 features). We observe an interesting effect: While the normalization slightly worsens the MCD score for the single-electrode sessions, which are relatively free of artifacts, it improves the MCD score for the Array-based sessions even for the offline evaluation mode.

An evaluation for the online evaluation mode, where the system has to compensate for time-related differences between training and evaluation data, is presented along with the introduction of the CSL-EMG_Array corpus in Section 4.3.3.

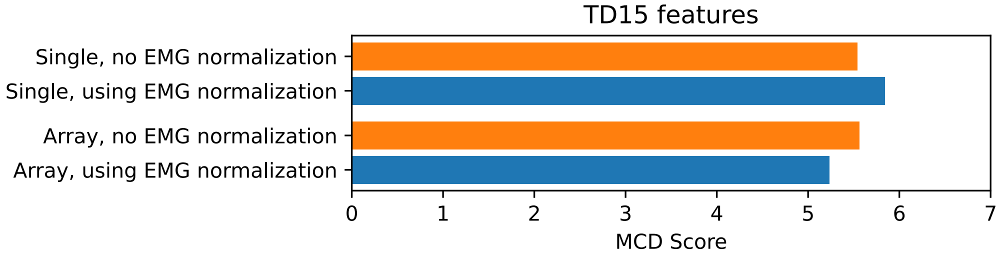


**Figure 5.3** – MCD scores for EMG-to-Speech conversion with and without EMG normalization for both the single-electrode as well as the array electrode sessions of the EMG-ArraySingle-A-500+ corpus, using TD15 features.

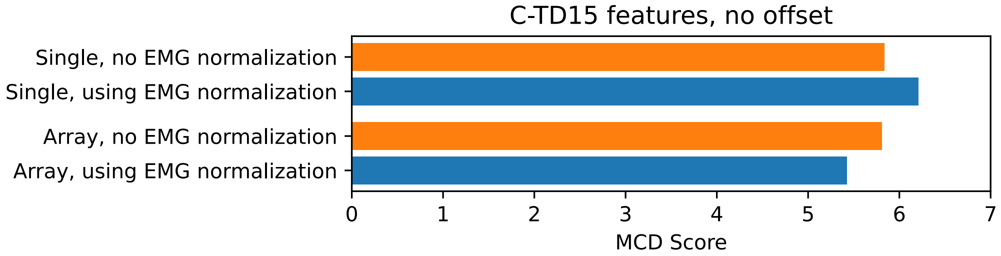


**Figure 5.4** – MCD scores for EMG-to-Speech conversion with and without EMG normalization for both the single-electrode as well as the array electrode sessions of the EMG-ArraySingle-A-500+ corpus, using CTD15 features.

## 5.2 Evaluation Measures

To improve EMG-to-Speech conversion, we need to measure its quality. This section briefly introduces the measures and methods used to do so in this dissertation: Objective scores such as the Mel-cepstral distortion and short time objective intelligibility as well as subjective listening tests.

### 5.2.1 Available Measures

First, we will review the MCD and STOI, two available measures used for comparing a distorted audio signal with a clean reference.

### Mel-Cepstral Distortion Score

The *Mel-Cepstral Distortion* (MCD) score [Kub93] is a measure of cepstral distance: It measures how much one cepstrogram differs from another. The MCD score is defined as a scaled Euclidean distance between Mel-frequency cepstral coefficient vectors excluding the first coefficient:

$$\mathbf{MCD} = 10/\ln 10 \sqrt{2 \cdot \sum_{k=2}^{25} (\mathbf{mfcc}_{\text{synthesized}}[k] - \mathbf{mfcc}_{\text{reference}}[k])^2} \qquad (5.1)$$

Given that the reference is intelligible, clean speech, a low MCD score indicates that the MFCC parameters of both speech sequences are similar and thus that the synthesized audio is also intelligible. Note that the MCD score only considers MFCC parameters – it does not consider the fundamental frequency and is therefore of limited use when trying to evaluate the naturalness of generated speech.

To be able to calculate the MCD score, reference audio that is exactly aligned frame by frame to the generated audio is required. Such an alignment is not always available – for example, in the case of silent speech, where it simply does not exist. For cases where we do not have aligned reference audio available, we calculate the MCD score by first aligning MFCC vectors using the DTW algorithm and then calculating the MCD score. The resulting score is called the "DTW-MCD" score.

One downside of the MCD score is that, being a score derived from MFCCs, it will tend to overestimate the performance of systems that internally use an MFCC speech representation, compared to systems that do not. It should also be noted that the MCD score depends on the range of in- and output data (as a lower overall range will also improve the MCD score without increasing quality) – so the MCD score can not be meaningfully compared between different recordings, only between different systems operating on the same data. Normalization can partially alleviate this problem, and all MCD scores reported in this paper were obtained from audio normalized to have 16 bit signed integer range ($-2^15$ to $2^15 - 1$).

### STOI

The *Short Time Objective Intelligibility* index (STOI) [THHJ10] is a measure designed specifically to evaluate how intelligible a distorted speech waveform is, given a clean reference waveform that is assumed to be perfectly intelligible. A STOI of 0 means "completely unintelligible" and a STOI of 1 meaning "perfectly intelligible" / identical to the reference.

The STOI operates on a spectrogram representation of the audio waveform, calculated by resampling the signals to 10 kHz and taking the Fourier transform with 512-length, 256-shift Hanning windows. The resulting spectra are grouped into 15 one third octave bands, with the lowest bands frequency centered on 150 Hz, and the spectral power for each of these bands is extracted. The resulting frames are then normalized to make the total energy of the distorted speech frames match the clean speeches in a context of 30 units around the frames to be normalized. They are then clipped such that the minimum signal to distortion ratio between clean and distorted speech is -15 dB to prevent any single frame from unduly affecting the result. The final STOI measure is then the mean of the linear correlations between the normalized clipped distorted speech frames and the clean speech frames.

Compared to the MCD score, the STOI was specifically designed to evaluate intelligibility and is known to correlate well with human assessments of intelligibility [THHJ10]. Additionally, unlike the MCD score, it is not affected by re-scaling of the data, which improves comparability. Also, since it is not dependent on an MFCC representation, applicable to comparing different audio representations.

## 5.2.2 Proposed Measure: TL-Acc

The previous two measures primarily deal with intelligibility. However, naturalness is also an important part of synthesizing speech. Our direct synthesis SSI approach aims to produce speech that can not only be understood, but that also sounds as much like human speech as possible. Thus, we need a measure that can be used to compare generated audio with reference audio and tell us whether the generated audio is similarly natural. As naturalness strongly depends on fundamental frequency, a natural approach is to compare $F_0$ trajectories.

Typical approaches for doing this are the $F_0$ *correlation* (the correlation between of two $F_0$ trajectories, often restricted to voiced sections – then called the *voiced section correlation*) or the *voicing accuracy*. The voicing accuracy is the ratio of frames correctly assigned a discontinuous $F_0$ of 0 (not voiced) or not 0 (voiced), respectively. Both approaches are lacking: The voicing accuracy only indirectly relates to naturalness, and the voiced section correlation ignores the fact that one utterance may have many similar $F_0$ trajectories are all correct and natural fits for the utterance.

To ameliorate these issues, we introduce a new objective measure for comparing generated $F_0$ trajectories to a reference, the *trajectory-label accuracy* (TLAcc) [DUS19]. It combines voicing accuracy and correlation, and abstracts $F_0$ trajectories of voiced sections by reducing them to their most basic components: going up, going down, or neither of the two. While – compared to other prosody annotation schemes such as ToBI [SBP+92] – this is a major simplification of the complexity of $F_0$ movement during speech, this allows comparisons of whether a generated trajectory is basically similar to another – unlike these other schemes, which may provide too much detail to allow for meaningful comparisons, or may not even be accurately extractable without human assistance.

Our results show that the TLAcc is more strongly correlated with subjective assessments of naturalness than voicing accuracy or voiced section correlation. At the same time, it is still easy to evaluate without human interference, potentially making it a better candidate for use during system development than the voicing accuracy or voiced section correlation.

The TLAcc is calculated as follows: First, the numerical gradient of the $F_0$ trajectory is calculated by subtracting the value of the frame right of the current frame from the value left of the current frame – however, if either the value of the frame to the left or to the right of the current frame is zero (unvoiced), the central value is used instead of that value. Then, labels are assigned:

- "unvoiced" (the $F_0$ value of this frame is zero / unvoiced),
- "rising" (the $F_0$ value rises by at least 5 Hz, according to the calculated gradient)
- "falling" (the $F_0$ value falls by at least 5 Hz, according to the calculated gradient)
- "flat" (otherwise)

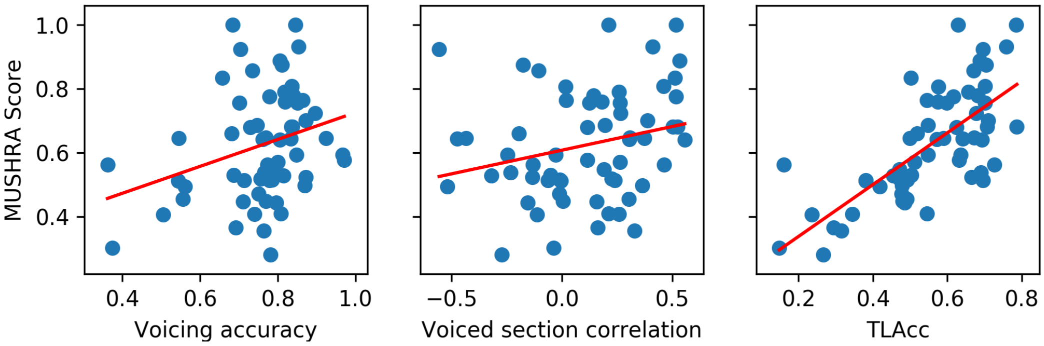


**Figure 5.5** – Scatter plots of utterance mean normalized MUSHRA scores against the ratings assigned to the same utterances by the voicing accuracy (left), the voiced section correlation (middle) and our proposed trajectory-label accuracy measure (right). Red lines indicate regression line.

The TLAcc is then the accuracy calculated between the reference and hypothesis trajectory labels. A reference python implementation of this measure is available online[1].

To verify that this is a reasonable approach, we compare the ratings produced by this new measure to human evaluations – the gold standard in rating the naturalness of speech – from a listening test using the MUSHRA [ITU01] method, evaluating output of EMG-to-Speech conversion on the EMG-ArraySingle-A-500+ Corpus. For more information on MUSHRA based subjective listening tests, see Section 5.2.3.

A scatter plot of the MUSHRA scores (average per utterance scores) versus the three different objective measures for all utterances from the listening test can be seen in Figure 5.5, showing that the trajectory label accuracy is strongly correlated with the listening test scores (Pearson's $r \approx 0.71$, compared to only $r \approx 0.3$ for the voicing accuracy and $r \approx 0.25$ for the voiced section correlation).

### 5.2.3 Subjective Listening Tests

While objective measures are convenient to use during system development to compare two methods, the gold standard in speech quality assessment continues to be subjective listening tests with human listeners. This dissertation uses two different approaches to listening tests: Comparative A/B preference tests, and mean opinion score (MOS) [IT06] based tests.

[1] https://github.com/cognitive-systems-lab/trajectory-label-accuracy

Comparative A/B preference tests are used to compare two systems directly: A test participant is presented with output from two systems A and B for the same utterance, and asked to choose whether they prefer system A, system B, or neither. This way of testing is most useful for evaluating whether a proposed system produces output that humans perceive as better than a baseline systems.

MOS based tests instead ask the user to rate utterances on a scale. The MOS testing method used in this dissertation is the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) method [ITU01]. Here, a test participant is presented with a reference audio file (the audible reference) marked as being the reference first. The participant is then asked to rate the output from multiple systems in addition to the reference (unmarked) and a low anchor on a continuous scale from 0 to 100 points. The anchor (generated by strongly distorting the reference with a low pass filter) and hidden reference are used to set and normalize for user expectations of quality. The reference is also used to verify test participants performed the task correctly by checking whether it was consistently assigned a score near the maximum.

All our subjective listening tests were performed using the web-based BeaqleJS framework [KZ14], with randomized order of presentation for utterances as well as different versions of each utterance. Details on prompting and instructions differ between tests and are given separately for each evaluation.

## 5.3 Baseline Offline EMG-to-Speech Conversion System

This section briefly describes the baseline offline EMG-to-Speech conversion system using neural networks, adapted from our previous work [DJS15b]. This system forms the basis for many of the evaluations that informed the design of our new online EMG-to-Speech conversion system, presented in Chapter 6. The model was first developed by the author of this dissertation when working as a student research assistant as part of the dissertation work of and together with Matthias Janke [Jan16].

The system uses a three-step approach to converting facial speech sEMG signals to audible speech. First, EMG TD15 features (see section 5.1.1) and, for training, parallel audio MFCC+F0 features are extracted. The EMG features are then converted to audio features by a feed-forward neural network.

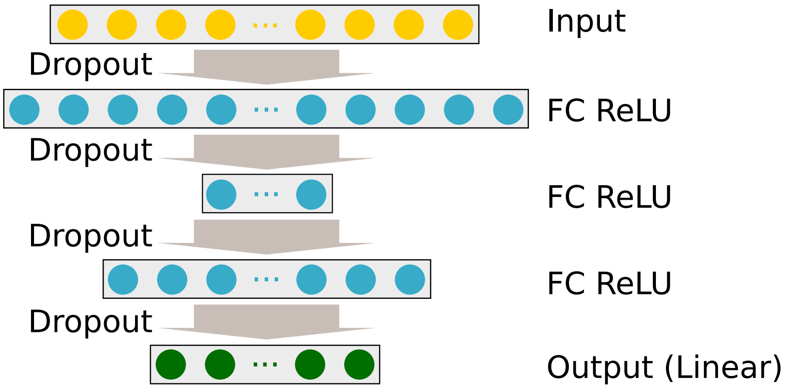


**Figure 5.6** – Structure of the feed-forward neural network used in the baseline EMG-to-Speech conversion system.

The network uses a three-hidden-layer bottleneck shape structure, with hidden layer sizes (from input to output) of 2048, 512 and 1024. The activation function is the Rectified Linear (ReLU) function, except for the output layer, which uses linear activation. After each layer (including the input layer), dropout regularization [SHK+14] (with a dropout rate of 0.5) is applied to prevent overtraining. Figure 5.6 shows an overview of the network structure. The network is trained for 250 epochs using stochastic gradient descent with a learning rate of 0.01 and a momentum term of 0.9.

Finally, after converting the EMG features to audio features, the final waveform output is generated by MLSA vocoding. Each of the steps described is performed for the data of a complete utterance at a time.

### 5.3.1 Baseline evaluation

To provide a baseline for the comparison with other approaches, we present an evaluation of the baseline offline EMG-to-Speech conversion system on the EMG-ArraySingle-A-500+ corpus. We perform this evaluation for different frame lengths that have been used in previous EMG-based speech processing work, 27 ms [WKJ+06] and 32 ms [DJS15b], as well as 30 ms as an intermediate step. The results are presented in Figure 5.7.

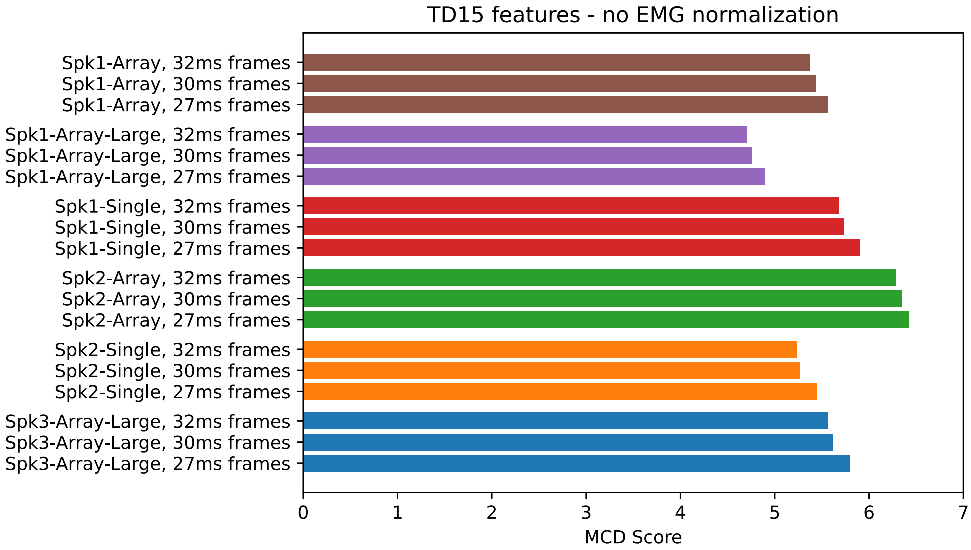


**Figure 5.7** – MCD score performance on the EMG-ArraySingle-A-500+ corpus of the baseline offline EMG-to-Speech conversion system using TD15 features.

## 5.4 Evaluating Aspects of EMG-to-Speech Conversion

This section presents different experiments performed as part of this dissertation to investigate specific issues related to EMG-to-Speech conversion and to inform the design of our real-time low latency EMG-to-Speech conversion system and study.

### 5.4.1 Paralinguistic Pre-Study on the EMG-UKA Data

While the main goal of an EMG-to-Speech system is to communicate the factual information that a user intended to convey, an important reason to prefer EMG-to-Speech conversion over methods employing a textual representation is that it can transport additional *paralinguistic* information – speaker attributes (i.e. speaker age, gender, personality, ...) or speaker states (i.e. mood, emotion, ...). This is, in fact, mentioned as an advantage of the direct-synthesis approach over other approaches to building SSIs – however, few studies try to verify that such information is present in biosignals used for SSI in practice.

To test whether EMG-to-Speech conversion is in fact capable of decoding information about the speaker and speech other than its text content, we attempt [DAB$^+$20] to recognize two such attributes: Speaker ID (a speaker attribute) and speaking mode (a speaker state). For this, we use the EMG-UKA corpus, since it contains a larger variety of speakers that have recorded multiple sessions than any other EMG corpora available.

For testing speaker recognition, we use the five speakers from the EMG-UKA corpus (see Section 4.2.1) for which more than one session is available. We then train speaker recognizers in a leave-one-session out setup and evaluate on the held-out session. This prevents our speaker ID system from simply latching on to session characteristics. We evaluate two basic classifiers: A linear discriminant analysis (LDA) classifier and a decision tree based random forest classifier. Both operate on a set of utterance summary statistic features calculated in the time domain (mean absolute value, root mean square, sum absolute values, variance, simple square integral, waveform length, average amplitude change, zero crossing rate, slope sign change) and frequency domain (median frequency, weighted mean frequency). We additionally evaluated LDA operating on speaker embeddings obtained from a neural network trained to replicate acoustic speaker embeddings (embedding transfer). The results of this evaluation in terms of speaker ID accuracy can be seen in Table 5.1. It can be seen looking at the range of results for each speaker that there is a clear bias towards speakers with larger amounts of training data that especially affects the more complex embedding transfer approach. However, all approaches manage to perform above chance level (0.55 – prevalence of the most common speakers utterances) averaging over all speakers.

One important caveat to consider with these results is the potential effect of time-related changes on the results: While we do take care to only evaluate on sessions not contained in training, the time at which each session was recorded is not known for the EMG-UKA corpus. It is therefore possible that multiple sessions for a single speaker were recorded on the same day or even without re-application of the electrodes, which may lead us to over-estimate the performance of speaker ID on this data.

