Robust Feature Extraction based on an Asymmetric Level-Dependent Auditory Filterbank and a Subband Spectrum Enhancement Technique

Md Jahangir Alam\textsuperscript{1}, Patrick Kenny\textsuperscript{2}, Douglas O’Shaughnessy\textsuperscript{3}

\textsuperscript{1,3}INRS-EMT, University of Quebec, Montreal, Quebec, Canada
\textsuperscript{1,2}CRIM, Montreal, Quebec, Canada
alam@emt.inrs.ca, Patrick.kenny@crim.ca, dougo@emt.inrs.ca

Abstract

In this paper we introduce a robust feature extractor, dubbed as robust compressive gammachirp filterbank cepstral coefficients (RCGCC), based on an asymmetric and level-dependent compressive gammachirp filterbank and a sigmoid shape weighting rule for the enhancement of speech spectra in the auditory domain. The goal of this work is to improve the robustness of speech recognition systems in additive noise and real-time reverberant environments. As a post processing scheme we employ a short-time feature normalization technique called short-time cepstral mean and scale normalization (STCMSN), which, by adjusting the scale and mean of cepstral features, reduces the difference of cepstra between the training and test environments. For performance evaluation, in the context of speech recognition, of the proposed feature extractor we use the standard noisy AURORA-2 connected digit corpus, the meeting recorder digits (MRDs) subset of the AURORA-5 corpus, and the AURORA-4 LVCSR corpus, which represent additive noise, reverberant acoustic conditions and additive noise as well as different microphone channel conditions, respectively. The ETSI advanced front-end (ETSI-AFE), the recently proposed power normalized cepstral coefficients (PNCC), conventional MFCC and PLP features are used for comparison purposes. Experimental speech recognition results demonstrate that the proposed method is robust against both additive and reverberant environments. The proposed method provides comparable results to that of the ETSI-AFE and PNCC on the AURORA-2 as well as AURORA-4 corpora and provides considerable improvements with respect to the other feature extractors on the AURORA-5 corpus.
1. Introduction

Speech intelligibility as well as the performance of speech recognition systems degrades in practical environments due to a variety of signal variabilities. Additive noise and reverberation are the important causes of signal variabilities. Additive noise from interfering noise sources and convolution noise arising from acoustic environments mainly cause a reduction of speech recognition performance.

The acoustic features most commonly used in speech recognition systems are Mel Frequency Cepstral Coefficients (MFCC) [1] and Perceptual Linear Prediction (PLP) [2] features. Both MFCC and PLP front-ends perform well in matched environments, where speech data are collected from reasonably clean environments. However, their performance degrades severely when the testing environment is different from the training environment.Degradation of performance due to mismatched environments has been a barrier for deployment of speech recognition technologies. Various sources give rise to this mismatch, such as background noise, channel/handset distortion, room reverberation. Most of the sources of speech variability produce additive distortion (e.g., background noise) and/or convolutional distortion (e.g., channel/handset mismatch) in the speech signal. Among all different additive noises, a multi-speaker or babble noise environment, where the interference is speech from speakers in the vicinity, is one of the most challenging noise conditions. Therefore, it is necessary to address this problem, i.e., performance degradation due to mismatch environments, to enable the deployment of recognition systems in real world conditions.

Various research has been reported in the literature to improve the robustness of speech recognition systems under additive noise and reverberation. The methods to compensate for the effects of environmental mismatch can be implemented at the front-end (feature domain techniques) or at the back-end (model domain techniques) or both. The model domain methods adapt each acoustic model to make it fit better to the mismatched acoustic environment so that the adapted models will be able to classify the mismatched speech features collected in the testing environment. The typical examples of this
category include the well-known noise masking [20], speech and noise decomposition (SND) [21], vector tailor series (VTS) [22], maximum a posteriori (MAP) [23-24], maximum likelihood linear regression (MLLR) [25], model-based stochastic matching [26], statistical reestimation (STAR) [27-29], parallel model combination (PMC) [30], optimal likelihood weighting based on the criteria of minimum classification error (MCE) and maximum mutual information (MMI) [31], etc.

