Speech Communication 106 (2019) 44–56

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

DNN-based performance measures for predicting error rates in automatic speech recognition and optimizing hearing aid parameters

Angel Mario Castro Martinez[a,b,*], Lukas Gerlach[c,b], Guillermo Payá-Vayá[c,b], Hynek Hermansky[d], Jasper Ooster[a,b], Bernd T. Meyer[a,b]

[a] *Department für medizinische Physik und Akustik, Carl von Ossietzky Universität Oldenburg, Germany*
[b] *Exzellenzcluster Hearing4all, Germany*
[c] *Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany*
[d] *Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA*

ARTICLE INFO

Keywords:
Automatic speech recognition
Performance monitoring
Spatial filtering
Hearing aids

ABSTRACT

In several applications of machine listening, predicting how well an automatic speech recognition system will perform before the actual decoding enables the system to adapt to unseen acoustic characteristics dynamically. Feedback about speech quality, for instance, could allow modern hearing aids to select a speech source in complex acoustic scenes with the aim of enhancing the speech intelligibility of a target speaker. In this study, we look at different performance measures to estimate the word error rates of simulated behind-the-ear hearing aid signals and detect the azimuth angle of the target source in 180-degree spatial scenes. These measures derive from phoneme posterior probabilities produced by a deep neural network acoustic model. However, the more complex the model is, the more computationally expensive it becomes to obtain these measures; therefore, we assess how the model size affects prediction performance. Our findings suggest measures derived from smaller nets are suitable to predict error rates of more complex models reliably enough to be implemented in hearing aid hardware.

1. Introduction

Speech awareness is the ability to detect and perceive spoken sounds as understandable language. Humans not only excel at recognizing speech but also identify how well are they understanding and producing the message (Postma, 2000) despite adverse conditions such as noise and reverberation (Assmann and Summerfield, 2004; Scheffers and Coles, 2000). Feedback about the speech quality could be exploited in several machine listening applications; in automatic speech recognition (ASR), for instance, this knowledge provides means to evaluate performance by estimating the word error rate (WER) with sufficiently low latency to dynamically adapt to challenging acoustic scenes and generalize better to new data. The methods used for WER prediction are often called confidence or performance measures (Kintzley et al., 2011; Mallidi et al., 2015).

In the past, performance measures have been used for various purposes such as evaluating the decision reliability of deep neural networks (DNN) classifiers. Mallidi et al. (2015) proposed two methods for measuring the uncertainty of posterior probabilities of speech classes as a function of the mismatch information between training and test sets. Their approach was used as the selection criterion for a multi-stream phoneme ASR system which yielded almost the same performance as including oracle information about the most similar acoustic condition of the test set. On the topic of multi-stream frameworks, Hermansky (2013) provides a comprehensive review; ranging from a biological motivation for recognizing speech in parallel interacting streams to current techniques for information extraction from multiple characterizations of the speech signal to deal with unexpected acoustic conditions.

Another application of performance measures is in the field of assistive technologies and hearing aids. Among the various features in modern multi-microphone hearing aids is the use of adaptive spatial filtering for noise reduction and speech enhancement (Rohdenburg et al., 2008; Souden et al., 2013), a method which requires the device to estimate the azimuth angle, also known as the direction of arrival, of the target source. Separation of individual sources, localization, and enhancement of individual sources (similar to the healthy human auditory system (Hawley et al., 1999)) is a powerful technique for information extraction as spatial separation increases speech intelligibility.

[*] Corresponding author at: Medizinische Physik and Cluster of Excellence Hearing4all, Carl von Ossietzky Universität, Küpkersweg 74, Oldenburg 26129, Germany.
E-mail addresses: angel.castro@uni-oldenburg.de (A.M. Castro Martinez), gerlach@ims.uni-hannover.de (L. Gerlach), guipava@ims.uni-hannover.de (G. Payá-Vayá), hynek@jhu.edu (H. Hermansky), jasper.ooster@uni-oldenburg.de (J. Ooster), bernd.t.meyer@jhu.edu (B.T. Meyer).

https://doi.org/10.1016/j.specom.2018.11.006
Received 17 October 2017; Received in revised form 28 September 2018; Accepted 26 November 2018
Available online 26 November 2018

Realistic environments, however, often involve two or more interfering sources (either noise or concurrent speakers), which can be moving or stationary, localized or diffuse; in hearing aids, these problems are particularly challenging as state-of-the-art processing deep learning schemes become unfeasible to implement due to the high computational resources required. A plethora of methods have been developed to tackle some cases making significant trade-offs: for instance, Cornelis et al. (2014) introduced a reduced-bandwidth filtering procedure for noise reduction; decreasing the amount of data needed to transfer between microphones without severely degrading performance. A comprehensive review of binaural signal processing algorithms can be found in Hamacher et al. (2008), Cornelis et al. (2014), Pertilä and Nikunen (2014) and Hamacher et al. (2008). For scenes with moving target sources, Adiloğlu et al. (2015) proposed a binaural steering beamformer able to track the target speaker in an anechoic diffuse noise setting.

As sophisticated as these algorithms are, some scenarios with interfering speech sources might conflict with the direction of arrival estimation, resulting in an incorrect beamforming direction, which could increase listening effort or reduce speech intelligibility. In these cases, more conservative but less fragile methods such as better-ear-listening should be preferred (Kayser et al., 2015). Inverse entropy was proposed by Spille et al. (2016) as a performance measure to evaluate how suitable the hearing aid parameters are in a given acoustic scene.

A phoneme posteriorgram is extracted from the softmax activations of a trained DNN classifier to compute the performance measures explored in this paper. The reference measure is the utterance average of the entropy per frame; intuitively, the entropy will increase if the posteriorgram becomes more uniformly distributed. Therefore a low value will correlate with the certainty of the decision made by the ASR system (Okawa et al., 1999). Also, frame-wise entropy allows us to compare our results to Spille et al. (2016) directly.

The mean temporal distance (dubbed M-Measure), proposed in Hermansky et al. (2013), measures the divergence of two probability vectors over a finite interval of time; the longer the span, the more certainty there is the two vectors come from different coarticulation patterns, the M-measure captures this behavior. Mallidi et al. (2015) implemented the M-measure as a multistream selector for phoneme recognition, outperforming entropy and even getting close to oracle selection of the best performing stream. For the last performance measure (referred to as MaP for matched phoneme filtering) (Kintzley et al., 2011), filters learned from clean posteriors are required to detect phonetic events when convolving them with the phoneme posteriorgrams. Filters matched to the average activation pattern of phonemes should produce a high peak for robust phoneme representations. For degraded posteriorgrams, these peaks should be less pronounced.

In previous related research, we explored these performance measures in binaural scenes obtained from behind-the-ear hearing aids (Meyer et al., 2016) and their relation to WER obtained in ASR experiments in different noise scenarios. A key finding was that the baseline measure (entropy) did not reliably capture the WER in the testing conditions, whereas M-Measure exhibit a better relation, which indicates its value can be used to deduce the WER. However, the actual difference between the WER and the estimation from the M-Measure (which we refer to as prediction error, PE) was not considered. Furthermore, the signal-to-noise ratio (SNR) of test data limited to the range of −10 and +10 dB SNR, and is therefore not representative of everyday listening situations that often include SNRs well above 10 dB.

A more extensive variety of noise types was explored in Meyer et al. (2017), and the PE derived from M-Measure or MaP was approximately 6% (in contrast to 11% for entropy). Nevertheless, this study was limited to single-channel microphone recordings only and did not explore potential applications for hearing aids.

The aim of this paper is to investigate performance measures both to predict WER in ASR and to test their applicability for optimizing hearing aid parameters (such as the beamforming angle in a spatial scene). The latter could be achieved if the performance measure assumes its maximum value for the current best parameter set (e.g., an on-speaker beamforming angle in a spatial scene). We take into account four different room scenarios over a realistic selection of SNRs to test the generalization capabilities of the performance measures despite the effects of spatial filtering, reverberation, and noise type.

Given that the aforementioned performance measures are extracted from the trained model to evaluate, potential challenges come from increasing the complexity of the acoustic model. For instance, making the forward pass to compute the posteriorgrams becomes more computationally expensive, thus increases latency. Depending on the ASR device and its use case it may or may not compromise performance. In hearing aids applications, on the other hand, the latency and computational cost are determining factors in the successful adoption of these measures due to the hardware limitations. Therefore, we assess how the size of the model affects the prediction performance and whether smaller networks could predict the error rates and correct azimuth angle of larger models without significant degradation.

In the following sections, the experimental setup and performance measures are described. In the results section, the estimation of WER and azimuth angle concerning each performance measure is shown as well as the model complexity versus prediction error analysis. Finally, we discuss the prediction error and how feasible the implementation of performance measures is in hearing aids.

2. Experimental setup

Fig. 1 shows an illustration of the main building blocks of the proposed model with the corresponding section in parenthesis. At its center are the speech recognition experiments and the different acoustic models with varying width and depth which maps single-channel acoustic observations to phoneme probabilities. The model is tested in different generated spatial conditions, all of which are processed by the beamformer speech enhancement algorithm of multi-channel hearing aid signals. The phoneme probabilities associated with these signals are quantified with the performance measures, which are related to the word error rate and speech representations in hearing aids.

2.1. ASR training and DNN architecture

The ASR systems used in this work were trained and tested on two databases. The first one was the English Aurora4 framework (Parihar et al., 2004) with the multi-condition set for training and the *eval92* for testing; the second one, a German small-vocabulary database (Ooster et al., 2018) based on the matrix sentence test for speech intelligibility – OLSA.[1] The database contains recordings of speakers performing a speech intelligibility test. The test uses a 50-word matrix to generate random sentences which are syntactically fixed but semantically unpredictable (Wagener et al., 1999). These sentences are presented in noise to a subject who responds with the words they recognized.

The Aurora 4 corpus comprises speech read from professionals, whereas the OLSA training data are read sentences from untrained subjects, and the *SPRINT* (SPontaneous Responses in Intelligibility Tests) test set contains the natural speech of participants in the hearing test.