To test speaking mode identification, we train an LDA classifier (using the same features that were used for speaker ID recognition) to decode speaking mode with three classes – audible (Aud), whispered (Whis) and silent (Sil). As session characteristics and speaking mode are not inherently linked, we train one classifier on the training set of all multi-mode sessions from all

**Table 5.1** – Session-wise minimum, maximum, and mean/standard deviation of the per utterance speaker ID accuracy from EMG using three different methods.

| | LDA | | | Random Forest | | | Embedding Transfer | | |
|---|---|---|---|---|---|---|---|---|---|
| Spk# | Worst | Best | Mean | Worst | Best | Mean | Worst | Best | Mean |
| 1 | 0.97 | 0.99 | 0.98±0.01 | 0.58 | 0.99 | 0.85±0.19 | 0.25 | 0.33 | 0.29±0.03 |
| 2 | 0.34 | 1.0 | 0.95±0.13 | 0.96 | 1.0 | 0.99±0.01 | 0.1 | 0.98 | 0.81±0.2 |
| 4 | 0.0 | 1.0 | 0.5±0.5 | 0.01 | 0.68 | 0.35±0.33 | 0.29 | 0.49 | 0.39±0.1 |
| 7 | 0.0 | 0.0 | 0.0±0.0 | 0.0 | 0.0 | 0.0±0.0 | 0.01 | 0.04 | 0.02±0.01 |
| 8 | 0.99 | 1.0 | 1.0±0.0 | 0.83 | 1.0 | 0.99±0.04 | 0.43 | 0.9 | 0.75±0.14 |
| All | 0.0 | 1.0 | 0.92±0.24 | 0.0 | 1.0 | 0.93±0.22 | 0.01 | 0.98 | 0.72±0.25 |

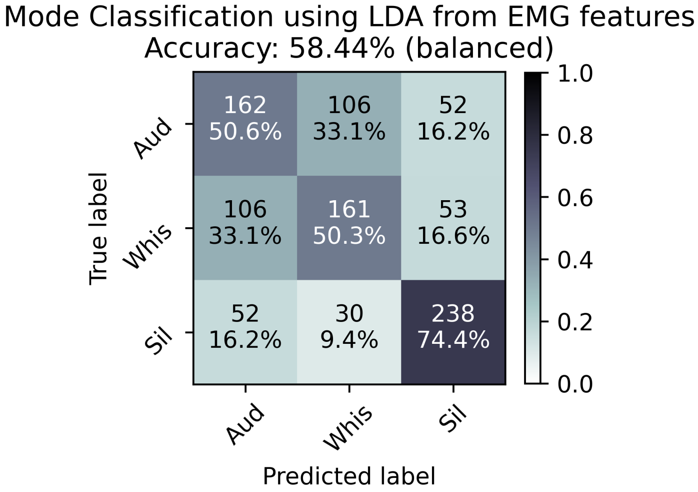


**Figure 5.8** – Confusion matrix of performing mode classification on EMG data using an LDA model.

speakers included in the EMG-UKA corpus and evaluate this classifier on the full multi-mode test set. The results of this evaluation are presented in Figure 5.8. As the multi-mode sessions are balanced with regard to mode, the chance level is be 33.33% accuracy, while the LDA mode classifier achieves 58.55% overall.

The accuracy with which the silent mode is recognized is even higher – 74.4% – and whisper and audible are confused at a higher rate than either is with silent. This matches expectations: While audible and whispered speech are similar insofar as they both involve the production of sound, silent mode speech does

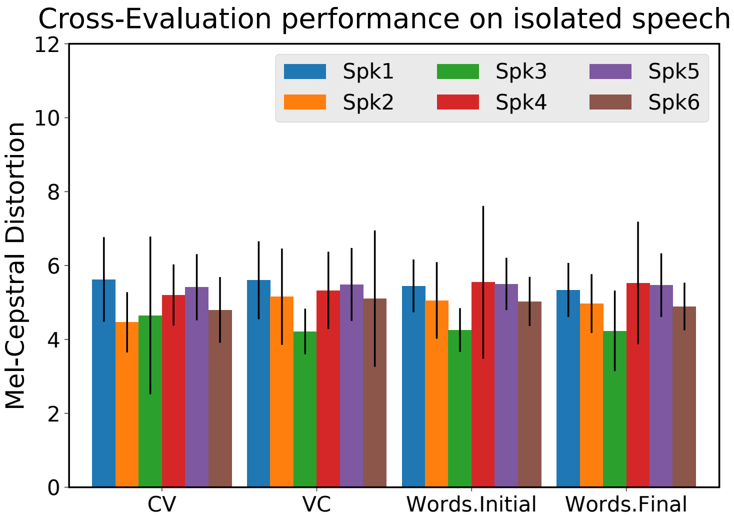


**Figure 5.9** – Results of EMG-to-Speech conversion on isolated speech, obtained using 8-fold cross evaluation. Bars indicate utterance standard deviation, lower is better.

not. The results once again illustrate the importance of compensating for a lack of audible speech production in silent speech processing – a goal of this dissertation.

### 5.4.2 Comparing Isolated and Continuous Speech

To investigate the differences between isolated speech and continuous speech, we perform EMG-to-Speech conversion on the CSL-EMG-Words-CVVC corpus (see Section 4.3.1) and analyze the results. To do this, we train a system based on the methods used for our baseline system described in Section 5.3 and compare the MCD scores of the systems output for different speaking styles.

To evaluate the performance of our system when both training on and converting on the same style (isolated words or CV/VCs), we perform 8-fold cross evaluation training on these subsets (splitting utterances into folds). The MCD scores of the resulting audio can be found in Figure 5.9.

Figure 5.10 shows MCD scores obtained on continuous speech (the sentences development set) using systems trained on different combinations of training data. We show scores when training on the sentences training set, the isolated CV/VCs, the isolated words, words + CV/VCs and finally, the sentences training set + words + CV/VCs all together. Figure 5.11 shows a similar

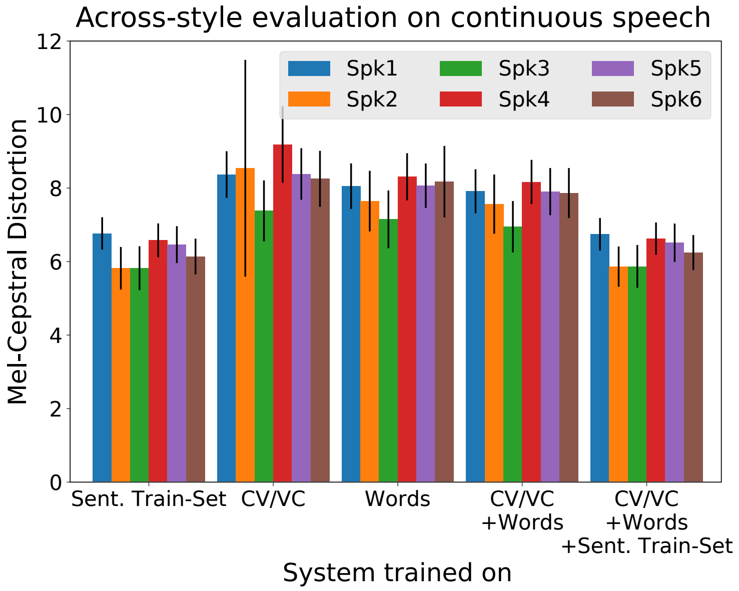


**Figure 5.10** – Results of EMG-to-Speech conversion on continuous speech (on the sentences development set of the CSL-EMG-Words-CVVC), with the system being trained on different combinations of training data. Bars indicate utterance standard deviation, lower is better.

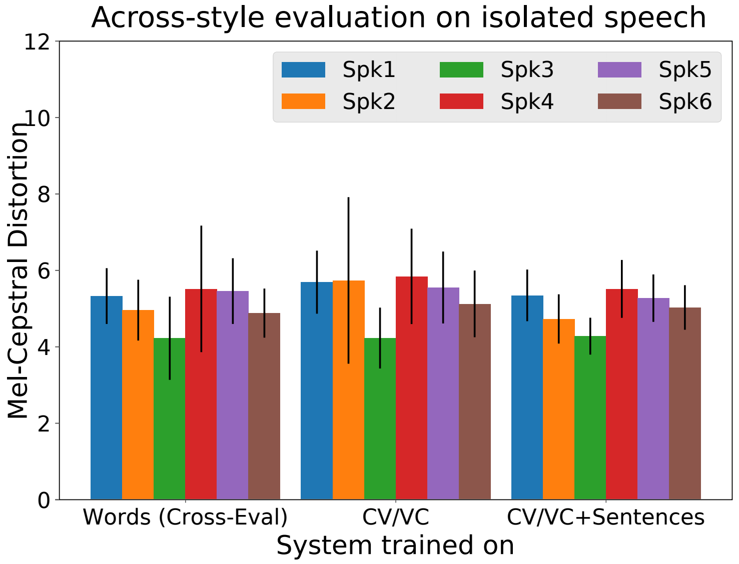


**Figure 5.11** – Results of EMG-to-Speech conversion on isolated speech (on the Words set of the CSL-EMG-Words-CVVC corpus), with the system being trained on different combinations of training data. Words 8-fold cross-evaluation provided for reference. Bars indicate utterance standard deviation, lower is better.

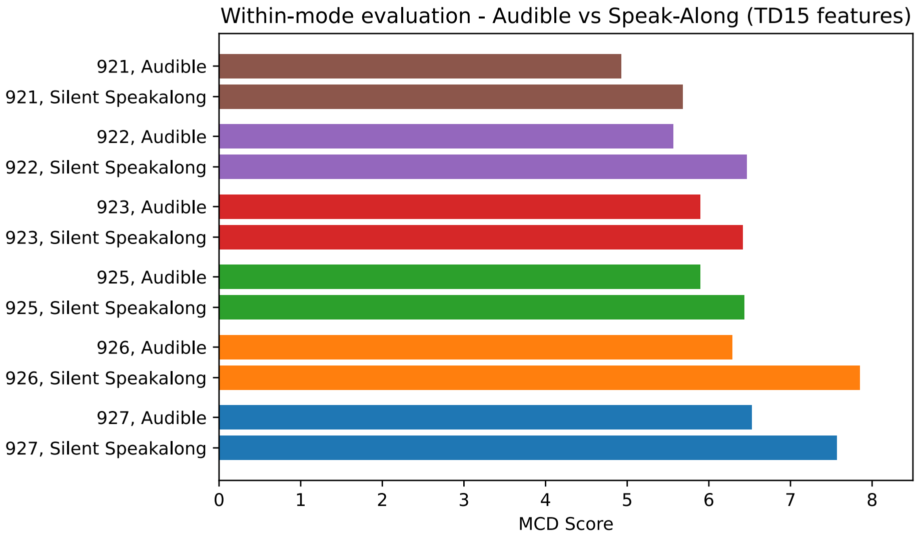


**Figure 5.12** – Evaluation on the CSL-EMG-Speak-Along Corpus - within-mode comparison conversion performance using TD15 features in a DNN+MLSA-based system.

evaluation for isolated speech (the Words set). Here, we use systems trained on CV/VCs and on CV/VCs + the sentences training set. The word cross-evaluation results are provided as a reference.

Comparing Figure 5.9 to Figure 5.10, we can see that there is a clear difference between isolated and continuous speech in terms of EMG-to-Speech conversion performance. Though the amount of training data available for continuous speech is much larger (see Section 4.2.2), better MCD scores are obtained for isolated speech. Additionally, when evaluating across styles, using a system across styles results in worse performance, as can be seen in Figure 5.10 and 5.11: When evaluating across styles, there is a larger drop in performance (Increase in MCD score) then when evaluating within a style (differences significant at $p < 0.05$).

### 5.4.3 Speak-Along Protocol Evaluation

To test whether training on speak-along data recorded according to the protocol described in Section 4.3.2, is feasible, we train and evaluate EMG-to-Speech conversion systems on both the audible and speak-along data of the CSL-EMG-Speak-Along corpus. To evaluate the protocol for both offline as well as online EMG-to-Speech conversion, we train systems using

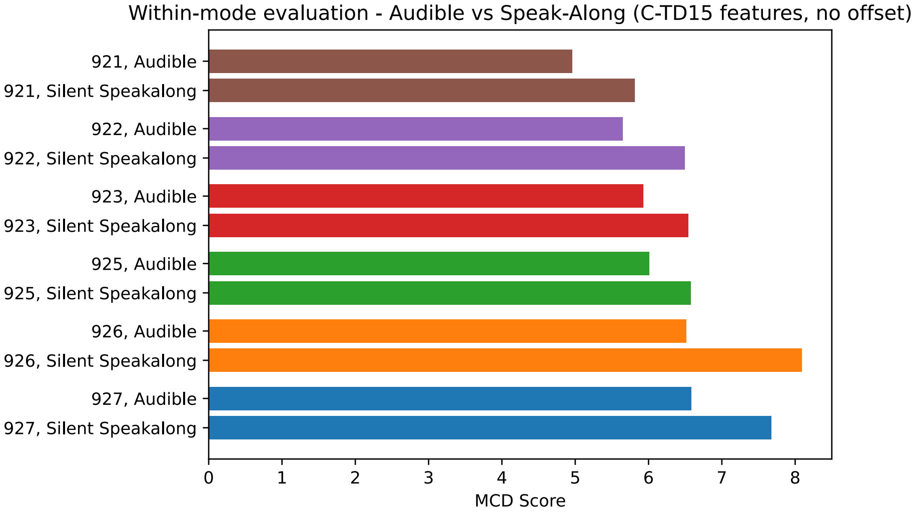


**Figure 5.13** – Evaluation on the CSL-EMG-Speak-Along Corpus - within-mode comparison conversion performance using C-TD15 features in a DNN+MLSA-based system.

both the baseline offline setup described in Section 5.3 as well as the online system described in Chapter 6, using C-TD15 features, both using MLSA vocoding. The MCD score results of this within-mode evaluations can be seen in Figure 5.12 and Figure 5.13 for the baseline TD15 and online C-TD15 systems respectively. It can be seen that, while the performance for the speak-along data is worse than performance on audible data, it is possible to train a working EMG-to-Speech system in this manner.

We additionally test how well both systems work on converting non-speak-along silent data, using the DTW-MCD score. The results are presented in Figure 5.14 (baseline TD15 system) and Figure 5.15 (online C-TD15 system). It can be seen that, while there are differences between the systems in terms of MCD score when evaluating within a mode, there is no significant MCD score difference between using an audible EMG based system and using a speak-along EMG based system for converting silent EMG data to audible speech. The conclusions we can draw from this are twofold: First, there appears to be no advantage to training a system for silent EMG-to-Speech conversion in terms of output quality – the signal is either no closer to the silent signal than an audible EMG signal, or any advantages are negated by issues unique to the speak-along recording method. However, second, a system trained on speak-along data performs no worse than a system trained

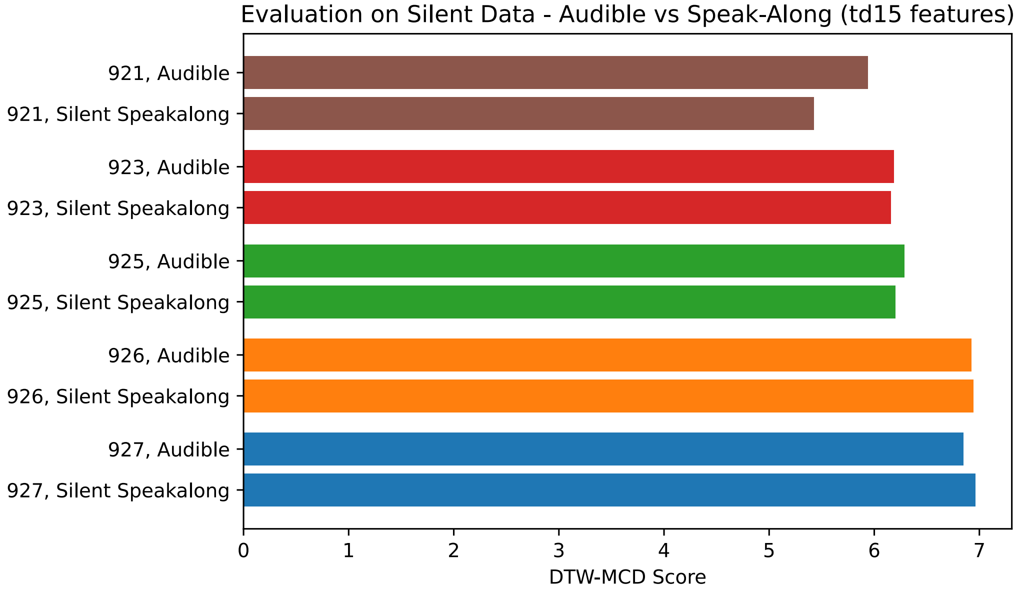


**Figure 5.14** – Evaluation on the CSL-EMG-Speak-Along Corpus - silent-mode comparison conversion performance using TD15 features in a DNN+MLSA-based system. Session 922 has been excluded due to broken Silent EMG data.

on audible data – so training a system using a speak-along recording protocol for people who have already lost their voice is feasible.

We additionally perform a signal-based evaluation and compare the mean *power spectral density* (PSD, calculated via Welchs method [Wel67]) of the EMG signals of each speaker. we calculate separate PSDs for the audible EMG data, silent EMG data with speak-along and silent EMG data without speak-along. The results of this evaluation can be seen in Figure 5.16.

The overall shape of the PSD curves is broadly similar between all sessions – note, however, that the average energy is much lower for some speakers than others. This is likely due to a combination of differences in speaking style as well as differences in electrode-skin contact. When comparing the audible EMG, silent EMG with speakalong and silent EMG without speakalong, there is a clear pattern. The overall energy of the audible EMG signal is always the highest, followed by the silent speakalong and silent EMG. This matches the results from previous work [JWS10, WJS11], where audible EMG was found to have an overall higher energy than silent EMG. Additionally, for speakers 922, 926 and 927, the energy of the speak-along signals is higher than that of the silent signals. This may indicate that the speak-along protocol

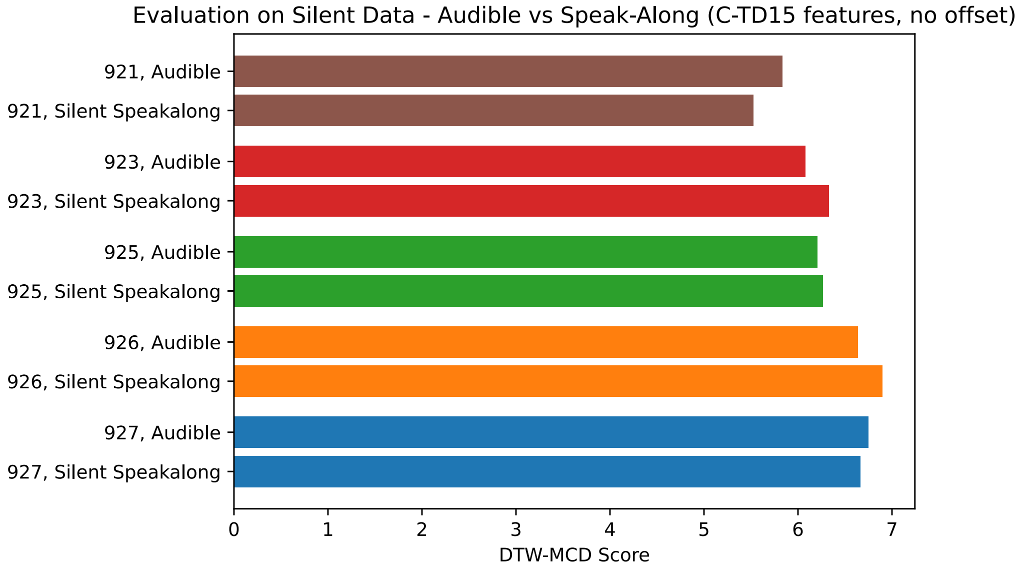


**Figure 5.15** – Evaluation on the CSL-EMG-Speak-Along Corpus - silent-mode comparison conversion performance using C-TD15 features in a DNN+MLSA-based system. Session 922 has been excluded due to broken Silent EMG data.

caused speakers to articulate in a manner that is a mix of audible and silent articulation. An alternate explanation could be speaker fatigue causing participants to articulate with less effort. To test whether this might be the case, we compare the PSD in the first half of the audible EMG recording with the PSD of the second half of the audible EMG recording. The results of this evaluation can be seen in Figure 5.17. It can be seen that there is a large reduction in energy from the first to the second half of the audible EMG data, indicating that the differences in PSD that we see are at least in part due to reasons other than speaking mode.

### 5.4.4 Initial EMG-to-Speech evaluation on the CSL-EMG_Array corpus

To provide a baseline for future results and to further illustrate the usefulness of the blockwise recording protocol of the CSL-EMG_Array corpus presented in Section 4.3.3, we calculate initial EMG-to-Speech conversion results on this corpus. We use C-TD15 features, as described in Section 5.1.2, with EMG normalization applied (Section 5.1.2), and perform the EMG to audio feature

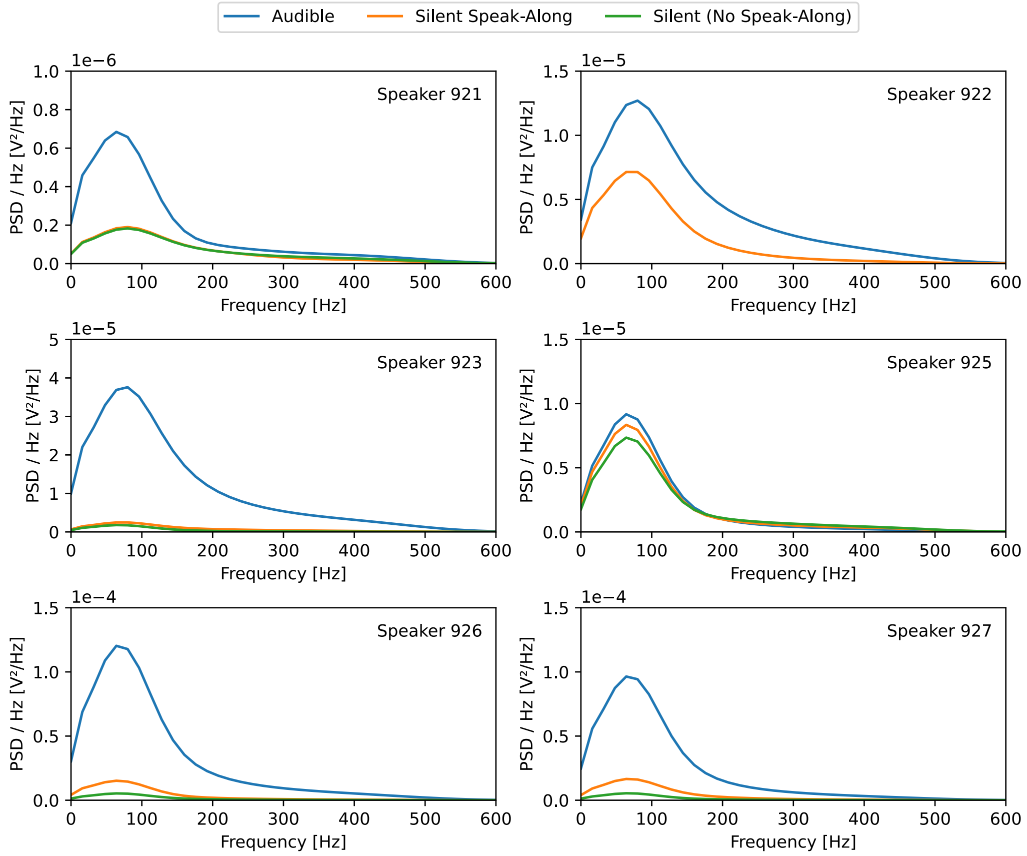


**Figure 5.16** – PSDs of the EMG signals of CSL-EMG-Speak-Along speakers, audible EMG, silent EMG with speak-along and silent EMG without speak-along. Note the different axis scales. Note the different y-axis scales.

