An Initial Investigation into the Real-Time Conversion of Facial Surface EMG Signals to Audible Speech

Lorenz Diener[1], Christian Herff[1], Matthias Janke[1], and Tanja Schultz[1]

***Abstract*— This paper presents early-stage results of our investigations into the direct conversion of facial surface electromyographic (EMG) signals into audible speech in a real-time setting, enabling novel avenues for research and system improvement through real-time feedback. The system uses a pipeline approach to enable online acquisition of EMG data, extraction of EMG features, mapping of EMG features to audio features, synthesis of audio waveforms from audio features and output of the audio waveforms via speakers or headphones. Our system allows for performing EMG-to-Speech conversion with low latency and on a continuous stream of EMG data, enabling near instantaneous audio output during audible as well as silent speech production. In this paper, we present an analysis of our systems components for latency incurred, as well as the trade-offs between conversion quality, latency and training duration required.**

I. Introduction

Silent Speech Interfaces [1] (SSIs) are speech interfaces that, instead of relying on an acoustic speech signal, use information gathered from other signals generated during various stages of the speech production process. A common property of all SSIs is that they can be operated without actually producing audible speech. This results in several key advantages of an SSI over a regular speech interface: It can be used even in very noisy areas – or, conversely, in areas where the noise from audible speech is not desirable, as well as in situations where private or confidential information needs to be protected from bystanders.

Several approaches to building silent speech interfaces, based on different signal modalities, have been explored in the past, among them ultrasound [2], permanent-magnet articulography [3] and electrocorticography [4]. Our approach to building an SSI uses facial surface electromyography to capture electrical signals generated by the articulatory muscles and directly (without first performing speech recognition) converts them to audible speech [5].

Fully silent operation, while desirable, is not without its challenges, however: It has been shown that humans articulate differently when merely mouthing words, compared to audible speech production [6]. Most SSIs trying to convert silent to audible speech use statistical models perform this conversion, which require parallel audible and non-audible training data. To properly convert fully silent speech signals, adaptation is required – of the system to the user, the user to the system, or both. Real-time silent-to-audible conversion is, therefore, a necessity not only for general usability, but also to improve systems, as an adaptation, especially of the user to the system, can only take place when feedback is present.

[1] Cognitive Systems Lab, University of Bremen, Bremen, Germany
lorenz.diener@uni-bremen.de

Fig. 1. Positioning of the electrodes, with one 4 × 8 electrode array on the cheek and an 8 electrode strip below the chin.

Our previous work focused on improving the *quality* of conversion of EMG signals to audible speech with no regards to performance. This paper reports our first results in building a real-time low latency capable EMG-to-speech conversion system, which will enable us to perform research into the effects of feedback and human-machine co-adaptation that was not possible with an offline system.

Written informed consent was obtained from every person whose data was used to obtain the results in this paper.

II. System description

There are large inter-session differences in EMG recordings, caused by factors such as differences in electrode positioning or skin condition. For this reason, we need to record data and train a session-dependent mapping before the system can be used. Our previous EMG-to-speech conversion systems performed all processing steps offline, on pre-recorded data corpora, including pre-processing and feature calculation, training of the EMG-to-speech mapping and synthesis of audio output. For a real-time system, the time taken to perform these steps has to be reduced as much as possible, requiring a completely different system design: EMG pre-processing, mapping and synthesis have to be performed both in *real-time* (i.e. with less time spent on each frame than the frames duration) as well as with *low latency* (i.e. with a low delay between user speech production and audio output). Audio pre-processing and the training of the mapping, while not necessarily required to be either real-time or low latency, still need to be fast enough so that a user does not have to wait for an unreasonable amount of time before the system can be used. The following sections introduce our system and its design in light of these requirements.

A. Hardware

To record EMG signals, our system uses an OT Bioelettronica EMG-USB2 EMG amplifier. With it, we record EMG signals from the user's cheek using an electrode array (4 x 8 electrodes, 10 mm inter-electrode distance) and the user's

chin using an electrode strip (8 electrodes, 5 mm inter-electrode distance), chosen and positioned in accordance with our previous work [7]. Fig. 1 illustrates the placement of the electrodes in the face. The signals are acquired and amplified with a gain of 5000 using bipolar derivation between neighbouring electrodes (resulting in 35 EMG channels), processed with a 1 Hz high-pass filter (HPF) and 900 Hz low-pass filter (LPF), sampled at 2048 Hz and converted to 16 bit integer values.