The Aurora 4 multi-condition set consists of 7137 utterances from 83 independent speakers. One half of the 16 kHz files were recorded with the close-talk Sennheiser HMD-414 microphone, the other half using one of 18 different types of microphones. Three-fourths of the training material was corrupted with one of six different types of noise (car, babble, restaurant, street, airport and train) at randomly selected SNR conditions between 10 and 20 dB. The test set *eval92* included in Aurora 4 comes from the WSJ0 5000 word closed-vocabulary task which consists of 330 utterances from 8 speakers repeated in the same conditions used

[1] The name derives from the German translation *Oldenburger Satztest.*

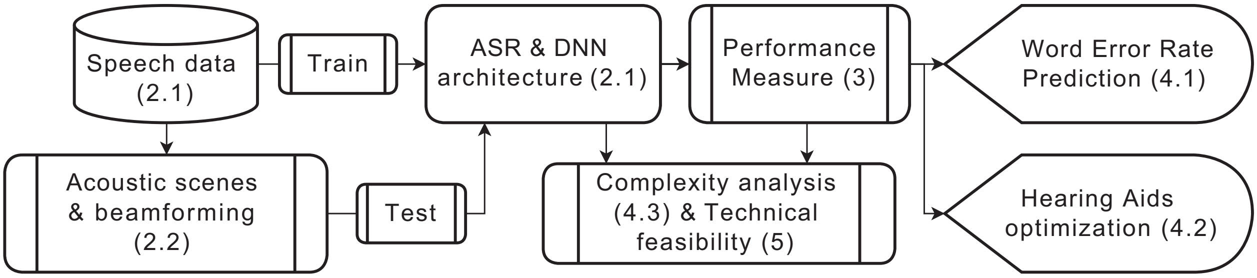
Speech data (2.1)
Train
ASR & DNN architecture (2.1)
Performance Measure (3)
Word Error Rate Prediction (4.1)
Acoustic scenes & beamforming (2.2)
Test
Complexity analysis (4.3) & Technical feasibility (5)
Hearing Aids optimization (4.2)

Fig. 1. Building blocks of the proposed study; starting with the speech material from which the acoustic scenes are generated to test the speech recognition experiments and architecture description, and finalizing with the results obtained and the feasibility analysis of the whole system to implement on hearing aids. The corresponding sections are shown in parenthesis.

for the training set at 5–15 dB SNR. The WSJ pruned trigram language model was used for decoding limited to the same 5000 vocabulary from the *eval92* test set.

The *OLSA* training data consist of 21,728 read matrix-sentence utterances (≈ 22.5 h) from 20 speaker. The data was recorded at 44.1 kHz with 32-bit resolution at a distance of approximately 0.5 m using a high-quality microphone (Neumann KM 184) in an isolated sound booth, and preprocessed in the same manner as the Aurora4 multi-condition set. First, the files were downsampled to 16 kHz and compressed to 16-bit resolution; then three quarters were mixed with the same six different types of noise at random SNRs between 10 and 20 dB and filtered according to the ITU-T recommendation P.341 (ITU-T, 2011). The *SPRINT* test set used in this work consists of 480 spontaneous responses from three randomly selected speakers.

Because the test data are responses to a matrix sentence, most utterances contain the words from the matrix test vocabulary. Nevertheless, considering the subjects respond spontaneously and have no restrictions regarding their reply, about 10% of the subject responses does not belong to the matrix test words. For decoding, we used a 0-gram language model limited to the 50 words of the matrix test and a phone-level 4-gram model trained for out-of-vocabulary words using a general German lexicon.

To evaluate the acoustic model complexity as a function of the number of parameters, we trained 20 feed-forward networks, namely every combination of 2–6 hidden layers with 256, 512, 1024 and 2048 (sigmoid) hidden units. To evaluate spontaneous speech data from listening subjects, we trained 3 different networks, i.e., the network with the most and fewest parameters, as well as the one that performed best on the Aurora 4 data. Following the training recipe from Vesely et al. (2013), each network was initialized with a stack of Restricted Boltzmann Machines trained one layer at the time via contrastive divergence (Hinton, 2010). The nets were then fine-tuned to classify frames into triphone states using the cross-entropy between the network output and the labels as a cost function. Every phoneme was modeled with three Hidden-Markov-Model (HMM) states except for the silence phone which has five states.

The training was done in up to 20 epochs (stopping when the relative improvement was lower than 0.001). The starting learning rate was 0.008 (halving it every time the relative improvement was lower than 0.01). 40-dimensional log-Mel-spectral coefficients, spliced ± 5 frames to provide temporal context, were fed to the input layer. A soft-max layer of approximately 2000 units was attached to the output to produce the most likely posterior probabilities of each context-dependent HMM state; we call this distribution over time a phoneme posteriorgram.

Even if the DNNs were trained to deliver scaled probabilities of the triphone HMM states, each transition could be seen as a branch of a correspondent decision tree. The roots of those trees correspond to the central phoneme of the trained triphones and were used to create 42-dimensional monophone posteriorgrams from the clustered branches by adding the corresponding activations, thus maintaining the distribution.[2] Monophone representations were used as they produce similar results to the triphone equivalent at a lower computational cost.

2.2. Generation of acoustic scenes

Four different acoustic scenes were generated to test how robust performance measures are at predicting word error rates and detecting the speaker in a horizontal half-plane in the presence of spatially localized and diffuse noise sources. As target speech, 100 random utterances were processed from Aurora 4 clean *eval92* test set as well as the half the *SPRINT* set. All scenarios were simulated using a subset of the database recorded by Kayser et al. (2009) featuring eight-channel head-related impulse responses (HRIR). For this study, we only selected the six signals from each of the three behind-the-ear (BTE) microphones attached to the hearing aids on the left and right ears of a head-and-torso simulator. Subsequently, a super-directive beamformer (Bitzer and Simmer, 2001) is jointly applied to the six-channel data. Then the resulting single-channel signals are used for the recognition experiments and posteriorgram calculation.

In the first and second acoustic scenes, the anechoic far-field HRIR were used to simulate a non-reverberant situation. These impulse responses were measured at 3m of separation between loudspeaker and microphones. Two different noises were used in each scene: OLN - the speech-shaped noise from the Oldenburg sentence test (Wagener et al., 1999) to recreate a spatially diffuse stationary noise and VC - a vacuum cleaner localized noise included in the *BBC Sound Effects Library,*[3] positioned at an azimuth angle of +40°.

The third and fourth scenes had the aforementioned noises in a *typical* office environment with a reverberation time T_{60} of 300ms measured by the HRIR at distance of 1 m between the source and target. For all the HRIR (anechoic and reverberated) from the database, the azimuth angle varied in 5° steps; to constrain the amount of ASR test sets. However, we only took 10° increments, resulting in 19 different azimuth angles in the range of $[-90^\circ \quad 90^\circ]$ to steer the beamformer. Table 1 summarizes the four generated scenes.

Spatial filtering of the speech signal takes place in the frequency domain: first the multi-channel short time Fourier transform $\mathbf{x}(\omega, t)$ of the six BTE microphones is multiplied with the complex conjugate of a binaural filter[4] $\mathbf{w}(\alpha, \omega)$, with the steering direction α. The resulting

[2] The list of phonemes are compliant with the ARPABET phonetic transcription code and include the /SIL/ and /NSN/ classes for silence and noise respectively.

[3] BBC03 81–203 34-1 HOUSEHOLD VACUUM CLEANER.

[4] The binaural multi-channel Wiener filter with preserved relative transfer functions used in this work was introduced in Marquardt et al. (2015).

Table 1
Description of the four generated acoustic scenes. As target speech, 100 random utterances from Aurora 4 clean *eval92* test set and 240 utterances from *SPRINT*. All scenarios were simulated using the six behind-the-ear channels of the HRIR database (Kayser et al., 2009). Subsequently, a super-directive beamformer (Bitzer and Simmer, 2001) is jointly applied to the six-channel data. The resulting single-channel signals are then used for the recognition experiments and posteriorgram calculation.

	Speech-shaped noise	BBC Vacuum cleaner
Non-reverberant room far-field HRIR $d = 3$ m	Anechoic OLN	Anechoic VC
Office setup $T_{60} = 300$ ms, HRIR $d = 1$ m	Office OLN	Office VC

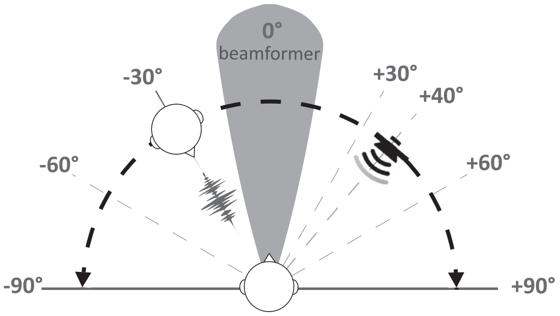

Fig. 2. Acoustic scene overview: the fixed target speaker is located at an azimuth angel of −30° with respect to the listener in the center. In first and third scenes diffuse noise OLN is added; in the other two localized noise VC is present at 40° 3*m* and 1*m* away respectively.

single-channel output $\mathbf{y}(\alpha, \omega, t)$ is defined by:

$$\mathbf{y}(\alpha, \omega, t) = \mathbf{w}^{\mathsf{H}}(\alpha, \omega)\,\mathbf{x}(\omega, t). \tag{1}$$

For every α, the filter coefficients $\mathbf{w}(\alpha, \omega)$ were obtained from the steering vector $\mathbf{d}(\alpha, \omega)$, using the *minimum variance distortionless response* (MVDR) minimization solution (Cox et al., 1987) to reduce the noise power spectral density. Steering vectors were computed from the anechoic head-related transfer functions of sound propagation in a given α direction with the front-left hearing aid microphone as the reference channel. In other words, relative transfer characteristics between microphones were exploited to obtain steering vectors. The noise covariance matrix $\mathbf{R}(\omega)$ was calculated from the whole set of anechoic head-related transfer functions serving as a model for a head-related isotropic noise field which only depends on the sensor configuration on the head, not taking room characteristics or coherent noise sources into account. With the parameters mentioned above, the solution is a function of the covariance matrix $\mathbf{R}(\omega)$:

$$\mathbf{w}(\alpha, \omega) = \frac{\mathbf{R}^{-1}(\omega)\,\mathbf{d}(\alpha, \omega)}{\mathbf{d}^{\mathsf{H}}(\alpha, \omega)\,\mathbf{R}^{-1}(\omega)\,\mathbf{d}(\alpha, \omega)}. \tag{2}$$

Fig. 2 shows a sketch of the acoustic scenes under consideration. Spoken utterances from a fixed azimuth angle of −30° were mixed with random parts of a spatially diffuse stationary speech-shaped noise at signal-to-noise ratios (SNR) from −10 to 30 dB in 5 dB steps plus clean speech in the end (~ 50 dB). The diffuse noise was replaced with the localized vacuum cleaner noise positioned at +40° azimuth using the same SNR range. The generated scenes comprise 232,560 utterances in total: 9 SNR x 19 beamformer azimuth angles x 2 rooms x 2 noise types x (100 + 240) utterances.

3. Performance measures

The performance measures obtained from DNN posteriorgrams are introduced in this section. Per-frame entropy is used as a baseline and compared to the mean temporal distance (M-Measure) and the *maximum a posteriori* (MaP) approach of phonetic event detection using matched filtering.

3.1. Phoneme entropy

The entropy of time frames from phoneme posteriorgrams has been suggested to estimate uncertainty in ASR (Okawa et al., 1999; Misra et al., 2003), which was motivated by the observation that for many types of noise, a posteriorgram approaches a uniform distribution for low SNR levels. In this work, given the 42 phonemes per posteriorgram, the maximum entropy value per frame is $H = 5.392$. On the other hand, the entropy should decrease the more sparse the posteriorgrams are. We, therefore, choose entropy as a baseline performance measure, which is calculated on a frame-by-frame basis for each K-dimensional vector $\mathbf{p}$:

$$H = -\sum_{k=1}^{K} p^{(k)}\, \log_2(p^{(k)}) \tag{3}$$

Frame-wise entropy values are averaged over time to obtain one scalar per utterance, which is then related to WER.