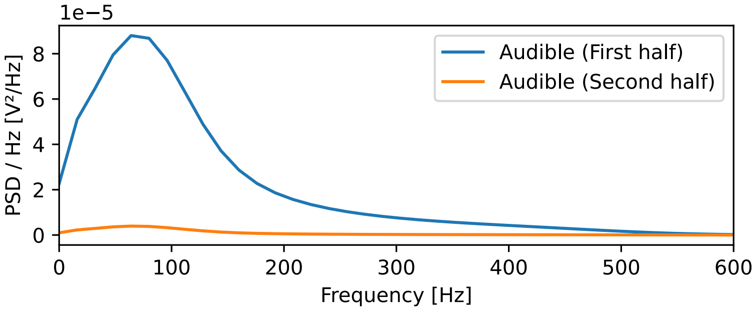


**Figure 5.17** – Mean PSD of the EMG signals of CSL-EMG-Speak-Along audible EMG data, split into first halves and second halves.

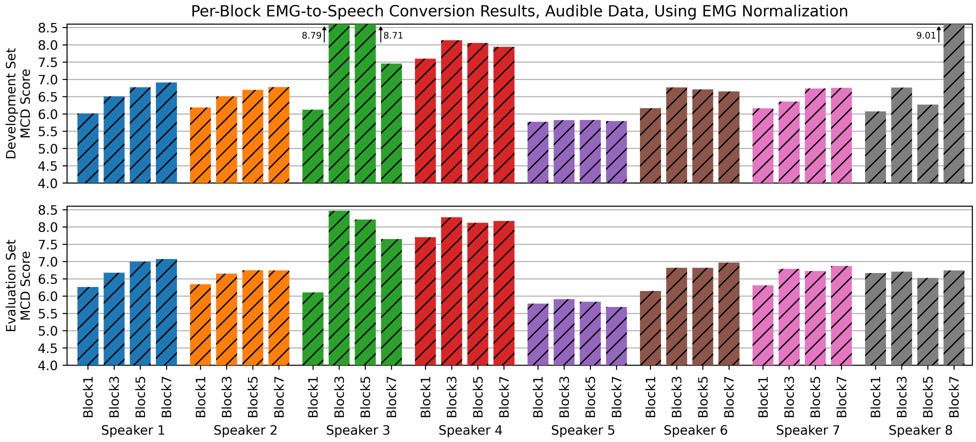


**Figure 5.18** – Baseline MCD scores (lower is better) for real-time session-dependent EMG-to-Speech conversion with EMG normalization, training on Block 1 and evaluating on Blocks 1, 3, 5 and 7.

mapping using a DNN-based online capable system as described in Chapter 6, trained on the block 1 training data. For evaluation, we converted the EMG features of the development and evaluation data of blocks 1, 3, 5 and 7 of the same session. The structure of the network (Bottleneck, hidden layer sizes of 2048, 512 and 1024, dropout regularization after each layer) is based on our previous work [DJS15b]. We trained the networks for 500 epochs using stochastic gradient descent with a learning rate of 0.01 and a minibatch size of 1024 (training times for these systems, on an Nvidia RTX2080Ti GPU, ranged between 20 and 25 minutes). The audio representation used as target for the feature mapping is the MFCC+F0, using the MLSA vocoder.

Results are provided in terms of the MCD score for the audible data, and the DTW-MCD score (calculated using the block 0 references) for silent data. They can be found in Figure 5.18 (Audible data) and Figure 5.19 (Silent data). Note that, due to the alignment, direct comparison of these scores to non-DTW MCD scores is not possible, however, it can be seen that the silent sessions follow the same general trends. The scores for the initial block (i.e. using the “offline” style evaluation where the test set is made of utterances recorded within the same recording block as the training data, with the order randomized) are better than those for the later blocks. This clearly illustrates the need for a blockwise recording protocol.

When the EMG normalization is omitted (results presented in Table 5.2, evaluated on audible sessions only), this becomes even clearer – while the performance for block 1 is still good, the mapping for the later blocks often

**Table 5.2** – MCD scores (lower is better) for an EMG-to-Speech conversion system without within-session EMG normalization, audible testing mode sessions only.

| | Block (Dev. set) | | | | Block (Eval. set) | | | |
|---|---|---|---|---|---|---|---|---|
| Session | 1 | 3 | 5 | 7 | 1 | 3 | 5 | 7 |
| Spk1 | 7.54 | 7.78 | 8.54 | 8.95 | 7.84 | 7.94 | 8.49 | 8.83 |
| Spk2 | 7.55 | 9.98 | 8.22 | 8.46 | 8.12 | 9.57 | 8.21 | 8.41 |
| Spk3 | 7.5 | 17.14 | 14.71 | 9.06 | 7.87 | 17.33 | 14.4 | 9.07 |
| Spk4 | 8.66 | 9.82 | 9.39 | 9.4 | 8.71 | 9.75 | 9.51 | 9.48 |
| Spk5 | 7.49 | 7.95 | 7.69 | 7.65 | 7.46 | 7.93 | 7.64 | 7.56 |
| Spk6 | 7.41 | 7.96 | 7.94 | 7.88 | 7.64 | 8.01 | 8.0 | 8.08 |
| Spk7 | 7.46 | 8.23 | 8.58 | 8.63 | 8.08 | 8.48 | 8.63 | 8.8 |
| Spk8 | 7.83 | 8.18 | 8.01 | 8.31 | 7.96 | 8.27 | 8.25 | 8.33 |

completely breaks down as the system has no way to compensate for even small time-related signal changes.

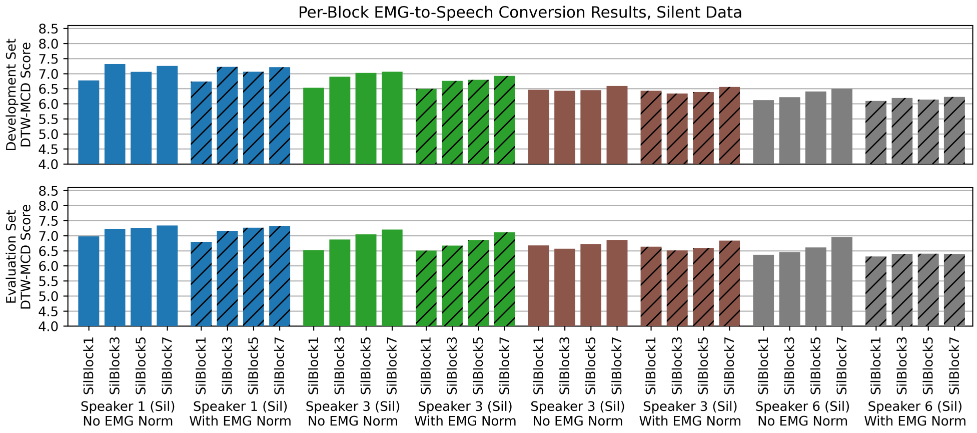


**Figure 5.19** – DTW-MCD scores (lower is better) for real-time session-dependent EMG-to-Speech conversion, training on Block 1 and evaluating on silent data from Blocks 1, 3, 5 and 7.

Chapter 6

# Low-Latency EMG-to-Speech Conversion

*This chapter details the design of our low-latency EMG-to-Speech conversion system. It presents the studies performed to inform design choices and the architecture used to achieve the goal of low-latency EMG-to-Speech conversion.*

To achieve EMG-to-Speech conversion with faster-than-realtime throughput and low latency, we have to make several trade-offs and design software in a way that allows for efficient operation. This chapter explains the architecture of our real-time low latency EMG-to-Speech conversion system and considerations behind architecture decisions.

## 6.1 Pipeline Structure

To perform EMG-to-Speech conversion in real time requires several different computation steps as well as input and output handling. All of these steps take varying amounts of computation time and sometimes require buffering and changes in frame rate. To implement a system that performs all of these steps in an efficient manner, we employ a pipeline structure that allows us to build the system as a sequence of modules that feed input to each other. Each module is either a source module (audio microphone source or EMG source), sink module (audio output), or a processing module that receives

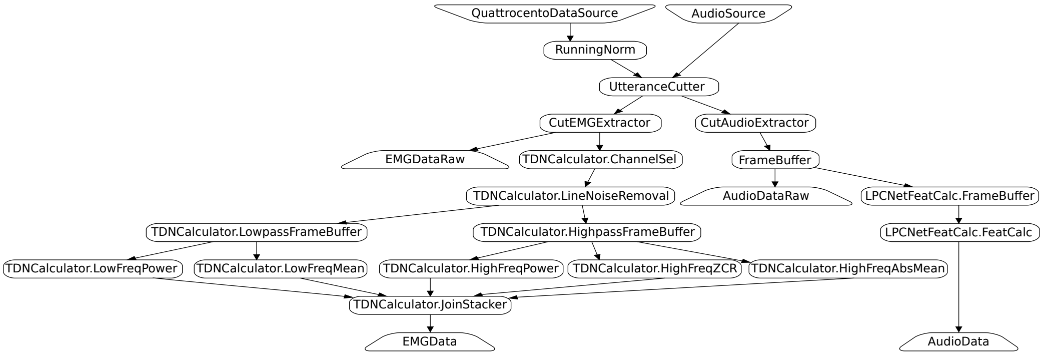

QuattrocentoDataSource
AudioSource
RunningNorm
UtteranceCutter
CutEMGExtractor
CutAudioExtractor
EMGDataRaw
TDNCalculator.ChannelSel
FrameBuffer
TDNCalculator.LineNoiseRemoval
AudioDataRaw
LPCNetFeatCalc.FrameBuffer
TDNCalculator.LowpassFrameBuffer
TDNCalculator.HighpassFrameBuffer
LPCNetFeatCalc.FeatCalc
TDNCalculator.LowFreqPower
TDNCalculator.LowFreqMean
TDNCalculator.HighFreqPower
TDNCalculator.HighFreqZCR
TDNCalculator.HighFreqAbsMean
TDNCalculator.JoinStacker
EMGData
AudioData

**Figure 6.1** – Structure of a system (built using our live framework) for recording, aligning and calculating features for parallel EMG and Audio data. The different modules are explained on the following pages.

input from one or more modules, performs computations on the input and feeds its output to one of more modules. The modules can then be connected in a directed acyclic graph structure to build EMG-to-Speech conversion systems in different configurations. Modules can run their computation in a separate process to allow for computationally expensive steps to run in parallel where possible – however, this introduces overhead for inter-process communication.

Two examples of system structures that are possible with this framework can be seen in Figures 6.1 and 6.2, for cutting and feature extraction during recording and live operation respectively. The following sections will introduce the different modules used in the system used to build the real-time feedback study presented in Chapter 7 of this Dissertation. Modules communicate by passing two-dimensional NumPy [Oli06] arrays between each other, with the first dimension representing time and the second dimension representing feature dimensions.

### 6.1.1 Input

The input modules interact with recording hardware and provide data from the recording hardware as outputs to modules further down the graph. Our system uses two different input modules: One for audio data from a microphone (for recording audible speech data for training and evaluation) and one for recording EMG data from an OT Bioelletronica Quattrocento multi-channel EMG amplifier.

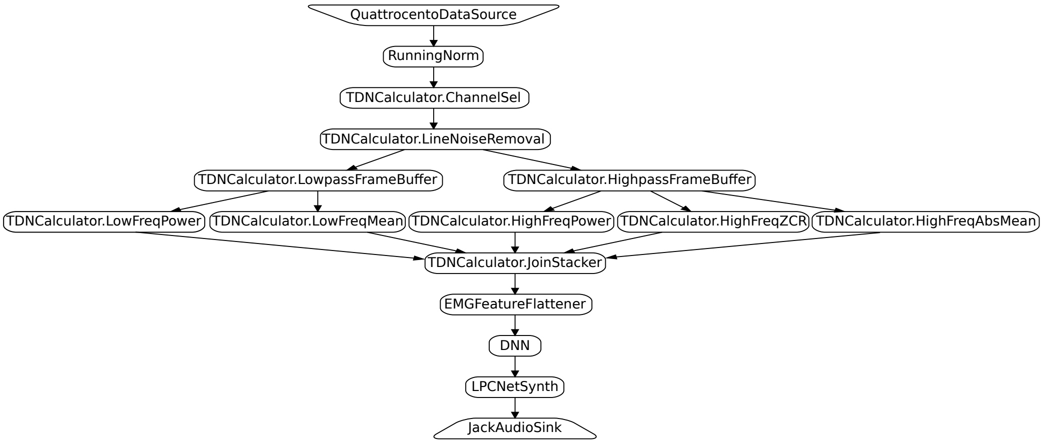

QuattrocentoDataSource
RunningNorm
TDNCalculator.ChannelSel
TDNCalculator.LineNoiseRemoval
TDNCalculator.LowpassFrameBuffer
TDNCalculator.HighpassFrameBuffer
TDNCalculator.LowFreqPower
TDNCalculator.LowFreqMean
TDNCalculator.HighFreqPower
TDNCalculator.HighFreqZCR
TDNCalculator.HighFreqAbsMean
TDNCalculator.JoinStacker
EMGFeatureFlattener
DNN
LPCNetSynth
JackAudioSink

**Figure 6.2** – Structure of a system (built using our live framework) for generation of low latency audio feedback. The different modules are explained on the following pages.

### Microphone Input

Our microphone input module (`AudioSource`) implements microphone recording using the PyAudio [Pha06] API. In a recording process, it records stereo audio from a specified audio interface with a fixed block size of 256 samples and resamples it to a sample rate of 16 kHz. The data is then passed to the following modules in a separate output process – this ensures that no frames are dropped in the input process.

### EMG Input

The EMG input module (`QuattrocentoDataSource`) records data from an OT Bioelletronica Quattrocento multi-channel EMG amplifier. To avoid latency introduced by third-party drivers or interface software, we implemented the binary network protocol used by the amplifier according to manufacturer specifications. This allows us to directly and programmatically control the recording parameters of the amplifier (active channels, sampling frequency, DC offset removal and anti-aliasing filter cutoff) and to receive data and pass it on to the rest of the system without delays caused by buffering.

The module uses two processes. The recording process receives raw sample data from the EMG amplifier, decodes it from the wire representation used by the amplifier to a multi-channel array of voltages, and collects sample values until a specified number of samples has been reached (to avoid overhead from too-frequent inter-process communication). The output process then sends these blocks of samples to the following modules. As in the audio input module, this design ensures that no samples are lost due to processing delays. Unlike the audio module, the EMG source can be configured to drop a number of samples on startup. This prevents "power-on" artifacts and transient effects from propagating further through the system.

**Synchronization Marker Generation** To ensure that audio and EMG signals are perfectly synchronized during processing, our system uses a marker. This marker is generated and output by the EMG amplifier, which has an electric output that can be programmatically configured to be pulled to either 0 V or 5 V using the EMG input module. In our system, this output is connected to the right channel of the audio interface. The actual synchronization is performed by the `UtteranceCutter` module, which will be described later in this chapter.

### 6.1.2 Output

Audio output is handled using the JACK [DLJ02] audio system. The JACK API allows very low-level access to audio hardware with fine-grained control over sample rates and buffer sizes, which is important for low latency audio playback.

Our `JackAudioSink` module performs two tasks in addition to passing the data on to the JACK system for output. First, it performs resampling and format conversion: Since the audio hardware sample rate and format are not necessarily the same as sample rate used by our system internally (16 kHz), the audio first has to be up-sampled to a rate and converted to a representation that the audio interface can process. Second, it performs output buffering. Since the JACK system does not internally buffer audio, any jitter in processing times would lead to undesirable audio drop-outs. To prevent this, the output module buffers a small amount of audio data (at minimum, 10 ms, at maximum, 30 ms – if the buffer ever grows beyond 30 ms, audio data is dropped until it is again below 30 ms to provide an upper bound on the latency introduced by audio buffering) of audio data to smooth over small intermittent delays in processing. The actual output is handled in a separate process to ensure that processing holdups cannot cause audio output issues. If the audio buffer runs empty, the latest block of audio samples is repeatedly output until new data becomes available.

### 6.1.3 Processing

The majority of the modules in our system are concerned with the processing of data. Broadly, there are three types of modules: Pre-processing / feature extraction modules, feature transformation modules, and an audio synthesis module.

#### Cutting and Synchronization

For training, we want to record separate utterances containing parallel EMG and Audio data. The `UtteranceCutter` module provides both splitting of incoming data streams at a given start and end time and synchronization of streams using the marker channel. It provides a function to mark an utterance start point in both the EMG and Audio input stream, and another function that returns all data since the start point. Utterances are synchronized so that the marker is at the same point in time at both streams (detected as the first sample for which the absolute value of the signal rises above 90% of the maximum signal amplitude in an utterance – this prevents electrical noise from interfering with marker detection) with an optional offset to compensate for electro-mechanical delay (set to zero in our system – the reasoning behind this is explained in Section 5.1.2). To ensure that no utterance data is cut off accidentally, the module includes 400ms of additional padding before and after the specified cut-off points.

#### Filtering and Windowing

The `FrameBuffer` module has two functions. First, it is able to apply low-high- and band-pass / stop filters in second order structure form to an incoming stream of data. Second, it is able to store this data in a ring buffer and split it into frames with a given frame size and shift (tracked in terms of seconds rather than full samples to prevent input signals with different sample rates from drifting apart over time due to roundoff error). All buffers required to perform filtering and windowing are pre-filled so that the first frame is output once one frame shift worth of input data has been received.

### EMG Feature Extraction

Our EMG feature extraction is implemented as a sequence of multiple modules. They take a stream of multichannel EMG data as input and output a sequence of C-TD15 EMG feature frames.

**EMG Normalization** The first step in calculating our EMG C-TD15 features is channel-wise EMG normalization, implemented by the `RunningNorm` module. This module implements the normalization presented in Section 5.1.2. To implement this procedure efficiently, we keep track of the per channel median by iteratively inserting new values into a sorted list of the last 250 ms of values, avoiding expensive re-sorting. Additionally, we only update median values every four samples.

**C-TD15 Feature Calculation** The normalized multichannel EMG signal is passed to the `TDNCalculator` module. This module implements the windowing, feature calculation and stacking as a sequence of sub-modules. We start by filtering the EMG signal using a 50 Hz notch filter for line noise rejection, followed by splitting the signal into high- and low frequency parts using a 3rd order Butterworth filter at 134 Hz and windowing into 32 ms windows with a 10 ms shift, both using the `FrameBuffer` module, resulting in a stream of low-frequency and high-frequency frames. We then use `LambdaNode` modules, which simply apply a single function to its input, to calculate the low-frequency signal power and mean as well as the high frequency signal power, rectified mean and zero-crossing rate for each frame. Finally, we use another module (`FrameJoinStacker`) that combines all the calculated features into one feature frame and stacks 15 frames into the past (padded with zeros for instant startup). For a more detailed description and evaluation of the C-TD15 feature set, refer to Section 5.1.2.

### Audio Feature Extraction

Our system implements three different types of audio features: Features based on the MLSA filter, features based on the LPCNet vocoder, and simple frame power features.

**MLSA Features** The MLSA features (Described in detail in Section 2.4.2) consist of MFCCs, calculated by the `WCEPFeatCalc` module, and $F_0$s, calculated by the $F_0$`Calc` module. The former uses the BioKIT [TWG+14] signal processing toolkit to perform MFCC calculation for a single frame of audio data, while the latter uses our implementation of the YIN algorithm. Both use the `FrameBuffer` module for windowing with a frame size of 32 ms and frame shift of 10 ms.