An audio signal, required for the training of our EMG-to-speech conversion system, is acquired in parallel with the EMG signal, using a RØDE NT-1 condenser microphone and a Behringer 302USB mixer. The audio signal is recorded at 16 kHz, at a bit depth of 16 bits.

B. System Architecture

To allow for the low latency extraction of features, we designed our feature extraction process to be as pipelined as possible: Every component of the process is implemented as a module which is fed data by either a recording source (the EMG amplifier or the sound card) or a preceding module, and feeds data to the next module in the processing chain as soon as it becomes available. This design also facilitates a high-throughput (and thus, real-time) multiprocessing-enabled implementation, as modules can trivially be run in different processes, connected by first-in-first-out pipes. It also allows us to perform computation not on the machine used for recording (and, later, audio output), but on another network-connected machine with better hardware.

C. Feature extraction

1) EMG features: As the input for our EMG-to-speech mapping, we use a set of time-domain features, modeled after the features introduced by Jou et al. [8]. The features used in our mapping are the *low-frequency (LF) power*, *LF mean*, *high-frequency (HF) power*, *HF zero crossing rate (ZCR)* and *HF mean of absolute values*, where the LF signal is obtained via a running mean LPF with a cut-off frequency of 134 Hz and the HF signal is obtained as the difference of the input signal and the LF signal. They are calculated separately for each EMG channel, on frames of a length of $f_l = 32\ ms$, with a frame shift of $f_s = 10\ ms$. To provide time context, the feature vectors are then stacked 15 frames into the past, yielding an overall 35 channels x 5 features x 15 frames context = 2625 dimensional feature vector. The feature calculation process can thus be split into three parts:

- **Framing:** This module collects incoming data in a ring buffer, and, once at least one complete frame worth (f_l) of data has arrived, passes on one complete frame every f_s. Framing is performed with a fractional frame shift, to prevent EMG and audio framing from drifting apart. As some of our features require the splitting of the signal into a low- and high frequency part, this module also allows running application of a causal FIR filter on the incoming data stream, delaying the first frame accordingly if the length of the filter is longer than the frame size.
- **Feature calculation:** The feature calculation module simply takes a stream of frames as its input and applies a given function to it to calculate one feature value for each channel, which it passes on to the stacking module. To calculate multiple features, multiple of these modules, with different feature calculation functions, are created.
- **Stacking:** The stacking module combines any amount of incoming feature streams, waits until each stream has delivered as many frames as are to be stacked, and then passes on the stacked, combined feature frame, ready for training or mapping.

2) Audio features: To represent the audio signal, our system uses a magnitude spectrogram. The calculation is performed much in the same way as for the EMG features: A framing module splits the audio into overlapping frames, on which a feature calculation module calculates the magnitude spectrum using a short-term Fourier transform (STFT) with a Blackman window, resulting in the stream of audio features. For training, the EMG and audio frames are then combined into one parallel feature stream.

D. Mapping

Following our previous work [9], we use a deep neural network (DNN) using rectified-linear units to perform the mapping of EMG to audio features. To lessen the impact of over-fitting when only a very small amount of training data is available, we added drop-out regularization between all layers [10]. Fig. 2 illustrates the structure of the neural network used in this paper.

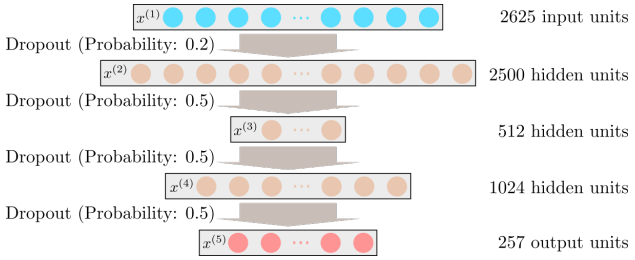

Fig. 2. Structure of the DNN used for EMG-to-audio feature mapping.