3.2. Mean temporal distance

The mean temporal distance or M-Measure (Hermansky et al., 2013) indicates the mean distance between probability estimations (speech features, posterior probabilities, etc...) over a specific time interval. It was initially proposed as an unsupervised performance monitoring tool for ASR. This measure takes into account the effect of phoneme coarticulation as its value tends to saturate around 200 ms. In clean phoneme posteriorgrams, isolated clear peaks can usually be observed, which are often temporally smeared in the presence of noise. This is the motivation for the M-Measure since different distinct phoneme vectors result in a large distance, while temporal smearing would result in similar vectors and therefore smaller distances. The M-measure accumulates the average divergences of two phoneme posteriors vectors $\mathbf{p}_{t-\Delta t}$ and $\mathbf{p}_t$ separated by a time interval of Δt and is defined as

$$\mathcal{M}(\Delta t) = \frac{1}{T - \Delta t} \sum_{t=\Delta t}^{T} \mathcal{D}(\mathbf{p}_{t-\Delta t}, \mathbf{p}_t), \tag{4}$$

Where T is the duration of the analyzed representation, in this case, a portion of the posteriorgram. As in Hermansky et al. (2013), the symmetric Kullback–Leibler divergence was chosen as distance measure $\mathcal{D}$ between phoneme posterior vectors $\mathbf{p}_{t-\Delta t}$ and $\mathbf{p}_t$.

$$\mathcal{D}(\mathbf{p}, \mathbf{q}) = \sum_{k=0}^{K} p^{(k)} \log \frac{p^{(k)}}{q^{(k)}} + \sum_{k=0}^{K} q^{(k)} \log \frac{q^{(k)}}{p^{(k)}} \tag{5}$$

As defined above, $p^{(k)}$ is the kth element of the posterior vector $\mathbf{p} \in \mathbb{R}^k$. We considered 20 values of Δt per utterance; the first five from 10 to 50 ms in steps of 10 ms and the rest from 100 to 800 ms in steps of 50 ms. For short Δt time spans, divergences are small indicating neighboring frames often correspond to the same phoneme. The value increases with the time span until the point both vectors $\mathbf{p}$, and $\mathbf{q}$ come from different coarticulation patterns, and the curve saturates.

Finally, the average M-measures from (4) were taken for each Δt value from 50 to 800 ms as a scalar performance measure.

3.3. Matched phoneme filtering

The core idea of the third measure is to search for phonetic activation patterns typical for clean posteriorgrams. This is achieved by first learning phoneme-specific activations from clean DNN output data, and by using the resulting curves as filters which are convolved with the temporal trajectories in the posteriorgram. This constitutes a matched phoneme filtering (MaP), which was first proposed in Kintzley et al. (2011). MaP suppresses degraded activations (i.e., from noisy speech) and therefore results in sparse representations of phonetic events. These events are defined as the local maxima of the filtered posteriorgram trajectories exceeding a certain threshold λ. When a filter matches the average activation pattern of a specific phoneme, the outcome should be a high peak and thus a robust representation of the given phoneme. In the case of noise-corrupted posteriorgrams, these peaks became less pronounced. This feature is exploited to assess the quality of the posteriorgrams.

We define the MaP measure as the rate of supra-threshold phonetic events, or peaks per second, from the filtered posteriorgram.[5] There are two pre-processing stages required to obtain this measure: calculating the phoneme filters and selecting an adequate λ threshold

3.3.1. Phonetic matched filters

Two filter banks were computed to estimate phoneme-specific filters from training data. One using 230 clean utterances[6] of the Aurora 4 *eval92* test set and the other one with the second half of the *SPRINT* utterances. Each subset was fed into a 5-hidden-layer (2048 hidden units per layer) DNN trained on clean data, as opposed to the multi-condition training for the rest of the models in this work. Monophone posteriorgrams were obtained from these activations in the same way described at the end of Section 2.1, every value lower than 0.1 was dropped to zero to reduce the noise from the softmax distribution. Afterward, phonetic events exceeding a λ threshold were extracted from the temporal trajectory for each utterance. This initial threshold was adopted from Kintzley et al. (2012) and later optimized to obtain *clean* filters.

For every phoneme, the peak frame of all consecutive events was aligned with the center of a 41-frame segment, subsequently averaged and normalized to use a single λ value for all filters, as described in Meyer et al. (2017). The amplitude of the filters depends on its shape and width. The filters were normalized with the 95th percentile of the maximum values to prevent clipping effects in the convolved posteriorgram. The resulting filterbank captures the average duration of each phoneme, for instance, vowel filters spread over several frames, in contrast, narrow filters represent voiceless stops such as /‘P’, ‘T’, ‘K’/.

3.3.2. Phonetic events threshold

The threshold λ determines which phonetic events from the filtered posteriorgram are included in the MaP measure count of peaks per second. This value needs to be carefully selected to estimate WER without prior knowledge reliably; selecting a very high threshold can lead to the exclusion of relevant events. Conversely, a low λ could increase the false positives particularly in deteriorated posteriorgrams or nearly uniform distributions. Another possibility to prevent the latter effect could have also been to apply a lower boundary to the posteriorgram before filtering as done in Kintzley et al. (2011). Preliminary experiments, however, deemed this strategy ineffective as thresholding early lead to missing segments of the activation patterns desired to match.

To determine the optimal threshold 17 values ranging from 0.15 to 0.95 in 0.05, we tested increments on all four acoustic scenes described in Section 2.2. Ideally, the relation between WER and MaP measure should be monotonic and exhibit small variances across different acoustic conditions. Fig. 3 shows the contrast between the worst ($\lambda = 0.2$) and best ($\lambda = 0.6$) values. Each curve represents the relation between WER and MaP measure for one acoustic scene averaged across all the DNN models.

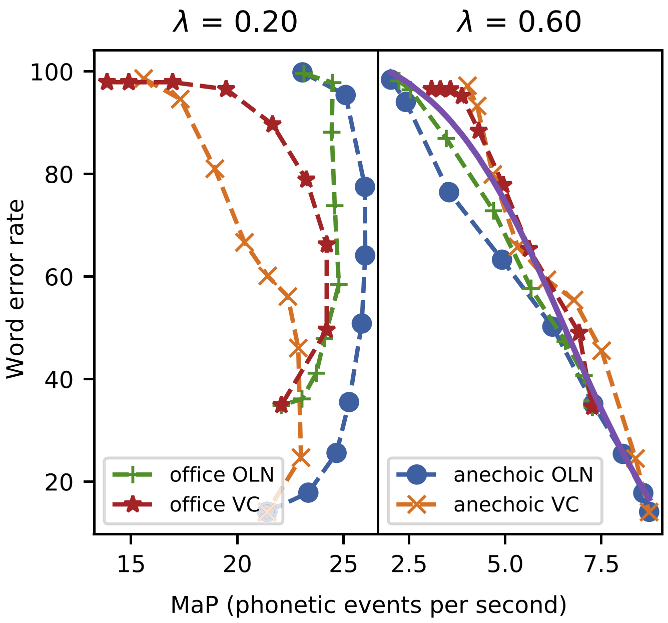

Fig. 3. Relation between WER and phonetic events per second for the worst (left) and best (right) thresholds. Each line represents a different acoustic scene. The thicker line is the resulting logistic function which is omitted in the left panel since the fitting algorithm did not converge.

For thresholds lower than 0.25 or higher than 0.85, the MaP values do not convey an appropriate measure for predicting WER and even appear to lose their predictability capacity after a particular SNR. This effect is especially noticeable in the scene with the diffuse noise type OLN. Conversely, for thresholds around 0.5, the WER/MaP curves are monotonic and are not so far apart from each other regardless of the condition or speech framework.

The following logistic function was fitted to all curves and used to linearize the WER/MaP relation, from which the Pearson correlation coefficient was calculated to quantify the range of thresholds.

$$S(x) = \frac{L}{1 + 10^{(x_0 - x)}} + y_0 \tag{6}$$

In this case, x represents the MaP measure, x_0 is the closest x value mapped to y_0, which is the median of a vector comprised of the WER from all 4 curves. L is the maximum value the function can assume and is the difference between the 5th and 95th percentiles of the same vector.

We found λ not to be very sensitive for values between 0.45 and 0.65, yielding correlation coefficients above 0.95. The highest value ($r = 0.98, p < .0001$) was found for $\lambda = 0.6$, which was therefore used in the subsequent experiments.

4. Results

4.1. Prediction of ASR error rates

In Figs 4 and 5, the three performance measures are plotted against the WER obtained from beamformer signals. Each of the four curves corresponds to one acoustic scene. Every data point corresponds to the average of the test utterances[7] for all azimuth angles at a single SNR

[5] Alternatively, other studies have collected the temporal position, and the magnitude of these phonetic events as an index searched table for fast keyword searching (Kintzley et al., 2011; 2012).

[6] The complete set consists of 330, but we excluded the 100 sentences used for creating the acoustic scenes. Additionally, upon comparing with the OLSA framework the optimal threshold did not change.

[7] 100 utterances in Fig. 4 for Aurora 4 and 240 in Fig. 5 for OLSA.

