Karlsruhe Institute of Technology

Improving Unit Selection based EMG-to-Speech Conversion

Masterarbeit am Cognitive Systems Lab
Prof. Dr.-Ing. Tanja Schultz
Fakultät für Informatik
Karlsruher Institut für Technologie

von

cand. inform.
Lorenz Diener

Betreuer:
Dipl. Inform. Matthias Janke
Prof. Dr.-Ing. Tanja Schultz

Tag der Anmeldung: 1. Februar 2015
Tag der Abgabe: 31. Juli 2015

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 31.07.2015

Zusammenfassung

Diese Masterarbeit stellt einen neuen Ansatz zur Verbesserung eines Unit-Selection-Systems zur Umwandlung myoelektrischer Signale der Gesichtsmuskulatur in hörbare Sprache vor.

Oberflächenelektromyographie ist das Aufnehmen der elektrischen Muskelaktivität mit an der Hautoberfläche angebrachten Elektroden. Es ist bekannt, das es möglich ist, aus solcher während der Sprachproduktion erzeugter Aktivität hörbare Sprache zu erzeugen. Dies wurde in vorhergegangenen Arbeiten mit verschiedenen Ansätzen erreicht.

Diese Arbeit konzentriert sich auf Unit Selection. Unit Selection erzeugt ein Sprachsignal, indem es Audio-Stücke, ausgewählt nach einem Kriterium, das auf parallelen EMG-Daten ausgewertet wird, konkateniert. In dieser Arbeit wird ein neuartiger Ansatz vorgestellt und evaluiert, der die Datenbank, aus der Units ausgewählt werden optimiert und so den Konvertierungsprozess qualitativ und quantitativ verbessert.

Insgesamt erreicht der neue Ansatz eine qualitative Verbesserung von bis zu 14.92 Prozent relativ gegenüber dem vorherigen System. Gleichzeitig wird die Zeit, die für die Konvertierung benötigt wird, um bis zu 98% verringert.

Abstract

This master's thesis introduces a new approach to improve the unit-selection based conversion of facial myoelectric signals to audible speech.

Surface electromyography is the recording of electric signals generated by muscle activity using surface electrodes attached to the skin. Past work has shown that it is feasible to generate audible speech signals from facial electromyographic activity generated during speech production, using several different approaches.

This work focuses on the unit-selection approach to conversion, where the speech signal is reconstructed by concatenating pieces of target audio data selected by a similarity criterion calculated on the parallel sequence of source electromyographic data. A novel approach, based on optimizing the database that units are selected from by using unit clustering to generate more prototypical units and improve the selection process, is introduced and evaluated.

In total, we obtain a qualitative improvement of up to 14.92 percent relative over a baseline unit selection system, while improving the time taken for conversion by up to 98%.

Contents

1. Introduction

Speech is the oldest and most natural form of human communication. We begin to acquire the ability to express ourselves in spoken language at an early age, naturally and without requiring special instruction. We use it every day with little thought given to the actual process, and communicate not only facts, but also paralinguistic information such as our mood or emotions.

Silent Speech Interfaces (SSIs) aim to allow us to use our natural communication skills even in situations where the actual production of audible speech is either not desirable (e.g. in situations where other people would be disturbed, such as in a library) or wholly impossible (e.g. in case of speech impediments that prevent proper articulation), or in noisy environments where environmental sounds may drown out anything spoken. They do this by using other modes of acquiring information about the speech production process instead of relying solely on acoustic signals recorded during normal audible speech.

There are several different modalities that can be used to construct SSIs, such as non-audible murmur [TS05] or the recording of lip movements using magnets attached to or implanted into a users lips [FEG+08]. This work focuses on speech synthesis based on facial *surface electromyography* (EMG): Surface electrodes on the users skin record muscle movement related electric potentials, which are then converted directly to audible speech.

This approach has several advantages: Unlike more invasive methods, surface EMG is easy and comparatively comfortable to use, and unlike SSIs based on speech *recognition*, a *synthesis* approach ideally enables the retention of information not directly contained in the textual content of what was spoken, such as prosody, speaker identity, mood and emotion.

To maximize the benefits of these advantages, a method that produces high-quality audio output is required. [ZJWS14] first introduced a conversion method based on unit selection, where the audio output is created by overlaying and concatenating small speech snippets. This allowed for output that sounds very natural - however, intelligibility remains a problem.

1.1 Goal of this work

In this work, we build on the work from [ZJWS14] and try to improve the output quality in relation to naturalness and intelligibility. We introduce an approach based on improving the *unit codebook* using a clustering technique.

1.2 Document overview

The remainder of this document is structured as follows: Chapter 2 introduces the biological and technical fundamentals of speech and facial electromyography and gives an overview over some of the work this thesis is based upon, chapter 3 introduces unit selection and explains how it can be used for EMG-to-Voice conversion and chapter 4 gives an overview over the improvements to the unit selection conversion process made in this work. Chapter 5 shows the results of experimental evaluations of the improved process with various sets of parameters, and chapter 6 finally provides some concluding remarks and an outlook on potential future work.

2. Fundamentals

This chapter introduces the physiological fundamentals of audible speech production, digital audio recording and electromyography. It explains the audible- and EMG features used in this work and introduces the basic principles behind unit selection based EMG-to-speech conversion. Finally, it gives an overview over previous related work.

2.1 Audible Speech

Audible speech is vocalized language. As an audio signal, it propagates through air as a longitudinal (compression) wave. An illustration of this can be found in figure 2.1 - sound propagates as a series of high-pressure and low-pressure areas in air, with spacing depending on sound frequency and intensity depending on the sounds volume.

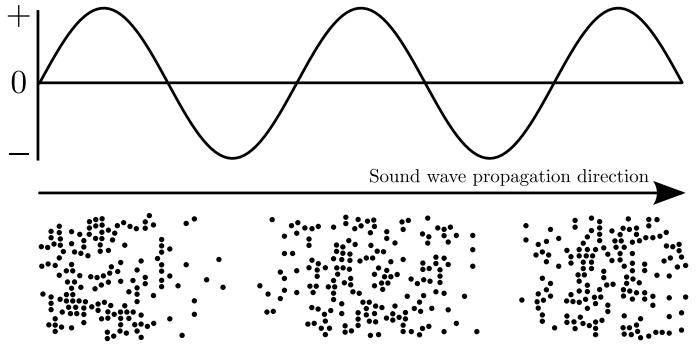

Figure 2.1: Sound propagation in air. Top: Pressure change relative to normal. Bottom: Particle movement.

2.1.1 Human speech production

Humans produce speech by exhaling air from their lungs, which passes by various obstacles that modify the stream of air before it leaves the body and thus change the produced sound. Together with the cavities that the air passes through on its way -

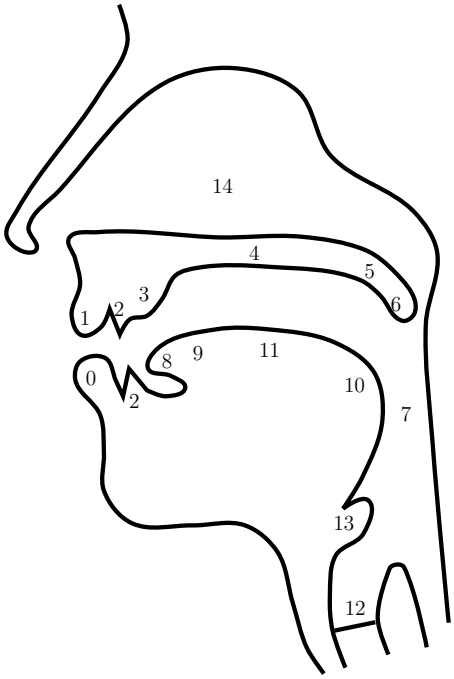

Figure 2.2: A cross-section of the human vocal tract, with places of articulation marked.

the *vocal tract* - these *articulators* form the *articulatory apparatus* with its various *places of articulation*. Sounds where the vocal tract is at least partially closed are called *consonants*, whereas sounds where no such obstruction occurs are called *vowels* [Can05, p. 58ff].

Figure 2.2 gives an overview of their locations within the vocal tract. The numbered locations correspond to the following places [Can05, p. 47]:

0 Lower lip
1 Upper lip
2 Teeth
3 Alveoles
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

Other than by their place of articulation, different sounds in human speech - called

phones - can be characterized by the articulators configuration, called the *manner of articulation* [Can05, p. 70ff]:

Stop - an occlusive sound, i.e. a sound where the airflow through the vocal tract stops completely before resuming again (e.g. the p in "pass").

Nasal - a sound where air flows primarily through the nasal cavity (e.g. the n in "nose")

Fricative - a sound resulting from turbulent airflow at a place of articulation due to partial obstruction (e.g. the f in "fricative").

Affricate - a stop changing into a fricative (e.g. the j in "jam").

Approximant - a sound where there is very little obstruction (e.g. the y in "yes").

Lateral - an approximant with airflow around the sides of the tongue (e.g. the l in "lateral").

Flap - a stop too brief to allow for buildup of air pressure (e.g. the t in butter in northern American English).

Trill - a sound resulting from the repeated opening and closing of the vocal tract, such as a "rolled r".

The final way of differentiating phones is by their degree of *phonation* or "voicing": 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 j in "Jane"), which would otherwise be *voiceless* (such as the ch in chain, which is otherwise identical to the aforementioned j).

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 an utterance is called *intonation* - except when used to differentiate or inflect words, which is done in some languages (e.g. Mandarin Chinese), in which case it is called the *tone*. For this work, it will be assumed from now on, however, that this is not the case, and that intonation solely carries *paralinguistic information* such as stress, mood and emotion.

2.1.2 Speech recording

The process of recording speech for digital processing is composed of three stages: The conversion of physical sound waves to an electrical signal, the sampling of this signal at discrete points in time, and the discretization of the sampled signal into intervals that can be digitally represented with a certain number of bits.

The first step is performed using a microphone. Different microphone technologies exist (e.g. induction microphones or condenser microphones), but they all achieve the same thing: The conversion of audible sound into an analog voltage signal.

As digital computers cannot directly process such a signal, *digitalization* is required before and further processing can take place. To this end, the signal is first *sampled* at a certain *sampling rate*, usually given in Hertz, resulting in one analog value per Hertz every second . The signal is then *quantized* by assigning discrete values to

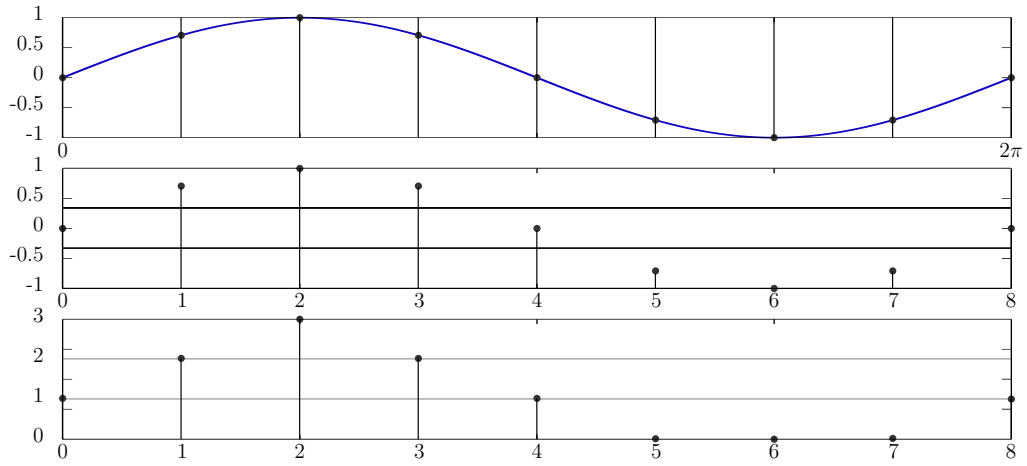

Figure 2.3: A continuous signal (top) is first sampled at a rate of 4πHz (middle) and then quantized with 2^2 steps (bottom).

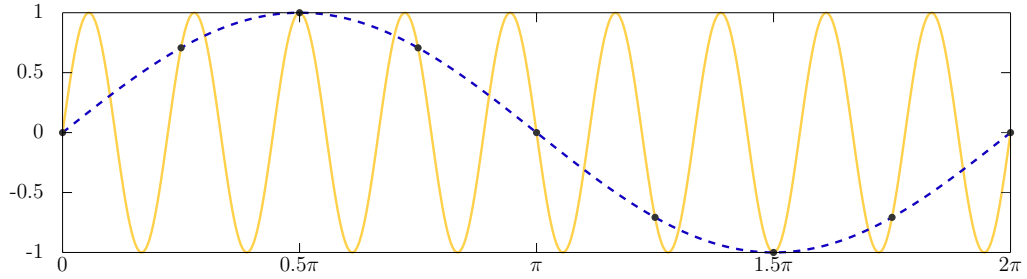

Figure 2.4: The two sine waves of different frequencies cannot be distinguished when only sampling at the marked positions. A signal reconstructed after sampling the yellow sine (frequency 4.5πHz) with this sampling frequency (4πHz) will contain only the blue dashed sine (frequency 1πHz).

analog voltage intervals, which finally gives a time series of discrete values that can be digitally processed. Figure 2.3 illustrates this process.

The digitalization process introduces error in each of its two steps. As can be seen in figure 2.4, sampling can introduce error when the signal varies with a frequency greater than half the sampling frequency (the *Nyquist frequency*). To avoid this problem, recording systems generally use a *low-pass filter*, a filter that attenuates signal components above a certain *cutoff frequency*, here set to be lower than the systems Nyquist frequency (with some room to account for imperfect low-pass filter behaviour). As long as a signal contains no components greater than the recording systems Nyquist frequency, it can be perfectly reconstructed from the sampled signal.

The speech signal is mostly concentrated between 250Hz and 4000Hz, which would allow for a sampling rate as low as 8000Hz, though higher rates are often used.

Quantization is the conversion of a real-valued into a discrete signal and thus necessarily introduces round-off error. While this error cannot be avoided, it can be managed by using a sufficient number of values (for digital speech processing, usually 16 to 32 bit numbers). The behaviour of the noise introduced by quantization is then similar to that of additive white noise and not correlated to the signal - and can be ignored.