**LPCNet Features** The `LPCNetFeatCalc` module calculates the Bark-scale cepstral coefficients, pitch period and pitch correlation features used by the LPCNet vocoder described in Section 2.4.2. The feature extraction is performed by passing the stream of input audio samples directly to the LPCNet reference implementation, which performs both framing (with a shift of 10 ms, matching the EMG feature frames) and feature calculation. We adapted the reference implementation for use via the Python native binary library interface for maximum performance. To compensate for the algorithmic lag of 25 ms in LPCNet synthesis, the first three frames of audio data are dropped in extraction, making the algorithmic lag effectively -5 ms at the cost of 30 ms of context. However, since we assume the EMD is 50 ms, a loss of 30 ms of context for improved latency is an acceptable trade-off.

**Frame Power features** The `SimpleFeatCalc` module can be used to extract the frame based power of audio data, once again with a frame size of 32 ms and frame shift of 10 ms. It also provides a function to quantize this power into three steps: Bottom 45th percentile, between 45th and 80th percentile, and above 80th percentile. These values were determined to roughly correspond to 'silence“, ”normal speech“ and ”loud speech“ on our dataset.

### Audio Synthesis

The synthesis modules perform the inverse of the audio feature extraction: They take a sequence of audio features and output an audio waveform.

**MLSA Synthesis** The `WCEPSynth` module performs MLSA synthesis using the HTS [ZNY+07] implementation of the MLSA filter, modified to be able to perform run-on synthesis instead of operating on an entire utterance of data at once.

**Frame Power Synthesis** The `SimpleSynth` module can be used to synthesize an audio from frame power features. It generates either silence, a buzzing signal at half volume, or a buzzing signal at full volume, depending on the input power. The buzzing signal is generated as a harmonic oscillation (150 Hz square wave) mixed with white noise at a 9:1 ratio, which results in a signal that sounds broadly similar to speech excitation.

**LPCNet Synthesis** The `LPCNetSynth` uses the LPCNet vocoder described in 2.4.2. Before passing the data to the LPCNet vocoder via the python binary interface, it performs two more tasks: First, the 0th Bark-scale cepstral coefficient is shifted down and stretched by a factor of 1.2. This cancels out low-amplitude noise while keeping the volume and dynamics of higher amplitude sound intact. Second, the features are clipped to the appropriate input range for LPCNet to avoid numerical explosion during synthesis.

**Comparing MLSA and LPCNet** We compare MLSA-based and LPCNet-based synthesis for EMG-to-Speech using the low-latency EMG-to-Speech conversion system described in this chapter, using the low latency capable C-TD15 features. We evaluate the quality of performing session dependent conversion with the feature transformation trained on the training set of the CSL-EMG_Array corpus and performing conversion of the Block 1 evaluation set (i.e. the evaluation sentences recorded in the same block as the training set).

To get an objective evaluation of the performance of both methods, we take the results of performing EMG-to-Speech conversion on the audible EMG sessions and calculate the STOI (we do not consider the MCD score, as the MLSA-based system directly optimizes for it and the LPCNet-based system does not, making it a bad choice for this comparison). The results can be seen in Figure 6.3. We find that the MLSA-based system on average generates output with a significantly ($p < 0.001$) higher STOI score.

As the gold standard in speech quality assessment is human listening tests, and the STOI scores obtained in our evaluation lie in an area where the resolution of the STOI is worst, we also perform a MUSHRA [ITU01] listening test. We evaluate a set of 48 randomly chosen utterances (4 utterances from each session), using both the audible EMG and silent EMG sessions. For the audible EMG sessions, the actual reference audio is used as the reference, whereas for the silent EMG sessions, we use the (non-aligned) Block 0 audible references of the same prompt. Listeners were prompted to rate the utterances similarity to

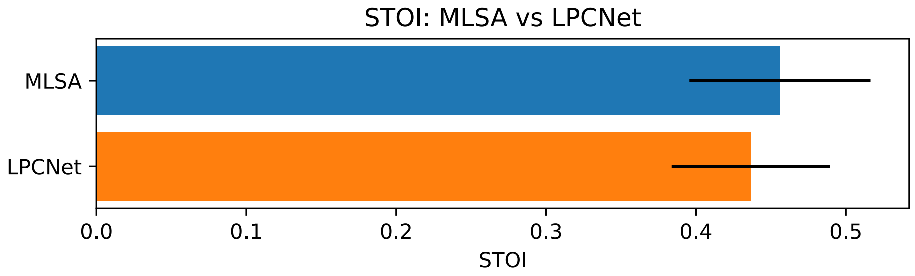


**Figure 6.3** – STOI evaluation of systems using MLSA and LPCNet based synthesis for EMG-to-Speech conversion using our low-latency system on the Block 1 evaluation set of the CSL-EMG_Array corpus. Error bars indicate standard deviation, higher is better.

the reference. As our anchor, we use a version of the reference low-pass filtered at 500 Hz. The result of this listening test (from 9 evaluators) can be seen in Figure 6.4. While for the silent sessions, there is no significant difference between MLSA and LPCNet-based synthesis, for the audible sessions (and for all sessions overall), the LPCNet synthesis is rated significantly higher by our evaluators than the MLSA synthesis.

### Feature Transformation

The feature transformation from EMG to Audio features is implemented by the `DNN` module. It uses the TensorFlow [AAB+15] framework to perform training of and inference using neural network models. The module supports the training of arbitrary TensorFlow models, either using standard backpropagation or model-agnostic meta learning training. Trained models can be saved and loaded, and the training of models can be resumed to adapt them using new data. The module also performs standardization (with parameters determined on the training data and saved along with the model) to ensure that in- and output feature dimensions are weighted equally in training.

We evaluate different neural network architectures using this system – this evaluation is presented in Section 6.2.

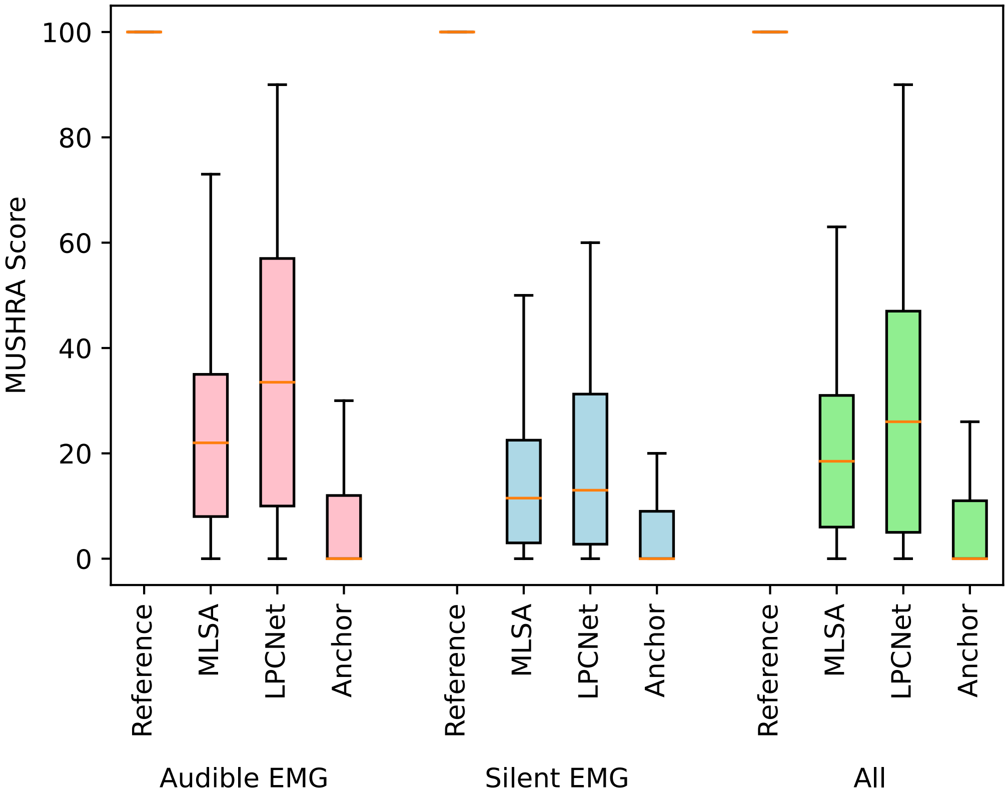


**Figure 6.4** – Results of a MUSHRA listening test comparison between MLSA and LPCNet based synthesis for EMG-to-Speech conversion using our low-latency system on the Block 1 evaluation set of the CSL-EMG_Array corpus audible and silent sessions as well as overall. Boxes indicate top/bottom quartile, whiskers indicate total range, horizontal bar indicates median. Higher is better.

## 6.2 Feature Transformation Approaches

Our EMG-to-Speech feature transformation is implemented using neural networks. The architecture of the neural network is critical for the system to function: It needs to be able to learn a good quality statistical mapping, be robust to noise, work under all conditions that we expect to be present in EMG data in a live experiment, and training and inference have to be efficient enough to be performed in a live evaluation setting.

We evaluate several different architectures: A basic feed-forward neural network, a convolutional neural network based EMG-to-Speech conversion (first

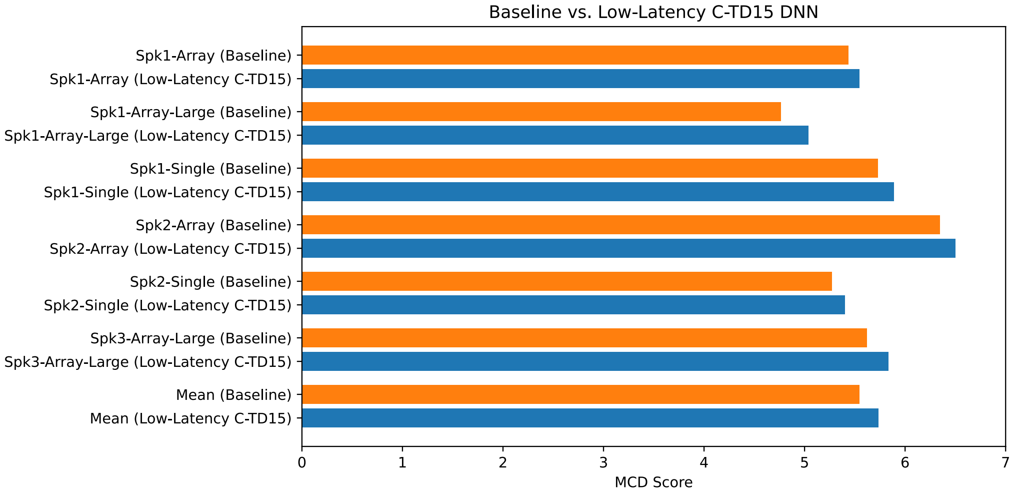


**Figure 6.5** – MCD score performance on the EMG-ArraySingle-A-500+ corpus: Baseline offline EMG-to-Speech conversion system using TD-15 features vs low-latency C-TD15 DNN based system.

presented in [DFAS18]) and EMG-to-Speech conversion with autoencoder-based speaker and content embeddings.

### 6.2.1 Feedforward Neural Networks (DNN)

The feedforward neural network architecture we use in low-latency EMG-to-Speech conversion is straightforward and adapted from the baseline system. The main difference to the baseline system is an overall lower input dimensionality (due to the use of C-TD15 features instead of TD15 features) and, when using LPCNet audio features, lower output dimensionality. Figure 6.5 shows an initial comparison of this system with the baseline (see Section 5.3) offline system performed on the EMG-ArraySingle-A-500+ corpus, using a frame size of 30 ms and MLSA synthesis (i.e. targeting MFCC features as the result of the feature transformation). It can be seen that the ability to perform low latency EMG-to-Speech conversion comes at the cost of significantly worse MCD scores (one-tailed t-test, p ¡ 0.05).

### 6.2.2 Convolutional Neural Networks (LeNet, Encoder-Decoder

Compared to feedforward networks, convolutional neural networks (see Section 2.5.1 for an introduction to such networks) have several properties that make them interesting for EMG-to-Speech conversion with array electrodes:

- Convolutional kernels are shift invariant. This may help reduce the influence of electrode position shifts.
- The lower number of parameters can improve training performance on low-data problems. Due to session dependence, EMG-to-Speech conversion is such a problem.

We evaluate convolutional neural networks for EMG-to-Speech feature transformation by comparing them to our baseline system, using TD15 features arranged in a 7x4 grid (dropping border channels) for the cheek array, resulting in an input tensor with dimensions $(7, 4, 155)$ and 7x1 for the chin array for an input tensor with dimensions $(7, 1, 155)$. As in the baseline system, the output layer is a 25 MFCC dimensions.

#### Network Architectures

We evaluate two different convolutional neural network architectures: One based on LeNet [LBBH98] and another based on an encoder-decoder structure [MSY16], both common structures for convolutional neural network. The hyper-parameters for the architectures presented were empirically tuned on the held out development data set. Both networks use average pooling instead of the more common max-pooling – we have found that average pooling seems to perform better, which we suspect is due to our task being a regression rather than a classification problem.

The structure of our LeNet-inspired network can be seen in Figure 6.6. It consists of a feature extractor part composed of three convolutional and two pooling layers and a regression part consisting of two fully connected layers. We train the network using the Adam optimizer with a learning rate of 0.003, $\beta_1$ of 0.9 and $\beta_2$ of 0.999.

Our encoder-decoder network employs a broadly similar structure, which can be seen in Figure 6.7 – a feature extractor, here subdivided into an encoder and decoder part, followed by a regression part. For training the

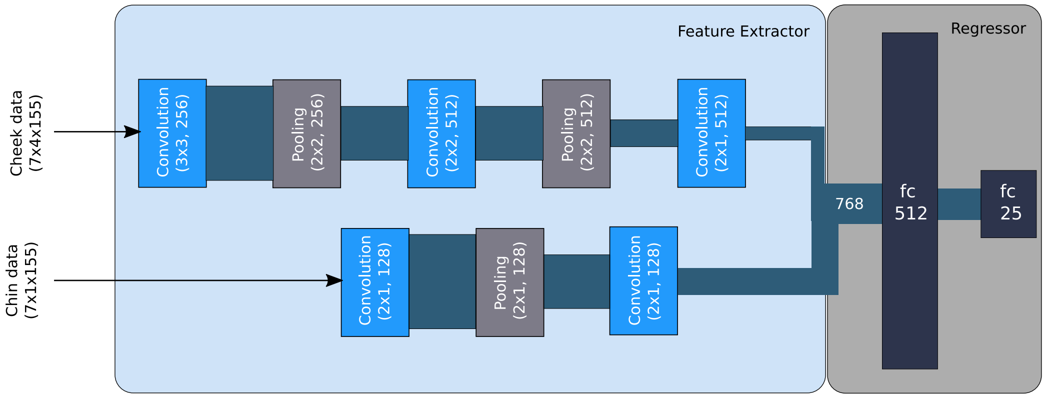


**Figure 6.6** – Architecture of the LeNet neural network used in our evaluations.

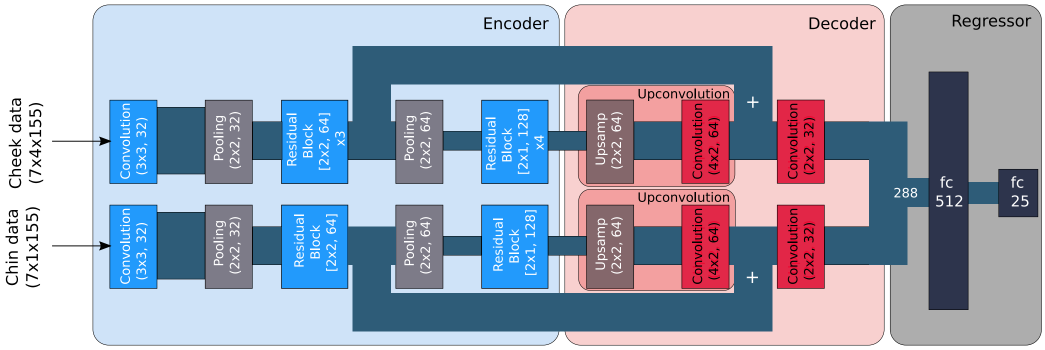


**Figure 6.7** – Architecture of the encoder-decoder neural network used in our evaluations.

encoder-decoder architecture an Adam optimizer with a learning rate of 0.002, $\beta_1$ of 0.9 and $\beta_2$ of 0.999 was used.

Both architectures were initialized using a He normal initializer [HZRS15]. The mean squared error was used as the loss function for parameter updates, and both networks were trained until the error on a held out validation set stopped decreasing.

### Evaluation

We compare the LeNet and Encoder-Decoder models both in a within-session as well as a cross-session (evaluating on a session held out from training) context, on the extended Speaker 1 data from the EMG-ArraySingle-A-500+ corpus, using the MCD score as the evaluation metric.

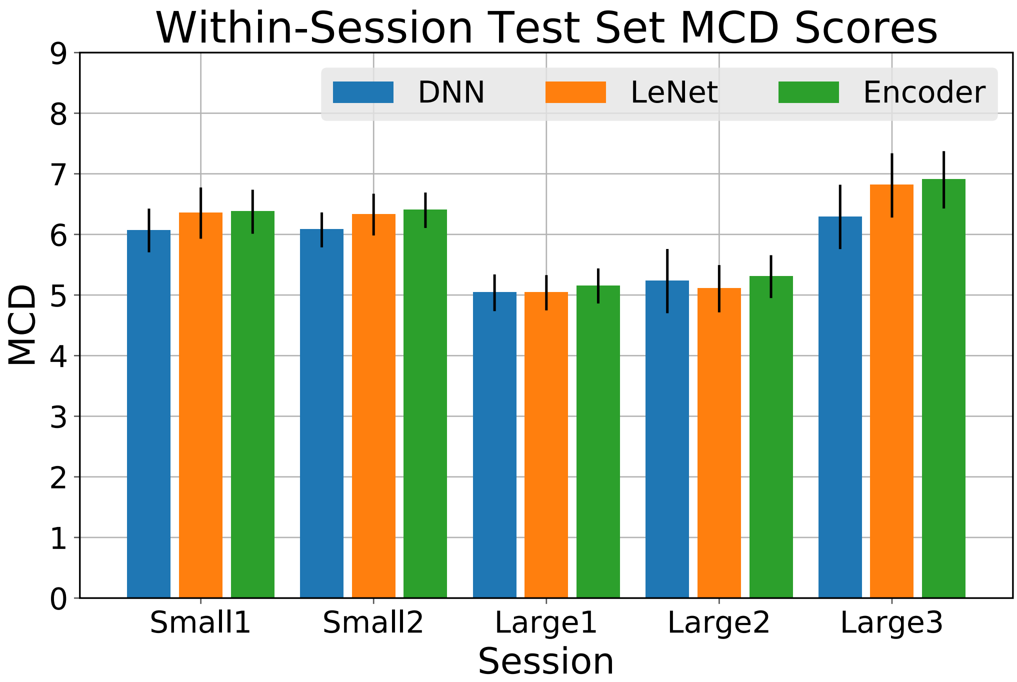


**Figure 6.8** – Within-Session Mel-Cepstral distortions for different CNN architectures.

To evaluate within-session performance, we train on the training set of one session and evaluate on the evaluation data of the same session. The results of this evaluation can be seen in Figure 6.8. It can be seen that for this evaluation mode, the convolutional architectures fail to outperform the Baseline DNN and perform significantly worse on most sessions (one sided dependent sample t-test, tested at $p < 0.05$, both models worse for Small1. Small2 and Large3, Encoder worse for Large1, other differences not significant). We suspect that this is because the CNNs strengths do not come into play in this evaluation mode: Within a session, there is no positional shift of the array, so the DNN highly tuned model generalizes to the relatively similar test set just as well as the CNN models.

For the cross-session evaluation, we train our networks on the training data of all but one session and then evaluate on the evaluation data of the held-out session. This means that, in effect, we are evaluating on unseen data from an unseen session. The results of this evaluation can be seen in Figure 6.9. While the encoder-decoder CNN still does not consistently outperform the DNN, the LeNet models performance is significantly better than that of the DNN (one sided dependent sample t-test, tested at $p < 0.05$).