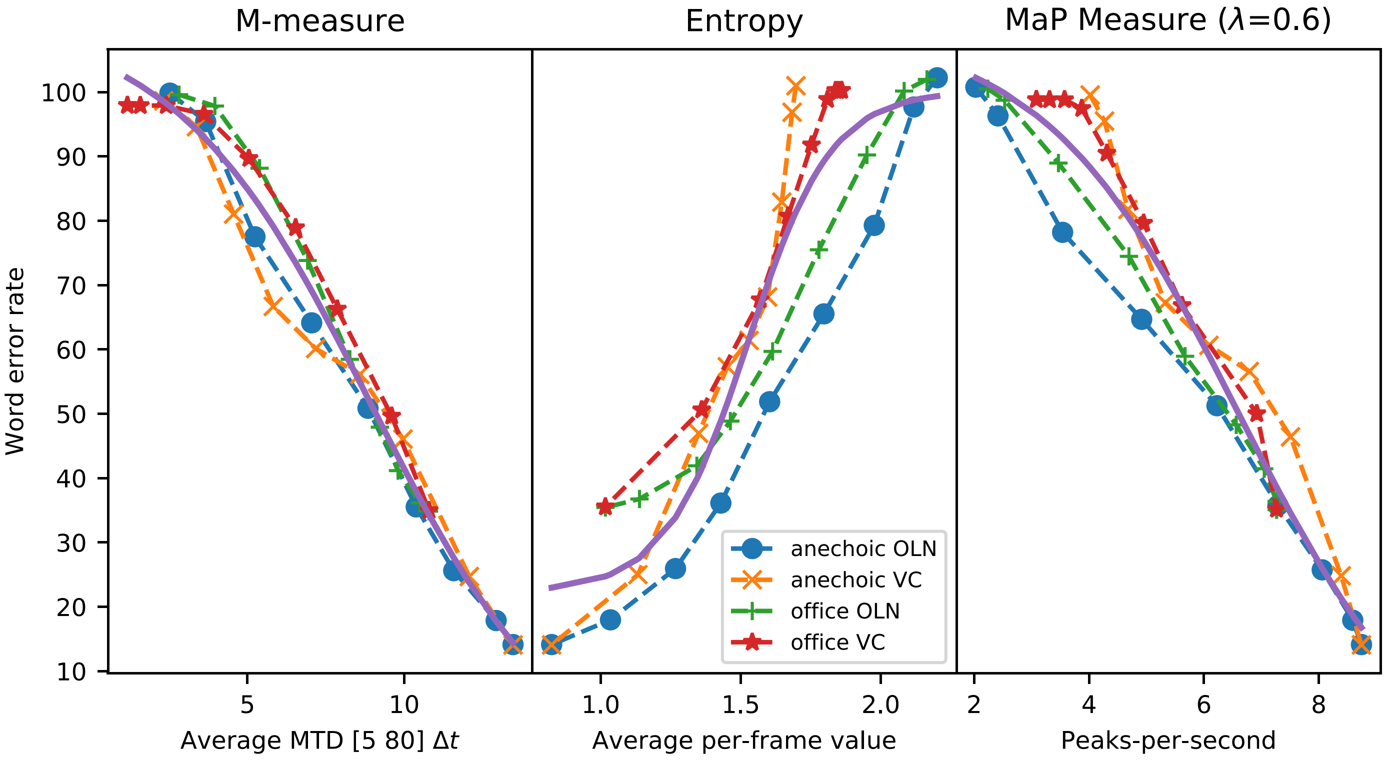

Fig. 4. Average WER against performance measures for all 4 acoustic scenes on the modified *eval92* test set. Each point along the curves indicates a different SNR ranging from −10 dB to clean. The data points in all curves are an average of all the utterances in every beamforming angle per condition. The logistic function used to fit the data is shown as a continuous thick line.

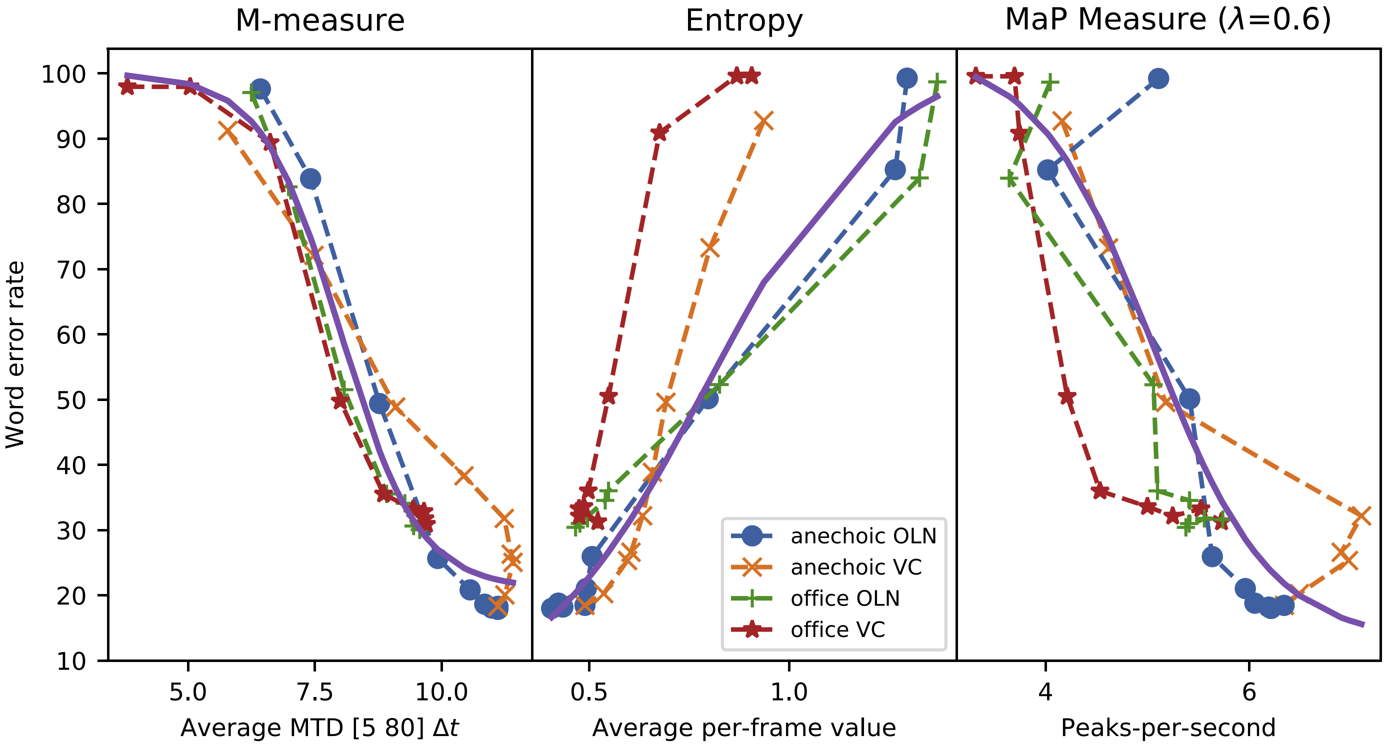

Fig. 5. Average WER against performance measures for all 4 acoustic scenes on the *SPRINT* test set. Each point along the curves indicates a different SNR ranging from −10 dB to clean. The data points in all curves are an average of all the utterances in every beamforming angle per condition. The logistic function used to fit the data is shown as a continuous thick line.

Table 2
Prediction errors calculated from each performance measures with respect to the beamformed signals for all the trained acoustic models. The WER of both the generated acoustic scenes and the Aurora 4 *eval92* test sets are shown for comparison.

Model	Parameters	WER (%)		Prediction error		
	10^6	Beamform	*eval92*	MTD	Entropy	MaP
2HL_256HU	0.78	48.22	15.39	4.80	9.97	5.97
2HL_512HU	1.81	49.62	14.02	**4.11**	**9.68**	**5.88**
2HL_1024HU	4.67	49.75	13.52	4.18	12.67	6.90
2HL_2048HU	13.53	51.39	13.39	4.73	13.49	7.70
3HL_256HU	0.84	44.30	15.22	5.38	11.89	10.07
3HL_512HU	2.07	46.07	13.71	5.27	14.33	8.64
3HL_1024HU	5.72	50.26	13.25	4.93	11.40	6.11
3HL_2048HU	17.73	50.64	**13.19**	5.18	13.63	6.40
4HL_256HU	0.91	47.92	14.88	5.38	13.85	8.73
4HL_512HU	2.34	49.21	13.77	5.73	13.59	8.71
4HL_1024HU	6.77	51.11	13.24	4.75	14.11	9.72
4HL_2048HU	21.92	50.08	13.25	4.54	10.38	6.34
5HL_256HU	0.97	47.72	14.98	5.00	13.67	7.81
5HL_512HU	2.60	45.61	13.52	6.02	16.16	13.82
5HL_1024HU	7.82	50.65	13.34	4.45	11.53	6.30
5HL_2048HU	26.12	46.03	13.38	4.58	15.56	8.47
6HL_256HU	1.04	**43.98**	15.00	5.17	13.36	9.45
6HL_512HU	2.86	47.53	13.66	5.76	18.53	14.50
6HL_1024HU	8.87	51.19	13.47	5.53	15.17	12.82
6HL_2048HU	30.31	47.70	13.36	4.69	12.02	8.32

value. As before, a logistic function was fitted to the data, followed by the correlation coefficient calculation (cf. Section 3.3.2).

Among all DNN models, the highest correlations were obtained by the architecture with 512 hidden units (HU) in each of its 2 hidden layers (HL), for M-measure ($r = 0.99$), Entropy ($r = 0.94$) and MaP ($r = 0.98$). Data obtained from this net is thus plotted in Figs. 4 and 5. In all cases, the correlation was highly significant ($p \leq 0.0001$).

The root-mean-square error between the curves and the logistic function represents the *prediction error*, which estimates how well a given performance measure predicts the WER. For both frameworks, the model which achieved the lowest prediction error was **2HL_512HU**.

4.1.1. Aurora 4

As shown in Fig. 4, besides the prediction capabilities of each performance measure, certain aspects can be noticed about the Aurora 4 acoustic scenes. For instance, there is a difference of approximately 22% between the lowest WER for the clean condition in scenes generated in anechoic and office conditions, which reflects the signal degradation due to reverberation.

Unlike the entropy, where the curves spread out making it hard for a single curve to fit all scenes, the other two measures were able to better adjust to the different acoustic conditions. For SNRs below 10 dB, the type of noise has a higher influence on entropy and MaP measure than the type of room. Evidence for this observation is the way the curves sharing the same noise type approach each other as the SNR decreases, whereas the curves sharing the same room converge towards the same value at higher SNRs. In contrast, all M-measure curve trajectories follow a similar path. The room appears to play a more determinant role in the highest variance region for the fit (from −5 to +5 dB).

For the MaP measure, the fixed threshold for phonetic events seems to be adequate for challenging acoustic scenes as the actual peaks per second seem to fall close to the logistic fit. Nevertheless, the WER of the signals in the anechoic room with the speech shaped-noise (*anechoic OLN*) is broadly overestimated. A possible solution will be to use a dynamic threshold which lowers depending on acoustic scene parameters such as the SNR or reverberation time.

Table 2 includes the prediction error for all setups (for the beamformed signals), the number of parameters on each DNN, as well as the WER on both test sets (beamformed signals from spatial acoustic scenes, labeled *Beamform* and the regular Aurora 4 test set *eval92*). The DNNs are ordered according to the number of hidden layers, not by model size.

The WER is averaged over all nine SNRs and four acoustic scenes, so even if the average values range from 43.98% to 51.39%, in some cases, the WER reaches even 100%; whereas the maximum WER yielded on the *eval92* test is 27.85%. When comparing the decoding scores on both test sets, the lowest average WER (43.98%) for beamformed signals was produced by the architecture **6HL_256HU**; on the *eval92* set the best performing model was **3HL_2048HU** yielding 13.19% WER. A more comparable setting is the WER at 15 dB of the signals with the speech-shaped OLN noise in the anechoic room which is 8.29%, against the Aurora 4 *babble* noise conditions whose utterances yield 10.84% WER using the **3HL_2048HU** model. For the localized VC noise against the analogous most impairing *street* noise from the *eval92* set, the corresponding errors are 14.53% and 27.24%. These numbers reflect mainly the role of speech enhancement by the beamformer. Comparing the clean conditions, the unprocessed utterances from Aurora 4 have the upper hand with comparable error rates of 3.6% for *clean* against 5.31%, which as pointed out in other studies such as Kayser et al. (2015) and Baumgärtel et al. (2015), one possible culprit could be the artifacts introduced by the spatial filtering.