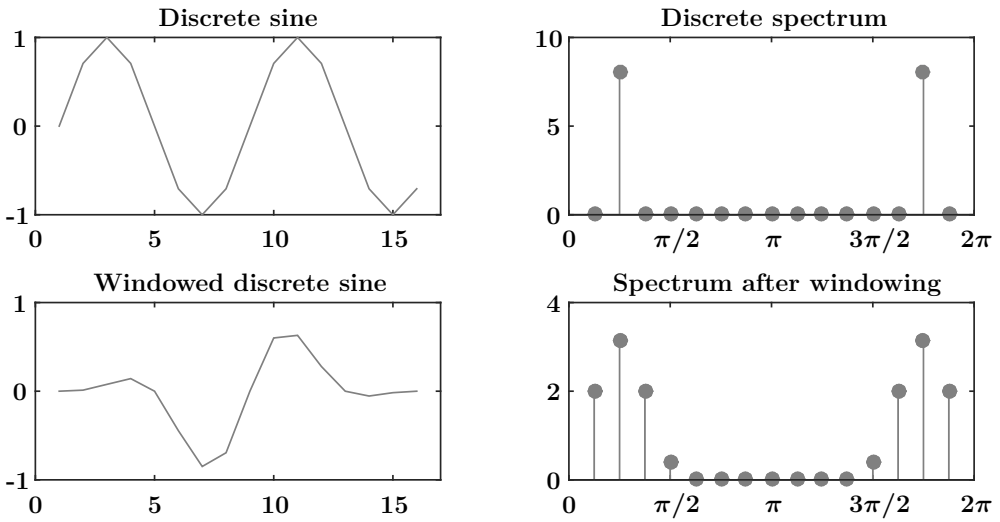

Figure 2.5: The windowing process (here, with a Blackman window) and its "washing-out" effect on the signal spectrum (Connecting lines for sine are drawn for illustration purposes only).

2.1.3 Speech representation

As explained in section 2.1.1, the process of human speech production can be understood as the generation of a base signal - either an oscillation of frequency F_0 or, in case of unvoiced phones, noise - followed by the application of a filter by the vocal tract.

The speech representation used in this work is similar: The signal is represented as the combination of F_0 representing the source signal (0 during unvoiced speech) and a set of *Mel-Frequency Cepstral Coefficients* (MFCCs) [Ima83] representing the filter. The feature extraction process for MFCCs comprises of 4 steps: Windowing, frequency analysis, cepstrum calculation and mel scale warping.

Windowing

Information about speech is temporally localized - extracting features from the entire signal at once does not yield much information. The process of cutting the signal into many smaller segments - called *frames* - for analysis is called *Windowing*.

This cutting process is equivalent to multiplying the time-domain signal with a function that is non-zero only for a certain duration - the *frame length* - starting at the beginning of the signal, shifting it forward a certain amount of time - the *frame shift* - for each successive frame. In the frequency domain, this multiplication becomes a convolution of the signal spectrum with the Fourier transform of the window, which causes spectral distortion (called *leakage*).

The impact of leakage can be reduced by not simply cutting the signal up into parts (i.e. windowing with a rectangular window, whose Fourier transform is the *sinc* function) but instead using a window function whose spectrum is more benign. In this work, a Blackman window is used. Figure 2.5 shows the windowing process and illustrates the leakage effect.

The Blackman window is defined as:

$$w(n) = \frac{1-0.16}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N-1}\right) + \frac{0.16}{2}\cos\left(\frac{4\pi n}{N-1}\right) \tag{2.1}$$

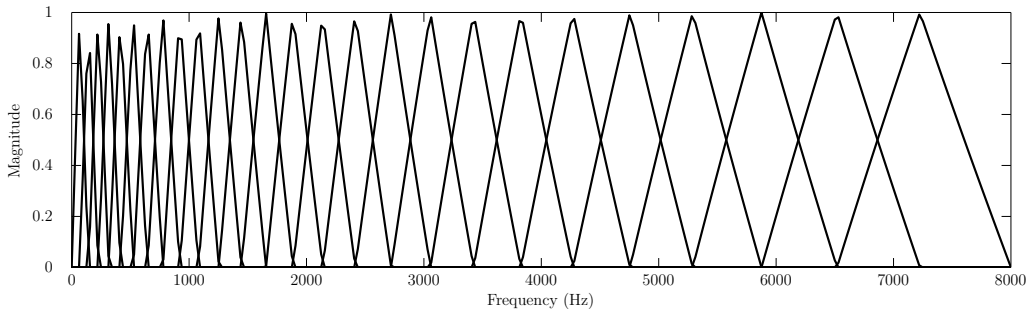

Figure 2.6: A 512-FFT to 26-Cepstral-Coefficients Mel filter bank.

Note that as the Blackman window rises and falls smoothly instead of infinitely fast (as the rect function does), a single Blackman windowed frame contains mostly information about the part of the signal it is centered on. To avoid losing information, it is necessary to have frames overlap - in that case, the frame shift is smaller than the frame length. This work uses a frame shift of 10ms and a frame length of 27ms, following the work of [ZJWS14].

Frequency Analysis

The human ear processes audio in the frequency domain. Similarly, in the frequency analysis step, a power-spectral representation of the signal - in this work, using a *fast Fourier transform* (FFT) - is computed. The power spectrum is the square of the absolute value of the complex spectrum.

Cepstrum Calculation

In the next step, the real part of the FFT of the logarithm of the frequency coefficients is calculated. The result of this calculation is called a *cepstrum*. The logarithm is, once more, for modeling human perception: Humans do not perceive volume linearly, but logarithmically. After the FFT application, the different cepstral coefficients provide information about the rate of change in any given frequency band.

Mel Filtering

The FFT produces a high number of cepstral components. To reduce the dimensionality of the feature sequence, neighbouring *quefrencies* can be binned together with a filter bank. The filter bank used in this work is the *Mel filter bank*. This filter bank again models human perception: It folds the spectrum into a set of coefficients that the human hearing apparatus is just able to distinguish. Figure 2.6 shows such a filter bank. In this work, the mel filtering is performed as a linear transform, after cepstrum calculation. [Ima83] The result of applying this frequency warping to the cepstrum are the final MFCCs.

2.1.4 F_0 Extraction

The fundamental frequency is, in this work, extracted using the YIN algorithm [DCK02], a modification of the auto-correlation method: The windowed signal is cross-correlated with itself, and the F_0 is extracted as the highest peak in the correlogram within a specified pitch range (manually set, depending on the speaker).

While it is possible to also estimate F_0 trajectories from EMG data [ZJWS14], this thesis focuses only on the mapping of MFCC vectors. For this reason, all audio used for subjective evaluations in this work (Compare section 3.2.4 for details about the audio synthesis and section 5.4.2 for details about the subjective evaluation performed in this work) was synthesized with the F_0 trajectories directly extracted from the reference audio file, preventing differences in F_0 mapping from affecting the comparison results.

2.2 Facial Surface Electromyography

Electromyography (EMG) is the recording of electrical signals generated by skeletal muscles in anticipation of and during movement. Surface EMG is the recording of such signals using surface electrodes attached to the skin. This section gives a brief introduction to the generation and recording of facial surface EMG signals related to speech production.

2.2.1 Biophysics of Skeletal Muscle Movement

Human movement is achieved through skeletal muscles - tissue that, upon receiving an electrical stimulus, contracts. Skeletal muscles normally come in antagonistic pairs: The contraction of a muscle relaxes its antagonist and vice-versa (i.e. the Biceps and Triceps).

Movement is initially initiated, voluntarily or reflexively, through the activation of at least one *motor neuron*. This motor neurons synapses (called *neuromuscular junctions*) are connected to many *myocytes* (muscle fibers). Together, a motor neuron and the muscle fibers it innervates are called a *motor unit*.

In rest, the muscle fibers membranes are negatively polarized with a potential difference of circa $-60mV$ due to a difference in the concentration of sodium (Na^+), potassium (K^+) and chloride (Cl^-) ions inside and outside of the cell: The membrane allows K^+ ions to pass to the outside, which happens 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 it is sufficiently so, 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]. Figure 2.7 illustrates this action potential generation and conduction.

Inside the cell, this causes (Via the secretion of another ion, Ca^{++} [MP04, 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*).

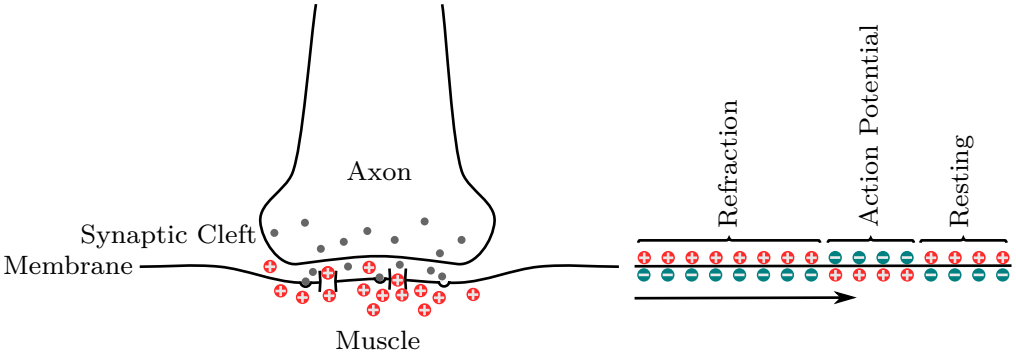

Figure 2.7: Generation (left) of action potentials and their conduction (right) along a muscle fiber.

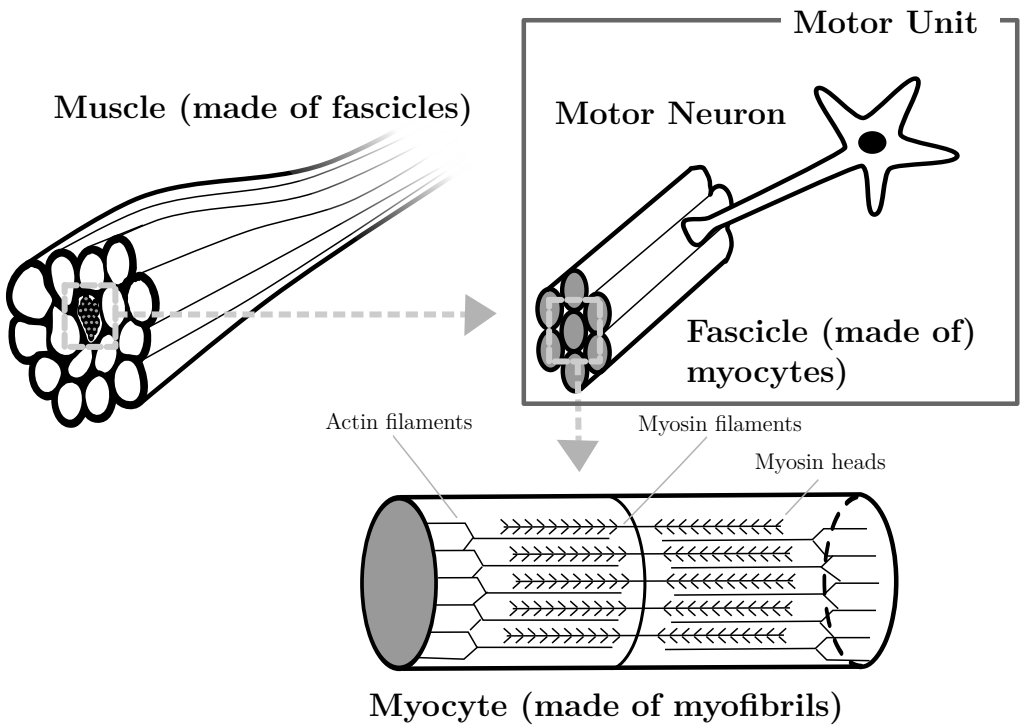

Figure 2.8: Schematic depiction of the structure of a skeletal muscle.

Figure 2.8 shows the structure of a skeletal muscle from the complete muscle over single muscle fibers - with a motor unit highlighted - down to myocytes and myofibrils with the actin fibers and myosin heads that are responsible for producing the actual cell contraction.

Electromechanical Delay

Muscles, when innervated, do not contract instantly - there is a small delay between membrane depolarization and movement onset, called the *electromechanical delay* (EMD) [CK79]. The exact delay depends on the muscle and innervation speed and strength. The model used for EMD in this paper is relatively simple: The EMG signal is universally delayed by 50 milliseconds, a value that was empirically determined in [JSW+06].

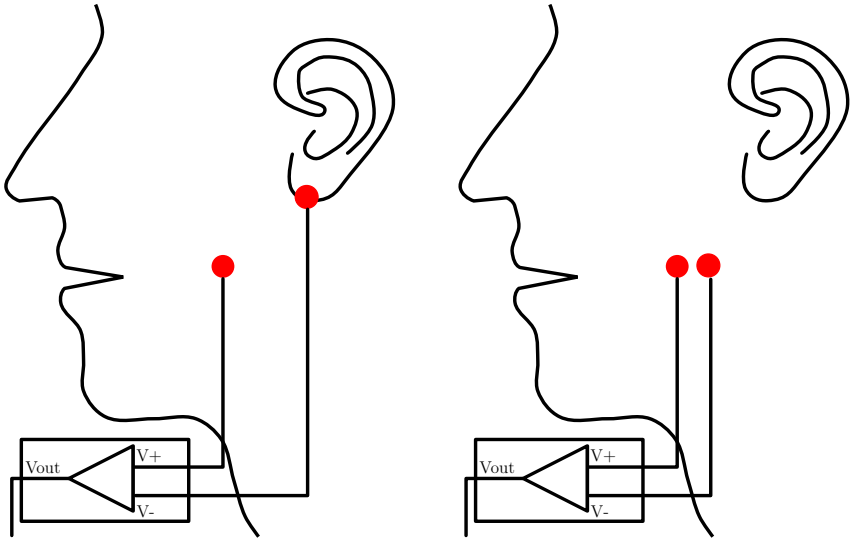

Figure 2.9: Unipolar (left, with the reference electrode on the electrically inactive ear lobe) versus Bipolar (right, deriving between two electrodes on electrically active territory) signal derivation.

2.2.2 Surface EMG Recording

As explained in the previous section, 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.

The amount of force a muscle generates depends on two factors: The amount of motor units recruited, and the rate at which they are fired [MP04, p. 97]. This gives five factors that determine what kind of signal arrives at an EMG electrode: Active motor units, the units firing rate, the position of the electrode relative to the units myocytes, the conduction between myocytes and electrodes, and body-internal as well as external noise.

The effect of *volume conduction* in tissue on the surface signal is two-fold: It leads to a spatial low-pass filtering (washing-out over space) of the signal [MP04, p. 89], as well as the summation of signals from multiple sources.

The actual signal measured is always a potential difference between two electrodes. This is done 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 (Compare figure 2.9). The advantage of bipolar measurement, coupled with *differential amplification*, is a much reduced sensitivity to *common-mode interference* (noise that affects both electrodes in the same way). Many common artifacts, such as power line noise or influence from non-target muscles, have this property.

For the recording of the EMG data used in this work, the process was as follows [JSW+06]:

- Unipolar or bipolar derivation with surface electrodes.
- DC offset removal.