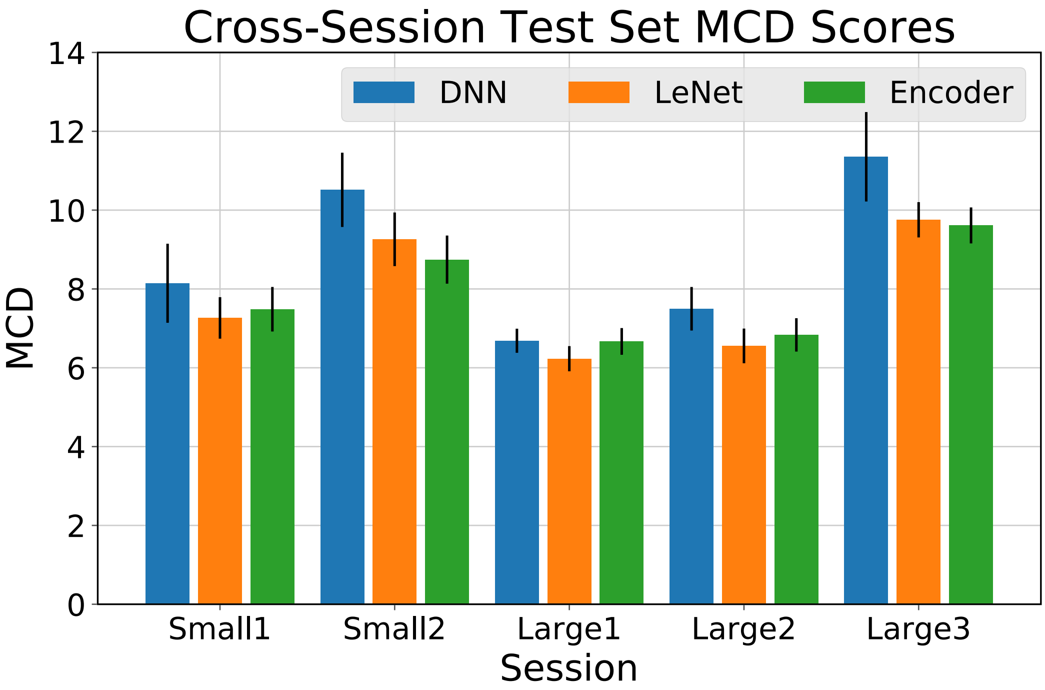


**Figure 6.9** – Cross-Session Mel-Cepstral distortions for different CNN architectures.

Note that even the LeNet model is still worse in the cross-session case than any within-session model – meaning that even with these improvements, the most promising approach for EMG-to-Speech conversion in general is still to record within-session training data.

### 6.2.3 Autoencoder-based speaker and content mappings

In Section 5.4.1, we have shown that it is possible to detect speaker identities from EMG data. One way in which this could be used to support EMG-to-Speech conversion is by training a model to factor EMG signals into speaker and content representations. To build such a model, we use an autoencoder – a model trained to map an input feature vector to itself – and train it to generate separate embeddings for speech content and speaker from audible speech. We then train a second model to generate content and speaker embeddings from EMG. One benefit of this approach is that it enables us to use large speech data sets (without an EMG signal) to train the output ("decoder") side of the network – using more data than we would ordinarily be able to collect for one EMG session. Since we require a large number of sessions, with the same, preferably array-based recording setup for evaluations, we evaluate this model using array sessions from the EMG-ArraySingle-A-500+ corpus (one session per speaker) combined with the audible EMG data from the EMG-Speakalong corpus (with two speakers from the EMG-Speakalong corpus – Speaker 921 and Speaker 923 – held out from all training and evaluation data unless otherwise mentioned). For training the speech encoder and decoder, we additionally mix this data with the data from the VCTK voice conversion corpus [YVM+19].

### Speech Autoencoder Structure and Training

The autoencoder neural network we use has a simple basic structure: The input features (MFCCs, MFCC deltas and $F_0$s) are first encoded into separate content and speaker embeddings by two separate encoder networks. These embeddings are then both used as inputs to a decoder network, that outputs $F_0$ values and MFCCs. We first train these networks using audible speech and then freeze the embeddings and decoder and re-train only the encoder networks to work from EMG data. Hyperparameters such as embedding and layer sizes were optimized on the development set. Figure 6.10 shows the overall structure of our autoencoder-based model.

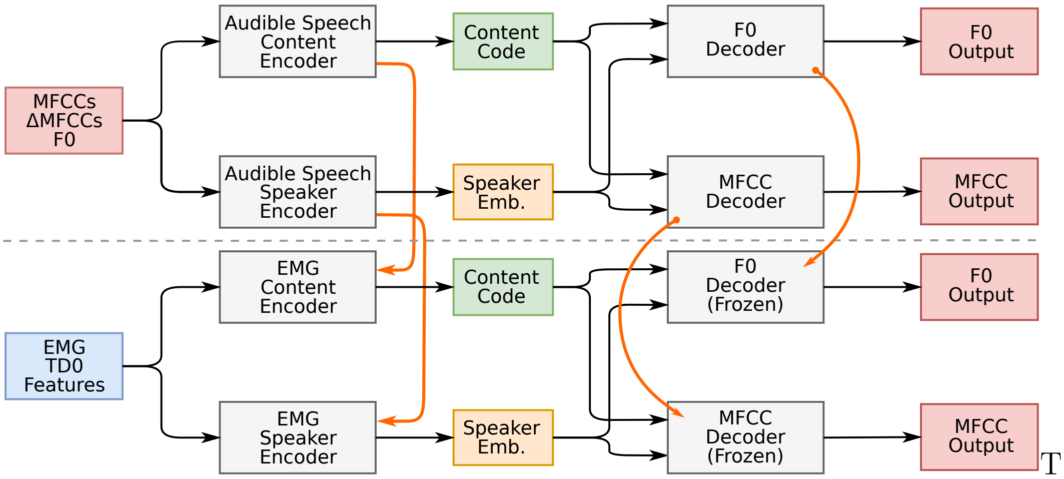


**Figure 6.10** – Structure of our autoencoder-based EMG-to-Speech conversion model: Audible speech autoencoder (above) and EMG-to-Speech feature transformation model trained using the pre-trained speech autoencoder (below). Orange arrows indicate how the audible speech autoencoder is used to train the EMG-to-Speech feature transformation model.

The training procedure begins by training the audible speech speaker encoder. The architecture of the encoder follows work by Wan et al. [WWPM18]. Input MFCC+Delta-MFCC+$F_0$-Frames are first passed through a three-hidden-layer LSTM. After processing one utterance of input data, the encoder produces the embedding by passing the final LSTM hidden state through a fully connected layer with linear activation and normalizing it to unit length. The dimensionality of the fully connected layer (and thus, the embedding) is 128, and the hidden layers use 384 LSTM cells each. The network is trained using the Generalized End-to-End loss [WWPM18]. This loss is computed by first calculating the per-speaker centroids of the embeddings (excluding the utterance itself when comparing it to the true speaker) for all utterances in a batch and then calculating the mean cosine distance of the embeddings to these centroids. The distances are then soft-maxed and the final generalized end to end loss is computed as the cross-entropy loss between the resulting likelihood vector and a one hot encoding of the true speaker of each utterance. This has the effect of pulling utterances of the same speaker towards each other while pushing them away from utterances of other speakers. The audible speech speaker encoder was trained on the VC using stochastic gradient descent with a learning rate of 0.02 and mini-batches containing 64 random slices of 32 consecutive MFCC+$F_0$ frames of 10 speakers each.

After the speaker encoder has been trained, the content encoder and speech decoder are trained jointly. The content encoder-decoder network broadly follows a vector-quantized variational autoencoder design, i.e. it is an autoencoder where the latent distribution is a set of discrete vectors. The encoder architecture is modeled after the networks used by van den Oord et al. [VDOV+17] and uses a 10 layer convolutional neural (with convolution over the time as well as input dimensions, using time slices of size 32) network with ReLU activation and skip connections – split into a convolutional stack with skip connections over the second, fourth and fifth layers, a residual stack with skip connections between the fifth and ninth as well as fifth and seventh as well as seventh and ninth layers, and finally a single convolutional layer with linear activation that downsamples the features to the dimensionality of the content embedding (64). The inner convolutional layers use 768 filters, while the embedding layer uses 64. As the final operation, these vectors are quantized using a codebook (The codebook size was determined using the development set – see the evaluation presented below) updated during training – the method used for this will be described below. The decoder network is split into an MFCC and an $F_0$ part. Each part takes the content and speaker embeddings and outputs MFCCs or $F_0$s. The architecture of the networks is similar to that of the content encoder: They are made from a main network that uses 9 convolutional layers with ReLU activation and skip connections over the second, third, fourth and fifth layers and a filter count of 1024 for all but the last layer, which uses linear activation that downsamples to the output dimensionality (25 for the MFCCs and 1 for the $F_0$s), followed by separate post-processing networks for MFCCs and $F_0$ that use five convolutional layers with 512 (MFCCs) or 128 ($F_0$s) filters before reducing back to the output dimensionality, tanh activation, batch normalization employed for every layer and a skip connection over the whole network.

To train the content encoder-decoder network, we use a loss comprised of multiple different parts: It is the sum of the mean squared error of the decoder network outputs before and after the post-processing networks and the commitment loss of the encoder network, calculated as the mean squared error of the quantization operation (weighted with a factor of 0.25). The codebook vectors are learned using exponential moving averaging: Each codebook vector is initialized randomly and then, whenever a pre-quantization feature vector is assigned to it set to a weighted average of its current value (weighted with a factor of 0.99) and the pre-quantization vector (weighted with a factor of 0.01). To prevent decoder model from overfitting on specific sequences of codebook vectors, we employ time jitter regularization [CWBvdO19]: With a probability of 0.12, each vector in the input sequence is replaced by either one of its neighbours. We train the joint encoder-decoder network using the Adam optimizer with a learning rate of 0.0001 and mini-batches of 64 slices of 32 consecutive speech frames each.

To verify that this approach is able to produce reasonable output, we evaluate the MCD score for different codebook sizes, using the development set of the VCTK and EMG-to-Speech audio data. We evaluate the MCD score, voiced section correlation, voicing error and TLAcc. The results of this evaluation can be found in Table 6.1. The best scores are obtained for a codebook size of 1024. They provide an upper bound for the performance the model could achieve on an EMG-to-Speech conversion task.

### Generating Embeddings from EMG

Given a trained speech autoencoder, we replace the encoder parts by encoders trained on EMG. The weights of the decoder networks are frozen during this training – they do not need to be retrained, since they have already been trained to work with the embeddings for the speakers present in our corpus. To do this, we use the parallel audio data of our EMG training

| | | $F_0$ | |
|---|---|---|---|
| Codebook size | MCD | Voicing error | TLAcc |
| 64 | 7.54 ± 0.45 | 14.85 ± 5.01 | 0.69 ± 0.09 |
| 128 | 5.45 ± 0.29 | 9.07 ± 3.96 | 0.75 ± 0.07 |
| 256 | 5.16 ± 0.27 | 7.36 ± 3.08 | 0.76 ± 0.07 |
| 512 | 5.46 ± 0.33 | 8.77 ± 3.36 | 0.76 ± 0.07 |
| 1024 | 4.74 ± 0.23 | 6.95 ± 3.21 | 0.78 ± 0.06 |
| 2048 | 5.25 ± 0.31 | 7.79 ± 3.10 | 0.77 ± 0.06 |

**Table 6.1** – Results of performing speech autoencoding with different codebook sizes in terms of mean and standard deviation over speakers of the MCD score (lower is better), $F_0$ voicing error (lower is better) and TLAcc (higher is better), on the VCTK and EMG-to-Speech data development sets.

data to generate speaker and content embeddings and then train networks to generate these embeddings from EMG TD features.

The architectures of the EMG encoder networks are similar to the audible speech encoder networks. For the EMG speaker encoder, we insert three convolutional layers (all using batch normalization and ReLU activation, with filter counts of 256 for the first two and 128 for the final convolutional layer) before the three-hidden-layer LSTM. The EMG content encoder uses four convolutional layers (with ReLU activation and 256 filters each) followed by three fully-connected layers (with 256 neurons and ReLU activation on the first two and 64 neurons and linear activation on the last). It uses skip connections over each layer other than the first and last, and uses batch normalization for every layer in addition to dropout regularization on the first two fully-connected layers. Both EMG encoder networks are trained to optimize the mean squared error between the EMG and (pre-trained, as described above) audible speech encoder network outputs, using an Adam optimizer with a learning rate of 0.0001 and mini-batches of 128 slices of 32 frames each.

### Evaluation

To evaluate the autoencoder model, we first perform EMG-to-Speech conversion, again with different codebook sizes, of the development set of our EMG-ArraySingle-A-500+ and EMG-Speakalong corpus development data to find the ideal codebook size. This evaluation can be found in Table 6.2, for

| | | $F_0$ | |
|---|---|---|---|
| Codebook size | MCD | Voicing error | TLAcc |
| 64 | 7.97 ± 0.69 | 28.05 ±10.69 | 0.70 ± 0.11 |
| 128 | 7.16 ± 0.64 | 27.26 ±10.65 | 0.71 ± 0.11 |
| 256 | 7.10 ± 0.64 | 26.78 ±10.70 | 0.71 ± 0.11 |
| 512 | 7.69 ± 0.70 | 27.09 ±10.70 | 0.69 ± 0.11 |
| 1024 | 7.62 ± 0.66 | 25.57 ±10.69 | 0.71 ± 0.11 |
| 2048 | 7.79 ± 0.63 | 24.50 ±10.43 | 0.71 ± 0.10 |

**Table 6.2** – Results of performing EMG-to-Speech conversion using our autoencoder-based model in terms of mean and standard deviation over speakers of the MCD score (lower is better), $F_0$ voicing error (lower is better) and TLAcc (higher is better), on the EMG-to-Speech data development set.

the same codebook sizes as before. We find that, for the EMG based system, the optimal codebook sizes is lower – 256 entries versus the audible speech autoencoder systems 1024 – and that quality overall decreases. An explanation for both these effects can be found in the trade-off between the accuracy of the content code and the difficulty of predicting the correct content code. As the codebook size increases, the quantization error decreases, but as there is now a larger number of content codes to pick from, predicting the correct one becomes more difficult.

We finally use the held-out evaluation data and perform EMG-to-Speech conversion using the model with the best-performing codebook size. We compare the results produced using this model to a multi-speaker baseline model – our baseline DNN model with an additional one-hot vector input for the speaker ID (including an extra value for unseen speakers), trained on the EMG-ArraySingle-A-500+ and EMG-Speakalong corpus training data with TD-15 features as input, using an Adam optimizer with a learning rate of 0.0001 and a mini-batch size of 1024. Figure 6.11 shows the results of performing EMG-to-Speech conversion using both the autoencoder-based as well as the baseline model. It can be seen that, while the autoencoder-based model is able to perform similarly to the baseline in terms of the $F_0$ metrics (differences not significant), the baseline model performs better in terms of MCD score. We also evaluate the ability of the model to convert data from new speakers. For this, we use the evaluation data of two speakers that, thus far, have been held out from all evaluations – speakers 921 and 923 from the EMG-Speakalong corpus. We obtain MCD scores of 7.68 (Speaker 921) and

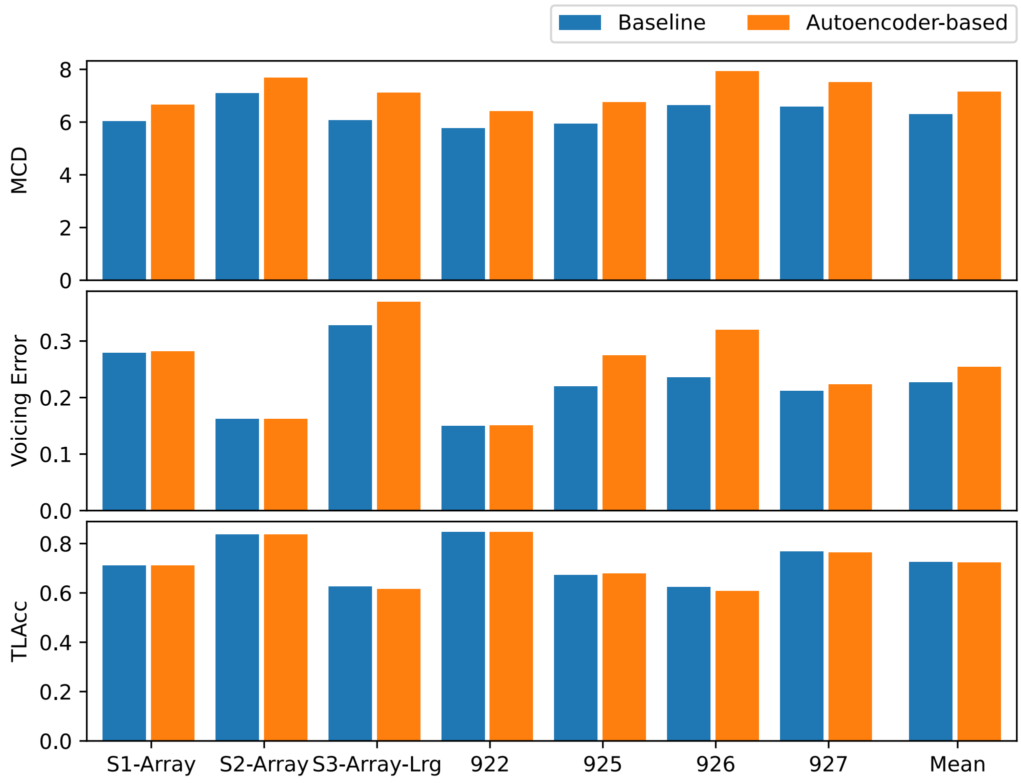


**Figure 6.11** – Comparison of our multi-speaker baseline and autoencoder-based model in terms of MCD score (lower is better), voicing error (lower is better) and TLAcc (higher is better).

7.85 (Speaker 923), compared to 6.94 (Speaker 921) and 7.04 (Speaker 923) for the baseline multi-speaker model on the same task.

In conclusion, we must note that models with more complex architectures that are able to incorporate more data, while promising, are currently not able to deliver state of the art performance in EMG-to-Speech conversion. Whether this would change if much more parallel EMG and audio data was available for model training remains to be investigated in future work.

## 6.3 System Adaptation

To be able to build EMG-to-Speech conversion systems with less data and time, it would be preferable to adapt a multi-session, multi-speaker base system to a new speaker instead of training a new system from scratch.

Additionally, it could be useful to adapt a system within one single session to make it able to perform better as the signal shifts over time due to changes in speaker and electrode conditions. We evaluate two adaptation methods for their potential in EMG-to-Speech conversion: Basic re-training adaptation, and model-agnostic meta learning based adaptation.

### 6.3.1 Retraining Adaptation

The procedure for retraining adaptation of neural network models is simple: Given a pre-trained model and new data that we wish to use to adapt this model, we simply perform gradient update steps using the existing model weights as initialization and our adaptation data as training data until the model has sufficiently re-converged, effectively fine-tuning it with new data. This process bears the risk of *catastrophic forgetting* – when a network is trained on one task and then adapted to perform another, it may forget how to perform the first task. In our case, since we only want our networks to perform well on data similar to the adaptation data, this is acceptable.

### 6.3.2 Model-Agnostic Meta Learning

*Model-Agnostic Meta Learning* (MAML) [FAL17] is a procedure for training neural networks in a way that makes them a good basis for retraining adaptation (as described above). The MAML method treats different tasks from the family of tasks that the model should later be adapted to as examples with which to train the weights of a model in such a way that very few gradient update steps lead to good performance on different domains. In each MAML optimization step, the method samples a number of tasks (in our case, sessions) and performs a back-propagation gradient update of a copy of the current model using a number of training examples from this task (effectively, performing one step of adaptation for this task). The MAML training algorithm then uses the model copies on which the gradient update has been performed to update the original model using a meta-loss: It updates the weights such that the loss of the adapted models becomes minimal. In this way, it directly trains the model so that it can be optimally adapted with few steps. The actual adaptation is then performed the same way as before: The model is simply re-trained with training examples from the new session.

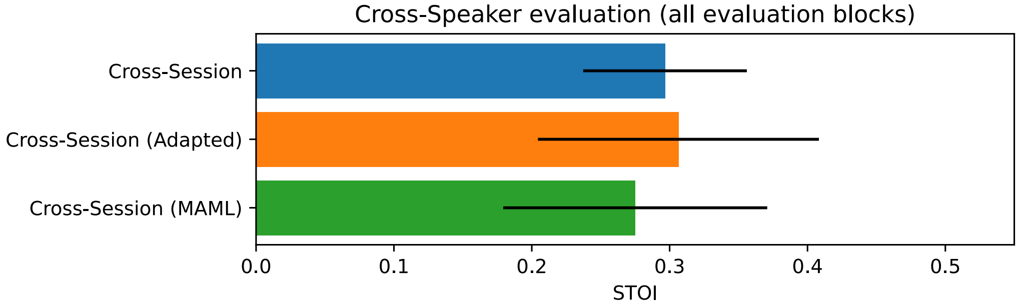


**Figure 6.12** – STOI results of performing cross-speaker evaluation of EMG-to-Speech conversion using our low-latency system. Error bars indicate standard deviation, higher is better.

### 6.3.3 Evaluation

While previous work has considered adaptation to a new session from previous sessions [PWSS19], we evaluate the presented adaptation approaches for the two modes of adaptation that are relevant for the study presented in Chapter 7: Adapting a system trained on multiple sessions from different speakers to an unseen session from a new speaker, and adapting a system trained on training data of one session to data recorded some time later in that same session, compensating for signal changes.