The main reason for such suboptimal decoding results in the generated acoustic scenes could be the mismatch between training and test sets. Besides the spatial processing, the former contains speech signals with an SNR range of 10–20 dB, and the latter was also designed to cover SNRs as low as −10 dB which are relevant in boisterous conditions. Additionally, reverberation effects introduced by the office acoustic scene and beamforming artifacts, all of which are only covered in the test data, make both sets differ widely.

We could presumably have yielded a lower WER by modifying either training or test data with the simulation framework to obtain a more homogenous training-test set. Post-processing strategies might be applied to the training or test sets to cope with these adverse effects, such as dereverberation schemes. Nonetheless, as we were interested in mismatched data as often encountered in realistic scenarios both in ASR and in hearing aid processing, we chose not to increase the similarity of both sets. Another reason for not covering spatial processing in training is to highlight the information provided by the analyzed posteriorgrams concerning the correct beamforming angle; our intuition is that the sensitivity towards suboptimal beamforming should be higher for the model trained on close-talk recordings only.

The exact model size does not seem to be a crucial parameter in predicting errors obtained from performance measures. Except for the two-hidden-layers architecture, 512 hidden units per layer were found to produce relatively high prediction errors for all measures.

4.1.2. OLSA

On the OLSA framework, depicted in Fig. 5, an optimal function able to adequately fit the data points of the four acoustic scenes was not found for the entropy and the MaP measures, the reason being the widely different responses to the different environments.

The Entropy values plotted against the WER show for all conditions a monotonic behavior, but in the scenes with VC noise, the entropy increase has a considerably higher rate than in the scenes with the speech-shaped noise. This effect results in highly underestimated WERs for lower SNRs in the acoustic scenes with the localized VC noise.

For the MaP measure, the curves do not exhibit a monotonic behavior for any condition. Even when the MaP values at high SNRs are in a narrow range in all conditions, the peaks-per-second spread increases in the lower SNRs for the acoustic scenes with the speech-shaped noise; whereas in the other two conditions, the opposite effect occurs for higher SNR values.

Table 3
Prediction errors calculated from each performance measures with respect to the beamformed signals for all the trained acoustic models. The WER of both the generated acoustic scenes and the *SPRINT* test sets are shown for comparison.

Model	Parameters	WER (%)		Prediction error		
	10^6	Beamform	*SPRINT*	MTD	Entropy	MaP
2HL_256HU	0.66	45.19	**16.36**	9.01	11.69	17.08
2HL_512HU	1.58	**44.92**	16.62	**4.37**	**10.08**	16.33
6HL_2048HU	29.39	45.71	16.78	5.39	15.17	**15.12**

Because of the small vocabulary size of the OLSA framework and the particular use case of the *SPRINT* test set, the WERs on these acoustic scenes were lower than the ones generated with Aurora 4. Conversely, the WERs yielded in the monaural version of the test set were higher than the clean *eval92* set because of the natural responses using several words not seen in the training set.

Analogous to Table 2 for the *eval92* test set, Table 3 shows the prediction errors for the *SPRINT* data. From the tested ASR architecture, the system with 512 HU and 2 HL showed the lowest WER and prediction error for MTD and Entropy. The lowest WER yields the system with 256 HU and 2 HL. The biggest network produces the best prediction error for the MaP measure; however, due to the effects mentioned above, the MaP measure produces an unreliable fitting curve which is reflected in the highest prediction error regardless of the network.

4.2. Performance measures for hearing aid signals

Performance measures were used to improve speech intelligibility in hearing aids by selecting speech-specific parameters for optimal beamforming and SNR estimation. Both approaches have been found to directly correlate with an increase of intelligibility, albeit in combination with speech enhancements and noise suppression techniques.

4.2.1. Aurora4

In Fig. 6 the trajectories of the three performance measures are plotted when varying the beam angle at five different SNRs in the most challenging acoustic scene: the office room with the VC noise. The values were scaled down with the maximum value to set the range between 0 and 1. Because the WER positively correlates with the entropy the minimum entropy should indicate the location of the target speaker. Conversely, in the other two measures, which negatively correlate with the WER, the indicator should be a peak in the correspondent values.

The M-measure is the only capable of detecting the target speaker region around −30°, at all but the highest SNR, where the value peaks at −20°. Furthermore, the steep descent taken by these trajectories from −30° to +10° reveals the approximate position of the noise source located at +40° and separates the two hemispheres even at the lowest SNR (−10 dB).

For the entropy trajectory for clean speech there is no evident distinct value to identify the target speaker in the acoustic scene, nor is there a discernible pattern, presumably because the beamformer does not provide a substantial gain, so relatively stable M-Measure values are obtained for a wide range of azimuth angles. Those values are elevated in noisy conditions for an azimuth of approx. -90 degrees. This phenomenon could arise from an early reflection in the office environment used for the corresponding acoustic scene. At −10 dB and 0 dB, there is an overlap between those curves, the lowest value at 0° and a maximum peak of +50° which might be due to the presence of the noise at +40°. At positive SNRs, the trajectories start separating the hemisphere where the target and noise sources are; the entropy curve at 20 dB is the only one accurately detecting the target source, by having the lowest value approximately at −30°, and even a peak precisely at +40°.

Both MaP and entropy measures are quite affected by reverberation effects, to the extent of being utterly unreliable for azimuth angle prediction at SNRs below 20 dB. Smoother patterns were found in the anechoic room conditions, although peaks and valleys still were shifted from the location of the target speaker and noise source should be. The M-measure positively correlates with the acoustical resemblance between testing and training data; for instance, if an ASR system were trained using clean speech, the lower the SNR was in the test utterances, the lower the value would be (Hermansky et al., 2013); therefore the M-measure could determine the highest SNR input from multiple channels. We investigated whether this correlation exists in the other two measures. Being able to estimate the SNR accurately in an unknown acoustic scene could determine the degree of enhancement or suppression applied to the signal and even the appropriate technique.

The relation between SNR and performance measures turned out to be monotonic (increasing for M-measure and MaP, decreasing for entropy). However, the MaP measure trajectory from the office room scenarios exhibited a peak at 20 dB and slightly decreased at 30 dB. Furthermore, the trajectories from the anechoic and office scenes diverged considerably from each other, indicating SNR estimation via performance measures is sensitive to room configuration.

Table 4
Prediction errors of the beamformer signals calculated from each performance measures for the least complex and the best performing models. * These prediction errors are computed with the lowest WER.

Model	MTD	Entropy	MaP
2HL_256HU	4.80	9.97	5.97
2HL_256HU*	5.00	10.03	6.14
2HL_512HU	4.11	9.68	5.88
2HL_512HU*	4.92	11.68	7.45

4.2.2. OLSA

The corresponding trajectories estimated with the *SPRINT* data is shown in Fig. 7. Same as with the Aurora 4 setup, the only performance measure able to capture the correct target speaker azimuth angle is the M-measure. It reflects increased values between −90° and 0°, with a peak in all SNR at −30°. At +40° the M-Measure also shows a small peak in the −10 dB SNR condition, but the overall values above 0° are distinct smaller which enables a secure selection of the correct angle.

The Entropy only shows a small peak for the conditions with positive SNRs. For the −10 dB SNR condition the lowest (i.e. best) Entropy value is at +40°.

The MaP measure at positive SNRs highlights the hemisphere where the speaker is located although the peaks do not correspond to the correct azimuth angle. This pattern gets lost as the SNRs diminishes until there is no clear relation between the measure and the target speaker.

4.3. Computational complexity analysis and optimization

The prediction errors from Table 2 are computed using the WERs from the same model from which the performance measures are extracted. In other words, these measures aim to predict the error rate yielded by the corresponding DNN. However, the more complex the model is, the more computationally expensive it becomes to obtain these measures. Therefore, the possibility of using a relatively compact architecture was tested to predict error rates yielded by more complex ones. For optimization purposes, we compared the change in prediction error between the best performing model **2HL_512HU** and the model with the least number of parameters **2HL_256HU**, using the oracle knowledge about the minimum WER obtained by any of the 20 acoustic models with the beamformed signal at −30°.

The results are shown in Table 4.

Unlike the rest of the models, the model with the fewer parameters seems to be robust to reparametrization, as the prediction errors

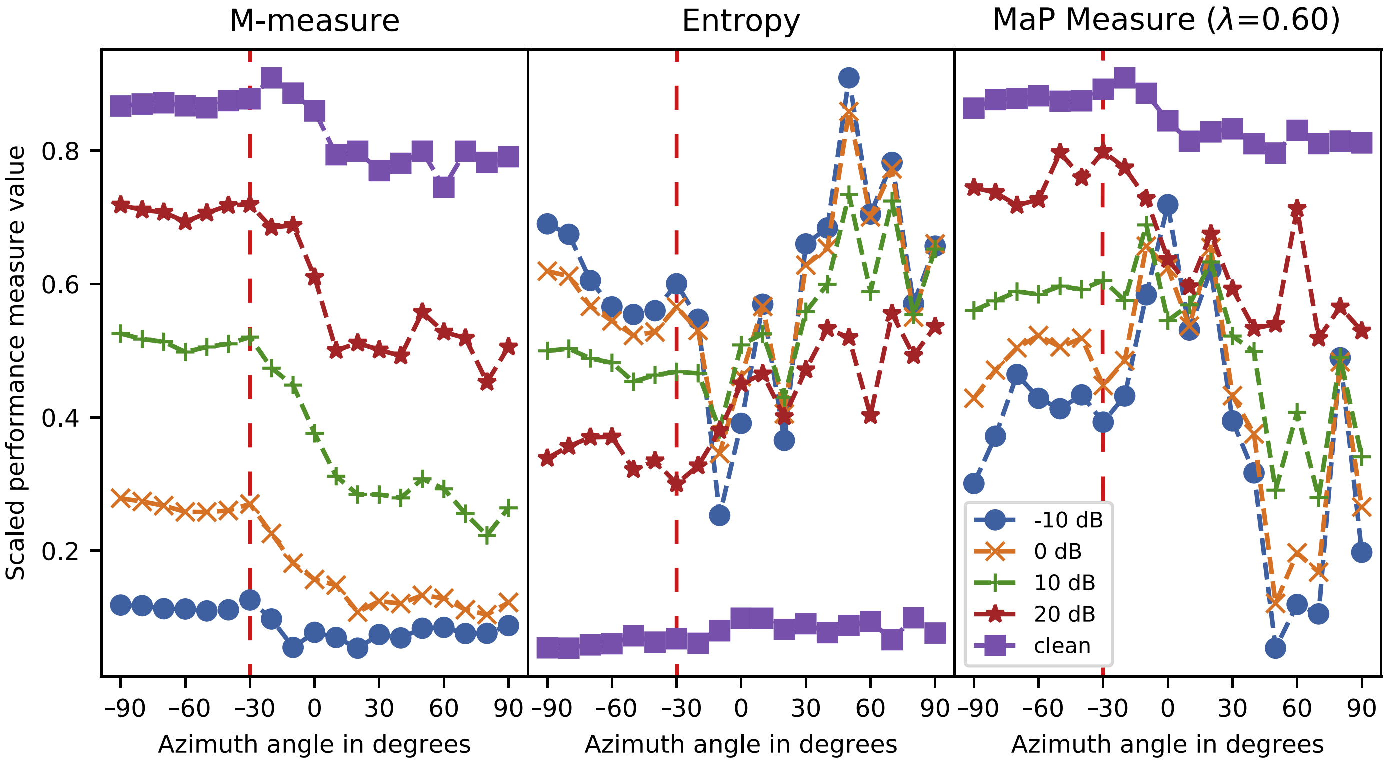

Fig. 6. Performance measures at every azimuth angle for the beamformer in the most acoustically challenging scene (office room with VC noise) for the *eval92* test data. The dashed vertical line indicates the target speaker position −30°. Each trajectory represents a different SNR from −10 dB to clean speech. For comparison the values are scaled down with the maximum, setting the range between 0 and 1.