- Differential amplification of the signal.
- Low- and high-pass filtering for artifact removal.
- Digitalization (as already described in section 2.1.2).
- Application of a 50ms delay to account for EMD.
- Windowing with a rectangular window (Refer again to section 2.1.2).
- Feature extraction
- Feature reduction

2.2.3 EMG Time-Domain Features

The relationship between facial surface EMG signal of the muscles controlling the articulatory apparatus and the generated speech is non-trivial. To learn useful relationships, in practice, features must be extracted from the signal in some way. The *time-domain features* used in this work are based on [JSW+06].

In the following description of the features, let $x[n]$ denote the n-th signal value. With $p[n]$ being the nine-point double averaged signal:

$$w[n] = \frac{1}{9}\sum_{k=-4}^{4} v[n+k], \quad \text{where} \quad v[n] = \frac{1}{9}\sum_{k=-4}^{4} x[n+k]. \tag{2.2}$$

we can define a high-frequency signal:

$$p[n] = x[n] - w[n] \tag{2.3}$$

as well as a rectified high-frequency signal:

$$r[n] = |p[n]| \tag{2.4}$$

Additionally, for any given feature $\mathbf{f}$, within a frame of size N, $\bar{\mathbf{f}}$ denotes the frame-based mean:

$$\bar{\mathbf{f}} = \frac{1}{N}\sum_{n=0}^{N-1} f[n] \tag{2.5}$$

$\mathbf{P_f}$ denotes the frame-based power:

$$\mathbf{P_f} = \sum_{n=0}^{N-1} f[n]^2 \tag{2.6}$$

and $\mathbf{z_f}$ the frame-based zero-crossing rate (the number of times the signals sign changed).

We can now define the $TD0$ features as a combination of all of the above:

$$\mathbf{TD0} = [\bar{\mathbf{w}}, \mathbf{P_w}, \mathbf{P_r}, \mathbf{z_p}, \bar{\mathbf{r}}] \tag{2.7}$$

and more generally the **TDN** features as a stacking with left and right context of size N each (for a total of $2 * N + 1$ sets of **TD0** features). In this work, $TD15$ features are used, giving a context of 15 frames to the left and right each for every feature frame.

2.2.4 LDA Feature Reduction

The TD15 features provide a great amount of context for every single feature frame. As a result, the TD15 feature vector is very large, which can lead to numerical and memory problems.

In order to address this and to improve selection performance, the dimensionality of the feature vector is reduced using *Linear Discriminant Analysis* (LDA) [Rao48]. LDA is a technique that linearly transforms a set of feature vectors so that each additional dimension contains the maximum amount of information that helps separate the data into a set of assigned classes - it maximizes the ratio of between-class variance to within-class variance.

In this work, the classes used are phone-substate labels (For each phone, the phones beginning, middle and end), force-aligned to the audible recordings using a speech recognizer. The LDA transformation matrix is calculated on the TD15 features of the training set of each session. This matrix is then used to transform all feature vectors of that session. The resulting transformed feature vector is finally cut at 32 dimensions, resulting in our final TD15-LDA features.

2.3 Related work

Using electromyography to infer information about speech, audible or silent, has been the subject of a growing body of past research. This section gives a short overview over some past results.

[MO86] is an early investigation into the relationship between EMG signals and speech. The authors, with the goal of working towards speech prostheses for people who have lost the ability to speak, show that a statistical relationship between facial electromyographic signals and audible speech exists.

[CEHL01] try to use facial EMG to enhance the performance of an (audible) speech recognition system and to this end, try to perform speech recognition on isolated words with a 10 word vocabulary, with accuracies of 89% to 98%, using an LDA classifier. They later improve on their results in [CEHL02], using a Hidden Markov Model classifier to achieve more stable recognition when some amount of temporal variation is introduced.

[JSW+06] introduce a first EMG-based continuous speech recognition system, based on modeling and recognizing the EMG signals of single phonemes. They evaluate multiple sets of features and the effect of delaying the EMG signal to account for EMD, finally achieving a word error rate of 32% on a 108 word vocabulary. In [SW10], they improve upon these results by modeling phoneme co-articulation, achieving similar performance in a multi-speaker system.

[TWS09] is the first paper that performs direct conversion of EMG signals to audible speech, without performing any recognition. They use feature extraction methods similar to [JSW+06] and map the feature set to a set of mel-cepstral coefficients using *Gaussian Mixture Model Mapping*, a technique previously used for voice-to-voice conversion. They achieve a best average Mel-Cepstral Distortion score (Compare section 5.2) of 6.37, though with large variations between utterances (with a standard deviation of 2.34).

[WSJS13] introduce the use of *electrode arrays* for EMG silent speech interfaces, replacing the complicated setup required for attaching a set of electrodes in certain places in the face with a more streamlined process. Using this new setup to perform speech recognition, they manage to achieve an average word error rate of 10.9% in a speaker-dependent system.

[ZJWS14] introduce the use of unit selection for EMG-to-speech conversion. As this work continues the investigation into unit selection based EMG to Speech conversion, the approach is explained in detail in chapter 3.

3. Basic Unit Selection

An EMG-to-Speech conversion system is a system that takes facial electromyographic recordings as its input and outputs audible speech synthesized from this EMG data that, ideally, closely matches the spoken or (in the case of a Silent Speech system) mouthed words.

This chapter introduces unit selection, as originally used for speech synthesis [HB96] as well as the changes necessary to use unit selection as a basic EMG-to-Speech conversion system, as introduced by [ZJWS14], which this work builds upon and uses as a baseline system for evaluation.

3.1 Unit Database

Unit selection, as the name implies, builds utterances by selecting and then concatenating *unit*s of sound from a unit database, called the *unit codebook*. To understand the makeup of this codebook, we must first look at what a single unit consists of.

A unit, in unit selection, is made of a *source feature sequence* and a *target feature sequence*, both of a certain length, the *unit width* (this work, following [ZJWS14], uses units with a width of 15 frames). The target feature sequence is made up of vectors of audio data, from which the output sequence is eventually generated. The source feature sequence contains the features used in the selection process to determine which unit fits best. In the case of EMG-to-Speech conversion, the source feature sequence contains the EMG-TD15-LDA data, as described in section 2.2.3 and the target feature sequence contains MFCC and, if F_0 data is mapped along with the MFCCs F_0 data, as introduced by section 2.1.3.

In this work, the basic unit codebook is created by extracting units from a corpus of synchronous recordings of EMG and speech data. This is done by first performing feature extraction on the complete data sequences, then extracting segments as long as the unit width from both, which together make up a single unit. This is done starting at every frame in every utterance in the training set for the corpus, resulting in a large codebook of units for the unit selection algorithm to choose from. Figure 3.1 illustrates the codebook creation process.

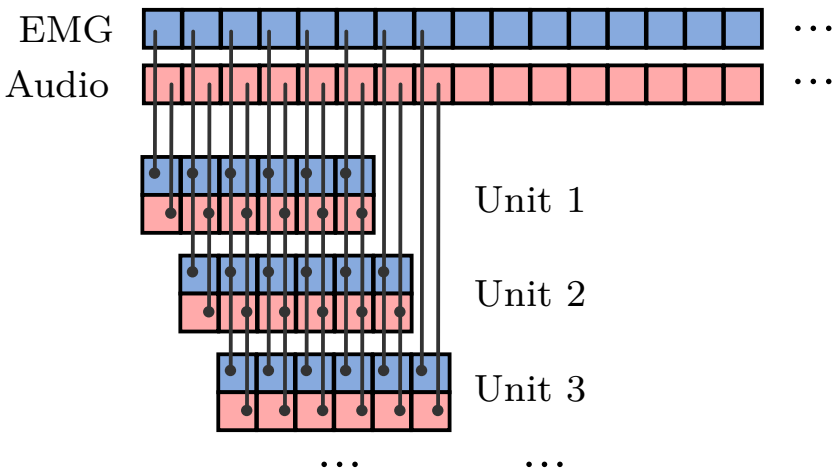

Figure 3.1: Creating a unit codebook from parallel feature sequences, with an exemplary unit width of 6. Note that each square represents an entire feature vector.

3.2 Conversion Process

To use a prepared unit codebook to convert a sequence of EMG feature vectors to an audible utterance, a four-step process is used: First, the EMG feature sequence is cut up into a sequence of *test units*, then, unit selection is performed on the test units, the resulting target sequences are overlapped to create a result audio feature sequence, and this audio feature sequence is then re-synthesized, creating audio output.

3.2.1 Test Unit Creation

The process of creating test units is similar to the codebook creation process: Segments as long as the unit width are extracted from the EMG feature sequence, though this time, the *unit shift* between segments is not necessarily chosen as 1 frame, but can be varied (In this work, a unit shift of 2 frames, empirically determined to be a good value by [Zah14], is used).

3.2.2 Unit Selection

The test units contain only a source feature sequence, so appropriate audio data has to be found. This is done by finding the sequence of units from the unit codebook which minimizes the average *cost* - made up of a weighted sum of a *target cost* and *concatenation cost* - given the test units.

The target cost is computed by evaluating a cost function for a test units source feature sequence and a codebook units source feature sequence. It ensures that units that match the input EMG data are selected.

The concatenation cost, on the other hand, makes sure that the resulting output sequence sounds reasonably natural. It is computed by evaluating a cost function for the overlapping frames of two adjacent candidate output units from the codebook. Figure 3.2 illustrates how the cost functions for selecting one unit are evaluated.

The cost function, in each case, is a single-valued function that takes as input two feature sequences and computes a similarity measure. Different cost functions can

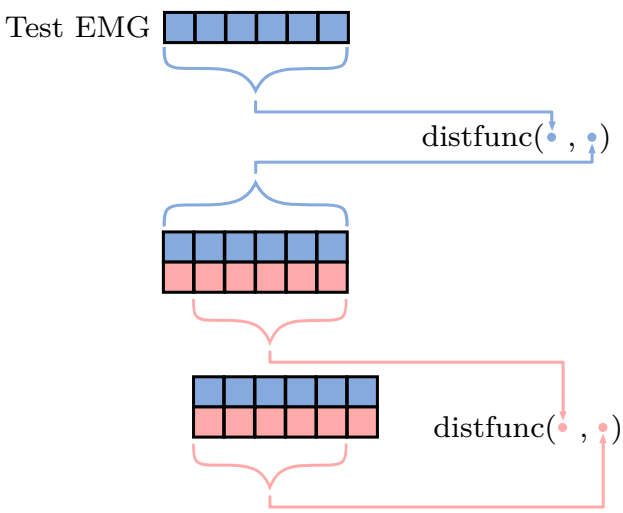

Figure 3.2: Illustration of how target (top) and concatenation (bottom) cost are calculated during unit selection.

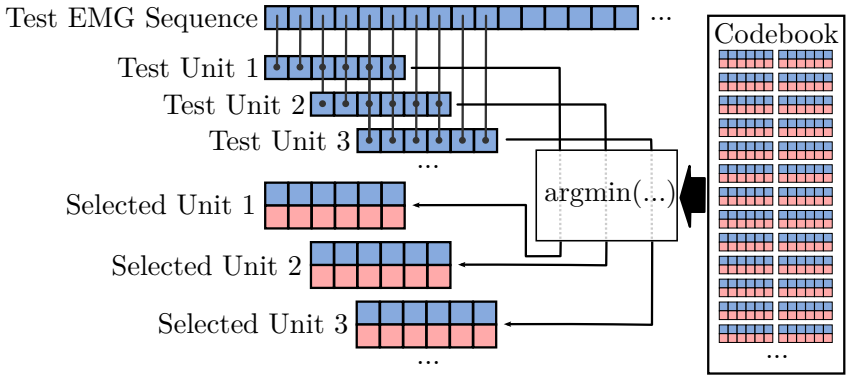

Figure 3.3: Creating a list of test units and selecting units from the unit codebook that match these test units, shown here with an exemplary unit shift of 2.

be used to select units (i.e. an euclidean distance, a cosine distance, ...), and section 5.3.2 explains some possible cost functions in more detail. Figure 3.3 shows the process of creating test units and selecting appropriate units from the unit codebook.

If the weight for the concatenation cost is non-zero, the minimization of the cost for an utterance requires finding the Viterbi path through a graph where each codebook unit is a state, with transition weights and observation likelihoods according to the cost functions used. As the graph is fully connected and can be relatively large, it is advisable to restrict the search beam to a limited number of active paths [ZJWS14].

If the concatenation cost is set to zero, leaving only the target cost as the selection criterion, the process becomes simpler: In this case, it is sufficient to independently select the unit with the lowest cost for each test unit. It is, of course, also possible to employ a similar strategy when the concatenation cost weight is more than zero: A greedy selection of units according to cost, while not as good as a Viterbi search, can deliver reasonable results very quickly.

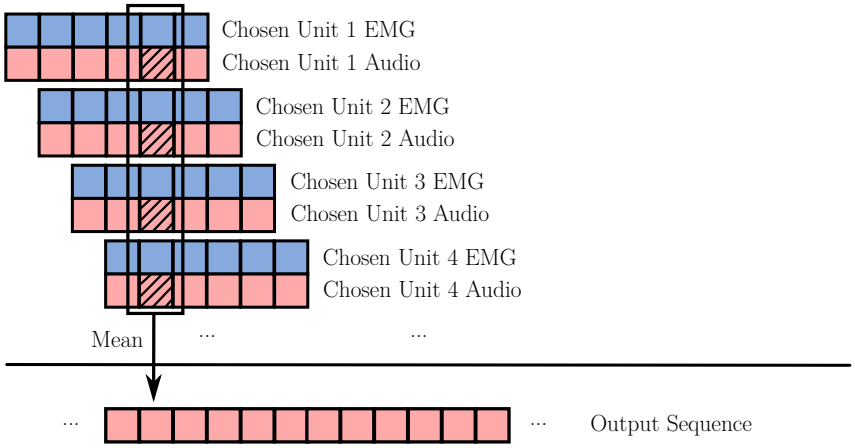

Figure 3.4: Creating audio output by overlapping of the target feature sequences followed by MLSA synthesis, shown here with an exemplary unit shift of 1.

3.2.3 Overlapping

The result of the unit selection process is a set of units with target (audio feature) sequences - however, due to the unit shift being smaller than the unit width, there is always (except for the first and last frames) more than one frame of audio data from the units target feature sequences for every frame of input data. To create the output feature sequence, all frames that match the same input frame need to be coalesced into a single frame of data. This is done by taking their mean.

3.2.4 Synthesis

Finally, after an audio feature sequence has been obtained, to generate actual audio output, the audio features have to be turned back into proper wave audio, essentially reversing the feature extraction process.

Analogous to the feature extraction process, this work uses *Mel-Log Spectrum Approximation* (MLSA) [FTKI92], using the synthesis tools from the HTS speech synthesis toolkit to perform this step [TZB02]. Figure 3.4 illustrates the process of creating audio output from a list of selected units.

4. Unit Clustering

Basic unit selection, as described in chapter 3, already provides a reasonable way to convert EMG data to Speech, but it is not without its faults. This chapter describes potential problems with the unit selection approach and the way this work tries to address them.