The objective of feature domain techniques is to make features more consistent in diverse environmental conditions. Feature domain methods can be classified into two subgroups. One subgroup of feature domain techniques aims at modifying the test features and making those features match the acoustic conditions better for the trained models: speech enhancement methods [32-37], codeword dependent cepstral normalization (CDCN) [38], feature-based stochastic mapping [26], multivariate Gaussian based cepstral normalization (RATZ) [39], feature normalization methods, such as cepstral mean normalization (CMN) [7, 11, 40], the stereo-based piecewise linear compensation for environment (SPLICE) [41-42]. The other subgroup of feature domain techniques, on the other hand, aims at making a special robust speech feature representation, which is used for both training and testing, to reduce the sensitivity to the various acoustic conditions.

This paper deals with the latter subgroup. Robust feature extractors are usually obtained either by appending a pre-processing step, like speech enhancement [8, 9, 14, 15], or by incorporating algorithms in an MFCC or PLP computation framework such as PNCC [3], amplitude modulation-based cepstral features [43-44], frequency masking [9], or by adding a post-processing step, like feature normalization techniques [7, 11], (e.g., cepstral mean normalization (CMN)) or by combining any two or all of the above mentioned steps [3, 4]. Most of the front-ends use, in addition to other techniques for environmental mismatch compensation, a feature normalization technique, at the least CMN, as a post-processing scheme.

Additive noise reduction approaches usually have a tradeoff between the amount of noise reduction and speech distortion induced due to processing of a speech signal. At very low SNR the intensity of this induced distortion is high, thereby deteriorating the performance of the speech recognition systems. Compensation of reverberant noise is usually done by dereverberation, which can be obtained by inverse filtering the impulse response of the
room [12, 13]. However, room impulse response is dependent on the distance between the speaker and the microphone and on the conditions of the room. Therefore, extracting a common set of robust features, which can perform well at low SNRs and also can handle various room impulse responses, is a difficult and challenging task.

To deal with additive noise distortion various speech enhancement methods have been proposed in [8, 9, 14, 15]. In [10, 12, 13] several approaches have been proposed for handling convolutional noise distortions. The ETSI-AFE, described in [4], uses a two-stage Wiener filter and a blind equalization technique, which is based on the comparison to a flat spectrum and the application of the LMS algorithm, for improving robustness of ASR systems against additive noise distortions and channel effects. The PNCC technique, proposed in [3], includes the use of a gammatone filter-bank (GTFB) and a power law nonlinearity in place of the Mel filter-bank and log nonlinearity, used in conventional MFCCs framework, a medium duration power bias subtraction technique, for noise reduction, based on the arithmetic mean (AM)-geometric mean (GM) ratio and cepstral mean normalization as a post-processing scheme for DC offset removal, for robust feature extraction.

In this work, for robust features extraction, we propose to enhance the speech auditory spectrum using a weighting rule based on the subband a posteriori signal-to-noise ratio (SNR). In order to allow a realistic and controllable frequency-domain asymmetry and to model most of the level dependency observed in basilar membrane (BM) filtering, the proposed method includes the use of a compressive gammachirp filter-bank (cGCFB) [16] for auditory spectral analysis. We use a power function nonlinearity as it has been found in [3] that it is more robust than the logarithmic nonlinearity used in a conventional MFCC framework. As a post-processing scheme for the normalization of the features, we use the short-time cepstral mean and scale normalization (STCMSN) technique, proposed in [7]. Feature normalization is normally performed over the whole utterance with the assumption that the channel effect is constant over the entire utterance, such as CMN (or CMVN). Also, normalizing a feature vector over the entire utterance is not a feasible solution in real-time applications as it causes an unnecessarily long processing delay. To relax this assumption and to reduce the processing delay, cepstral features in the proposed method are normalized over a sliding window of 1.5s duration. In this paper, we denote
this proposed feature extractor as the Robust Compressive Gammachirp filterbank Cepstral Coefficients (RCGCC). Replacing cGCFB with the GTFB in the proposed RCGCC feature extraction framework, we also present Robust Gammatone Filterbank Cepstral Coefficient (RGFCC) features to show the effectiveness of using cGCFB for the auditory spectral analysis in the proposed front-end. Experimental recognition results on the AURORA-2, AURORA-4, and AURORA-5 corpora demonstrate that the proposed RCGCC feature extractor outperforms the MFCC and PLP front-ends and provides comparable (and sometimes better) results to most state-of-the-art front-ends used in this work.