The fast training of neural networks requires a large amount of processing power, but parallelizes well. Our system uses the Brainstorm [11] neural network library to perform mini-batched stochastic gradient descent training on a GPU, with a mini-batch size of 512, a training momentum of 0.9 and a learning rate of 0.001 for the first three epochs and 0.01 for 27 epochs afterwards.

Fig. 3 shows an overview of the system setup as used for training. Note that feature calculation can be performed during recording, minimizing the time the feature calculation step adds to system training.

E. Synthesis

As our audio features are magnitude spectrograms, the problem of synthesizing a waveform is equivalent to estimating a matching phase. To this end, we use the method proposed by Griffin and Lim [12], implemented to operate on a continuous stream of data. It reconstructs the input

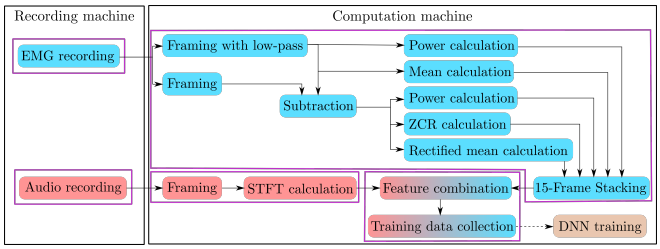

Fig. 3. The different modules of our EMG-to-speech conversion system during the data recording and system training stage. Inner borders indicate modules running within one process.

signal by first computing the Blackman-windowed STFT of a random signal, replacing its magnitude spectrogram with the input spectrogram, estimating the signal that best matches this new spectrogram by overlap-adding the spectrogram frames' inverse STFTs, and then iteratively repeating these steps with the estimated signal instead of a random signal as input. This estimation is performed on overlapping blocks of data large enough to fully determine every new spectrogram frames estimated signal as soon as the input data is available (a total block size of $\lfloor f_l/f_s \rfloor * 2 + 1$ frames, centered on the current frame). Fig. 4 illustrates the complete pipeline during real-time conversion.

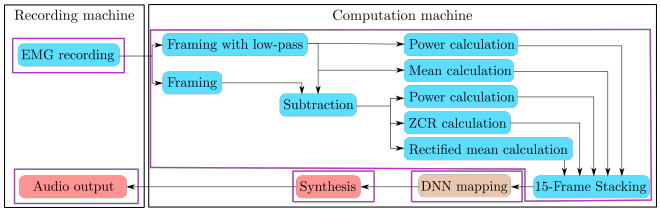

Fig. 4. The different modules of our system during real-time EMG-to-speech conversion. Inner borders indicate modules running within one process.

III. System Analysis

This section presents an analysis of our system for two factors: We analyze the latency incurred by different components of our system, and present an evaluation of how the amount of training data (and consequently, the amount of time spent recording and training at the start of a session) affect output quality.

A. Component latency

The conversion latency in our system can be divided into four categories.

- **Hardware latency:** Latency inherent to the hardware used in our system – EMG amplifier and sound card.
- **Network latency:** Latency induced by the the transfer of data over the network, from the machine running the recording and output to the computation machine and back. In our wired network, the round-trip time of a packed between recording and computation machines (mean, measured once per second for one minute) is $l_{\text{net}} = 0.13\ ms$
- **Buffer latency:** Latency induced by the keeping of temporary buffers in processing pipelines. This is alleviated by properly pre-filling the conversion pipeline to generate output as soon as possible, leaving only the length of the longest buffer as added latency – in our system, this is during synthesis, with a buffer length of $l_{\text{buf}} = (\lfloor f_l/f_s \rfloor + 1) * f_s = 40\ ms$.
- **Computation latency:** Latency due to the computation time taken up by feature calculation, mapping and synthesis. As every module depends on the previous modules output, the total computation latency l_{comp} is the sum of all modules computation times. This is analyzed further below.