4.1 Problems with Basic Unit Selection

Unit selection, in speech synthesis as well as in EMG-to-speech conversion, creates an output sequence by direct concatenation of data seen in the training set. Unlike parametric synthesis, which can potentially create good output even when the test data does not match the training set particularly well, unit selection tends to break down in the presence of data for which the unit database does not contain well-matching units.

Another problem can occur when a unit has a low target cost for test units that do not actually have a sequence similar to the desired target sequence, which can occur either due to inadequacy of the target cost function or in the presence of artifacts (e.g. a weak signal due to bad electrode contact might be confused with silence, similar externally induced EMG or audio artifacts might be present in two otherwise unrelated units, ...). This can lead to *outlier units* which, due to the nature of unit selection, can lead to "stuttering" over a large section of the output.

It is for these reasons that, given an utterance to convert, unit selection generally produces either very good and natural sounding results, or (still very natural sounding) unintelligible nonsense. For unit selection based EMG-to-speech conversion, this can be such a problem that it is in fact better to not even consider the concatenation cost at all during selection, as even with very little weight assigned to it, it can dominate the selection process [ZJWS14].

This leads to two obvious ways in which unit selection might be improved: Increasing the size of the unit codebook, and improving the quality of the codebook units.

Adding more units to the codebook seems like an easy way to achieve better results, but is difficult in practice: Having a larger number of units does not necessarily

improve the quality of the unit selection conversion output unless they match the units used in testing. Due to inter-person and even inter-session differences in the EMG signal, caused by variation in electrode positioning and skin properties [WSJS13], simply creating a very large codebook from multiple recording sessions is insufficient.

4.1.1 Preliminary Experiment: Codebook Size Reduction

Even if differences between sessions were accounted for, it is not at all clear that simply adding more units to the codebook actually improves the output results: If the added units are very similar to units already in the codebook, all that their addition achieves is greater redundancy, which does nothing to improve the quality of the output. Another downside of simply adding more units is that - since performing unit selection requires the calculation of the cost function at least once for every pair of test- and codebook units - a large codebook is undesirable when working towards real-time conversion.

In a preliminary experiment, we evaluated the effect of reducing the codebook size dramatically. The experiment was performed on a single session, Spk1-Array - for details on the data and basic unit selection setup, refer to section 5.1 in the next chapter of this work.

Unit selection conversion was performed within the training set in a cross-evaluation configuration: For every training utterance, a unit codebook was created from the training set with that utterance held out. Unit selection conversion was then performed for the held out utterance. Finally, the list of units used to create the output was saved. The process was then repeated for with audio features as both source and target feature sequence and using an euclidean distance as the cost function (Since source and target feature sequences are identical, this configuration outputs the best possible audio sequence that can be created with the units from the unit codebook).

By combining the lists of used units and then culling units that were used very little or not at all, it is possible to reduce the codebook to a fraction of its size. This culled codebook was then used to evaluate conversion quality on the development set. Figure 4.1, a histogram of unit counts (i.e. with the bars of bin number n showing how many units were used n times when converting an utterance in during the cross-evaluation process), clearly shows that the majority of units is hardly used at all during conversion.

In our preliminary experiment, we reduced the codebook to only units that were used at least once both in the audio-audio conversion (leaving only units that provide good output) and emg-audio conversion (leaving, of those, only those units that the unit selection is actually likely to pick). The size of this new codebook is 17836 units, down from 159987.

This near tenfold reduction leads to a much quicker conversion and hardly affects quality at all: The reduced codebook achieves a frame-based MCD score (Compare section 5.2) of 5.9 (Std. deviation 0.38), compared to 5.85 (Std. deviation 0.34) with the original codebook. Reducing the list of units even further by using only those of the previous 17836 units that were used at least twice in the audio-audio conversion results in a codebook with only 6931 units that still performs adequately (MCD

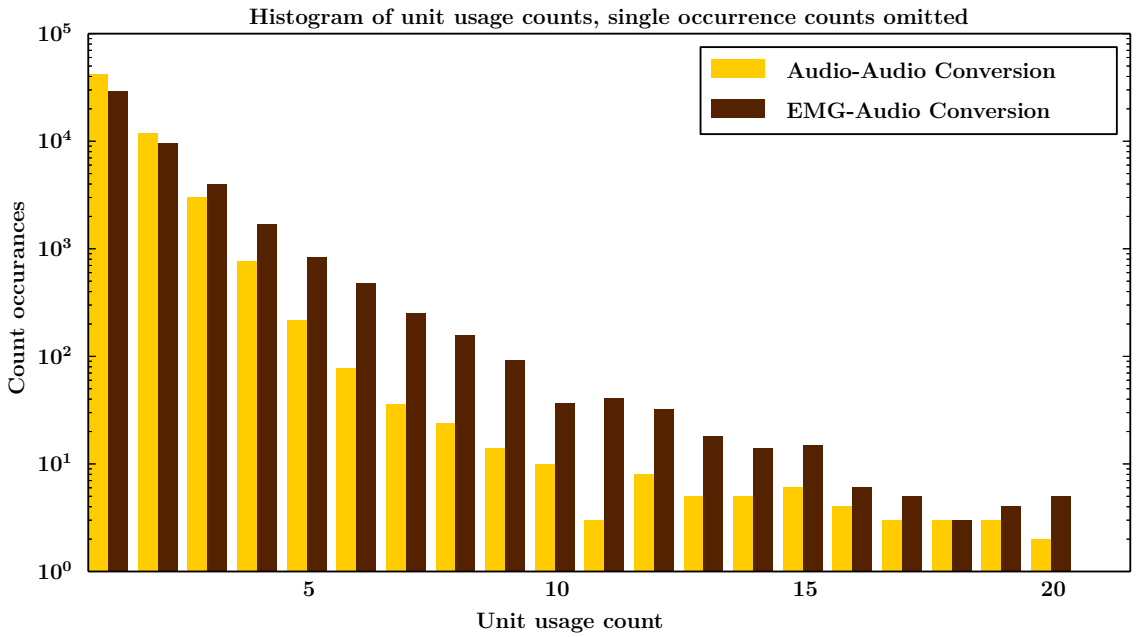

Figure 4.1: Histogram of unit usage counts. Note the logarithmic scaling of the occurrence axis. Counts that occur only once and non-occurring units omitted for clarity (Counts above 21 occur at most once, and no single unit occurs more than 45 times).

score 5.98, standard deviation 0.34), while only taking a fraction of the time the original system needed for conversion.

From these results, it is clear that reducing codebook redundancy can be done without taking a hit to quality. As the rest of this work will show, it is possible to reduce the codebook size even further than this while actually *improving* the quality of the converted speech.

4.2 K-Means Clustering

For the reasons presented in the previous section, reducing the size of the unit codebook is a desirable goal. In this work, that goal is achieved through unit clustering using the k-means algorithm. This section explains that algorithm and how it can be applied to improve the quality and performance of unit selection emg-to-speech conversion.

4.2.1 Algorithm

The K-Means algorithm [Mac03, p. 284ff] is an algorithm for clustering a set of $N > K$ D-dimensional vectors into K clusters. The algorithm consists of three steps:

1. Initialize the set of cluster centroids.
2. Assign each input vector to a cluster.
3. Recompute centroids as cluster means and repeat from step 2 until some condition is met.

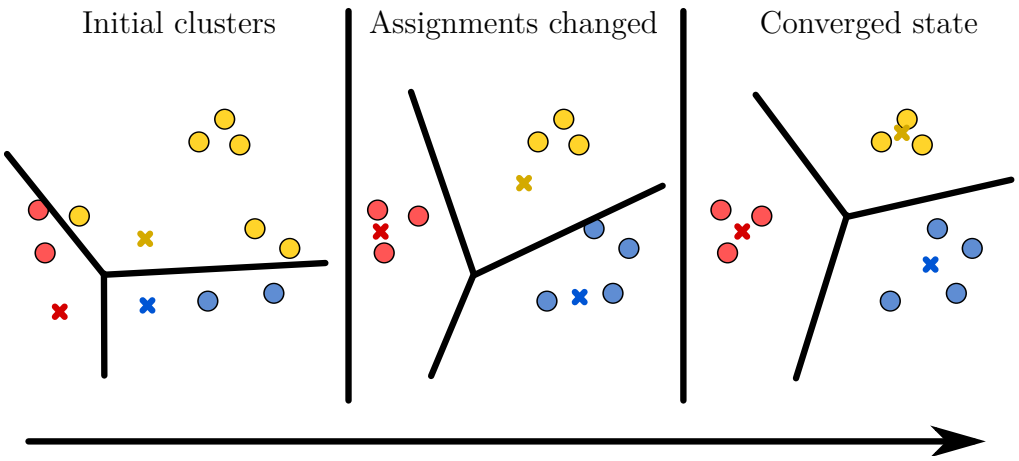

Figure 4.2: Clustering 2-dimensional data into 3 clusters using the K-Means algorithm, aborting after unit assignments stop changing.

Figure 4.2 shows an example of the K-Means algorithm clustering a set of 2-dimensional vectors into three clusters. A more detailed description of the algorithm follows. For this purpose, let x_n be the nth of N input vectors, with the per-element variance of the data normalized to 1, and $d(x, y)$ a D-dimensional metric (In this work and from now on, the euclidean distance). Let m_k denote the kth of K current cluster centroids and $A_n \in \{1..K\}$ the index of the cluster that the nth vector is currently assigned to.

Initialization

The initialization step can be performed in several different ways as long as all m_k are initialized to different values. A common method is initializing the centroids randomly. Another common method, the method used in this work, is picking the first K input vectors as initial centroids:

$$m_{1..K} = x_{1..K} \tag{4.1}$$

Assignment

In the assignment step, a new cluster assignment is computed for each input vector. This is done by finding the centroid that the current input vector has the smallest distance to:

$$A_n = \arg\min_{1..K} d(x_n, m_k) \tag{4.2}$$

Centroid Re-Computation

The new assignments are now used to re-compute the centroid for each cluster as the arithmetic mean of all the vectors assigned to it:

$$m_k = \frac{\sum_{n \in \{m | A_m = k\}} x_n}{\sum_{n \in \{m | A_m = k\}} 1} \tag{4.3}$$

This is repeated until a termination condition is met. One possible termination condition is running the algorithm until only some fraction of cluster assignments change. The output of the algorithm is the cluster assignments for all vectors.

It is possible to let the algorithm run until none of the assignments change at all: While the K-Means algorithm is not guaranteed to find the assignment that minimizes the distance of the vectors to their assigned centroid globally, it is guaranteed to converge to a local optimum as long as the re-computation step only ever reduces average distance of vectors in a cluster to the centroid according to the metric used (this is the case for the euclidean metric).

In this work, clustering was terminated when the assignment of less than 0.1 percent of vectors changed during an iteration. To be able to run experiments quickly, a parallelized GPU implementation of the K-Means algorithm, using the NVidia CUDA toolkit, was used [Giu].

4.3 K-Means Codebook Clustering

As the preliminary experiment in section 4.1.1 has shown, reducing the codebook size without impacting quality is feasible and does not require the use of any particularly complicated method - simply restricting the codebook to units that are, in some sense, good, is already sufficient. However, this simple method only achieved the same output quality and did not, in fact, improve it.

The goal of our *k-means codebook clustering* approach is, then, to not only reduce the size of the codebook, but also to improve the makeup of each single unit. In this way, we hope to be able to use less units to generate better output.

4.3.1 Principle

The main reasons why unit selection sometimes produces bad results, as explained in section 4.1, are the lack of a suitable unit in the codebook and the selection of a wrong unit from the codebook even though a better one would have been available. Codebook clustering addresses both of these problems by turning the *basic units* from standard unit selection, as introduced in chapter 3, into *cluster units*, which are more prototypical of the audio- and EMG snippets which they represent.

4.3.2 Cluster Unit Creation

Cluster units are created by first applying the k-means algorithm to some combination of the feature sequences of each unit from the set of basic units. The cluster assignments resulting from this are then used to generate cluster units - one cluster is turned into one cluster unit by computing the dimension-wise arithmetic mean of all source- and target feature sequences from units assigned to that cluster (Compare figure 4.3). The resulting units are output, and can be used as a new codebook to perform unit selection as before (refer to section 3.2.2 for details about the selection process).

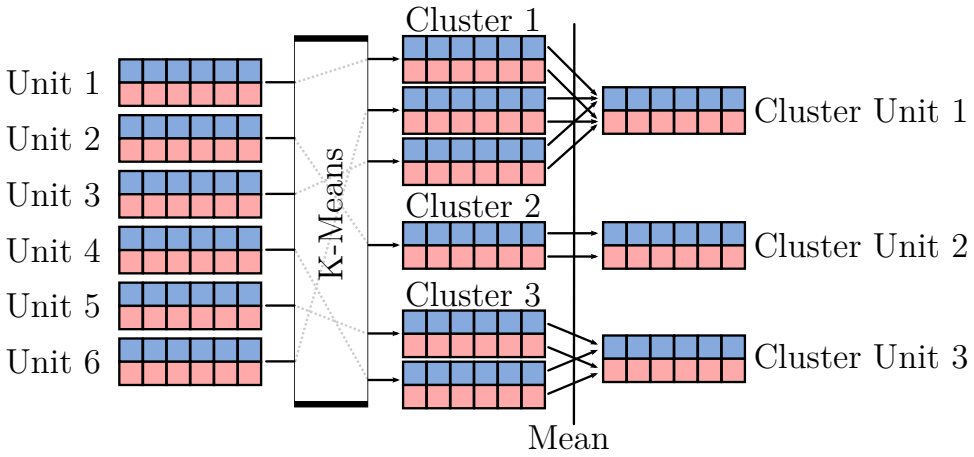

Figure 4.3: Creation of cluster units from basic units. The units are first assigned a cluster by performing K-Means clustering on some unit features, then new *cluster units* are created by taking the column-wise mean of the unit features.

4.3.3 In-Cluster Selection

While cluster unit selection can deliver improved selection quality and speed, it is not without problems. The clustering process averages many units that may in some cases not be perfectly aligned. This can cause a blurring of phones at phone boundaries and an overall less sharp pronunciation in the conversion output.

To address this, we can use a technique similar to how unit selection is used originally in text-to-speech synthesis. [BT97] use clustering (there, based on automatic creation of decision trees) to find similar units. They then select a cluster of units based on a target cost measure, and units from *within* the correct cluster based on concatenation cost.

The same technique can be applied with little modification to cluster unit selection: Unit clusters are selected from the codebook according to target cost between test unit EMG features and mean codebook unit EMG features, then the Viterbi path through a graph where only transitions to units from the correct cluster are allowed is computed to generate the output unit sequence, making the selection a two-step process. The rest of the conversion process is kept as before.