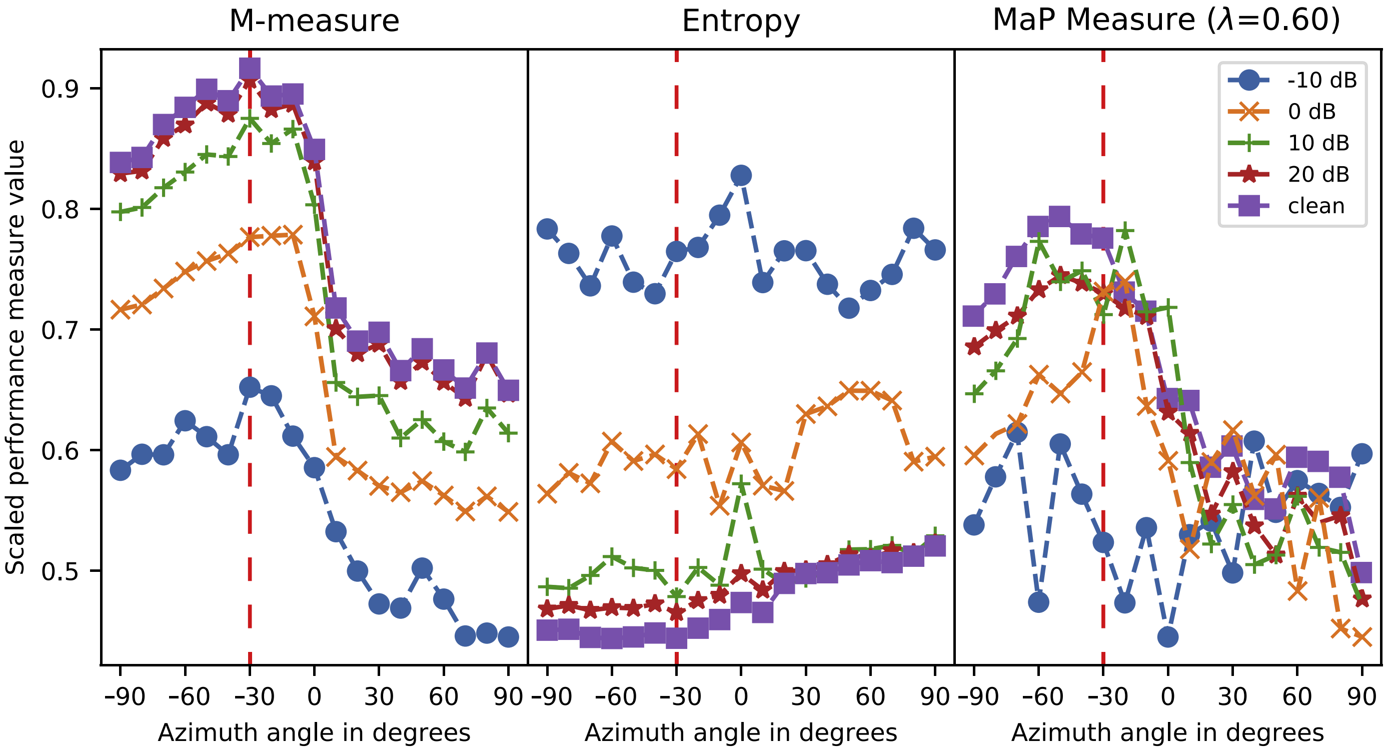

Fig. 7. Performance measures at every azimuth angle for the beamformer in the most acoustically challenging scene (office room with VC noise) for the *SPRINT* test data. The dashed vertical line indicates the target speaker position −30°. Each trajectory represents a different SNR from −10 dB to clean speech. For comparison the values are scaled down with the maximum, setting the range between 0 and 1.

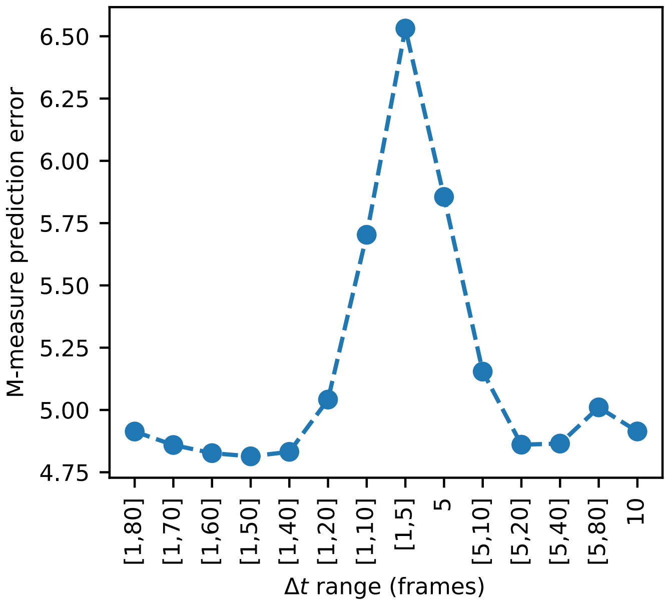

Fig. 8. M-measure prediction error averaged across all models for different Δt ranges. The first five values from 1 to 5 are separated in steps of 1 frame and the rest from 10 to 80 in steps of 5 frames.

rose relatively 4.2% and 2.9% for the M-measure and MaP respectively, whereas there was almost no increase for entropy. For other models, a much broader relative increase in prediction error was obtained when using the minimum WER: the average relative increases were approximately 15% for M-measure, 12% for Entropy and ∼19% for MaP measure.

Decreasing computational complexity while preserving the prediction capabilities is a crucial trade-off for performance monitoring adoption in ASR enhanced technologies; mainly when memory restricts the size of the architecture. Hardware, in general, determines the latency of computing performance measures. Even if the M-measure excels both in predicting WER and in accurately estimating the desired azimuth angle for the beamformer, there are some potential challenges to consider before implementing this procedure in hearing aids. The most relevant is the calculation that requires a broad temporal context for averaging the symmetric KL-divergence (5) between frames.

Based on preliminary experiments for a small amount of speech data, a Δt range of 50 to 800 ms was taken into account for calculating the M-measure. Results for the complete data set with varying temporal context are shown in Fig. 8. Despite not resulting in the lowest prediction error, being off by 0.19 from the absolute minimum, the selected window is a relatively stable parameter across all models. Due to these restrictions present in hearing aids, particularly the computational complexity, obtaining the mean temporal distance for $\Delta t = 80$ (from equation (4)) is prohibitively expensive. A possible compromise to keep latency low and still get reliable estimates is restricting to a maximum window of 100 ms. The eligible ranges are: [1,10], [1,5], [5,10], 5, and 10 frames, among which the lowest prediction error corresponds to using the single value of $\Delta t = 10$. Further details about the technical analysis are presented in the following section.

5. Technical feasibility for hearing devices

Hearing devices and in particular, hearing aid processors are limited concerning hardware capabilities and resources. These limitations are mainly due to the small size of the hearing aids and the requirement for battery operation. Therefore, strict power consumption constraints and silicon area restrictions must be met by the processors used in hearing aid devices. On the contrary, the architectures of these processors have to be highly optimized for parallel processing on the instruction- and data-level to meet the performance requirements for real-time processing of computationally complex hearing aid algorithms (Chen et al., 2016; Roeven et al., 2004; Hartig et al., 2014).

In this section, the technical feasibility of implementing the DNN-based performance measures on hearing aid processors is discussed. For this purpose, processing performance and latency were evaluated. The processing performance is determined in cycle counts per frame to verify the feasibility of the proposed algorithm. An appropriate operation frequency of 50 MHz for the hearing aid processor was selected to meet real-time processing, also taking into account resulting power consumption, silicon area, and processing delay constraints (Roeven et al., 2004; Qiao et al., 2011).

The hearing aid processor adopted in this study is described in Hartig et al. (2014) and shown in Fig. 10. This processor supports the parallel processing on the instruction- and data-level by executing two SIMD (single instruction multiple data) very long instruction words (VLIW) in parallel. Commonly used specialized functional units for processing audio data are available like a multiply-accumulate unit, arithmetic units, barrel-shifters, and permutation units. Furthermore, 64 general purpose registers are implemented, in order to keep the number of required main memory accesses low.

The first part of DNN-based performance measures is the extraction of the 40-dimensional log-Mel-spectral coefficients, which are fed to the input layer of the DNN. This part consists mainly of fast Fourier transforms, logarithmic functions, and discrete cosine transforms. The processing of this part requires 32958 cycles, which were measured in a cycle accurate profiling step using an HDL (hardware description language) simulator. The processor is equipped with some specialized functional hardware units, like a complex-valued MAC unit (Gerlach et al., 2015), for speeding up the computation of a fast Fourier transform. Nevertheless, the computational complexity of these part is low, compared to the processing of the forward path of the DNN, described in the following. At a clock frequency (f_C) of 50 MHz, which partly accounts for a processing time (T_P) of 0.659 ms for one frame, according to the following equation:

$$T_P = \frac{cycles}{f_C} \tag{7}$$

The assessment of the technical feasibility depends on the computational complexity of the different DNNs. An extrapolated estimate of the processing cycle counts required for a forward pass on every evaluated DNN is shown in Fig. 9. This calculation is based on a hand-optimized assembly implementation on the same exemplary processor architecture shown in Fig. 10.

The forward passes are mainly matrix multiplications. The activation functions are sigmoid for the hidden units and softmax for the output layer. Arithmetic operations like exponential and division are needed to compute the activation functions. These were taken into account with a cycle count equivalent to the word width (32-bit), under the assumption that a single coordinate rotation digital computer (CORDIC) hardware accelerator was used to compute these operations (Tiwari and Khare, 2015). Such a hardware accelerator is a possible extension for a hearing aid processor; due to its flexibility, it can increase the performance of other algorithms as well (Gerlach et al., 2017).