**Adapting across speakers** To gauge the performance of our system when trying to adapt to a new speaker, we train systems on data from all but one speaker from the audible EMG set of the CSL-EMG_Array corpus. The results of this evaluation, in terms of the STOI, can be seen in Figure 6.12. The mean STOI for the system adapted with basic retraining adaptation is slightly (though not significantly, using a two-tailed t-test and a significance level of p ¡ 0.05) better than that for the basic system, while the average STOI for the MAML-adapted system is slightly (though again, not significantly) lower. A possible reason why the MAML-based adaptation is unable to improve on basic adaptation could be that it only works well when there are many different tasks to sample from – which is not the case for our problem (only 8 speakers with one session each).

**Adapting within a session** In addition to this, we also test our adaptation on within-session data, again using the audible sessions from the CSL-EMG_Array corpus. We adapt a system trained on the Block 1 training

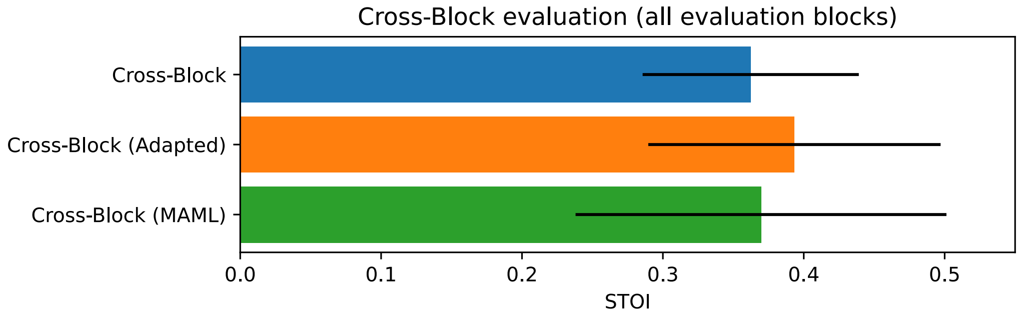


**Figure 6.13** – STOI results of performing cross-block evaluation of EMG-to-Speech conversion using our low-latency system. Error bars indicate standard deviation, higher is better.

data of a session with the same sessions adaptation data (Blocks 2 / 4 / 6) and evaluate on the eval data from the evaluation blocks of the session that follow each adaptation block (Blocks 3 / 5 / 7). Figure 6.13 shows the results of this cross-block adaptation evaluation. While both adapted systems obtain better average scores than the non-adapted systems, the differences are once again not statistically significant (using a two-tailed t-test and a significance level of p ¡ 0.05), and the mean STOI for the system adapted with basic retraining is still better than the system using MAML based adaptation.

### 6.3.4 Discussion

We have evaluated two adaptation methods – retraining adaptation and MAML-based adaptation – for two modes of adaptation – adapting to a new session, and adapting within a session. The results are, overall, weak: The differences we find do not rise to a statistically significant level. We can speculate about the causes for this. For cross-speaker adaptation, a likely reason is the difficulty of the task compared to the relatively low amount of data. While cross-speaker adaptation in neural network based audible speech processing is typically performed using background models trained on multiple hundreds of hours of speech by thousands of different speakers, our evaluation used much less. The low speaker count may especially impact MAML-based adaptation: Training a model that can be adapted to a new task well is difficult if there is only a limited number of tasks to sample from in training. The adaptation within a session, compensating for time-related differences within a session, seems to have worked relatively better, though here, too, basic retraining adaptation results in a better mean STOI. Overall, adaptation seems promising, though larger corpora with more speaker variety may be needed to fully take advantage of it.

Chapter 7

# User-in-the-Loop Study

*This chapter describes the EMG-to-Speech live feedback study performed as the culmination of the system design and experiments performed in this thesis. It describes and motivates the parameters and design of the study and discusses its results and the implications these results have for future EMG-to-Speech conversion systems.*

## 7.1 Study Setup

To, finally, evaluate the effects of live audible feedback on the performance of EMG-to-Speech conversion, we perform a live feedback experiment where the user is put in a feedback loop with the system: The user speaks silently, and the silent EMG signals are converted into audible speech, potentially allowing the user to adjust their articulation depending on system output. We evaluate two different types of feedback: Simple feedback consisting of speech-power related speech-like buzz noise (*Simple* system, prioritizing generating very accurate system output) and complex feedback consisting of speech (*Complex* system, prioritizing generating speech-like output).

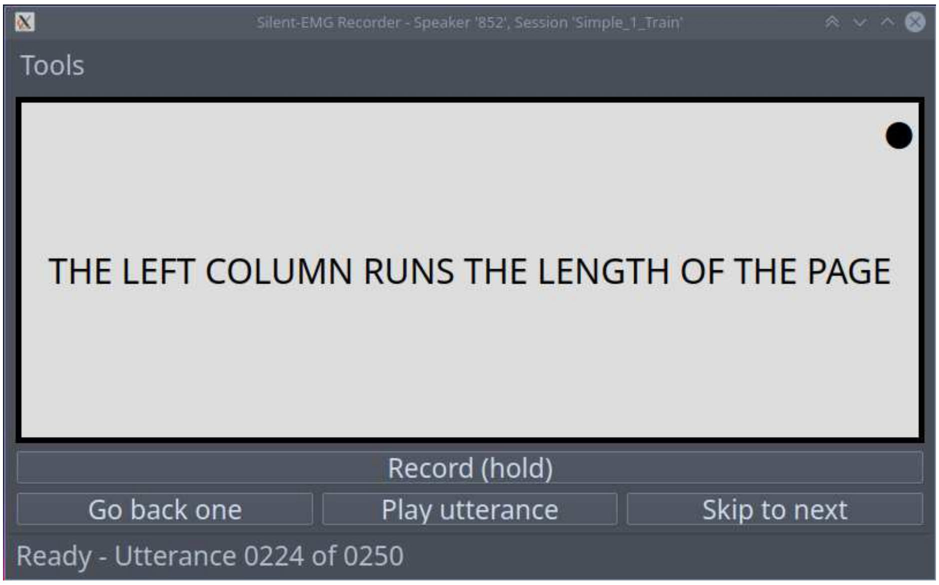
### 7.1.1 Hardware and Electrodes

As we are evaluating a practical setup that could eventually be used by non-expert users without expert assistance, we use the array-based electrode montage (4x8 cheek array, 1x8 chin array, chained differential amplification) described in detail in Section 4.1.2 for our study. For recording, we use an OT Bioelletronica Quattrocento multi-channel EMG amplifier with a 10 Hz high-pass for DC offset removal and a 500 Hz low-pass filter anti-aliasing filter, sampling at 2048 Hz at a bit depth of 16 bit at an amplification factor of 150 V/V. Audio is recorded at 16 kHz using a RODE NT-1 condenser microphone and a Behringer Xenyx 302 audio interface, and the EMG and audio signals are synchronized using a marker channel.

### 7.1.2 Software

Both the Simple and Complex system are built using our pipeline framework – for an overview of the configuration used, refer back to Figure 6.2. It uses C-TD15 features with a frame size of 32 ms, running EMG normalization and a feedforward neural network mapping with the structure described in Section 6.2.1. For the Simple system, the final layer of that network is replaced with a soft-max layer and the network is trained to optimize the categorical cross-entropy of the frame based speech power quantized into three levels (As described in Section 6.1.3). In addition, for the simple feedback the synthesis module is replaced with the SimpleSynth module. Neural network training is ran for 350 iterations, and adaptation is ran for 50 iterations, both using stochastic gradient descent with a learning rate of 0.01 and using momentum, as described in Section 2.5.1. Feedback audio is output via the recording computers integrated sound card, and can be recorded back via the audio interface.

The front-end for our experiments is built using the Qt framework. It consists of a simple GUI leading through the stages of the experiment and a recording window. The recording window (see Figure 7.1) contains a label displaying the utterance prompt, a recording push-to-talk button, controls for going forward and backward by one utterance. In addition to this, the user interface includes a signal review window which shows the EMG and audio signals that are being received from the microphone and EMG amplifier at the current moment in time.



Silent-EMG Recorder - Speaker '852', Session 'Simple_1_Train'

THE LEFT COLUMN RUNS THE LENGTH OF THE PAGE

Ready - Utterance 0224 of 0250

**Figure 7.1** – Recording window of our EMG-to-Speech GUI.

### 7.1.3 Experiment Structure

The design of our experiment is designed to let us evaluate the effect of feedback (i.e. potential user-to-system adaptation) both with and without system-to-user adaptation. The experiment consists of 10 steps: 8 recording steps, one training step and one adaptation step (both ran in parallel to recordings). Figure 7.2 shows an overview of the experiment procedure. The steps are as follows:

**Step 1: Record training data -** The first step of an experiment session in our study is recording training data to later train an EMG-to-Speech feature transformation neural network. We record 250 utterances (english language, broadcast news domain, subset of the prompts for the training set of the EMG-ArraySingle-A-500+ corpus) of audibly spoken parallel EMG and Audio data. EMG and audio features (either LPCNet or quantized frame-based speech power) are extracted in parallel while data is recorded so that training can begin immediately after recording has finished.

**Step 2.1: System training -** After recording the training data, we start training an EMG-to-Speech feature transformation neural network. The network is trained for 350 iterations in parallel with the next three recording blocks (2.2, 2.3 and 2.4). The training runs in parallel to all steps labeled 2.x, and is required to finish before step 3 can begin.

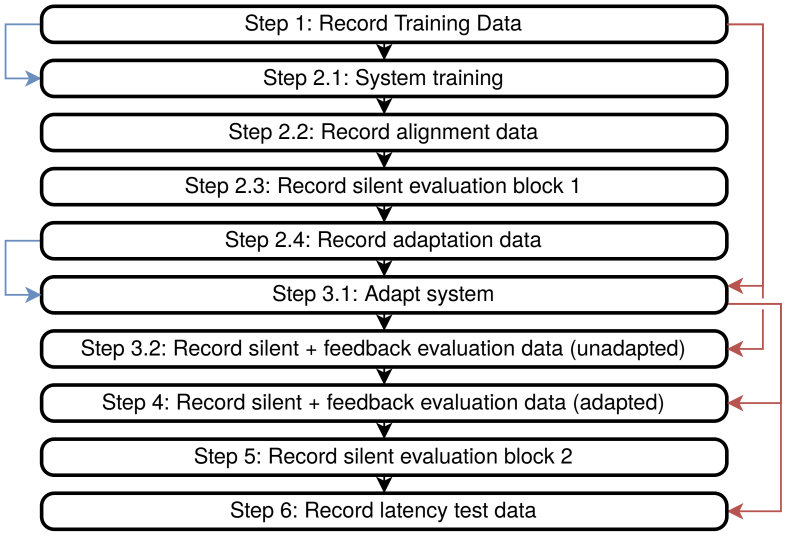

Step 1: Record Training Data
Step 2.1: System training
Step 2.2: Record alignment data
Step 2.3: Record silent evaluation block 1
Step 2.4: Record adaptation data
Step 3.1: Adapt system
Step 3.2: Record silent + feedback evaluation data (unadapted)
Step 4: Record silent + feedback evaluation data (adapted)
Step 5: Record silent evaluation block 2
Step 6: Record latency test data

**Figure 7.2** – Procedure of our user in the loop study. Blue arrows (left) indicate data recorded in a step is being used in another step. Red arrows (right) indicate system trained in a step is required for another step.

**Step 2.2: Record alignment data -** Next, we perform an audible recording of the 40 utterance evaluation set (using the same prompts as the MG-ArraySingle-A-500+ corpus). This data will be used during evaluation for calculating the DTW-MCD score.

**Step 2.3: Record silent evaluation block 1 -** This block is the initial evaluation block ("Initial (No Feedback)"). It is used to establish baseline EMG-to-Speech conversion performance on silent speech. No feedback is provided during the recording.

**Step 2.4: Record adaptation data -** This block is, once again, audible EMG data, same as the training data before. It is used as adaptation data for the feedback system.

**Step 3.1: Adapt system -** Immediately after recording the adaptation data, it is used to adapt the feedback system (adaptation for 60 iterations), in parallel with step 3.2. The training runs in parallel to all steps labeled 3.x, and is required to finish before step 4 can begin.

**Step 3.2: Record silent + feedback evaluation data (unadapted) -** The system trained in step 2.1 is used to perform a recording of silent speech EMG data with audible feedback ("Feedback").

**Step 4: Record silent + feedback evaluation data (adapted) -** The system trained in step 2.1 is used to perform a recording of silent speech EMG data with audible feedback, using an adapted system ("Feedback (Adapted)").

**Step 5: Record silent evaluation block 2 -** The final evaluation block is once again recorded as silent speech EMG with no feedback. Its purpose is to test whether the study participant was able to adapt their behaviour to the system to produce better output and sustain that adaptation.

**Step 6: Record latency test data -** We finally record 5 utterances of audible speech from the training set in the left audio channel and adapted feedback system output in the right audio channel. This data can be used to determine the effective latency of our system by finding the shift that gives the maximum correlation between the two channels.

For the silent speech EMG blocks without feedback, the participant is instructed to speak as naturally as possible, while for the blocks with feedback, the participant is encouraged to use the feedback to adjust their behaviour to produce output that better matches their speech production. After the conclusion of the recordings, participants were asked to fill a self-assessment questionnaire.

### 7.1.4 Participants

Our study contains data from a total of 6 participants (4 male, 2 female) aged 24 to 56 years. Participants were recorded on a voluntary basis and were not compensated for participation in the study. One participant (Participant 2) participated in both a recording with simple feedback and complex (full speech) feedback (performed on separate days), one (Participant 6) participated in only a simple feedback recording, and the remaining participants participated only in a complex (full speech) feedback recording. During the recording with Participant 5, technical issues resulted in no feedback being emitted during the adapted-system feedback block – this block has therefore been excluded from the analysis.

## 7.2 Results

Having concluded the study, we can now evaluate what the effects of our feedback on speech production were, if any. If there was a positive effect, we would expect a lower DTW-MCD score (indicating higher quality) for when performing EMG-to-Speech conversion for recordings with feedback compared to performing EMG-to-Speech conversion for recordings without feedback. We would additionally expect the EMG signal of recordings where feedback was present to be more similar to sessions where feedback was not present. For silent evaluation block 2, we would expect to see these changes being sustained to some extent, if a learning effect was present.

### 7.2.1 DTW-MCD

To evaluate our system, we use the DTW-MCD (as described in Section 5.2.1) score, computed between the EMG-to-Speech output of the block to be evaluated and the alignment audible speech data from Step 2.2. For the blocks where speech feedback output was generated, we use the feedback systems output. For blocks with no feedback, we use the unadapted (Initial block) and adapted (Final block) feedback system to generate audio output after the recording has concluded to be able to compare the performance on non-feedback blocks with the performance on feedback blocks. Similarly, to evaluate the performance on recordings using the simple feedback, we train systems as if complex speech feedback had been used and use the output of these systems for our evaluation.

Figure 7.3 shows the DTW-MCD scores obtained for the simple feedback recordings, while Figure 7.4 shows the results for complex (speech) feedback recordings.

For the simple feedback, there is a significant improvement of participant 2s performance between the initial silent recording, and the recording with feedback, though the same does not hold for the final recording and the adapted feedback block. For participant 6, there is no significant difference between feedback and non-feedback blocks using the same system (all tests: related sample t-test, $p < 0.05$, corrected for multiple comparisons).

For the complex feedback recordings, a similar picture emerges for participant 1 and 2: The performance on feedback blocks is significantly improved for both the adapted and unadapted system for participant 1, and for the adapted system feedback for participant 2. For the remaining participants, feedback

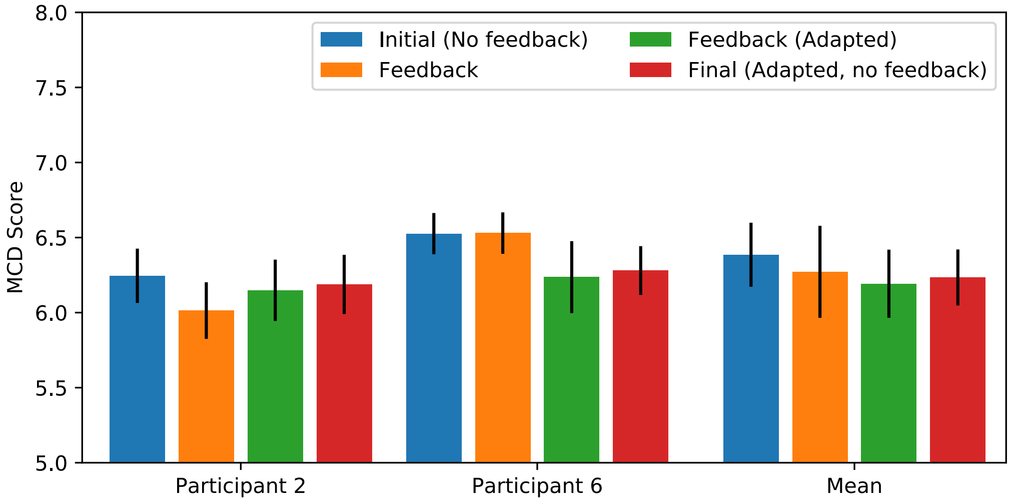


**Figure 7.3** – DTW-MCD scores for EMG-to-Speech conversion performed on the recordings using simple feedback. Lower is better, error bars indicate standard deviation.

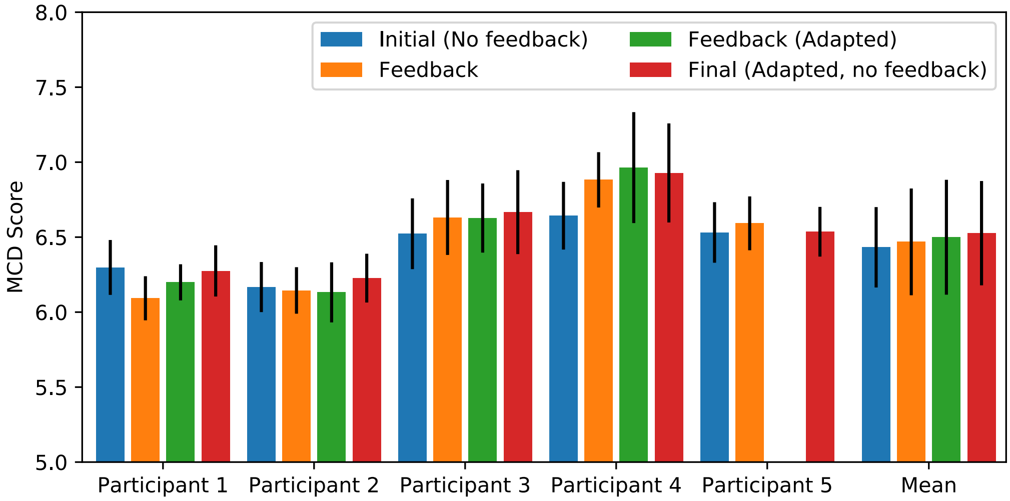


**Figure 7.4** – DTW-MCD scores for EMG-to-Speech conversion performed on the recordings using complex (speech) feedback. Lower is better, error bars indicate standard deviation.

block performance did either not differ significantly from the associated non-feedback block, or was significantly worse (unadapted systems for participants 3 and 4, all tests: related sample t-test, $p < 0.05$, corrected for multiple comparisons). We can also see that there are large differences between participants: The scores for participant 1 and 2 are, across all blocks, much lower than those for the remaining participants.

### 7.2.2 Power Spectral Density

We compare the EMG signals recorded during different steps of our study by calculating the mean *power spectral density* (PSD) for each speaker and step using Welchs method [Wel67]. The results of this evaluation can be seen in Figure 7.5.

For participants 1, 2 and 4, we see a large difference between the overall energy in the training data versus all other blocks. Interestingly, for these participants, the audible EMG adaptation data recorded in step 2.4 shows a relatively low overall energy. A possible explanation for this is fatigue causing participants to articulate with less effort – as in the PSD evaluation of the CSL-EMG-Speak-Along Corpus (see Section 4.3.2), we compare halves of the audible EMG from step 1 and find that the energy decreases from the first half of the audible EMG training data to the second half. For the remaining two speakers, we see a different picture. The EMG PSD of participant 3 is much more consistent than that of the remaining speakers. This may be due to a conscious effort by the speaker, who reported on the self-assessment questionnaire (for details on the questionnaire, see subsection 7.2.3) to have made a strong effort to get “good feedback from the system”. Finally, for participant 5, who reported being an experienced speaker and who has had previous experience producing silent speech, the PSD for audible EMG from step 1 and silent EMG from step 2.3 seem similar as well. Unfortunately, the similarity in PSD did not translate to better EMG-to-Speech performance for either participant 3 or 5. Additionally, the overall energy of the adaptation session is much higher. Though the origin of this difference is not clear (there are no obvious artifacts present in the EMG signal), the difference in EMG could help explain the difficulties in obtaining a usable adapted system for participant 5.