Figure 4.4 illustrates the entire in-cluster selection process. Compare this to figure 3.3, which demonstrated the unit selection process in basic unit selection. The selection process for cluster unit selection without in-cluster selection is the same as this basic unit selection process, except with the unit codebook filled with the cluster units.

This in-cluster selection allows to retain the benefits of clustering while still generating crisp audio output. An additional benefit gained is that, as the usage of units that match the test units is ensured by selecting only from matching clusters, it is now feasible to use a non-zero weight for the concatenation cost during conversion without overly biasing the process towards well-concatenating, but ultimately unfitting units.

In addition to qualitative benefits, the restriction of the Viterbi search to fitting units, while still being slower than a simple greedy algorithm, allows the search for

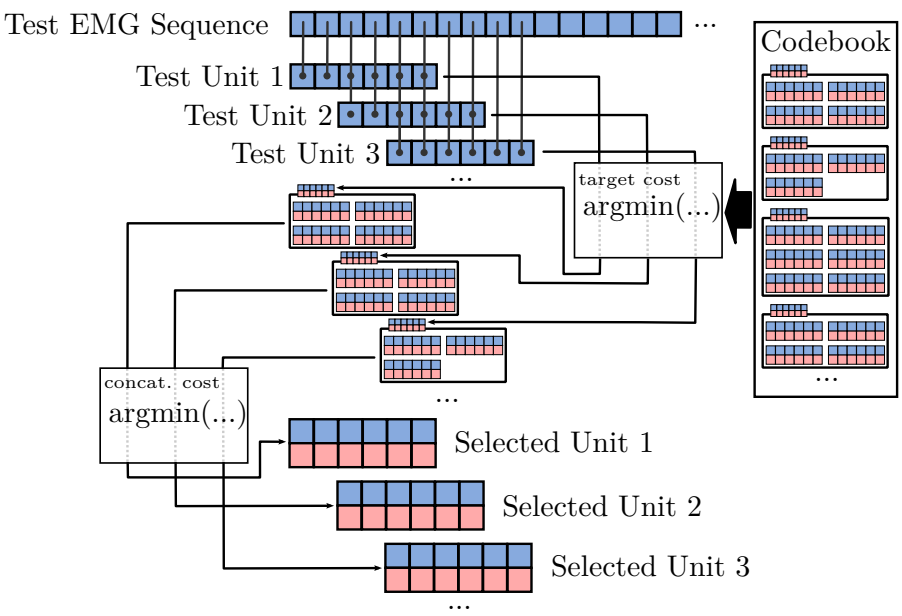

Figure 4.4: Cluster unit selection with in-cluster selection: In step 1, clusters are selected by the target cost of the mean unit. In step 2, units assigned to the selected clusters are selected by concatenation cost.

the optimal unit to complete in less time than if the entire unit codebook would have to be considered.

5. Evaluation

This section describes the corpus used in our experiments and the process and results of tuning the parameters of the proposed cluster-based unit selection method, followed by a final evaluation of the method on unseen data, objectively using an audio distance measure and subjectively using a listening test.

The baseline system, in this chapter, is always the basic unit selection EMG-to-speech conversion system [ZJWS14], as described in section 3, using the best-performing distance from that work (the cosine similarity) except in section 5.3.2, where two new distance functions are compared to the cosine similarity.

5.1 Data Corpus

The corpus used in this work is based on the corpus used in [ZJWS14], to allow for the comparison of results. In addition to the four sessions used there, this work also uses two new, larger sessions to evaluate the performance of unit selection and cluster-based unit selection on large training data corpora.

The corpus consists, in total, of six sessions of phonetically balanced English speech, recorded by two male speakers and one female speaker. For each utterance, the corpus contains synchronously recorded EMG signals and audible speech.

For the EMG recordings, two separate recording setups were used: A single-electrodes setup, and an electrode array setup, first used for EMG-to-speech conversion by [WSJS13]. The audio signal was, in both cases, recorded with a standard close-talking microphone and sampled at 16 kHz.

5.1.1 EMG Recording Setups

The single-electrodes setup uses a set of 10 electrodes, 3 of which are used in a unipolar configuration with a reference electrode on the mastoid portion of the temporal bone (behind the ear) and 6 of which are used in a bipolar configuration, for a total of 6 channels. The electrodes were, based on the work of [MHMSW05],

Figure 5.1: The single-electrodes recording setup, with electrode positions marked. White numbers indicate bipolar derivation, whereas black numbers indicate unipolar derivation against a reference electrode attached behind the ear.

positioned to capture signals from important articulatory muscles (compare figure 5.1, numbers here correspond to numbered electrodes or electrode pairs):

2, 3 the levator anguli oris,
2, 3 the zygomaticus major,
4, 5 the platysma,
5 the depressor anguli oris,
1 the anterior belly of the digastric and
1, 6 the tongue.

The signals from these electrodes were captured using a 6 channel EMG recording system (*Becker-Meditec Varioport*), filtered with a high-pass filter at 1 Hz for DC offset removal and then sampled at 600 Hz.

For the array recording setup, we used two electrode arrays: One large array, with 4x8 electrodes arranged in a regular grid with an inter-electrode distance of 10 mm, and one small array with 8 electrodes in a linear configuration, with an inter-electrode distance of 5 mm.

The large array was positioned on the cheek, capturing signals from muscles responsible for tooth and lip movement, while the small array was positioned under the chin, capturing signals from the tongue. Figure 5.2 illustrates the placement of the electrode arrays.

The electrodes in the arrays were used in a bipolar derivation configuration, with one channel recording the difference between two adjacent electrodes within a row. This gives 7 channels per row for the large array, and 7 channels for the small array, for a

Figure 5.2: The array recording setup, with a 4x8 electrode array on the cheek and a smaller 1x8 electrode array under the chin.

total of $4 * 7 + 7 = 35$ channels. The signals for all channels were recorded using an *OT Bioelettronica EMG-USB2* EMG recording system, sampled at 2048 Hz.

During a recording session, each single utterance to be recorded was displayed (as text) to the speaker being recorded, who was allowed to record the utterance at their leisure, using a push-to-talk setup to start and end utterance recording. Speakers were allowed to re-record utterances if desired. The recording was supervised by a recording assistant, who also attached the recording electrodes before the start of the recording session and verified that electrodes did not detach from the skin during recording by visual inspection of the recorded EMG signal.

5.1.2 Signal Synchronization

Training an EMG-to-speech mapping system requires parallel EMG and speech data. As delays introduced by the signal processing and recording hardware could cause one signal to lag behind the other, one channel in each setup was used to record a hardware marker, to allow for synchronization of the signals after recording. This was achieved by simultaneously pulling the marker channel for all setups high at the start of each utterance.

After recording and before any further pre-processing, this marker was used to determine a common cutting point for each utterances EMG and audio signal, compensating for any lag that different recording devices and software may have caused.

5.1.3 Session Details

The corpus consists mostly of sessions of around 500 phonetically balanced English utterances, based on [SW10]. The two larger sessions additionally incorporate utterances from the Arctic [KB04] and TIMIT [GC+93] corpora, giving a total of

Session	Accumulated data length, in (mm:ss)			# of train/eval/dev utterances		
	Train	Eval	Dev	Train	Eval	Dev
Spk1-Single	27:10	01:19		500	20	
Spk2-Single	26:54	00:49		496	13	
Spk1-Array	31:01	01:59	00:47	500	30	10
Spk2-Array	25:44	01:10		500	20	
Spk1-Array-Large	76:44	00:48		1093	10	
Spk3-Array-Large	123:04	00:45		1968	10	
Total	310:37	06:50	00:47	5057	103	10

Table 5.1: Data corpus information for the recorded utterances, including speaker/session breakdown. Speaker 1 and 2 are male, speaker 3 is female.

1103 utterances for the smaller and 1978 utterances for the bigger of these two new sessions.

The sessions were split into a training set, used to train the mapping system, and an evaluation set, used in the final evaluation to judge system performance. For one session, a development set was held out from the training set. This development set was used during the development and parameter tuning of the system.

Table 5.1 gives a detailed overview of the sessions and their breakdown into training, evaluation and development sets.

5.2 MCD Score

To objectively evaluate conversion results, a measure of the difference between conversion result and reference is needed. Ideally, this measure should be strongly correlated with the *naturalness* and *intelligibility* of the converted utterance. This work uses the *mel-cepstral distortion* (MCD) score [Kub93] to perform such evaluations. The MCD score is defined as a scaled Euclidean distance between MFCC vectors excluding the first coefficient:

$$\mathbf{MCD} = 10/\ln 10 \sqrt{2 \cdot \sum_{k=2}^{25} (\mathbf{mfcc}_{conv}[k] - \mathbf{mfcc}_{ref}[k])^2} \tag{5.1}$$

Calculating an MCD score frame by frame requires that the audio files being compared are properly aligned to each other. When evaluating frame-based conversion methods, this is easily ensured. In unit selection, it might be desirable to compute a better alignment. For this reason, this work considers not only the MCD score computed frame by frame, as above (Referred to as the "Frame-Based MCD" from now on), but also the "DTW-MCD" score after the MFCC sequences of conversion result and reference have been aligned using the dynamic time warping (DTW) algorithm [Vin68] minimizing MCD.

5.3 Cluster Unit Selection Parameters

Unit selection based on unit clustering, as described in chapter 4, has many parameters that affect the performance of the system. To choose reasonable parameters for the evaluation of the system, we ran several series of experiments on our development set. This section will describe the parameters that needed to be set, the experiments we ran to determine them, and the parameters chosen for the evaluation on the evaluation data.

5.3.1 Clustering Features

The clusters found by the K-Means algorithm depend on what set of features is passed in - it will find different cluster assignments when clustering by audio, by EMG data, or by both. This section evaluates five different ways of clustering units for unit selection: Sequential Audio-EMG clustering, combined Audio-EMG clustering, EMG-only clustering, Audio-only clustering and label-assisted clustering. All evaluations in this section are performed with the basic cluster-based unit selection, i.e. without in-cluster selection.

Sequential Audio-EMG clustering

The sequential Audio-EMG clustering approach was the first approach that this work evaluated. As the name implies, clustering is done sequentially in two steps. First, all units are clustered based on the contained MFCC feature vectors. Within each cluster, a second iteration of clustering is performed, this time according to the contained units EMG-TD15-LDA feature vectors.

This approach of splitting the clustering into two steps has several advantages. One is that the quality improvements of clustering on the audio side versus clustering on the EMG side can be individually evaluated, with different cluster counts. Another is that, since the complete unit set only has to be considered in the first step when clustering with the comparatively low-dimensional MFCC vectors, clustering is quick.

The results of performing unit selection based on the sequential clustering of units, for a large selection of cluster counts for MFCC and EMG, can be see in figure 5.3 (Note that, while figure 5.3 reports, for illustration purposes, both frame-based and DTW-MCD, both MCD scores are proportional, which was also the case for all other series of experiments. Therefore, for brevity, the rest of this section will only report the frame-based MCD).

One thing that can be seen right away is that the clustering unit selection approach beats the baseline systems MCD score by a large margin even with a relatively small amount of units: The system using 150 MFCC clusters and then, within each MFCC cluster, 5 EMG clusters, for a already beats out the baseline systems frame-based MCD score by around 0.35 points - with only 750 units, an order of magnitude less than the baseline system.

Looking more closely, two trends can be identified: Using more audio clusters improves the MCD score, while using more EMG clusters worsens it. This raises the question of whether it is better to cluster based on audio data alone, something that will be investigated in section 5.3.1.

Unit Length

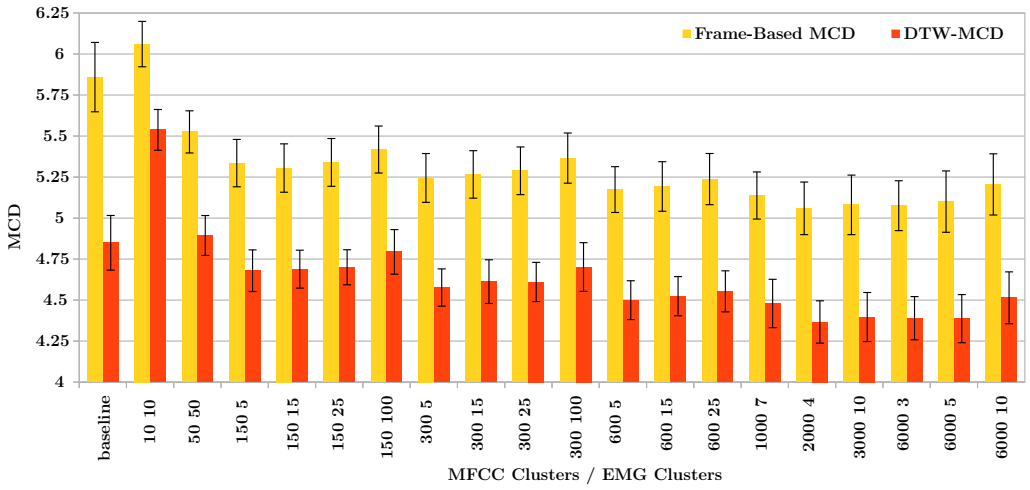

Figure 5.3: Cluster unit selection with split clustering, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

To verify the results of [ZJWS14] concerning optimal unit length on our new cluster-based unit selection system, we tested the system, using sequential Audio-EMG clustering, with units of a shorter length than the usually used 15 frames per unit. The results of this evaluation, using 7 frame units, can be seen in figure 5.4.

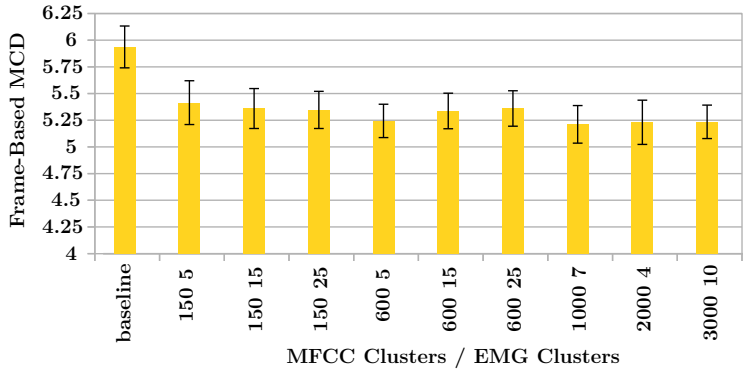

Figure 5.4: Cluster unit selection on 7 frame units with split clustering, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

Our results are similar to those of [ZJWS14]: The system using the shorter units performs worse than the system using the long units. We therefore decided to use long units in the rest of our evaluation.

Combined Audio-EMG clustering

While the sequential clustering approach is very interesting for an initial investigation into the impact of using clustered units in unit selection, it has two obvious drawbacks when trying to achieve good conversion quality. One is that there are two cluster counts to optimize instead of one, complicating the search for good parameters. The other is that, even with very good cluster counts, a best cluster assignment obtained by clustering only on parts of the vectors to be clustered is not necessarily even locally optimal for the entire vector.