Despite the high cycle counts per operation, in this case, 32 cycles, the required number of cycles needed for the multiplications and additions within the matrix multiplication computations in the forward pass is by far greater. Hence, the computation of the forward path dominates the total cycle count. Even if the exponential and division operations could be computed within one cycle, the total cycle count would be decreased by just 1.8 % on average. The matrix multiplications in the

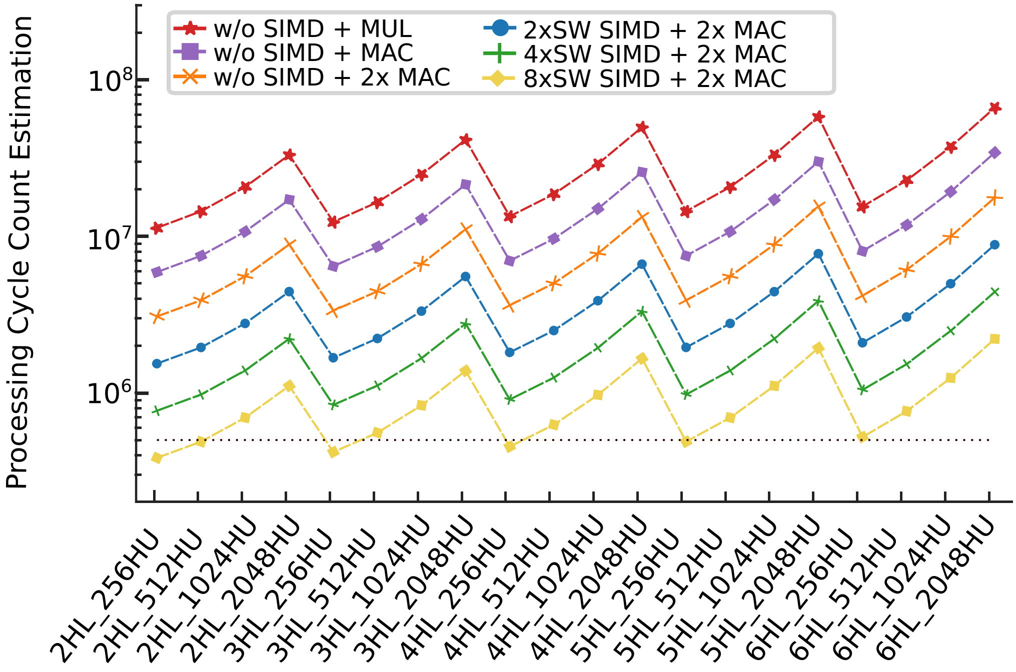

Fig. 9. Forward pass cycle count estimation for all DNN models with 256 up to 1024 hidden units (HU) and 2 –6 hidden layers (HL). The number of cycles are estimated for the use of SIMD (single instruction, multiple data) on 2–8 data points or subwords (SW) and either multiply-only (MUL) or multiply-accumulate (MAC) units. The black dashed line represents the maximum cycle count for online processing.

forward pass are of the following form:

$$\begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix} \times \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \tag{8}$$

where $\boldsymbol{v}$ and $\boldsymbol{u}$ represent the values of the neurons, $\boldsymbol{w}$ is the weight matrix and $\boldsymbol{b}$ is the bias vector. The processing time to compute these matrix multiplications depends on the parameters m and n.

In order to meet the performance requirements for real-time processing of these matrix multiplications, the required cycles have to be processed in time by a processor with a given clock frequency (fC) of 50 MHz. The available instruction level parallelism can be enhanced with specialized function units or even with data level parallelism mechanism to decrease the processing cycle count required.

A specialized but commonly used hardware unit is the multiply-and-accumulate (MAC) unit (Arm et al., 2009; Gerlach et al., 2015), which performs a product of two numbers and adds the result to the accumulator in one cycle. This feature can be effectively used for these matrix multiplications, instead of using a multiply instruction (MUL) followed by an add instruction (ADD).

Instruction-level parallelism is typically exploited by using a very long instruction word (VLIW) instruction set architecture (Qiao et al., 2011; Hartig et al., 2014). This process enables the execution two or more instructions in parallel per cycle; for example, two MAC units can be used by processor architectures (Arm et al., 2009), which execute two instructions in parallel. This processor feature can be used for matrix multiplications nearly halving the cycles required.

A well-known and applied data level parallelism mechanism is the so-called single instruction multiple data (SIMD), which is applicable to hearing aids (Ku et al., 2013; Gerlach et al., 2017). Typical SIMD mechanisms perform the same operation in parallel on multiple input data organized in various subwords (SW) of equal size (2xSW, 4xSW, ...).

Therefore, due to few dependencies in the calculation of the DNN forward path, the computation is nearly fully parallelizable for both levels: instruction and data. The inner loop of the handwritten scheduled assembler code for the matrix multiplications given in Eq. (8) is shown in Fig. 11. The processor architecture depicted in Fig. 10 supports instruction level parallelism, by processing two instructions per cycle on the issue slot #0 and issue slot #1. The specific multiply-and-accumulate operations (MAC_32 instructions) are SIMD instructions, processing two subwords (SW) with 32 bits each in parallel. These MAC_32 instructions are scheduled in parallel to the memory accesses (MV instructions). The MAC_32 instruction writes the result in two target registers, which combined double the data width to avoid overflow during fixed-point operations. Loop unrolling was used to achieve 1.88 instructions per cycle (IPC) on average. As a result, 10 vector elements can be multiplied and added in 9 cycles.

For the best performing model **2HL_512HU**, in terms of prediction error, and the model with the least number of parameters **2HL_256HU**, the number of cycles required can be reduced with the aforementioned hardware architectures by a factor of approximately 30 (Fig. 9) compared to the sequential implementation without SIMD instructions and a single multiply instruction (w/o SIMD + MUL). Based on these results, the lowest processing time achievable according to (7) of the complete DNN is 7.69 ms for 384634 cycles and a clock frequency of 50 MHz.

Following (4), the temporal distance between 2 propagated frames separated by 100 ms can be calculated as soon as the last frame is processed. This computation includes several independent division and logarithmic operations. These operations were also computed with the CORDIC hardware accelerator within 32 cycles each, resulting in 88000 estimated cycles in addition to the forward pass of the second frame. Using two hardware accelerators in parallel (Gerlach et al., 2017) the processing time would be approximately 0.88 ms for a processor/co-processor clock frequency of 50 MHz.

The total estimated processing time of this algorithm would be dependent on the number of temporal distance values to be averaged. Finding an optimal observation window falls out of the scope of this study as it involves a detailed memory analysis of the particular setup to be implemented on. However, holding up to 20 values in memory would not add up a considerable amount of processing cycles. Taking into account the impact of the reduced fixed-point accuracy in the DNN and

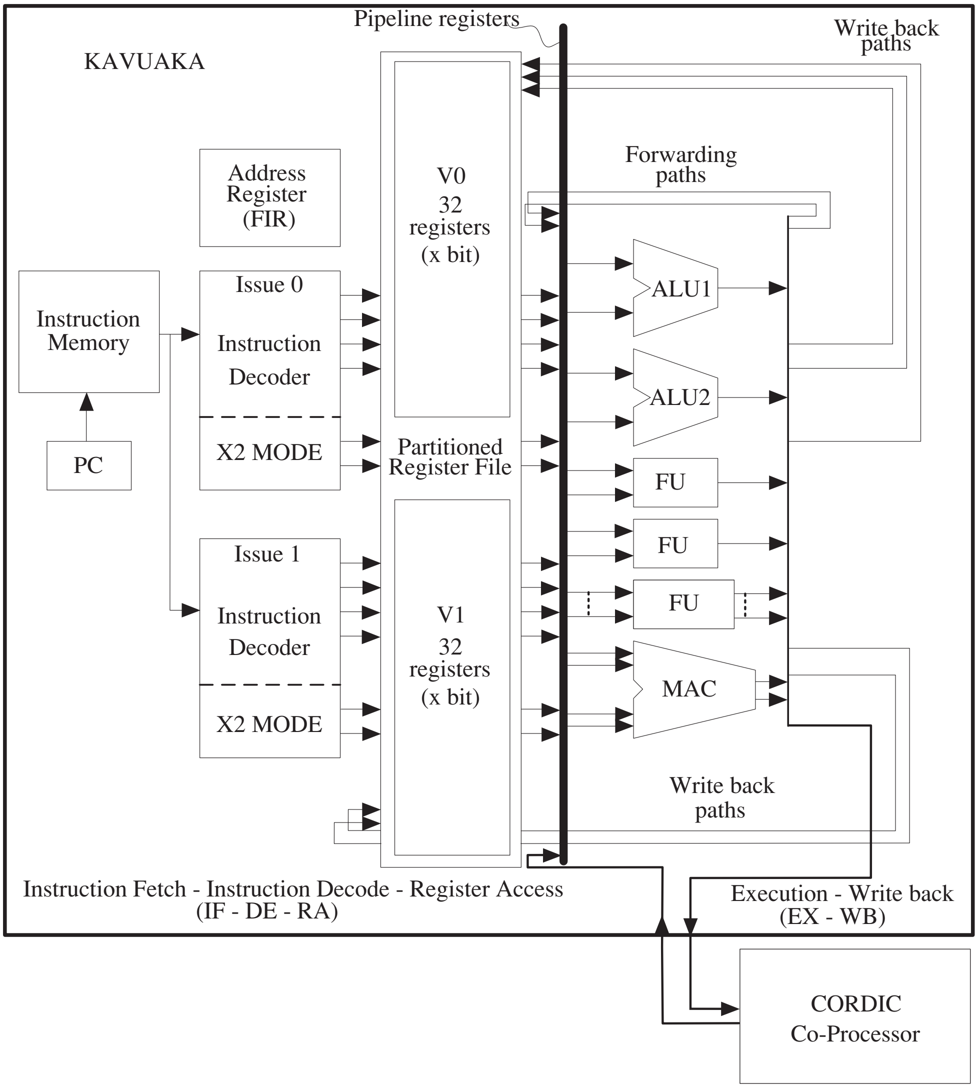

Fig. 10. Very Long Instruction Word (VLIW) processor architecture (Hartig et al., 2014) for hearing aid devices. The processor is divided into two pipeline stages. Two instructions are decoded and then executed by the specialized functional units, like a complex-valued MAC unit (Gerlach et al., 2015). An external CORDIC co-processor is attached.

```
//Issue-slot  #0           ;  Issue-slot  #1
:LOOP
MV       V1R0,   FIR0+   ; MV       V1R1,       FIR1+
MV       V1R2,   FIR0+   ; MV       V1R3,       FIR1+
MV       V1R0,   FIR0+   ; MAC_32  V0R2+V0R3,  V1R0,   V1R1
MV       V1R1,   FIR0+   ; MAC_32  V0R2+V0R3,  V1R2,   V1R3
MV       V1R2,   FIR1+   ; MV       V1R3,       FIR0+
MV       V1R0,   FIR1+   ; MAC_32  V0R2+V0R3,  V1R0,   V1R2
MV       V1R2,   FIR1+   ; MAC_32  V0R2+V0R3,  V1R1,   V1R0
LOOPR   V0R1,   LOOP    ; NOP
NOP;  MAC_32  V0R2+V0R3,  V1R3,   V1R2
```

Fig. 11. Scheduling of handwritten assembler code of the inner loop for the matrix multiplications. Two instructions are processed in parallel. Up to three registers (V0R0-V1R31) are addressed per instruction, whereas the first one is the target register. Move instructions (MV) copy data from the main memory, using pointers stored in FIR registers. Multiply-and-accumulate (MAC) are executed in parallel. The suffix _32 indicates the subword operation mode (SIMD). The MAC_32 instruction writes the result in two target registers, to avoid overflow during fixed-point operations.

the performance of the measure, implementing the M-measure could be feasible in hearing aid hardware, and the latency would be less than the frame shift of 10 ms.