2. Proposed Feature Extractor

The complete block diagram of the proposed robust compressive gammachirp filterbank cepstral coefficient (RCGCC) feature extractor for robust speech recognition is shown in fig. 1. In the RCGCC feature extractor, processing of a speech signal begins with pre-processing (including DC removal and pre-emphasis, typically using a first-order high-pass filter). Short-time Fourier Transform (STFT) analysis is performed using a finite duration (25 ms) Hamming window with a frame shift of 10 ms to estimate the power spectrum of the signal. Compressive gammachirp filter-bank (cGCFB) integration is performed on both speech and noise power spectra for auditory spectral analysis. A sigmoid-shaped weighting rule is applied to enhance the auditory spectrum. The 13-dimensional static features, obtained after applying a power function nonlinearity with a coefficient of 1/15 and the discrete cosine transform (DCT), are normalized using the short-time cepstral mean and scale normalization (STCMSN) technique. A detailed description of various stages of the proposed method is given below.

2.1 Short-time Spectrum Estimation

A Hamming-windowed direct spectrum estimator (i.e., the squared magnitude of the Fourier transform of the signal) is the most often used power spectrum estimation method for speech processing applications. For the $m$-th frame (frame length is 25 msec with a
frame shift of 10 msec) and kth frequency bin an estimate of the windowed periodogram can be expressed as:

\[
\hat{X}(m,k_i) = \left| \sum_{j=0}^{N-1} w(j) x(m, j) e^{-i(2\pi)j k_i/N} \right|^2,\tag{1}
\]

where \( k_i \in \{0,1,\ldots,K-1\} \) denotes the frequency bin index, \( i \) is the imaginary unit, \( N \) is the frame length, \( x(m,j) \) is the time domain speech signal and \( w(j) \) denotes the time domain window function called a taper, which usually is symmetric and decreases towards the frame boundaries (e.g., Hamming). Direct spectrum estimators, such as a Hamming-windowed periodogram, are attractive as these estimators are completely independent of data and therefore do not suffer from problems arising due to modeling deficiencies. However, these methods are not robust to noise environments.

### 2.2 Estimation of Noise Power Spectrum

The estimation of a noise power spectrum from the noisy speech signal plays a very important rule in noise reduction/speech enhancement algorithms. For relatively stationary noises, an accurate estimation of the noise spectrum can be done using minimum statistics (MS)-based approaches [49-50]. These algorithms lead to less satisfying results for rapidly changing noises. In this paper, for accurate estimation of noise power spectra, we employ a minimum mean square error (MMSE) - soft speech presence probability (SPP) (MMSE-SPP)-based noise estimation approach, proposed in [17]. In this method, the initial estimate of the noise power spectrum is computed by averaging the first ten frames of the speech spectrum. The advantage of this method is that it does not require a bias correction term as required by a MMSE-based noise spectrum estimation method; it also results in less overestimation of noise power and is computationally less expensive [51]. The MMSE-SPP based noise estimation procedure is summarized in algorithm 1. The reason behind choosing MMSE-SPP - based noise spectrum estimation method is that it is computationally simple and requires only one parameter to be tuned.
Algorithm 1 [17]

1: for all signal frames $m$ do

2: Compute a posteriori SPP (signal presence probability) $P(H_1|S(m))$ as

$$P(H_1|S(m)) = \frac{1}{1 + (1 + \xi_{opt}) e^{-\frac{\|S(m)\|^2}{\hat{N}^2_{ns}}}}$$

where $\hat{N}^2_{ns}$ is the noise spectrum estimate of the previous frame, $m$ is the frame index, $|S(m)|^2$ is the noisy speech spectrum, and $\xi_{opt} = 31.62$.

3: Compute a smoothed a posteriori SPP $\bar{P}(m)$, as

$$\bar{P}(m) = 0.9\bar{P}(m-1) + 0.1P(H_1|S(m)).$$

4: Avoid underestimation

$$P(H_1|S(m)) \left\{ \begin{array}{ll} \min\{0.99, P(H_1|S(m))\} & \bar{P}(m) > 0.99 \\ P(H_1|S(m)), & \text{otherwise.} \end{array} \right.$$ 

5: Update the noise power spectrum estimate as

$$\left[\hat{N}^2_{ns}(m)\right]^2 = P(H_0|S(m))|S(m)|^2 + P(H_1|S(m))\hat{N}^2_{ns}(m-1).$$