For this reason, we evaluated combined clustering: Clustering the units based on the entire combined feature vector of MFCC and TD-15 data. The results of unit selection based on units obtained through combined clustering can be seen in figure 5.5.

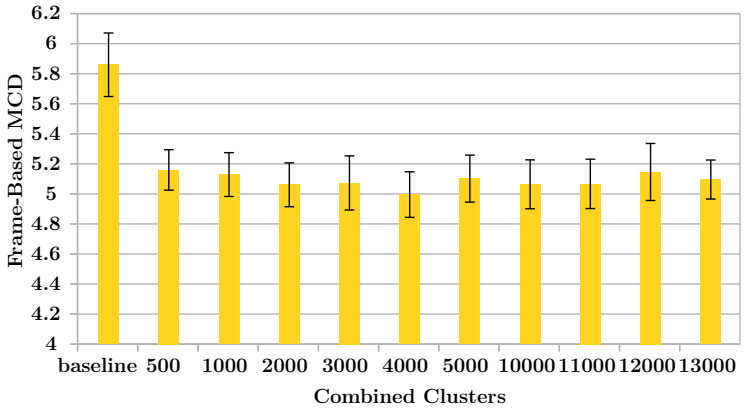

Figure 5.5: Cluster unit selection with combined EMG-MFCC clustering, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

As can be seen, the more computationally expensive combined clustering approach does pay off in quality: The best combined clustering based system manages to achieve a better MCD score than any of the split clustering systems evaluated.

Another trend that can be seen in the combined clustering is that of an u-shape curve when evaluating for different cluster counts: Using too few clusters does not yield optimal results, but after a certain point, using any more units diminishes the effects of the clustering, and the MCD score begins to rise again.

EMG-only clustering

In section 5.3.1, a larger number of EMG clusters actually *worsened* the resulting output. To further investigate this phenomenon, we performed clustering based on the units EMG feature vectors only. The result of using thusly clustered units in unit selection are shown in figure 5.6.

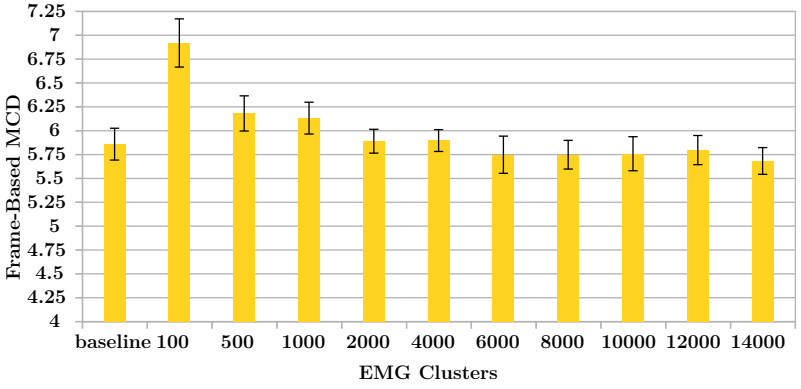

Figure 5.6: Cluster unit selection with EMG (TD15-LDA) only clustering, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

As expected, the performance is sub-par, with none of the systems achieving a significant improvement over the baseline, and some actually being worse. We suspect that this is due to clustering based on euclidean distances not being able to capture similarities in the EMG-TD15-LDA data well. This is supported by findings from [Zah14], which found that the euclidean distance was not a good target cost for unit selection based EMG-to-speech conversion (For further evaluation of target cost functions, refer to section 5.3.2).

Audio-only clustering

In the sequential clustering approach in section 5.3.1, a larger number of MFCC-side clusters seemed to improve performance, while a larger number of EMG-side clusters seemed to worsen it. Taken to it's logical conclusion, this gives MFCC-only clustering. The results of performing unit selection with cluster units obtained by clustering based on only the MFCC vectors of the units can be found in figure 5.7.

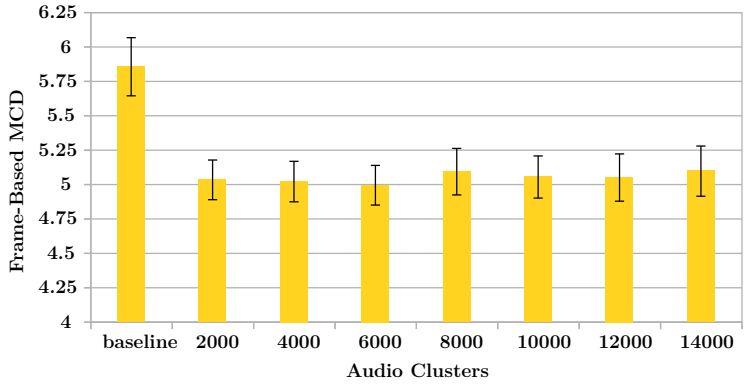

Figure 5.7: Cluster unit selection with MFCC-only clustering, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

It can be seen that the audio-only clustering system achieves a quality of output that is similar to that of the combined clustering system. While the performance is not better, audio-only clustering does have one major advantage over combined clustering: As the features used in clustering are only the 25 MFCCs, clustering is much less resource-intensive. With quality being the same and clustering requiring far less memory and time, we decided to use audio-only clustering as the basis for all further investigations.

Label-assisted clustering

The idea behind the label-assisted clustering approach is to use additional information to aid the clustering process. In unit selection based speech synthesis, the source features for unit selection are phone-based - either plain phone label or phonetic features (such as the the position where a phone is formed or its voicedness, compare section 2.1.1).

To perform label-assisted clustering, we first generated boolean phonetic feature vectors from the label data aligned to the MFCC feature vectors - as described in section 2.2.4 discussing LDA prepossessing.

For the phone-assisted clustering, this was done by generating a 45-element vector for each frame where each vector element corresponds to one of the 45 English phones

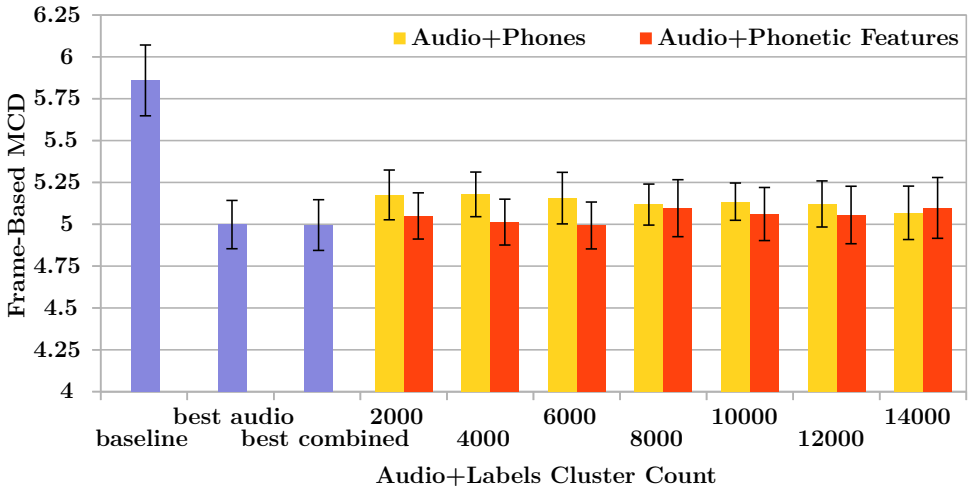

Figure 5.8: Cluster unit selection with clustering based on MFCCs augmented with labels, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

used for labeling in this work. The vector entries were set to 1 if the frame was labeled as the corresponding phone and to 0 otherwise. For phonetic-feature-assisted clustering, similar vectors - this time of length 78 - were generated, where each vector element corresponds to one phonetic feature and is set to 1 if the phone the frame is labeled as is marked as exhibiting that feature and 0 if not.

For evaluation, we performed clustering on the concatenation of the MFCC feature vectors contained in the base units (as before, in audio-only clustering) and the aforementioned boolean vectors indicating each frames phone or each frames phones phonetic features, still based on the k-means algorithm using the euclidean distance. The results of using units clustered in this way in unit selection, compared to the best combined-clustering and best audio-only clustering systems, can be seen in figure 5.8.

The phonetic-feature augmented clustering achieves a performance similar to that of the best previous systems, while the phone augmented clustering does slightly worse. This could be due to the fact that, as the label information is already used in the LDA pre-processing, using that very same information to try and further improve the selection process is not beneficial.

5.3.2 Target Cost Functions

[Zah14] found that the euclidean distance is not a good target cost function to use in unit selection based EMG-to-speech conversion. They instead found that, out of the target cost functions evaluated (euclidean distance, cosine distance), the cosine distance provided the best results.

In this work, we evaluated two additional target cost functions, evaluating them on the audio-only clustering system: The maximum-norm distance, and the sum-norm distance. Figure 5.9 illustrates these two in comparison to the euclidean norm and the cosine distance. To calculate the target cost for a single codebook unit given a test unit, these distances are evaluated between all pairs of parallel EMG frames of the test- and codebook unit being compared (compare figure 3.2 earlier in this work) and then averaged.

Note that this evaluation concerns the *target cost* only: The clustering method remains unchanged.

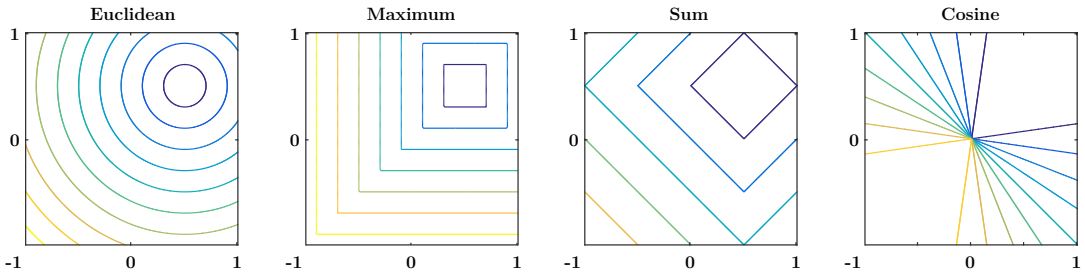

Figure 5.9: Different distance functions evaluated for the target cost, illustrated as lines of equal distance to the vector $[0.5, 0.5]$.

The maximum norm distance $L_\infty(a, b)$ between two N-dimensional vectors a and b is obtained by calculating the difference of the two vectors being compared, then finding the absolutely largest element of the difference:

$$L_\infty(a, b) = \max\{|a_1 - b_1|, ..., |a_N - b_N|\} \tag{5.2}$$

The Manhattan distance $L_1(a, b)$ of two vectors a and b of dimensionality N is obtained by taking the difference of the two vectors and summing their elements absolute values:

$$L_1(a, b) = \sum_{n=1}^{N} |a_n - b_n| \tag{5.3}$$

We compared these two to the best performing target cost from [Zah14], the cosine similarity $\mathrm{cs}(a, b)$, defined based on the angle between the two vectors a and b being compared:

$$\mathrm{cs}(a, b) = 1 - \frac{a \cdot b}{|a| * |b|} \tag{5.4}$$

The result of our evaluation of these functions as the target cost in unit selection with audio-only clustering can be seen in figure 5.10. Clearly, the maximum norm distance performs adequately, but not better than the cosine similarity. The sum norm distance does not produce good results.

It being clear that none of the new evaluated functions outperforms the cosine similarity, we decided to use the cosine similarity as the target cost in our final evaluation.

5.3.3 In-Cluster Selection

The averaging of units which is necessary to create the cluster units used in clustering-based unit selection has implications for output audio clarity, which can be addressed by using in-cluster selection, as described in section 4.3.3. The result of an evaluation series using in-cluster selection, with audio-only clustering and emg-only clustering,

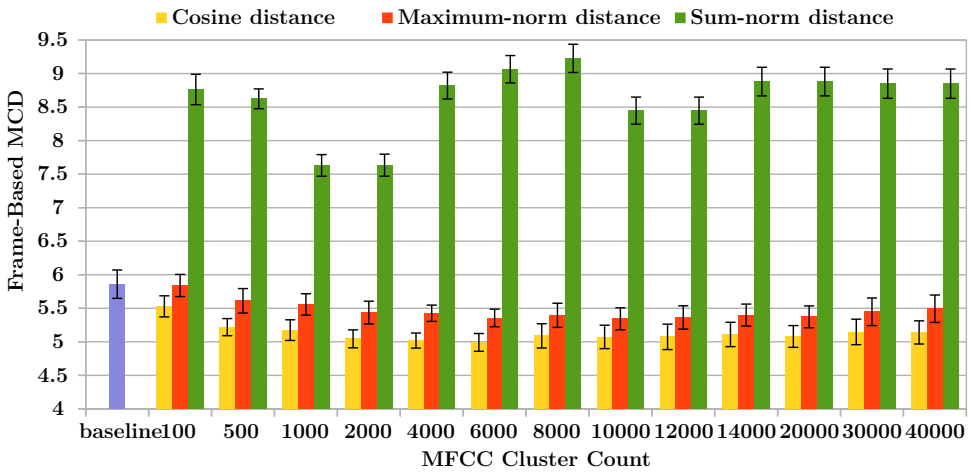

Figure 5.10: Cluster unit selection with different target cost functions - the cosing distance as the best performing target cost from [ZJWS14] as well as the two new functions L_∞ and L_1 - and cluster counts. Lower is better, error bars indicate 95% confidence interval.

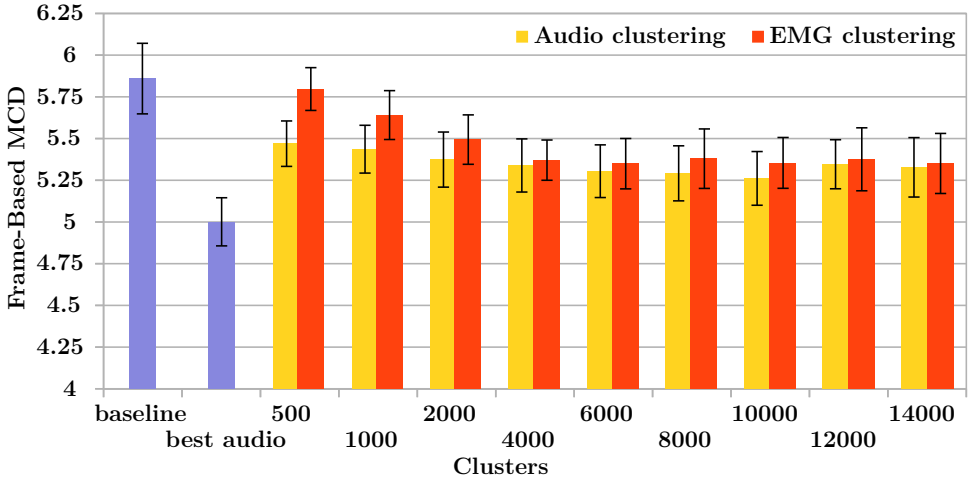

Figure 5.11: Cluster unit selection with in-cluster selection, using different cluster counts. Lower is better, error bars indicate 95% confidence interval.

cosine similarity for the target cost and euclidean distance for the concatenation cost, can be seen in figure 5.11.