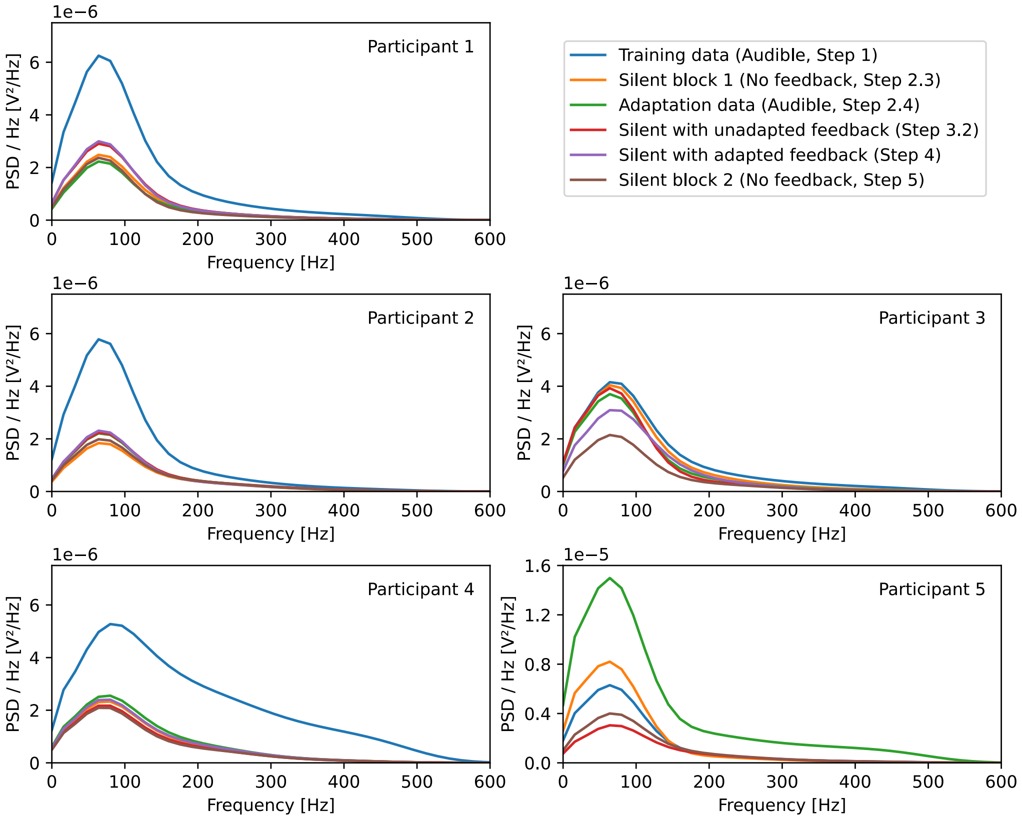


**Figure 7.5** – PSDs of the EMG signals of user-in-the-loop study participants, separated by recording step.

### 7.2.3 Self-Assessment

To assess the participants subjective experience using our system, we use a self-assessment survey presented to participants after the conclusion of a recording session. Participants were asked to rate their agreement with several statements about the session on a 5 point Likert scale, with 1 point being low agreement ("disagree completely") and 5 points being full agreement ("agree completely"). Figure 7.6 shows a summary overview of the questions and responses. The statements fall into four categories: Output quality, control over the system, latency of the output and user experience.

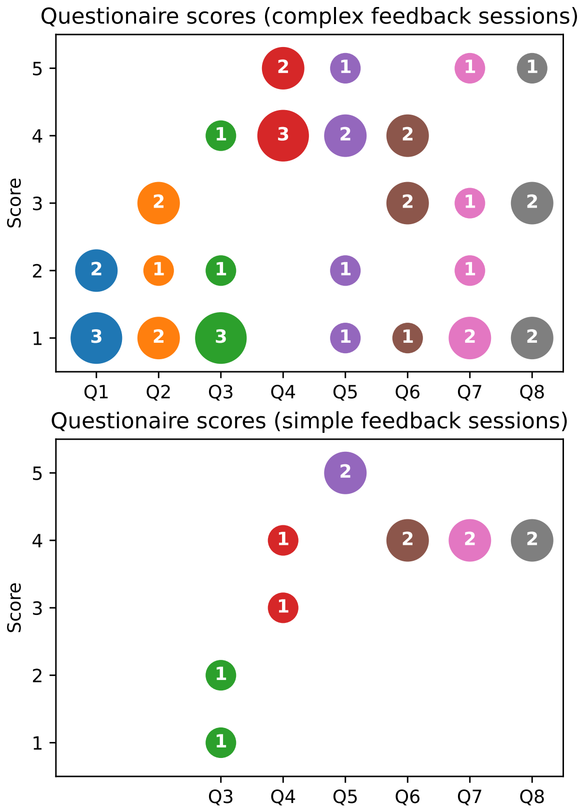
**Output quality:** To evaluate how output quality of the complex feedback system was perceived, we asked participants using the complex feedback system whether they thought the output was intelligible (**Q1**, Mean agreement: 1.4 points) and whether it matched what they had said

(**Q2**, Mean agreement: 2.75 points). Interestingly, Participant 1 and 2, who had both better scores overall and significant improvements with feedback, gave higher scores on both questions (Mean agreement of 2 for the first and 3 for the second question) than the other participants (Mean agreement of 1 for the first and 1.33 for the second question, respectively).

**Control:** Independent of speech quality, participants were asked to rate whether they felt that they could control the system well (**Q3**). For the systems using simple feedback, the mean agreement was 3.5 points, whereas for the complex system, it was 2.2 points. Here, we once again, we find a large difference between participants 1 and 2 (Mean agreement of 3.5) and the other participants (Mean agreement of 1.33 points).

**Latency:** Participants were asked whether they felt that the system output was generated without delay (**Q4**). The mean agreement with this statement was high at 4.4 points for the complex feedback and 5 points for the simple feedback. We additionally perform a signal-based estimation of the lag. Since we need a good match between system output and produced speech if we want to avoid noise dominating our estimate, we use the data recorded in step 6 from the participants for which conversion yielded good quality output – Participants 1 and 2. We calculate the offset that maximizes a correlation of the audio envelopes of the original audible signal and converted signal power (calculated with a 1 ms frame shift), resulting in an estimated median latency of ~15.6 ms for Participant 1 and ~13.1 ms for Participant 2 (Overall median 15.5 ms). While it should be noted that the quality of the results introduces large amounts of jitter into our latency estimate (estimated latencies vary between ~11.0 ms and 40.6 ms), the result matches our users perception of the output having little to no delay. Additionally, even the high estimate of 40.6 ms is still below the limit of 50 ms after which delayed feedback would start to cause an increased rate of disfluencies [SKRL02].

**User experience:** Participants were asked to rate whether they felt that the feedback made silent speaking easier (**Q5**, Mean agreement: 3.2 points on complex feedback, 4 points on simple), whether it made silent speaking more pleasant (**Q6**, Mean agreement: 3 points complex feedback, 4 points simple), whether they thought it improved their ability to speak silently (**Q7**, Mean agreement: 2.4 points complex feedback, 4 simple), and whether they thought they had learned to use the system better during use (**Q8**, Mean agreement 2.6 points complex



Users were asked to rate their agreement with the following statements from 1 (completely disagree) to 5 (completely agree):

**Q1** Speech output was intelligible.

**Q2** Speech output matched what I said.

**Q3** System output was easy to control.

**Q4** System output happened without latency.

**Q5** Feedback made silent speaking easier.

**Q6** Feedback made silent more pleasant.

**Q7** Feedback improved my ability to speak silently.

**Q8** I learned to use the system better during the recording.

**Figure 7.6** – Bubble strip plot of responses to our user questionnaire. Area of bubble and number in bubble indicate number of responses. Questions 1 and 2 omitted for simple feedback system (not applicable).

feedback, 4 points simple). Overall, we again see higher agreement for better performing sessions.

## 7.3 Discussion

Overall, the result that is most clear from our study is that EMG-to-Speech conversion under conditions approaching "real-world" usage is still extremely inconsistent. While we were able to train systems that worked well for three of the recording sessions, the results for the three other recording sessions were not satisfactory. This may be in part due to differences between speakers, but also due to the large variation in signals and signal quality that should be expected when recording EMG data in an out of the lab setting. Even for the better sessions, the speech feedback output, while broadly matching the participants speech intentions, was not intelligible speech. However, encouragingly, for the three participants with better results, both the DTW-MCD score and the subjective evaluation show feedback – both the simple and complex versions – improving performance. Additionally, these participants also felt that their control of the system improved through use – however, we were not able to verify this using our objective evaluation. There are several possible explanations for this: It is possible that the DTW being an imperfect evaluation metric smooths over an actual effect that may or may not emerge on a much larger sample, that artifacts or shifts in the signal in the later blocks counteracts the learning effect, or that despite users believing that they improved over time, there is no effect.

Finally, latency is clearly low enough that output was perceived as having very little or no delay by our participants, and our signal-based evaluation provides additional evidence that our system is able to output data with a latency well below 50 ms.

Chapter 8

# Conclusion

*This chapter summarizes the work shown in this dissertation. It presents the key results and takeaways and provides an outlook on potential future applications and avenues for further research.*

## 8.1 Summary of Results

In this dissertation, we've presented different approaches to advancing EMG-to-Speech conversion closer to practical, out of the lab settings. We have specifically established for the first time solutions to the following problems:

**Real-time low-latency EMG-to-Speech conversion:** We have built a system that allows for very low latency EMG-to-Speech conversion. The system is based on a new framework that allows us to flexibly construct EMG-to-Speech conversion systems using different modules for feature extraction, feature transformation and synthesis, which allows for flexible iteration and experimentation. The system is built to take advantage of multi-processing to allow for high throughput even when some operations are computationally expensive, and flexible enough to easily allow for the real-time processing of biosignals other than speech EMG. We have implemented and evaluated modules for this system that allow for very low latency EMG-to-Speech conversion, including a new causal EMG feature set (C-TD15 features). We've also introduced a method for EMG signal normalization that greatly lessens the impact of time-correlated drift in the makeup of the signal.

**Online EMG-to-Speech conversion in practice:** We've performed evaluations to determine good hyperparameters for online EMG-to-Speech conversion feature transformation models. To this end, we have evaluated many new conversion approaches, including methods based on convolutional neural networks and autoencoders as well as neural network adaptation with and without model-agnostic meta learning. We have recorded several new data corpora (CSL-EMG-Words-CVVC, EMG-Speakalong, CSL-EMG_Array) to better evaluate different aspects of EMG-to-Speech conversion, including one corpus (The CSL-EMG_Array corpus) that allows for a realistic evaluation of online EMG-to-Speech conversion performance in practice and of within-session as well as between-session adaptation. We have made this corpus as well as our evaluations available as an open access data corpus to foster further innovation in the field.

**Neural vocoding for EMG-to-Speech conversion:** We have evaluated different means of representing audio and performing synthesis in the context of EMG-to-Speech conversion, including a real-time low-latency capable neural vocoder (LPCNet). In doing so, we have found that while there was no significant improvement in terms of STOI, human listeners express a clear preference for the LPCNet vocoder compared to MLSA. We also implemented and tested a simple frame-power-based method that only generates buzzing feedback, but presents an easier machine learning problem.

**New evaluation metric:** We've introduced the TLAcc a method to better compare $F_0$ trajectories of two pieces of speech for similarity, and evaluated this method using human listening tests, finding that it showed better correlation with human ratings than other common $F_0$ evaluation measures.

**Silent EMG vs. audible EMG and the effect of feedback:** We have evaluated different methods for improving the performance of EMG-to-Speech conversion on silent speech EMG data. We've evaluated a speak-along approach to generating parallel EMG and Audio data that can be used for system training and evaluation. We've found this approach to be viable for training systems for speakers that cannot produce audible speech anymore. Finally, we have performed a live feedback study, using both simplified as well as complex (full speech) to determine what the effect, if any, of audible feedback on EMG-to-Speech.

While we were unable to confirm the existence of a learning effect that persists past the presence of feedback, we have found limited evidence in support of the hypothesis that the presence of feedback improves EMG-to-Speech conversion quality – but only for participants for whom an EMG-to-Speech conversion system was able to produce output that the participants could control well.

## 8.2 Outlook on Potential Future Research

EMG-to-Speech conversion remains a difficult and elusive research problem. While we have succeeded in challenging work towards building more practical EMG-to-Speech conversion systems, there still remains much to be done, and many of our results suggest that it might be best reconsider some of the decisions made earlier.

**Carefully constructed electrode montages:** Our feedback study has shown that consistency is a large problem especially in an online setting. While array EMG electrodes are superior to single electrode montages in convenience, in practice, they often have channel detachments during recording, more variability in position relative to muscles between speakers, and use smaller electrode surfaces. Approaches that use carefully chosen electrode positions, with electrodes possibly integrated into a mask harness, might be able to deliver much more consistent results.

**Large data sets and multi-speaker systems:** In many fields that work with machine learning, the greatest improvements have not come from building better algorithms, but from simply giving clever enough algorithms vast amounts of data to work with. Both the autoencoder based method we evaluated for building multi-speaker systems as well as the model agnostic meta-learning method we have evaluated for building speaker-adaptive systems generally perform best when a very large and varied training set is available. It may be worth performing a large number of recordings with many different speakers to see if this holds true for our problem – and with the CSL-EMG_Array recording procedure we have laid groundwork that could inform such recordings.

**Improvements in recording technology:** EMG recording technology continues to improve. While all our recordings were performed using a desktop EMG amplifier, portable amplifiers that may be capable enough to be used in EMG-to-Speech conversion have now become commercially available. The utility of such devices in EMG-to-Speech conversion should be thoroughly investigated. Another avenue for hardware improvement might be to use EMG together with other signal modalities such as ultrasound, microwave radar or permanent magnet articulography to capture a broader view of the speech production process.

## 8.3 Closing Remarks

While we have made EMG-to-Speech conversion capable of running in real time and with low latency for the first time – opening up new avenues for research – much work remains to be done before practical usage of EMG-to-Speech conversion in a realistic setting with intelligible output can become a reality. We hope that the results presented as part of this dissertation of moving the field of EMG-to-Speech conversion further towards this goal.

# Glossary

**audible EMG** EMG recorded during audible (modal) speech.. 53, 87–90, 100, 107, 115, 120, 124, 130

**direct synthesis** A silent speech interface that converts a speech-related biosignal directly to audible speech without an intermediate textual representation.. 2–4, 6, 76

**EMG-to-Speech** The direct conversion of surface electromyographic signals to an audible speech waveform.. viii, xii, 3–7, 9, 14, 17, 30, 31, 34, 36, 47, 49, 53, 54, 57–59, 61, 64–67, 70–72, 74, 78–82, 84–87, 89, 93, 94, 100–104, 106, 107, 110–113, 117, 119, 120, 122, 123, 127

**feature transformation** The computation of audio features from EMG features.. 36, 70, 97, 100, 101, 103, 104, 119

**modal speech** "Normal" audible acoustic speech – the way healthy individuals produce speech when not specifically prompted to speak in any other way.. 23

**silent operation** Operating on a biosignal recorded during speech production with no audible acoustic component, i.e. during silent speech, as opposed to operating without using the acoustic speech signal but on biosignals recorded during modal speech.. 6, 23, 44

**silent speech** Speech produced without an audible acoustic component, i.e. "mouthing" words.. 23, 44, 63, 120, 121

**Silent Speech Interface** SSI, A speech interface that continues to function even when an acoustic audible signal is not present.. 2, 3, 6, 23, 49

**silent EMG** EMG recorded during silent speech.. 64, 87, 88, 100, 117, 124, 130

# Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[BB16] Bruce Bartlett and Jenny Bartlett. *Practical Recording Techniques: The step-by-step approach to professional audio recording*. CRC Press, 2016.

[BHG+16] Florent Bocquelet, Thomas Hueber, Laurent Girin, Christophe Savariaux, and Blaise Yvert. Real-time control of an articulatory-based speech synthesizer for brain computer interfaces. *PLoS computational biology*, 12(11):e1005119, 2016.

[BM14] Martin J Ball and Nicole Muller. *Phonetics for communication disorders*. Psychology Press, 2014.

[BSW+18] Peter Birkholz, Simon Stone, Klaus Wolf, Dirk Plettemeier, Klaus Wolf, Dirk Plettemeier, Simon Stone, and Peter Birkholz. Non-invasive silent phoneme recognition using microwave signals. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 26(12):2404–2411, 2018.

[Can05] Luciano Canepàri. *A handbook of phonetics: natural phonetics: articulatory, auditory and functional*. LINCOM textbooks in linguistics ; 10. LINCOM Europa, 2005.

[CEHL01] Adrian D C Chan, Kevin Englehart, Bernard Hudgins, and Dennis Lovely. Myoelectric Signals to Augment Speech Recognition. *Medical and Biological Engineering and Computing*, 39:500–506, 2001.

[CK79] PR Cavanagh and PV Komi. Electromechanical delay in human skeletal muscle under concentric and eccentric contractions. *European Journal of Applied Physiology and Occupational Physiology*, 42(3):159–163, 1979.

[CP14] Luigi Cattaneo and Giovanni Pavesi. The facial motor system. *Neuroscience & Biobehavioral Reviews*, 38:135–159, 2014.

[CSM+19] Beiming Cao, Nordine Sebkhi, Ted Mau, Omer T Inan, and Jun Wang. Permanent magnetic articulograph (pma) vs electromagnetic articulograph (ema) in articulation-to-speech synthesis for silent speech interface. In *Proceedings of the Eighth Workshop on Speech and Language Processing for Assistive Technologies*, pages 17–23, 2019.

[CTB07] John T Cacioppo, Louis G Tassinary, and Gary Berntson. *Handbook of psychophysiology*. Cambridge university press, 2007.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CWBvdO19] Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053, 2019.

[DAB+20] Lorenz Diener, Shahin Amiriparian, Catarina Botelho, Kevin Scheck, Dennis Küster, Isabel Trancoso, Björn W. Schuller, and Tanja Schultz. Towards silent paralinguistics: Deriving speaking mode and speaker id from electromyographic signals. In *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020.

[DB06] Emanuela D'Andrea and Erik Barbaix. Anatomic research on the perioral muscles, functional matrix of the maxillary and mandibular bones. *Surgical and radiologic anatomy*, 28(3):261–266, 2006.

[DBS18] Lorenz Diener, Sebastian Bredehöft, and Tanja Schultz. A comparison of EMG-to-Speech Conversion for Isolated and Continuous Speech. In *13th ITG Conference on Speech Communication*, 2018.

[DCK02] Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[DFAS18] Lorenz Diener, Gerrit Felsch, Miguel Angrick, and Tanja Schultz. Session-Independent Array-Based EMG-to-Speech Conversion using Convolutional Neural Networks. In *13th ITG Conference on Speech Communication*, 2018.

[DJS15a] Lorenz Diener, Matthias Janke, and Tanja Schultz. Codebook clustering for unit selection based emg-to-speech conversion. In *Sixteenth Annual Conference of the International Speech Communication Association*, pages 2420–2424, 2015. Interspeech 2015.

[DJS15b] Lorenz Diener, Matthias Janke, and Tanja Schultz. Direct conversion from facial myoelectric signals to speech using deep neural networks. In *International Joint Conference on Neural Networks*, pages 1–7, 2015. IJCNN 2015.

[DLJ02] Paul Davis, Stéphane Letz, and JACK team. JACK audio connection kit, 2002. [Online; accessed November 11nd, 2020].

[Doe42] E Doehne. Bedeutet Flüstern Stimmruhe Oder Stimmschonung? Technical report, 1942.

[DRVS20] Lorenz Diener, Mehrdad Roustay Vishkasougheh, and Tanja Schultz. CSL-EMG_Array: An Open Access Corpus for EMG-to-Speech Conversion. In *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020.

[DRY$^+$09] Srinivas Desai, E Veera Raghavendra, B Yegnanarayana, Alan W Black, and Kishore Prahallad. Voice conversion using artificial neural networks. In *2009 IEEE International*

*Conference on Acoustics, Speech and Signal Processing*, pages 3893–3896. IEEE, 2009.

[DSLD10] Siyi Deng, Ramesh Srinivasan, Tom Lappas, and Michael D'Zmura. EEG Classification of Imagined Syllable Rhythm using Hilbert Spectrum Methods. *Journal of Neural Engineering*, 7(4), 2010.

[DUS19] Lorenz Diener, Tejas Umesh, , and Tanja Schultz. Improving fundamental frequency generation in emg-to-speeech conversion using a quantization approach. In *Automatic Speech Recognition and Understanding*, 2019.

[FAE58] Knud Faaborg-Andersen and A Edfeldt. Electromyography of Intrinsic and Extrinsic Laryngeal Muscles During Silent Speech: Correlation with Reading Activity: Preliminary Report. *Acta oto-laryngologica*, 49(1):478–482, 1958.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[Fan81] Gunnar Fant. The source filter concept in voice production. *STL-QPSR*, 1(1981):21–37, 1981.

[FEG+08] Michael J Fagan, Stephen R Ell, James M Gilbert, E Sarrazin, and Peter M Chapman. Development of a (silent) speech recognition system for patients following laryngectomy. *Medical engineering & physics*, 30(4):419–425, 2008.

[FHG+17] Diandra Fabre, Thomas Hueber, Laurent Girin, Xavier Alameda-Pineda, and Pierre Badin. Automatic animation of an articulatory tongue model from ultrasound images of the vocal tract. *Speech Communication*, 93:63–75, 2017.