6: Obtain the final noise power spectral estimate by temporal smoothing as

$$\hat{N}^2_{ns}(m) = 0.8\hat{N}^2_{ns}(m-1) + 0.2\left[\hat{N}_u(m)\right]^2.$$ 

7: end for

2.3 Computation of Auditory Filterbank Gain

Dominant speech analysis techniques for speech recognition, such as MFCC [1] and PLP [2], try to emulate human auditory perception. The widely used MFCC front-end [1] was proposed more than two decades ago as a representation based on a crude approximation of the auditory response, with little rigorous justification for its implementation [45]. A more general approach for auditory spectral analysis is based on gammatone filterbanks, which involve dedicated design of mathematical forms of frequency response that match physiological experimental results. The gammatone filterbank (GTFB) consists of a series of non-uniform overlap bandpass filters, which model the frequency selectivity property of human hearing. Its name is due to the nature of its impulse response, which is a gamma envelope modulated by a tone carrier centered at $f_c$ Hz.
where \( a, b, n, t, \varphi \) and \( f_c \) are the amplitude normalization constant, bandwidth of the filter in Hz, order of the filter, time, phase shift and center frequency, respectively.

\[
g_r(t) = a t^{n-1} e^{-2b t} \cos(2\pi f_c t + \varphi) \quad \text{for} \quad t > 0, \tag{2}
\]

Figure 1: Block diagram showing various stages of the proposed robust compressive gammachirp filterbank cepstral coefficient (RCGCC) feature extractor.

Some attempts have been made to utilize GTFB in ASR for auditory spectral analysis of the speech signal, for instance [3, 43-44, 46-48], which show significant performance gains in various noise types and signal-to-noise ratio (SNR) levels. The compressive gammachirp filter bank (cGCFB), introduced by Irino et al. in [16], is a generalization of the GTFB. The frequency response of a gammatone filter is very nearly symmetric, which is not a good match to the auditory data, and it cannot model the level-dependent properties observed in basilar membrane (BM) filtering. The cGCFB allows a realistic
and controllable frequency-domain asymmetry and has also been shown to model most of the level dependency observed in the BM filtering [16].

![Diagram](image)

**Figure 2**: Block diagram showing various steps for the computation of the compressive gamma-chirp filter-bank gain from the gammatone filterbank using a high pass asymmetric function (HP-AF) and a low pass asymmetric function (LP-AF) with $b_1 = 1.81, c_1 = -2.96, b_2 = 2.17, c_2 = 2.20$.

In this work, we have adopted cGCFB for the auditory spectral analysis of the speech signal. The frequency response of the cGCFB is given by:

$$H_{cgb} = a_r |H_{gt}(f)| \cdot e^{\theta_1} \cdot e^{\theta_2},$$

(3)

where $H_{gt}(f)$ is the frequency response of the GTFB, $\theta_1 = \tan^{-1}\left(\frac{f - f_{r1}}{b_1 B(f_{r1})}\right)$, $\theta_2 = \tan^{-1}\left(\frac{f - f_{r2}}{b_2 B(f_{r2})}\right)$, $f_{r1}$ and $f_{r2}$ are the center frequencies of the low-pass asymmetric function (LP-AF) $e^{\theta_1}$ and high-pass asymmetric function (HP-AF) $e^{\theta_2}$, respectively. $a_r$ is an amplitude normalization factor, $f_{r2}$ is defined in terms of a frequency ratio $f_{ratio}$ as $f_{r2} = f_{ratio} \cdot f_{p1}$, with $f_{p1} = f_{r1} + \frac{c_1 b_1 B(f_{r1})}{n}$, the peak frequency of the passive gammachirp filter [16]. So, the frequency response of the cGCFB can be obtained from the GTFB by multiplying the frequency response $|H_{gt}(f)|$ of the GT filter.
with HP- and LP-AFs. The high-pass asymmetric function makes the pass-band of the cGCFB more symmetric at lower levels. Fig. 2 shows how cGCFB gains (or weights) are computed from the GTFB weights. The number of filters used in this work is 64 and $b_2 = 2.17, c_2 = 2.20, b_1 = 1.81, c_1 = -2.96$. Fig. 3 presents a comparison of shapes and weights of the Mel filterbank (MelFB), GTFB and compressive GTFB (cGCFB).