The MCD score of the in-cluster selection is worse than that of the basic audio-only cluster unit selection in either case (this difference is statistically significant at a significance threshold of p=0.05).

As expected, the EMG-only clustering, which did not perform well at all in basic clustering unit selection, provides output of a comparable quality to that of audio-only in-cluster unit selection, however, the audio-only system still has a slight edge.

5.4 Final evaluation

Having evaluated several different sets of parameters and systems for our clustering-based unit selection system, this section will provide a final evaluation of the new system on the evaluation set of our data corpus.

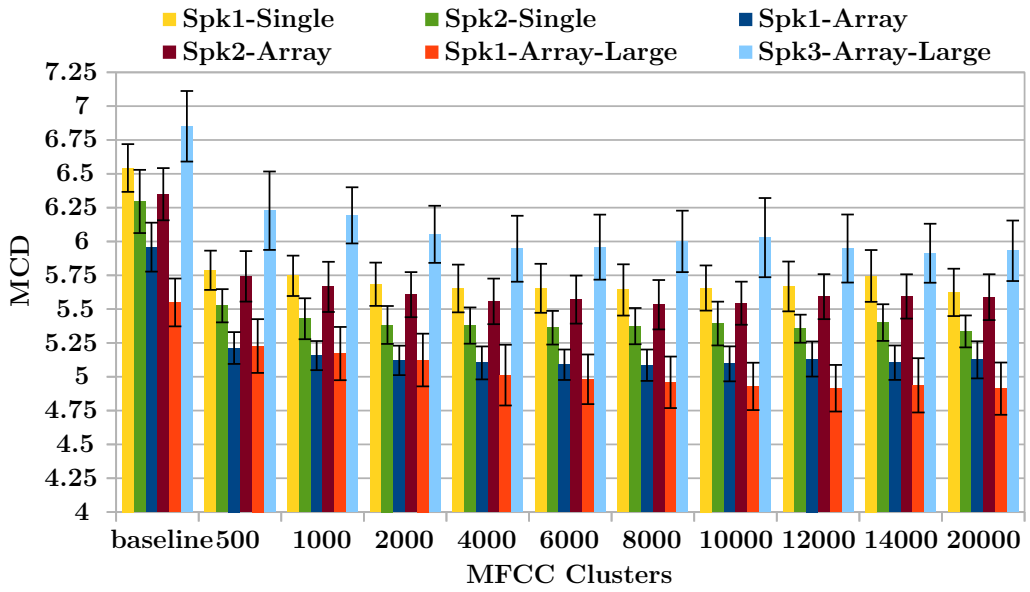

Figure 5.12: Frame-Based MCD for cluster unit selection on the evaluation set, with different cluster counts. Lower is better, error bars indicate 95% confidence interval.

We decided to perform this evaluation on the audio-only cluster unit selection system, using the cosine similarity as the target cost, and evaluating both basic cluster unit selection as well as in-cluster selection.

While the following charts will, for reference, show an entire series of evaluated cluster counts, the results reported, and the subjective evaluation, will always be using 6000 clusters, the amount of clusters determined to give the best MCD scores on the development set.

5.4.1 Objective Evaluation

Qualitative improvements

For the objective evaluation, this work uses, as in the previous section, the MCD score, this time computed on the results of performing unit selection EMG-to-speech conversion on the evaluation set data.

The results of the evaluation for the basic cluster unit selection can be seen in figure 5.12 (Frame-based MCD) and, for reference, 5.13 (DTW-MCD).

It can be seen that, though there are large differences between sessions, the new cluster unit selection system achieves a significant improvement in MCD score over the baseline system for every session (at a significance threshold of p=0.05, verified using a two-tailed paired t-test).

The cluster count selected for the evaluation - 6000 clusters - does provide good results for all sessions, and for the session whose size is similar to that of the development set, seems close to optimal. For the larger sessions, it appears that using more clusters can still improve results - this may have to be investigated further in future work.

One thing which stands out is the mean MCD score of session Spk3-Array-Large: Despite the large amount of data, which should help to improve the quality of the

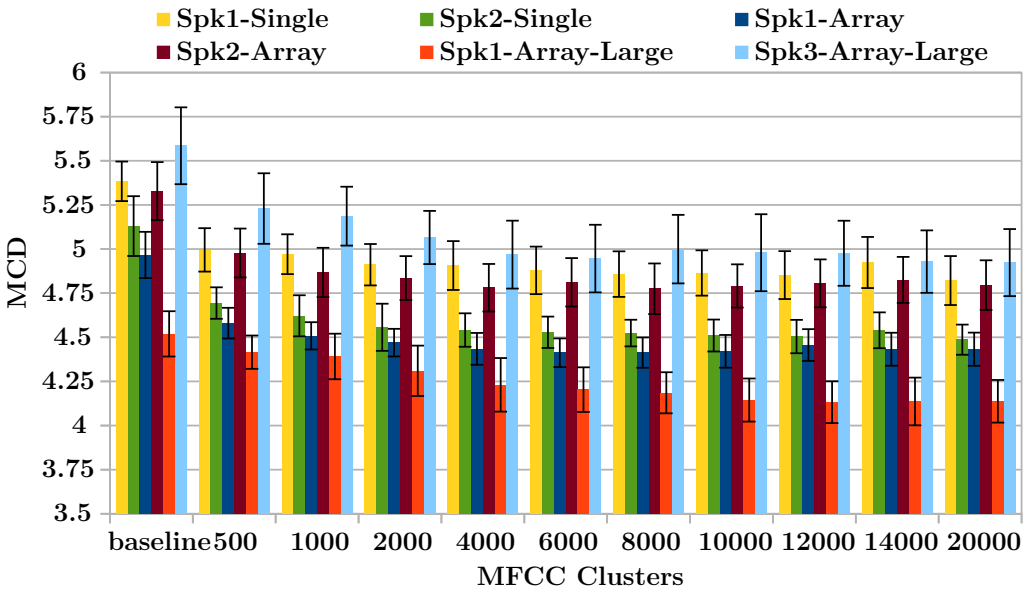

Figure 5.13: DTW-MCD for cluster unit selection on the evaluation set, with different cluster counts. Lower is better, error bars indicate 95% confidence interval.

unit selection output, the MCD is a lot worse than the other sessions for both the baseline and the cluster unit selection system. This could be due within-session variance as skin condition and speaker behaviour during speech production change during the very long recording session, as well as the larger amount of experience of speakers 1 and 2 with the recording system. The cluster unit selection approach is not able to fully compensate for this, however, it still performs much better than the baseline system.

The results for the unit selection with in-cluster selection, shown in figure 5.15 and figure 5.16, paint a similar picture: The baseline system is outperformed by the new system by a significant margin for every session evaluated (again at a significance threshold of p=0.05, verified using a two-tailed paired t-test). In addition, comparing the MCD scores of the basic cluster unit selection and unit selection with in-cluster selection shows that the basic approach consistently yields a significantly better MCD score for every session, even though the in-cluster selection system does seem to achieve the goal of creating less time blurring (illustrated for an example utterance in figure 5.14).

A noticeable difference between the basic unit selection and the in-cluster unit

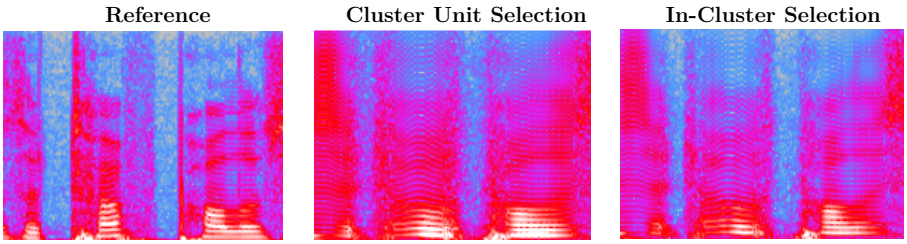

Figure 5.14: Spectrograms of a snippet of audio from utterance Spk3-Array-Large-Utt4, showing the blurring effect of cluster unit selection, and the less blurry results obtained with in-cluster selection.

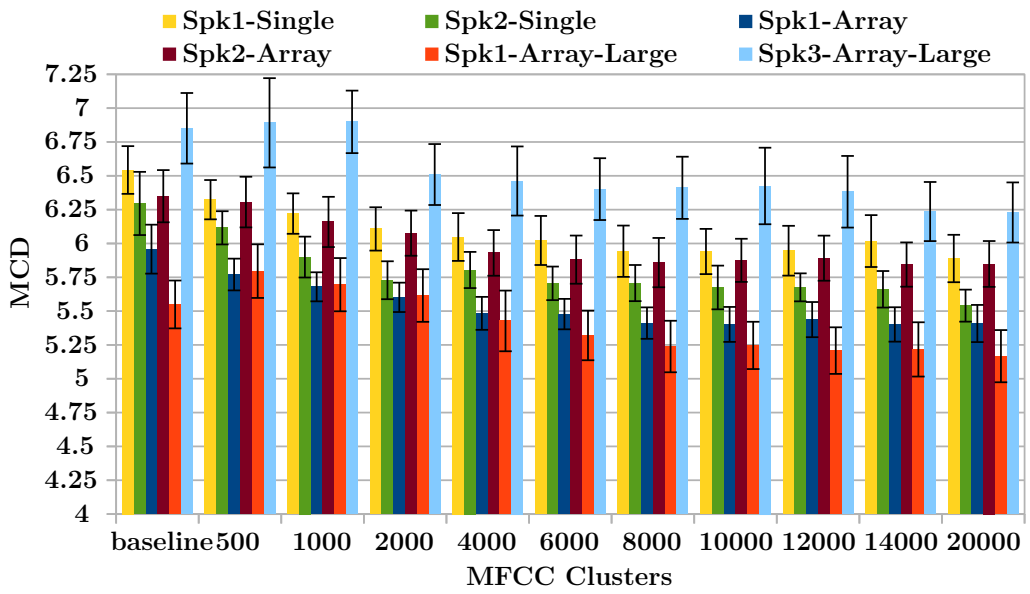

Figure 5.15: Frame-Based MCD for cluster unit selection with in-cluster selection on the evaluation set, with different cluster counts. Lower is better, error bars indicate 95% confidence interval.

selection approaches concerning cluster count seems to be that while for the cluster unit selection, 6000 clusters seems to be a good value, the same is not true for in-cluster selection: Here, for the majority of sessions, the MCD score has not stopped falling at the end of the evaluation run. It might be possible to improve the performance of cluster unit selection with in-cluster selection by using a larger amount of clusters, especially for larger sessions.

Overall, basic cluster unit selection achieves an improvement of the mean frame based MCD score (over all sessions) from 6.26 to 5.44, a 13.11 percent relative decrease (absolute improvement of the frame based MCD score of 0.82). In-cluster selection manages to improve the mean frame-based MCD by 7.27 percent relative (an absolute improvement of 0.46), down to 5.8.

The mean MCD scores for each session (for the baseline system, the 6000 unit cluster unit selection system and the 6000 unit in-cluster selection system) can be found in table 5.3 in the next section of this work.

The largest improvements were achieved for the single-electrodes sessions (with a 14.92 percent relative improvement for session Spk2-Single in the basic cluster unit selection case, from 6.54 down to 5.65), while the large sessions benefit the least (with only a 10.27 relative improvement on session Spk1-Array-Large, from 5.55 to 4.98). This relatively smaller improvement could be due to, as previously mentioned cluster count differences.

The situation is the same for in-cluster selection, with a maximum relative improvement of 9.52 percent for session Spk2-Single (6.54 down to 6.02) and a minimum improvement of 4.14 percent relative on session Spk1-Array-Large (5.55 down to 5.32).

Quantitative improvements

While the quality improvements provided by cluster unit selection are already significant, the quantitative improvements are even larger. Unit selection needs to

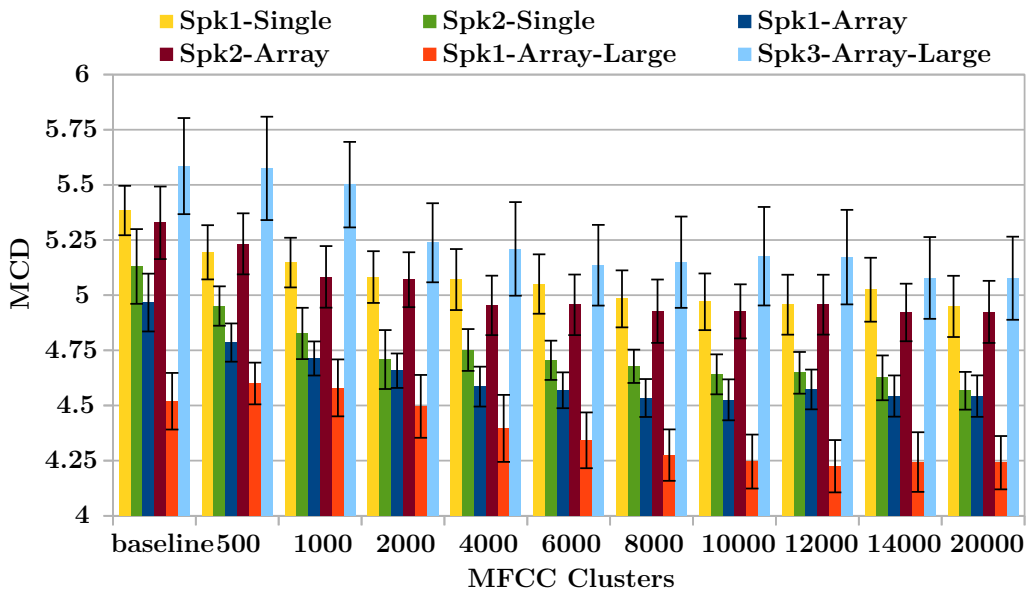

Figure 5.16: DTW-MCD for cluster unit with in-cluster selection selection on the evaluation set, with different cluster counts. Lower is better, error bars indicate 95% confidence interval.

compute, at the very least, the target cost for every pair of test unit and codebook unit. This makes converting with a large unit codebook computationally expensive, which is a problem when real time performance is desired.

Cluster unit selection can help here: As it creates a smaller codebook of cluster units (compare table 5.2 for unit counts of the baseline system), the time taken for conversion is also greatly reduced. Even with in-cluster selection, the expensive Viterbi in-cluster concatenation step is restricted to relatively few units, found during the less expensive first phase of the selection process.

The result of a quantitative evaluation of cluster-based unit selection methods for EMG-to-speech conversion can be found in table 5.3 (All times were obtained on an Intel Core i7-3770 CPU running at 3.40GHz).