6. Conclusions

Performance measures have been previously used for various purposes such as evaluating the decision reliability of DNN classifiers. In this study, the main aim was to use these measures for predicting word error rates in ASR systems and optimize the beamforming angle in hearing aids. We took into account four different room scenarios over a realistic selection of SNRs and two different speech frameworks to test the generalization capabilities of the performance measures despite the effects of spatial filtering, reverberation, and noise type.

Given the low prediction error concerning WER and the accurate estimation of the desired azimuth angle for the beamformer, the M-measure appears to be the most suitable candidate for performance monitoring. Furthermore, this measure is robust against new acoustic scenes and noise types even when the beamformer is steered in wrong directions; yielding an accurate prediction in the presence of unseen data. The model which produced the lowest prediction error was **2HL_512HU**.

Likewise, when observing the performance measures at different azimuth angles, the M-measure provided the most accurate position of the target speaker and the noise source. In contrast, both the MaP measure and the entropy severely degraded in low SNR setups.

The exact model size does not seem to be a crucial parameter in predicting errors obtained from performance measures; it does play a roll in computational complexity and latency. Both factors are decisive in online applications such as hearing aid parameter optimization. As the most time-consuming step in the M-measure calculation is the symmetric KL-divergence, the best trade-off between performance and latency was taking a temporal context of 100ms ($\Delta t = 10$).

Finally, the M-measure implementation in hearing aids was deemed feasible with a combination of parallel architectures, hardware acceleration and model weights quantization. The lowest possible processing time of the full forward pass was 7.69 ms with a clock frequency of 50 MHz. The optimized M-measure calculation would add 0.88 ms when using 10 frames as temporal separation. In other words, the M-measure could be potentially integrated into hearing aid hardware for parameter optimization.

Acknowledgment

This work was funded by the DFG (Cluster of Excellence 1077/1 Hearing4All (http://hearing4all.eu)).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.specom.2018.11.006.

References

Adiloğlu, K., Kayser, H., Baumgärtel, R.M., Rennebeck, S., Dietz, M., Hohmann, V., 2015. A binaural steering beamformer system for enhancing a moving speech source. Trends Hear. 19. 2331216515618903.

Arm, C., Gyger, S., Masgonty, J.-M., Morgan, M., Nagel, J.-L., Piguet, C., Rampogna, F., Volet, P., 2009. Low-power 32-bit dual-mac 120 w/mhz 1.0 v icyflex1 dsp/mcu core. IEEE J. Solid-State Circuits 44 (7), 2055–2064.

Assmann, P., Summerfield, Q., 2004. The perception of speech under adverse conditions. In: Speech Processing in the Auditory System. Springer, pp. 231–308.

Baumgärtel, R.M., Hu, H., Krawczyk-Becker, M., Marquardt, D., Herzke, T., Coleman, G., Adiloğlu, K., Bomke, K., Plotz, K., Gerkmann, T., et al., 2015. Comparing binaural pre-processing strategies ii: speech intelligibility of bilateral cochlear implant users. Trends Hear. 19. 2331216515617917.

Bitzer, J., Simmer, K.U., 2001. Superdirective microphone arrays. In: Microphone Arrays. Springer, pp. 19–38.

Chen, C., Chen, L., Fan, J., Yu, Z., Yang, J., Hu, X., Hei, Y., Zhang, F., 2016. A 1V, 1.1 mW mixed-signal hearing aid SoC in 0.13 μm CMOS process. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, pp. 225–228.

Cornelis, B., Moonen, M., Wouters, J., 2014. Reduced-bandwidth multi-channel wiener filter based binaural noise reduction and localization cue preservation in binaural hearing aids. Signal Process. 99, 1–16.

Cox, H., Zeskind, R., Owen, M., 1987. Robust adaptive beamforming. IEEE Trans. Acoust. 35 (10), 1365–1376.

Gerlach, L., Marquardt, D., Payá Vayá, G., Liu, S., Weibrich, M., Doclo, S., Blume, H., 2017. Analyzing the trade-off between power consumption and beamforming algorithm performance using a hearing aid asip. In: Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XVII). IEEE.

Gerlach, L., Payá-Vayá, G., Blume, H., 2015. An area efficient real- and complex-valued multiply-accumulate SIMD unit for digital signal processors. In: Proceedings of the IEEE Workshop on Signal Process. Systems (SiPS). IEEE, pp. 1–6.

Hamacher, V., Kornagel, U., Lotter, T., Puder, H., 2008. Binaural signal processing in hearing aids: technologies and algorithms. Adv. Digital Speech Transm. 14, 401–429.

Hartig, J., Gerlach, L., Payá-Vayá, G., Blume, H., 2014. Customizing a vliw-simd application-specific instruction-set processor for hearing aid devices. In: Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS). IEEE, pp. 1–6.

Hawley, M.L., Litovsky, R.Y., Colburn, H.S., 1999. Speech intelligibility and localization in a multi-source environment. J. Acoust. Soc. Am. 105 (6), 3436–3448.

Hermansky, H., 2013. Multistream recognition of speech: dealing with unknown unknowns. Proc. IEEE 101 (5), 1076–1088.

Hermansky, H., Variani, E., Peddinti, V., 2013. Mean temporal distance: predicting asr error from temporal properties of speech signal. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7423–7426.

Hinton, G., 2010. A practical guide to training restricted boltzmann machines. Momentum 9 (1), 926.

ITU-T, 2011. Recommendation P.341 Transmission Characteristics for Wideband Digital Loudspeaking and Hands-free Telephony Terminals, p. 30.

Kayser, H., Ewert, S.D., Anemüller, J., Rohdenburg, T., Hohmann, V., Kollmeier, B., 2009. Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses. EURASIP J. Adv. Signal Process. 2009, 6. doi:10.1155/2009/298605.

Kayser, H., Spille, C., Marquardt, D., Meyer, B.T., 2015. Improving automatic speech recognition in spatially-aware hearing aids. In: Proceedings of Interspeech, pp. 175–179.

Kintzley, K., Jansen, A., Hermansky, H., 2011. Event selection from phone posteriorgrams using matched filters. In: Proceedings of Interspeech, pp. 1905–1908.

Kintzley, K., Jansen, A., Hermansky, H., 2012. Map estimation of whole-word acoustic models with dictionary priors. In: Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, pp. 787–790.

Ku, Y., Sohn, J., Han, J., Baek, Y., Kim, D., 2013. A high performance hearing aid system with fully programmable ultra low power dsp. In: Proceeedings of the IEEE International Conference on Consumer Electronics (ICCE), pp. 352–353.

Mallidi, S.H., Ogawa, T., Hermansky, H., 2015. Uncertainty estimation of dnn classifiers. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, pp. 283–288.

Marquardt, D., Hadad, E., Gannot, S., Doclo, S., 2015. Theoretical analysis of linearly constrained multi-channel wiener filtering algorithms for combined noise reduction and binaural cue preservation in binaural hearing aids. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (12), 2384–2397.

Meyer, B.T., Mallidi, S.H., Castro Martínez, A.M., Payá-Vayá, G., Kayser, H., Hermansky, H., 2016. Performance monitoring for automatic speech recognition in noisy multi-channel environments. In: Proceedings of the Spoken Language Technology Workshop (SLT) IEEE. IEEE, pp. 50–56.

Meyer, B.T., Mallidi, S.H., Kayser, H., Hermansky, H., 2017. Predicting error rates for unknown data in automatic speech recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5009–5013.

Misra, H., Bourlard, H., Tyagi, V., 2003. New entropy based combination rules in hmm/ann multi-stream asr. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), 2. IEEE, pp. II–741.

Okawa, S., Nakajima, T., Shirai, K., 1999. A recombination strategy for multi-band speech recognition based on mutual information criterion. In: Proceedings of Eurospeech, 99, pp. 603–606.

Ooster, J., Huber, R., Kollmeier, B., Meyer, B.T., 2018. Evaluation of an automated speech-controlled listening test with spontaneous and read responses. Speech Commun. 98, 85–94. doi:10.1016/j.specom.2018.01.005.

Parihar, N., Picone, J., Pearce, D., Hirsch, H.-G., 2004. Performance analysis of the aurora large vocabulary baseline system. In: Proceedings of the 12th European Signal Processing Conference. IEEE, pp. 553–556.

Pertilä, P., Nikunen, J., 2014. Microphone array post-filtering using supervised machine learning for speech enhancement. In: Proceedings of Interspeech, pp. 2675–2679.

Postma, A., 2000. Detection of errors during speech production: a review of speech monitoring models. Cognition 77 (2), 97–132.

Qiao, P., Corporaal, H., Lindwer, M., 2011. A 0.964 mw digital hearing aid system. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2011. IEEE, pp. 1–4.

Roeven, H., Coninx, J., Ade, M., 2004. Coolflux dsp-the embedded ultra low power c-programmable dsp core. In: Proceedings of the International Signal Processing Conference (GSPx). Citeseer.

Rohdenburg, T., Goetze, S., Hohmann, V., Kammeyer, K.-D., Kollmeier, B., 2008. Combined source tracking and noise reduction for application in hearing aids. In: Proceedings of the ITG Conference on Voice Communication (SprachKommunikation). VDE, pp. 1–4.

Scheffers, M.K., Coles, M.G., 2000. Performance monitoring in a confusing world: error-related brain activity, judgments of response accuracy, and types of errors. J. Exp. Psychol. Hum. Percept. Perform. 26 (1), 141.

Souden, M., Araki, S., Kinoshita, K., Nakatani, T., Sawada, H., 2013. A multichannel mmse-based framework for speech source separation and noise reduction. IEEE Trans. Audio Speech Lang. Process. 21 (9), 1913–1928.

Spille, C., Kayser, H., Hermansky, H., Meyer, B.T., 2016. Assessing speech quality in speech-aware hearing aids based on phoneme posteriorgrams. In: Proceedings of Interspeech, pp. 1755–1759.

Tiwari, V., Khare, N., 2015. Hardware implementation of neural network with sigmoidal activation functions using cordic. Microprocess. Microsyst. 39 (6), 373–381.

Vesely, K., Ghoshal, A., Burget, L., Povey, D., 2013. Sequence discriminative training of deep neural networks. In: Proceedings of Interspeech.

Wagener, K., Brand, T., Kollmeier, B., 1999. Entwicklung und evaluation eines satztests für die deutsche sprache ìli; evaluation des oldenburger saiztests. ZAudiol 1999c; 38: 86 95.