**Figure 3**: Comparison of the shapes of Mel-filterbank (MelFB), gammatone filterbank (GTFB) and compressive gammachirp filterbank (cGCFB).
2.4 Auditory Spectrum Enhancement (ASE)

In this work, we propose a sigmoid-shaped weighting rule \( H(k,m) \) to enhance the auditory spectrum \( S_{aw}(k,m) \). Proposed weighting rule is based on the subband \( a posteriori \) SNR (in dB) \( \gamma_{aw}(k,m) \) and is formulated as [18]:

\[
H(k,m) = \frac{1}{1 + e^{-\tau \vartheta(k,m)}},
\]

where \( k \) is the subband index, \( m \) is the frame index, \( \tau \) is a parameter that controls the lower limit of the weighting function and \( \vartheta(k,m) \) is the subband instantaneous SNR (in dB) defined as:

\[
\vartheta(k,m) = \gamma_{aw}(k,m) - 4.5,
\]

where \( \gamma_{aw}(k,m) = \max \left( \frac{S_{aw}(k,m)}{N_{aw}(k,m)} - 4.0 \right) \), \( N_{aw}(k,m) \) is the noise power spectrum mapped onto the auditory frequency axis. Here, \( \tau = 4.5 \) is chosen experimentally.

In order to remove the outliers from the weighting function, as given by eqn. (4), due to noise variability, we use a two-dimensional median filter. For smoothing the decision regions, a two-dimensional moving average filter is also applied [18]. In order to achieve optimal performance it is desirable to have a feature extractor that is well suited both for clean and adverse acoustic conditions.

While the auditory filtering (e.g., Mel-scale filterbank) approximates the nonlinear characteristics of the human auditory system in frequency, the natural logarithmic nonlinearity or power function nonlinearity deals with the loudness nonlinearity. It approximates the relationship between a human’s perception of loudness and the sound intensity [62]. Power function nonlinearity or root compression (which can be expressed as \( y_i = y^a \), \( 0 < a < 1 \)) followed by DCT leads to better compaction of energy and hence more robust to mismatch condition [3, 63-64]. In this paper, similar to [3], we use a power function nonlinearity with a coefficient of \( a = 1/15 \) to approximate the loudness nonlinearity of human perception. The static features, obtained after applying the discrete
cosine transform (DCT), are normalized using the short-time cepstral mean and scale (STCMSN) approach [7].

2.5 Feature Normalization (FN)

In order to compensate for the effects of environmental mismatch feature normalization strategies are employed in speech (and speaker) recognition systems. These techniques are preferred because a priori knowledge and adaptation are not required under any environment. Most of the normalization techniques are applied as a post-processing scheme on the cepstral coefficient features. Normalization techniques can be classified as model-based or data distribution-based techniques. In model-based normalization techniques, certain statistical properties of speech such as mean, variance, and moments, are normalized to reduce the residual mismatch in feature vectors, e.g., short-time mean and variance normalization (STMVN). Data distribution-based techniques aim at normalizing the feature distribution to the reference, such as short-time Gaussianization (STG) [54, 55]. Some feature normalization techniques have been proposed in the past for speech and speaker recognition systems, including feature warping [54], STG [55], cepstral mean normalization (CMN) [56, 57], cepstral variance normalization (CVN) [58], histogram equalization [59] and RASTA filtering [60, 61]. Almost all the feature extractors include feature normalization technique as a post-processing scheme. Feature normalization is normally performed over the whole utterance with the assumption that the channel effect is constant over the entire utterance, such as CMN, mean and variance normalization (MVN). Also, normalizing a feature vector over the entire utterance is not a feasible solution in real-time applications as it causes unnecessarily long processing delay. To relax this assumption and to reduce the processing delay MFCC features are normalized over a sliding window of more than 1s duration. The feature vector to be normalized is located at the centre of the sliding window.

In the proposed feature extractor the static features, are normalized using the short-time cepstral mean and scale normalization (STCMSN) approach [7] over a sliding window of 1.5 seconds duration (i.e., 150 frames for a frame shift of 10 milliseconds). In STCMSN approach the $n$-th feature space and $m$-th frame $c(n,m)$ are normalized as
where $\mu_{st}(n,m)$ and $d_{st}(n,m)$ are the short-time mean and short-time difference between the upper and lower bound, respectively. Short-time mean $\mu_{st}(n,m)$ of the $n$-th feature space is defined for a short-time window of $L = 150$ frames as:

$$
\mu_{st}(n,m) = \frac{1}{L} \sum_{j=m-L/2}^{m+L/2} c(n,j)
$$

(7)

Short-time difference between the upper and lower bound $d_{st}(n,m)$ of the $n$-th feature space is defined for a short-time window of $L = 150$ frames as:

$$
d_{st}(n,m) = \max_{(m-L/2) \leq j \leq (m+L/2)} (c(n,j)) - \min_{(m-L/2) \leq j \leq (m+L/2)} (c(n,j)).
$$

(8)