Session	Training set length (mm:ss)	Baseline unit count
Spk1-Single	27:10	139542
Spk2-Single	26:54	137909
Spk1-Array	31:01	159987
Spk2-Array	25:44	119949
Spk1-Array-Large	76:44	403047
Spk3-Array-Large	123:04	634031

Table 5.2: Length of the training set for all sessions and the number of units in the baseline unit codebook created from that training set.

The improvements are drastic: While the baseline system takes 123.6 times realtime to process the evaluation set, basic cluster unit selection manages to do the same in a much better 2.33 times realtime, an improvement of 98 percent. For in-cluster selection, where, like in the baseline system (and unlike in the basic cluster unit selection system) the time taken for conversion depends in part on the training set size, the time taken for conversion is larger - 19.5 times realtime - but still a lot better than the time taken by the baseline system (a 84 percent improvement).

Session	MCD Score			Time taken for conversion (mm:ss)		
	Baseline	Clustering	In-cluster	Baseline	Clustering	In-cluster
Spk1-Single	6.54	5.65	6.02	72:33	2:25	14:56
Spk2-Single	6.30	5.36	5.70	44:02	2:06	12:30
Spk1-Array	5.96	5.09	5.48	154:39	3:47	17:16
Spk2-Array	6.35	5.57	5.88	50:31	2:41	9:31
Spk1-Array-Large	5.55	4.98	5.32	150:50	2:11	28:14
Spk3-Array-Large	6.85	5.96	6.40	241:57	1:58	28:29

Table 5.3: Mean evaluation set frame-based MCD scores and computation times for the baseline unit selection system as well as cluster unit selection without and with in-cluster selection, using 6000 cluster units.

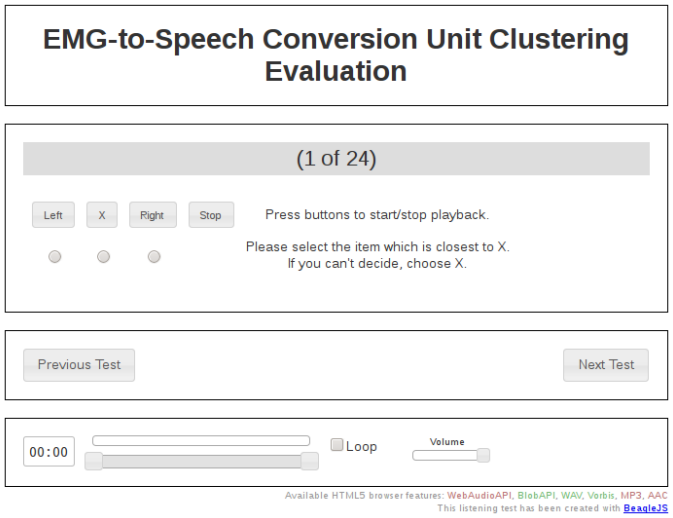

Figure 5.17: The listening test system setup used to perform subjective comparisons of our systems.

5.4.2 Subjective Evaluation

The MCD score is correlated with human perception of speech quality [Kub93]. It is, however, not a perfect measure. For this reason, we performed a set of subjective evaluations, where humans were asked to listen to the outputs of our systems and compare them. This section will describe these subjective evaluation tests.

Test Setup

All listening tests were performed as A/B comparison tests with reference: Each participant was given the ability to listen to the output of both systems being compared, along with reference audio output, and asked to decide which output they considered to be closer to the reference. A neutral option - "no preference" was provided in case a participant did not find any of the outputs to be substantially better than the other. A screenshot of the web-based system used to perform the listening tests - the BeaqleJS system [KZ14] - can be found in figure 5.17.

Utterances for the tests were selected at random from the evaluation set, selection 4 utterances from each session for a total of 24 utterances. In each test run, the utterances were presented in a randomized order and with the systems randomly being assigned to the "Left" and "Right" sides, to avoid any potential bias. As the

goal of this work was to evaluate the MFCC conversion performance only, both systems outputs were synthesized using F0 trajectories extracted from the reference audio.

Basic Cluster Unit Selection versus In-Cluster Selection

In our first listening test, we compared the output of two cluster unit selection systems to decide which system to compare with the baseline system: The basic cluster unit selection system using 6000 units and the in-cluster selection system using 6000 units. The result of this evaluation, in which 6 people participated (For a total of 144 listened utterances), can be found in figure 5.18.

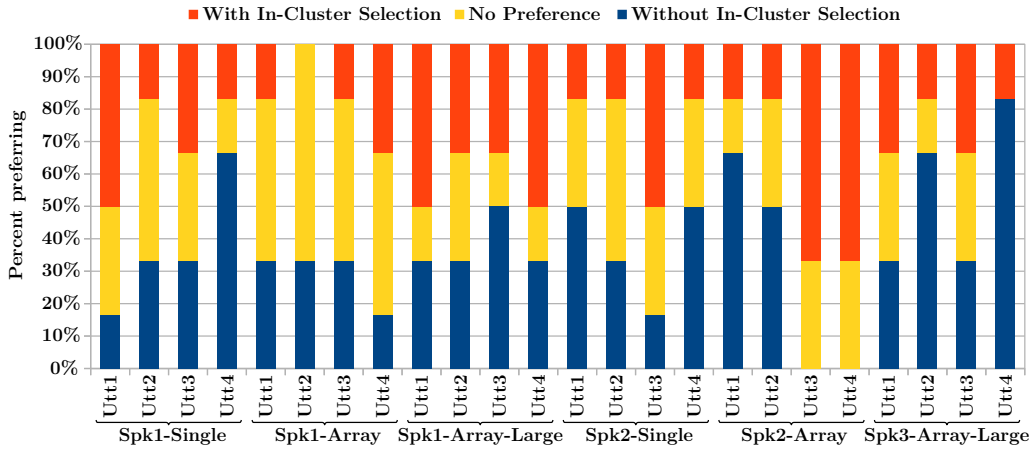

Figure 5.18: Results of comparative listening test with cluster unit selection not using in-cluster selection versus cluster unit selection using in-cluster selection.

The results are close, and there is (as expected) no clear preference for any system over the other for many utterances, there was a trend towards the in-cluster selection system (contradicting the MCD score results): The in-cluster system was preferred for 37.50% of all listened utterances, while the basic cluster unit selection system was the preferred system for only 29.86% (with the remaining 32.64% of listens resulting in no preference for either system). We decided, for this reason, to perform the final subjective evaluation comparison against the baseline system using the in-cluster selection system.

Baseline system versus In-Cluster Selection

While the results were very close for the initial comparison of the two cluster-based unit selection systems, this is not the case for the evaluation comparing the baseline system with the in-cluster selection system. A total of 11 people participated in this listening test, giving a total count of 264 listened utterances. The results can be seen in figure 5.19.

It is clear that, in the subjective evaluation, the new cluster-based system is superior by a large margin: It was the preferred system for two-thirds of listened utterances, and the preferred system for 21 of 24 utterances (with the improvement being so strong for two utterances that every participant preferred the cluster-based system over the baseline).

Out of the remaining three utterances, participants showed no clear preference for two. The baseline system was the preferred system for only for one utterance, Spk2-Single-Utt4. It is unclear what caused this outlier, and further investigation may reveal avenues enabling further performance improvements for cluster-based EMG-to-speech conversion.

In total, the cluster-based system was preferred for 66.67% of listened utterances, the baseline system in only 12.12%, with participants not preferring either listened utterance in 21.21%.

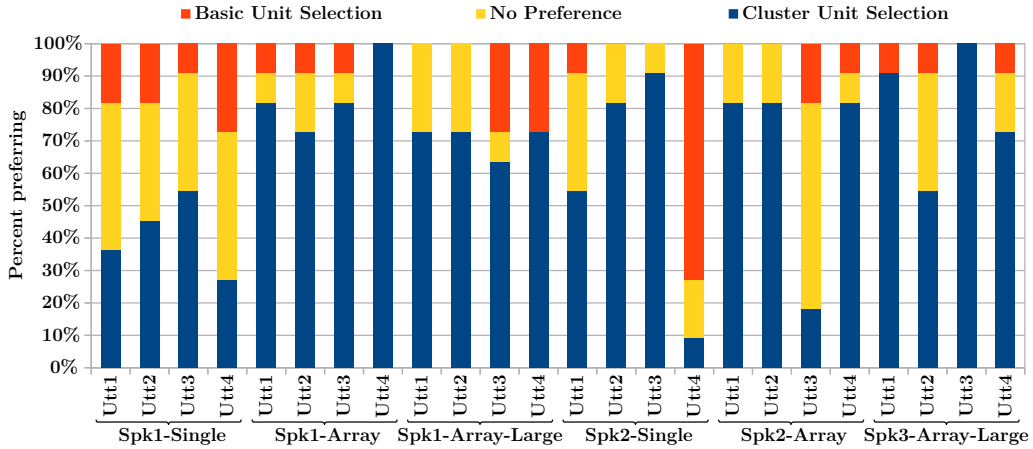

Figure 5.19: Results of comparative listening test with basic unit selection versus cluster unit selection with in-cluster selection.

6. Conclusion and Final Remarks

This work introduced a novel k-means clustering based method for improving unit selection based EMG-to-speech conversion, attempting to improve the unit codebook by generating more prototypical units. It then introduced a variation of this technique that tries to compensate for some of its problems.

Several sets of parameters for this method were evaluated on a development data set before settling on one to perform a final round of evaluations, objective and subjective, on an evaluation set.

The original method, cluster unit selection, resulted in an average improvement in frame-based MCD score of 13.11%. The variant, in-cluster selection, only achieved an improvement 7.27%, but was the preferred system in a subjective comparison of the two new systems.

In the subjective comparison of the baseline unit selection system from [ZJWS14] to the in-cluster selection system, the output of the new system was overwhelmingly preferred by participants, with the new system being the preferred system for a full two thirds of listened utterances, compared to the old system being only preferred in only 12.12% of all listened utterances.

6.1 Future Work

While the output of the new cluster-based unit selection EMG-to-speech conversion is a significant improvement over the previous conversion systems, the output remains unsatisfactory. Several issues need to be addressed before such a system can approach practicality.

For silent speech interfaces in general, one large issue not addressed in this work is between-session and between-speaker variance. Currently, most systems can only work session-dependently, requiring time-consuming enrolment every time electrodes are reattached. For practical use, this is not acceptable - a session-independent or at least session-adaptive system will have to be designed.

Such a system will also have to output good F_0 trajectories, a task not considered in this work. [ZJWS14] already use unit selection to create such trajectories, and

clustering-based methods or methods using very wide units might be able to improve upon the results presented there.

Another large issue is the operation on silent speech. This work only concerned itself with operating on EMG data recorded during audible speech. A transfer of this system to silently recorded EMG data remains challenging.

Within cluster-based unit selection, several things remain to be investigated, as well. One is how cluster count relates to the amount of training data available: Right now, the cluster count is fixed according to validation on a development set. A good automated procedure to find an optimal cluster count, or potentially alternate clustering algorithms, might improve performance especially for larger sessions.

The issue of selecting the best unit out of all those available also remains unsolved: Oracle experiments have shown that MCD scores as low as 3 should be possible even with basic unit selection, if only the right unit was selected every time. In this work, some additional target cost functions were evaluated, but a good procedure for a statistical optimization of the selection process through a learned target cost remains to be found.

6.2 Closing Remarks

The performance of silent speech interfaces based on the EMG modality has slowly, but steadily been improving for the last years. While no system is, as of yet, ready for use even in a clinical setting, this goal is beginning to look more achievable, and even though there is much work still to be done, it is our hope that the findings presented in this hope have contributed a small amount towards achieving it.

Bibliography

[BT97] Alan W. Black and Paul A. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Proceedings of Eurospeech*, volume 2, pages 601–604. International Speech Communication Association, 1997.

[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 F. Lovely. Myo-electric signals to augment speech recognition. *Medical and Biological Engineering and Computing*, 39(4):500–504, 2001.

[CEHL02] Adrian D. C. Chan, Kevin Englehart, Bernard Hudgins, and Dennis F. Lovely. Hidden markov model classification of myoelectric signals in speech. *Engineering in Medicine and Biology Magazine*, 21(5):143–146, 2002.

[CK79] Peter R. Cavanagh and Paavo V. 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.

[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.

[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 and Physics*, 30(4):419–425, 2008.

[FTKI92] Toshiaki Fukada, Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 137–140, 1992.

[GC+93] John S. Garofolo, Linguistic Data Consortium, et al. *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.

[Giu] Serban Giuroiu. Cuda k-means clustering. http://github.com/serban/kmeans. Accessed: June 25, 2015.

[HB96] Andrew J. Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 373–376, 1996.

[Ima83] Satoshi Imai. Cepstral analysis synthesis on the mel frequency scale. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 8, pages 93–96, 1983.

[JSW+06] Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. Towards continuous speech recognition using surface electromyography. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pages 573–576, 2006.

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

[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 Zoelzer. Beaqlejs: Html5 and javascript based framework for the subjective evaluation of audio quality. In *Linux Audio Conference*, 2014.

[Mac03] David J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[MHMSW05] Lena Maier-Hein, Florian Metze, Tanja Schultz, and Alex Waibel. Session independent non-audible speech recognition using surface electromyography. In *2005 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 331–336, 2005.

[MO86] Michael S. Morse and Edward M. O'Brien. Research summary of a scheme to ascertain the availability of speech information in the myoelectric signals of neck and head muscles using surface electrodes. *Computers in biology and medicine*, 16(6):399–410, 1986.

[MP04] Roberto Merletti and Philip A. Parker. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.

[Rao48] C. Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

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

[SW10] Tanja Schultz and Michael Wand. Modeling coarticulation in emg-based continuous speech recognition. *Speech Communication*, 52(4):341–353, 2010.

[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 (Interspeech)*, pages 1957–1960, 2005.

[TWS09] Arthur R. Toth, Michael Wand, and Tanja Schultz. Synthesizing speech from electromyography using voice transformation techniques. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pages 652–655, 2009.

[TZB02] Keiichi Tokuda, Heiga Zen, and Alan W. Black. An hmm-based speech synthesis system applied to english. In *Proceedings of the 2002 IEEE Workshop on Speech Synthesis*, pages 227–230. IEEE, 2002.

[Vin68] Taras K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, 4(1):52–57, 1968.

[WSJS13] Michael Wand, Christopher Schulte, Matthias Janke, and Tanja Schultz. Array-based electromyographic silent speech interface. In *International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS)*, pages 89–96, 2013.

[Zah14] Marlene Zahner. Konvertierung von myoelektrischen signalen der gesichtsmuskulatur zu sprache: Ein unit selection-ansatz. Master's thesis, KIT, 2014.

[ZJWS14] Marlene Zahner, Matthias Janke, Michael Wand, and Tanja Schultz. Conversion from facial myoelectric signals to speech: A unit selection approach. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

Index