Table I shows the computation times required for different parts of the real-time EMG-to-speech mapping to produce one output frame (10 ms of audio), measured averaged over 1000 input frames. Total computation latency is $l_{\text{comp}} = 9.34\ ms$, making the total overall latency $l = l_{\text{net}} + l_{\text{buf}} + l_{\text{comp}} = 0.13\ ms + 40\ ms + 9.34\ ms = 49.47\ ms$. With the largest computation delay stemming from necessary buffering, further optimization might require redesigning the system to operate on smaller and more closely spaced frames – requiring tighter computation time constraints, as well.

TABLE I
Per-output-frame computation latencies

Module	Computation time	
	Absolute	Relative to total
Framing	0.16 ms	1.71%
Framing with LPF	0.44 ms	4.71%
Subtraction	0.1 ms	1.07%
Power (LF)	0.67 ms	7.17%
ZCR (HF)	0.16 ms	1.71%
Mean (LF)	0.1 ms	1.07%
Power (HF)	0.58 ms	6.21%
Rectified Mean (HF)	0.14 ms	1.5%
Stacking	0.02 ms	0.21%
DNN Mapping	4.03 ms	43.15%
Synthesis	2.94 ms	31.48%
Total	9.34 ms	

In trying to estimate the time between when speech output is expected and when the system actually produces output, we have to take *electromechanical delay* into account: There is some time between the moment an electrical excitation of the muscle tissue can be measured and movement onset [13]. While the exact delay may vary between different phones, 50 ms has been found to be a good average estimate [8]. This leaves our system with an estimated time lag of close to zero milliseconds between user-expected and actual sound output, plus hardware delays. In practice, we have observed latencies that appear to be in excess of this, hinting at an influence of hardware latencies on the system – how these are distributed and how they could be mitigated remains to be investigated.

B. Quality versus training time

Session-independence in EMG-based speech processing remains an unsolved research problem [14]. Our conversion system, therefore, operates session-dependently, necessitating training with newly acquired data in each session before it can be used. From a quality standpoint, it would be desirable to have more training data rather than less, however, more data requires more recording, processing and training time. Finding a good balance between the amount of training data and the

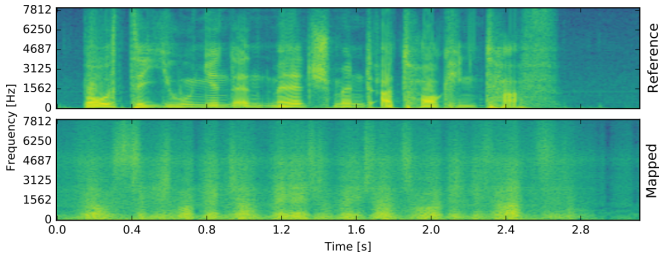

Fig. 5. A comparison of reference and mapped audio of the utterance "Both the union and management are talking tough.", converted using a system trained with 450 training utterances.

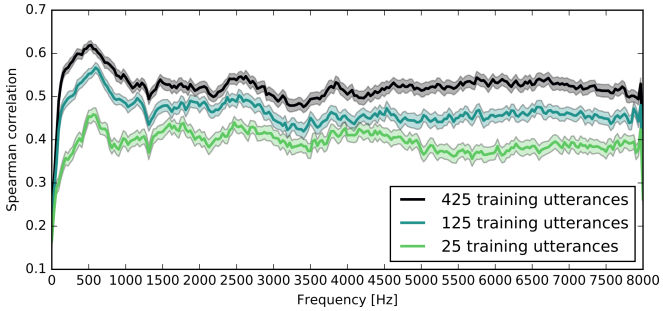

Fig. 6. Mean spectrogram band correlations between reference and mapped audio (larger is better). Coloured areas indicate standard error of the mean.

quality of the system output is, then, an important trade-off. For this reason, we evaluated our system's performance on unseen data depending on the amount of data used in training.

To perform this evaluation, we trained our system with a varying number of training utterances, from 25 up to 450 in steps of 25, from a pre-recorded corpus from our previous work [9]. We then used the system to map the held out set of test utterances, and compared the rank correlation of the spectrum (Spearman's rho) between reference audio and mapping result, aligned to maximize that correlation.

Fig. 5 shows a set of aligned spectrograms. It can be seen that the mapping manages to capture the overall structure of the spectrum, though fine detail is lost, and a large noise floor remains. This can also be seen in Fig. 6, which shows the mean correlations of the spectrograms bands for three systems – the mapping works particularly well for low frequencies, with the maximum correlation of 0.62 in the 496 Hz to 527 Hz band for the best performing system.