The main idea behind STCMSN technique is that under mismatched conditions a difference of log spectrum between the training and test environments is removed by adjusting the short-time mean and short-time scale. The advantage of short-time feature normalization technique is that it does not use constant channel assumption as used in the full utterance-based feature normalization method, and reduces the unnecessary long processing delay [7]. Fig. 4 presents the zero-th cepstral coefficient feature ($c_0$) of a clean speech signal and that of a noisy speech signal, having a signal-to-noise ratio (SNR) of 5 dB, (a) before applying feature normalization and (b) after normalizing by the STCMSN method. It is observed from fig. 4 (a) that the effects of additive noise on a clean speech signal are:

- The minimum values of the cepstral features are elevated.
- The valleys of the cepstra are affected by the additive noise energy while the peaks remain almost unaffected.

The larger difference on valleys leads to a mismatch between the clean and noisy speech. It is clear from fig. 4 (b) that the STCMSN method is able to reduce the mismatch between the clean and noisy speech features.
Figure 4: The zero-th cepstral coefficients (c0) of clean and noisy speech (street noise, SNR = 5 dB) when (a) no normalization technique is applied, and (b) normalized by the STCMSN technique.

3. Performance Evaluation

The AURORA-2 [5], AURORA-4 [52] and AURORA-5 [6] corpora are used for the performance evaluation of the proposed RCGCC (robust compressive gammachirp filterbank cepstral coefficients) feature extractor and for comparing its performances to the conventional MFCC, PLP (HTK version of PLP), ETSI-AFE [4], and PNCC [3] features, in the context of speech recognition. The robust gammatone filterbank cepstral coefficient (RGFCC) feature extractor is also presented and compared with the proposed RCGCC front-end. For performance evaluation we have considered both matched (training and test environments are the same) and mismatched (test environment is
different from the training environment) environments. A brief description of the corpora, mentioned above, used for the performance evaluation are given in the following section.

3.1 Speech corpora

**AURORA-2**: The AURORA-2 database, widely used for noise robust front-end evaluation, consists of connected digit (English) task spoken by American English talkers [5]. There are two training sets (clean training set and multi-condition training set) and three test sets (test sets A, B and C). The clean training set, constitutes mismatched training/testing conditions [7], used in this work consists of 8440 clean speech recordings only from 55 male and 55 female adult speakers. The noisy training and test utterances are obtained by artificially corrupting the clean digit utterances with 8 different additive noises (subway, babble, car and exhibition hall, restaurant, street, airport and train-station) and 2 convolutional noises (MIRS (modified intermediate reference system) filtered subway and street noise) at SNRs of 20, 15, 10, 5, 0, and -5 dB. The number of test utterances for each noise type and SNR level (clean, 20, 15, 10, 5, 0, and -5 dB) is 1001. Test sets A, B and C are composed of 28028, 28028, and 14014, speech recordings, respectively. The speech recordings in the three test sets are uttered by 52 female and 52 male speakers.

**AURORA-5**: The AURORA-5 corpus comprises real time recording in a meeting room, recorded in a hands-free mode at the ICSI (International Computer Science Institute) in Berkeley [6]. The database consists of 2388 utterances from 24 speakers with 7800 digits in total. The speech was captured with four different microphones, labeled as mic 6, mic 7, mic E and mic F, placed at the middle of the table in the meeting room. The recordings contain only a small amount of additive noise, providing the typical effect of hands-free recording in a reverberant room.

**AURORA-4**: The AURORA-4 continuous speech recognition corpus, derived from the Wall Street Journal (WSJ0) corpus, is divided into 3 sets, namely, training, development (dev test) and evaluation (eval or test) sets. This task is often referred to as the 5k closed vocabulary task, i.e., there are no out-of-vocabulary words (OOVs) in the evaluation set. The training set contains 7138 utterances from 83 speakers, totaling 14 hours of speech data. 14 evaluation sets were defined in order to study the degradations in speech
recognition performance due to microphone conditions, filtering and noisy environments. Each of the filtered versions of the evaluation set recorded with a Sennheiser microphone and secondary microphone was selected to form the two evaluation sets. The remaining 12 subsets were defined by randomly adding each of the 6 noise types (car, babble, restaurant, street traffic, airport, and train-station noises) at a randomly chosen SNR between 5 and 15 dB for each of the microphone types as mentioned above. The goal was to have an equal distribution of each of the 6 noise types and the SNR with an average SNR of 10 dB [52]. Each of the test sets contains 166 utterances from 8 speakers, totaling 20.69 minutes of speech data. The 14 test sets are grouped into the following 4 families A, B, C, and D [52, 53] as described in table 1. The number inside the brackets represents the test set number defined in the AURORA-4 corpus [52]. More details about the AURORA-4 database (AURORA-2 as well as AURORA-5) can also be found in [19].