[FT92] Toshiaki Fukada and Keiichi Tokuda. an Adaptive Algorithm for Mel-Cepstral Analysis of Speech. In *Proc. ICASSP*, pages 137–140, 1992.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GCG+16] Jose A Gonzalez, Lam A Cheah, James M Gilbert, Jie Bai, Stephen R Ell, Phil D Green, and Roger K Moore. A silent speech system based on permanent magnet articulography and direct synthesis. *Computer Speech & Language*, 39:67–87, 2016.

[GG18] Jose A Gonzalez and Phil D Green. A real-time silent speech system for voice restoration after total laryngectomy. *Revista de Logopedia, Foniatría y Audiología*, 38(4):148–154, 2018.

[GGT+18] Tamás Grósz, Gábor Gosztolya, László Tóth, Tamás Gábor Csapó, and Alexandra Markó. F0 estimation for dnn-based ultrasound silent speech interfaces. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 291–295. IEEE, 2018.

[GGT+20] Gábor Gosztolya, Tamás Grósz, László Tóth, Alexandra Markó, and Tamás Gábor Csapó. Applying dnn adaptation to reduce the session dependency of ultrasound tongue imaging-based silent speech interfaces. *Acta Polytechnica Hungarica*, 17(7), 2020.

[GL18] Henry Gray and WH Lewis. Anatomy of the human body. 20th edition. *Philadelphia and New York, Lea & Febiger*, 1918.

[GLF+93] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus. *NASA STI/Recon technical report n*, 93, 1993.

[GRHF+20] Marion Girod-Roux, Thomas Hueber, Diandra Fabre, Silvain Gerber, Mélanie Canault, Nathalie Bedoin, Audrey Acher, Nicolas Béziaud, Eric Truy, and Pierre Badin. Rehabilitation of speech disorders following glossectomy, based on ultrasound visual illustration and feedback. *Clinical Linguistics & Phonetics*, pages 1–18, 2020.

[HB96] Andrew J Hunt and Alan W Black. Unit Selection in a Concatenative Speech Synthesis System using a Large Speech Database. In *Proc. ICASSP*, pages 373–376, 1996.

[HB16] Thomas Hueber and Gérard Bailly. Statistical conversion of silent articulation into audible speech using full-covariance hmm. *Computer Speech & Language*, 36:274–293, 2016.

[HCD+08] Thomas Hueber, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. Towards a Segmental Vocoder Driven by

Ultrasound and Optical Images of the Tongue and Lips. In *Proc. Interspeech*, 2008.

[HHdP+15] Christian Herff, Dominic Heger, Adriana de Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. Brain-To-Text: Decoding Spoken Phrases from Phone Representations in the Brain. *Frontiers in Neuroscience*, 9:1–11, 2015.

[HJD+16] C. Herff, G. Johnson, L. Diener, J. Shih, D. Krusienski, and T. Schultz. Towards direct speech synthesis from ecog: A pilot study. In *Engineering in Medicine and Biology Society (EMBC), 2016 38th Annual International Conference of the IEEE*, Aug 2016.

[HN11] John F Houde and Srikantan S Nagarajan. Speech production as state feedback control. *Frontiers in human neuroscience*, 5:82, 2011.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hue13] Thomas Hueber. Ultraspeech-player: intuitive visualization of ultrasound articulatory data for speech therapy and pronunciation training. In *INTERSPEECH*, pages 752–753, 2013.

[HWHK63] Arthur S. House, Carl Williams, Michael H. L. Hecker, and Karl D. Kryter. Psychoacoustic speech tests: A modified rhyme test. *The Journal of the Acoustical Society of America*, 35(11):1899–1899, 1963.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ArXiv e-prints*, February 2015.

[Ima83] Satoshi Imai. Cepstral Analysis Synthesis on the Mel Frequency Scale. *Proc. ICASSP*, pages 93–96, 1983.

[Int99] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[IT06] ITU-T. Itu-T Recommendation P.10. Technical report, 2006.

[Ita75] Fumitada Itakura. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67 – 72, 1975.

[ITU01] ITU Radiocommunication Bureau. Method for the subjective assessment of intermediate quality level of coding systems. *Recommendation ITU-R BS. 1534*, 2001.

[Jan16] Matthias Janke. *EMG-to-Speech: Direct Generation of Speech from Facial Electromyographic Signals*. PhD thesis, Karlsruher Institut für Technologie, 2016.

[JD17] Matthias Janke and Lorenz Diener. Emg-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(12):2375–2385, nov 2017.

[JMHSW06] Szu-Chen Jou, Lena Maier-Hein, Tanja Schultz, and Alex Waibel. Articulatory Feature Classification using Surface Electromyography. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[JSW$^+$06] Szu-Chen (Stan) Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. Towards continuous speech recognition using surface electromyography. In *In proceedings of the 9th ISCA International Conference on Spoken Language Processing*, 2006.

[JWS10] Matthias Janke, Michael Wand, and Tanja Schultz. a spectral mapping method for emg-based recognition of silent speech. *Proc. B-INTERFACE*, pages 2686–2689, 2010.

[KB04] John Kominek and Alan W Black. The cmu arctic speech databases. In *Fifth ISCA workshop on speech synthesis*, 2004.

[KB14] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

[KES$^+$18] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*, 2018.

[KHJG01] George Karpati, David Hilton-Jones, and Robert C Griggs. *Disorders of voluntary muscle.* Cambridge University Press, 2001.

[KKM18] Arnav Kapur, Shreyas Kapur, and Pattie Maes. Alterego: A personalized wearable silent speech interface. In *23rd International Conference on Intelligent User Interfaces*, pages 43–53. ACM, 2018.

[Kub93] Robert F Kubichek. Mel-Cepstral Distance Measure for Objective Speech Quality Assessment. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 125–128, 1993.

[KZ14] Sebastian Kraft and Udo Zölzer. BeaqleJS : HTML5 and JavaScript Based Framework for the Subjective Evaluation of Audio Quality. In *Linux Audio Conference (LAC-2014)*, 2014.

[Lag02] Terrence D Lagerlund. Volume conduction. *CONTEMPORARY NEUROLOGY SERIES*, 66:28–40, 2002.

[LB$^+$95] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lom11] E Lombard. Le Signe De L'elevation De La Voix. *Ann. Mal. Oreille Larynx*, 37:101–119, 1911.

[Mon04] C Montgomery. Vorbis i specification, 2004.

[Moz95] Michael C Mozer. A focused backpropagation algorithm for temporal pattern recognition. *Backpropagation: Theory, architectures, and applications*, 137, 1995.

[MP13] Roberto Merletti and Philip Parker. *Electromyography: Physiology, Engineering, and Noninvasive Applications.* John Wiley and Sons, Inc., 2013.

[MSY16] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[MTS+13] Takuto Moriguchi, Tomoki Toda, Motoaki Sano, Hiroshi Sato, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. A digital signal processor implementation of silent/electrolaryngeal speech enhancement based on real-time statistical voice conversion. In *INTERSPEECH*, pages 3072–3076, 2013.

[Nyq28] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[Oli06] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006. [Online; accessed May 2nd, 2020].

[OSH08] Makoto Otani, Shota Shimizu, and Tatsuya Hirahara. Vocal Tract Shapes of Non-Audible Murmur Production. *Acoustical science and technology*, 29(2):195–198, 2008.

[OT 19] OT Bioelettronica. Matrices datasheet. 2019.

[Pha06] Hubert Pham. PyAudio, 2006. [Online; accessed November 11nd, 2020].

[PWSS19] Krsto Proroković, Michael Wand, Tanja Schultz, and Jürgen Schmidhuber. Adaptation of an emg-based speech recognizer via meta-learning. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[QZC+19] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. AutoVC: Zero-shot voice style transfer with only autoencoder loss. volume 97 of *Proceedings of Machine Learning Research*, pages 5210–5219, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[RG11] Lee Ann Remington and Denise Goodwin. *Clinical anatomy of the visual system E-Book*. Elsevier Health Sciences, 2011.

[SAD+19] Tanja Schultz, Miguel Angrick, Lorenz Diener, Dennis Küster, Moritz Meier, Dean Krusienski, Christian Herff, and Jonathan Brumberg. Towards restoration of articulatory movements: Functional electrical stimulation of orofacial muscles. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3111–3114, July 2019.

[SB20] Simon Stone and Peter Birkholz. Cross-speaker silent-speech command word recognition using electro-optical stomatography. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7849–7853. IEEE, 2020.

[SBP+92] Kim Silverman, Mary Beckman, Janet Pierrehumbert, Mari Ostendorf, Colin Wightman, Patti Price, and Julia Hirschberg. Tobi: A standard scheme for labeling prosody. In *Proceedings of the Second International Conference on Spoken Language Processing*, pages 867–879, 1992.

[SCM98] Yannis Stylianou, Olivier Cappe, and Eric Moulines. Continuous Probabilistic Transform for Voice Conversion. *IEEE Transactions on Speech and Audio Processing*, 6(2):131–142, 1998.

[SHB+06] David Sundermann, Harald Hoge, Antonio Bonafonte, Hermann Ney, Alan Black, and Shri Narayanan. Text-independent voice conversion based on unit selection. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[SKRL02] Andrew Stuart, Joseph Kalinowski, Michael P Rastatter, and Kerry Lynch. Effect of delayed auditory feedback on normal speakers at two speech rates. *The Journal of the Acoustical Society of America*, 111(5):2237–2241, 2002.

[SL11] Johann S Schwegler and Runhild Lucius. *Der Mensch-Anatomie und Physiologie*. Georg Thieme Verlag, 2011.

[Sok12] Aleksandr Sokolov. *Inner speech and thought*. Springer Science & Business Media, 2012.

[SW10] Tanja Schultz and Michael Wand. Modeling Coarticulation in EMG-Based Continuous Speech Recognition. *Speech Communication*, 52(4):341–353, 2010.

[SWH+17] Tanja Schultz, Michael Wand, Thomas Hueber, Krusienski Dean J., Christian Herff, and Jonathan S. Brumberg.

Biosignal-based spoken communication: A survey. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(12):2257–2271, nov 2017.

[TBT07] Tomoki Toda, Alan W Black, and Keiichi Tokuda. Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory. *IEEE Transactions on Audio, Speech and Language Processing*, 15(8):2222–2235, 2007.

[THHJ10] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. a Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4214–4217, 2010.

[TKI94] Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai. Recursive calculation of mel-cepstrum from lp coefficients. *Trans. IEICE*, 71:128–131, 1994.

[TMB12] Tomoki Toda, Takashi Muramatsu, and Hideki Banno. Implementation of Computationally Efficient Real-Time Voice Conversion. *Proc. Interspeech*, pages 94–97, 2012.

[TS05] Tomoki Toda and Kiyohiro Shikano. Nam-to-speech conversion with gaussian mixture models. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1957–1960, 2005.

[TWG$^+$14] Dominic Telaar, Michael Wand, Dirk Gehrig, Felix Putze, Christoph Amma, Dominic Heger, Ngoc Thang Vu, Mark Erhardt, Tim Schlippe, Matthias Janke, Christian Herff, and Tanja Schultz. BioKIT - Real-Time Decoder for Biosignal Processing. *Proc. Interspeech*, pages 2650–2654, 2014.

[TWS09] Arthur Toth, Michael Wand, and Tanja Schultz. Synthesizing Speech from Electromyography using Voice Transformation Techniques. In *INTERSPEECH 2009*, pages 652 – 655, 2009.

[UCL02] UCLA phonetics laboratory. *Dissection of the speech production mechanism.* UCLA, 2002.

[vdODZ$^+$16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[VDOV+17] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[VRPG90] AC Metting Van Rijn, A Peper, and CA Grimbergen. High-quality recording of bioelectric events. *Medical and Biological Engineering and Computing*, 28(5):389–397, 1990.

[VS19a] Jean-Marc Valin and Jan Skoglund. Lpcnet: Improving neural speech synthesis through linear prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895. IEEE, 2019.

[VS19b] Jean-Marc Valin and Jan Skoglund. A real-time wideband neural vocoder at 1.6 kb/s using lpcnet. *arXiv preprint arXiv:1903.12087*, 2019.

[VSJV13] Koen Vos, Karsten Vandborg Sørensen, Søren Skak Jensen, and Jean-Marc Valin. Voice coding with opus. In *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.

[Wel67] Peter D Welch. the Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, jun 1967.

[WJH+14] Michael Wand, Matthias Janke, Till Heistermann, Christopher Schulte, Adam Himmelsbach, and Tanja Schultz. Application of Electrode Arrays for Artifact Removal in an Electromyographic Silent Speech Interface. *Biomedical Engineering Systems and Technologies*, 452:300–312, 2014.

[WJS11] Michael Wand, Matthias Janke, and Tanja Schultz. Investigations on Speaking Mode Discrepancies in EMG-Based Speech Recognition. In *Proc. Interspeech*, pages 601–604, 2011.

[WKJ+06] Matthias Walliczek, Florian Kraft, Szu-Chen Jou, Tanja Schultz, and Alex Waibel. Sub-Word Unit Based Non-Audible Speech Recognition using Surface Electromyography. In *Proc. Interspeech*, pages 1487–1490, 2006.

[WS11] Michael Wand and Tanja Schultz. Analysis of Phone Confusion in EMG-Based Speech Recognition. In *Proc. ICASSP*, 2011.

[WTJS09] Michael Wand, Arthur Toth, Szu-Chen (Stan) Jou, and Tanja Schultz. Impact of different speaking modes on emg-based speech recognition. In *10th Annual Conference of the International Speech Communication Association*, 2009.

[WWPM18] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[XWG19] Kele Xu, Yuxiang Wu, and Zhifeng Gao. Ultrasound-based silent speech interface using sequential convolutional auto-encoder. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2194–2195, 2019.

[YVM+19] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). 2019.

[ZJWS14] Marlene Zahner, Matthias Janke, Michael Wand, and Tanja Schultz. Conversion from Facial Myoelectric Signals to Speech: A Unit Selection Approach. In *Proc. Interspeech*, pages 1184 – 1188, 2014.

[ZNY+07] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda. The hmm-based speech synthesis system (hts) version 2.0. In *SSW*, pages 294–299. Citeseer, 2007.

[Zwi61] Eberhard Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.

# Appendix

## List of Supervised Student Theses

The following four tables present a list of all student theses (Bachelorarbeit or Masterarbeit) that were (co-)supervised by the author in the context of this work. All of these were completed under the primary supervision of Prof. Dr.-Ing. Tanja Schultz.

| Student | Year | Thesis Title |
|---|---|---|
| Daniel Hackbarth | 2020 | Echtzeitfähiger Neuronaler Vocoder für die EMG-basierte Sprachsynthese |
| Mehrdad Roustay | 2020 | Ein Array-basierter Korpus zur Evaluation von adaptiver echtzeitfähiger EMG-basierter Sprachsynthese |
| Magomed Ibragimov | 2020 | Klassifikation paralinguistischer Merkmale aus EMG-Daten: Sprecheridentität und Sprachmodus |
| Darius Ivucic | 2020 | Real-Time Speech Synthesis from Invasively Measured Neural Signals using a Unit Selection Approach |
| Gabriel Ivucic | 2020 | Real-Time Speech Synthesis from Electrical Activity of Facial Muscles using a Unit Selection Approach |
| Sebastian Bredehöft | 2018 | Aufzeichnung und Auswertung eines Datenkorpus für EMG-zu-Sprache-Konvertierung |
| Gerrit Felsch | 2018 | EMG-to-Speech Conversion using Convolutional Neural Networks |

**Table 8.1** – List of all supervised “Bachelorarbeiten”.

| Student | Year | Thesis Title |
|---|---|---|
| Sebastian Kühl | 2020 | Zielgerichtete Generierung von Text mittels Variational Autoencodern am Beispiel von Memes |
| Kevin Scheck | 2020 | EMG-to-Speech Conversion using Deep Generative Models |

**Table 8.2** – List of all supervised “Masterarbeiten”.

# List of Publications

The following bibliography lists all publications of which this documents author is an author or co-author.

[1] Inma Hernaez, Jose Andrés González-López, Eva Navas, Jose Luis Pérez Córdoba, Ibon Saratxaga, Jon Olivares, Gonzalo abd Sánchez de la Fuente, Alberto Galdón, Víctor García Romillo, Míriam González-Atienza, Tanja Schultz, Phil Green, Michael Wand, and Lorenz Diener. Voice restoration with silent speech interfaces (ReSSInt). In *IberSpeech*, 2020 (submitted).

[2] Miguel Angrick, Maarten Ottenhoff, Lorenz Diener, Darius Ivucic, Gabriel Ivucic, Sofoklis Goulis, Jeremy Saal, Albert J. Colon, Louis Wagner, Dean J. Krusienski, Pieter L. Kubben, Tanja Schultz, and Christian Herff. Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity. *bioRxiv (preprint, submitted for peer review)*, 2020.

[3] Christian Herff, Lorenz Diener, Emily Mugler, Marc Slutzky, Dean Krusienski, and Tanja Schultz. Towards speech synthesis from intracranial signals. In *Brain–Computer Interface Research*, pages 47–54. Springer, 2020.

[4] Lorenz Diener, Mehrdad Roustay Vishkasougheh, and Tanja Schultz. CSL-EMG_Array: An Open Access Corpus for EMG-to-Speech Conversion. In *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020.

[5] Lorenz Diener, Shahin Amiriparian, Catarina Botelho, Kevin Scheck, Dennis Küster, Isabel Trancoso, Björn W. Schuller, and Tanja Schultz. Towards silent paralinguistics: Deriving speaking mode and speaker id from electromyographic signals. In *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020.

[6] Catarina Botelho, Lorenz Diener, Dennis Küster, Kevin Scheck, Shahin Amiriparian, Björn W. Schuller, Tanja Schultz, Alberto Abad, and Isabel Trancoso. Toward silent paralinguistics: Speech-to-emg – retrieving articulatory muscle activity from speech. In *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020 (to appear).

[7] Lorenz Diener, Tejas Umesh, , and Tanja Schultz. Improving fundamental frequency generation in emg-to-speeech conversion using a quantization approach. In *Automatic Speech Recognition and Understanding*, 2019.

[8] Tanja Schultz, Miguel Angrick, Lorenz Diener, Dennis Küster, Moritz Meier, Dean Krusienski, Christian Herff, and Jonathan Brumberg. Towards restoration of articulatory movements: Functional electrical stimulation of orofacial muscles. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3111–3114, July 2019.

[9] Lorenz Diener, Gerrit Felsch, Miguel Angrick, and Tanja Schultz. Session-Independent Array-Based EMG-to-Speech Conversion using Convolutional Neural Networks. In *13th ITG Conference on Speech Communication*, 2018.

[10] Lorenz Diener, Sebastian Bredehöft, and Tanja Schultz. A comparison of EMG-to-Speech Conversion for Isolated and Continuous Speech. In *13th ITG Conference on Speech Communication*, 2018.

[11] Lorenz Diener and Tanja Schultz. Investigating Objective Intelligibility in Real-Time EMG-to-Speech Conversion. In *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association*, 2018.

[12] Matthias Janke and Lorenz Diener. Emg-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(12):2375–2385, nov 2017.

[13] Jochen Weiner, Lorenz Diener, Simon Stelter, Eike Externest, Sebastian Kühl, Christian Herff, Felix Putze, Timo Schulze, Mazen Salous, Hui Liu, Dennis Küster, and Tanja Schultz. Bremen big data challenge 2017: Predicting university cafeteria load. In Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm, editors, *KI 2017: Advances in Artificial Intelligence: 40th Annual German Conference on AI, Dortmund, Germany, September 25–29, 2017, Proceedings*, pages 380–386, Cham, 2017. Springer International Publishing.

[14] C. Herff, G. Johnson, L. Diener, J. Shih, D. Krusienski, and T. Schultz. Towards direct speech synthesis from ecog: A pilot study. In *Engineering in Medicine and Biology Society (EMBC), 2016 38th Annual International Conference of the IEEE*, Aug 2016.

[15] L. Diener, C. Herff, M. Janke, and T. Schultz. An initial investigation into the real-time conversion of facial surface emg signals to audible

speech. In *Engineering in Medicine and Biology Society (EMBC), 2016 38th Annual International Conference of the IEEE*, Aug 2016.

[16] Lorenz Diener, Matthias Janke, and Tanja Schultz. Codebook clustering for unit selection based emg-to-speech conversion. In *Sixteenth Annual Conference of the International Speech Communication Association*, pages 2420–2424, 2015. Interspeech 2015.

[17] Lorenz Diener, Matthias Janke, and Tanja Schultz. Direct conversion from facial myoelectric signals to speech using deep neural networks. In *International Joint Conference on Neural Networks*, pages 1–7, 2015. IJCNN 2015.

[18] Tim Reiner, Sylvain Lefebvre, Lorenz Diener, Ismael García, Bruno Jobard, and Carsten Dachsbacher. A runtime cache for interactive procedural modeling. *Computers & Graphics*, 36(5):366–375, 2012.