Fig. 7 shows mean correlations depending on the amount of training data. As expected, increasing the amount of training data improves output quality. With our current mapping setup, however, improvement seems to slow down after 125 to 150 and seems to saturate at around 200 to 250 training utterances, with very little improvement thereafter. This leads us to believe that 150 training utterances (circa 9.5 min of audio recording and, with our setup, circa 5.5 min of training time), with a mean spectral correlation of 0.48, are sufficient for initial feedback experiments.

IV. Conclusion

We have presented an initial system for real-time low latency EMG-to-speech conversion and shown evaluations of the system towards potential use in real-time feedback experiments. In the future, we hope to further improve

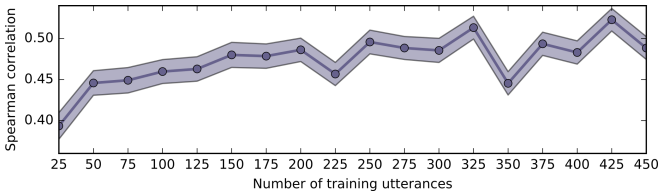

Fig. 7. Average spectral correlation between reference and mapped audio depending on training set size (larger is better). Coloured area indicates standard error of the mean.

this system and to use it to investigate questions related to machine-human co-adaptation in silent speech. We plan to investigate the effect of real-time audio feedback with varying quality and delay on silent- and audible mode speech, especially if conversion quality improves after prolonged system use. We plan to investigate hardware latency. We hope to improve *quality* by starting the training procedure from models pre-trained with large amounts of data, possibly performing data adaptation on the session recordings.

References

[1] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[2] T. Hueber, E.-L. Benaroya, G. Chollet, B. Denby, G. Dreyfus, and M. Stone, "Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips," *Speech Communication*, vol. 52, no. 4, pp. 288–300, 2010.

[3] J. A. Gonzalez, L. A. Cheah, J. M. Gilbert, J. Bai, S. R. Ell, P. D. Green, and R. K. Moore, "Direct speech generation for a silent speech interface based on permanent magnet articulography," in *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*, vol. 4, 2016, pp. 96–105.

[4] C. Herff, D. Heger, A. de Pesters, D. Telaar, P. Brunner, G. Schalk, and T. Schultz, "Brain-to-text: Decoding spoken phrases from phone representations in the brain," *Frontiers in Neuroscience*, no. 217, 2015.

[5] A. R. Toth, M. Wand, and T. Schultz, "Synthesizing speech from electromyography using voice transformation techniques," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2009, pp. 652–655.

[6] M. Janke, M. Wand, and T. Schultz, "Impact of lack of acoustic feedback in emg-based silent speech recognition," in *11th Annual Conference of the International Speech Communication Association*, 2010.

[7] M. Wand, C. Schulte, M. Janke, and T. Schultz, "Array-based electromyographic silent speech interface," in *International Conference on Bio-inspired Systems and Signal Processing*, 2013, pp. 89–96.

[8] S.-C. S. Jou, T. Schultz, M. Walliczek, F. Kraft, and A. Waibel, "Towards continuous speech recognition using surface electromyography," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2006, pp. 573–576.

[9] L. Diener, M. Janke, and T. Schultz, "Direct conversion from facial myoelectric signals to speech using deep neural networks," in *International Joint Conference on Neural Networks*, 2015.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[11] K. Greff and R. Srivastava, "Brainstorm," 2015, IDSIA. [Online]. Available: https://github.com/IDSIA/brainstorm

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

[13] P. Cavanagh and P. Komi, "Electromechanical delay in human skeletal muscle under concentric and eccentric contractions," *European Journal of Applied Physiology and Occupational Physiology*, vol. 42, no. 3, pp. 159–163, 1979.

[14] M. Wand, C. Schulte, M. Janke, and T. Schultz, "Compensation of recording position shifts for a myoelectric silent speech recognizer," in *The 39th International Conference on Acoustics, Speech, and Signal Processing*, 2014, iCASSP 2014.