Table 1: Evaluation conditions for the AURORA-4 LVCSR task.

<table>
<thead>
<tr>
<th>Test set</th>
<th>description</th>
<th>sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clean data in training and test (same channel)</td>
<td>set 1</td>
</tr>
<tr>
<td>B</td>
<td>Clean data in training and noisy data in test (same channel)</td>
<td>sets 2-7</td>
</tr>
<tr>
<td>C</td>
<td>Clean data in training and test (different channel)</td>
<td>set 8</td>
</tr>
<tr>
<td>D</td>
<td>Clean data in training and noisy data in test (different channel)</td>
<td>sets 9-14</td>
</tr>
</tbody>
</table>

Additive noises added in test sets 2-7 & 9-14 are: car, babble, restaurant, street, airport, and train-station.

3.2 Experimental setup

For the performance evaluation of our proposed feature extractor, we have chosen matched and mismatched conditions. Features extracted from the clean training data are used for training the recognizer and decoding is performed on both matched and mismatched (due to additive noise, channel/handset distortion and reverberation) environments. For our experiments, we use 13 static features (including the 0th cepstral coefficient) augmented with their delta and double delta coefficients, making 39-dimensional feature vectors. The analysis frame length is 25 ms with a frame shift of 10 ms. The Delta and double features were calculated using a 7-frame (lag size 3-frame) and 5-frame (lag size 2-frame) window, respectively. For all the feature extractors, the
features, before appending delta and double delta features, are normalized using the feature normalization technique: MVN is used for MFCC, and PLP features, CMN is used for PNCC, ETSI-AFE uses a blind equalization technique (for PNCC and ETSI-AFE exactly the same normalization methods are used as mentioned in the respective references, i.e., in [3] & [4]) and in the proposed RCGCC method STCMSN is used to normalize the feature vectors. In order to provide a comparison of performances, in terms of speech recognition accuracy, of using logarithmic nonlinearity versus power-law nonlinearity and MVN versus STCMSN, we considered the following versions of MFCC front-ends:

**MFCC**: Conventional Mel-frequency cepstral coefficient (MFCC) features that utilize logarithmic nonlinearity for compression and the MVN method for feature normalization.

**MFCC_I**: Mel-frequency cepstral coefficient (MFCC) features that utilize a power-law nonlinearity for compression and the MVN method for feature normalization.

**MFCC_II**: Mel-frequency cepstral coefficients (MFCC) features that utilize a power-law nonlinearity for compression for compression and the STCMSN method for feature normalization.

To show the reason behind choosing a compressive gammachirp filterbank (cGCFB) instead of a gammatone filterbank, we also compared the performance of the RCGCC feature extractor to that of the RGFCC (robust gammatone filterbank cepstral coefficients). The only difference between the RCGCC and RGFCC front-ends is in the auditory filterbank integration. In the RGFCC, a gammatone filterbank is used instead of cGCFB.

For the recognition task we use the HTK speech recognizer. For the connected digit recognition task on the AURORA2 and AURORA-5 corpora the backend configuration in the standard AURORA evaluation framework [5] was used: 16-state whole-word HHM models for the digits, 3-state and 1-state HMM models for silence (sil) and short pause (sp), respectively. Each whole-word HMM model was associated with a 3-component GMM (Gaussian mixture model) for the state emitting probability density. 6-component GMM is used for the sil/sp states.
For the continuous speech recognition task on the AURORA-4 corpus, all experiments employed state-tied crossword speaker-independent triphone acoustic models with four Gaussian mixtures per state. A single-pass Viterbi beam search-based decoder was used along with a standard 5K lexicon and bigram language model with a prune width of 250 [52, 53].

For the performance evaluation of the proposed feature extractor method and comparing the recognition performances of the feature extractors considered in this paper, the percentage word accuracy (%WAcc) is used. WAcc is a more representative figure of the recognizer performance. WAcc is defined as

\[
WAcc = \frac{N - D - S - I}{N} \times 100\% ,
\]

where \(N\) is total number of words in the test database, \(D\) is the number of deletions (i.e., words of the original sentence missing in the recognized sentence), \(S\) is the number of substitutions (i.e., words of the original sentence each appearing substituted by a different word in the recognized sentence), and \(I\) represents the number of insertions (i.e., the recognized sentence includes a new word between two words of the original sentence).

The standard way to evaluate ASR performance of a new system (\(WAcc_{new}\)) is by measuring relative improvement (RI) with the baseline system (\(WAcc_{baseline}\)). RI can be defined

\[
RI = \frac{WAcc_{new} - WAcc_{baseline}}{100 - WAcc_{baseline}} \times 100 .
\]

A positive RI indicates how much performance is improved by the new system than the baseline and a negative value indicates how much degradation of performance is provided by the new system over the baseline.

### 3.3 Results & discussion

In order to achieve optimal performance it is desirable to have a feature extractor that is well suited both for clean and adverse acoustic conditions. Figures 5, 6, and 7 present the auditory spectrum of the clean speech signal (fig. 4) and noisy speech signals (figs. 5-7) obtained (a) without and (b) with applying the sigmoid-shaped weighting rule of the
proposed method. It is observed from figs. 5-7 that the auditory spectrum enhancement (ASE) method of the proposed feature extractor did not introduce any distortion to the clean auditory spectrum and enhanced the noisy auditory spectrum. The effectiveness of using a power-law nonlinearity and short-time cepstral mean and scale normalization (STCMSN) for noise robust performance is demonstrated in table 2, in the context of speech recognition experiments on the AURORA-2 & -5 corpora, by incorporating them in the MFCC feature extraction framework. It is observed from table 2 that use of the power-law nonlinearity (MFCC_I) provided an average gain of 0.77% and 4.5% in word recognition accuracy on the AURORA-2 and AURORA-5 corpora, respectively, over a logarithmic compression (e.g., MFCC). Normalizing the cepstral features by the STCMSN method (e.g., MFCC_II) instead of the MVN method (e.g., MFCC_I) yielded an additional average gain of 3.1% and 2.8% in recognition accuracy on the AURORA-2 and AURORA-5 corpora, respectively.

The most important modules of the proposed RCGCC feature extractor are: compressive gammachirp filterbank, noise spectrum estimation - subband a posteriori SNR-based auditory spectrum enhancement, power-function nonlinearity and STCMSN. In order to show how much each of these modules contributes to the noise robust performance, we compared the performance of the RCGCC and RGFCC and RCGCC_I (here, for feature normalization CMN is utilized instead of the STCMSN) front-ends on the AURORA-4 LVCSR task. Comparing the performances of the RCGCC and RCGCC_I, from the presented results in table 3, it is evident that the contribution (in terms of recognition accuracy) of STCMSN as the feature normalization module in the proposed feature extractor is about 3.8% on the average. In the LVCSR task on the AURORA-4 corpus (tables 3 and 6), there is not much difference in performance between the RCGCC and RGFCC, i.e., the contribution of the compressive gammachirp filterbank is almost the same as that of the gammatone filterbank. In the case of a small-vocabulary connected-digits task on the AURORA-2 and -5 corpora (see tables 4 and 5), the proposed RCGCC method performed better than the RGFCC in terms of word recognition accuracy. Since our previous work [18] was based on the small-vocabulary connected-digits task and we found that the compressive gammachirp filterbank performed better than the gammatone
filterbank, we chose the compressive gammachirp filterbank for auditory spectral analysis in the proposed front-end.

Table 4 presents the average word accuracy (in %), averaged over the 10 noise scenarios, mentioned in section 3.1, of all the three test sets A, B, and C of the AURORA-2 connected digit corpus. In additive noise conditions on the TIdigits task, the proposed RCGCC feature provided comparable results to that of the PNCC and ETSI-AFE features and outperformed the MFCC and PLP features. Average relative improvements ($RI$ in %) obtained by the proposed feature are presented in the last column of table 4.

In order to demonstrate the performance of the proposed feature extractor in real-time reverberant environments (i.e., not simulated reverberant environments), experiments were conducted on the meeting recorder digits (MRDs) subset of the AURORA-5 corpus. Table 5 presents a comparison of the performances in terms of word accuracy (in %) of the proposed feature extractor with the other considered feature extractors, including the RGFCC, in a real-time reverberant environment on the AURORA-5 corpus. In convolutive noise conditions, the proposed method provided consistently better word accuracy on all microphones of the AURORA-5 corpus than all other methods. The average relative improvements obtained by the proposed method over the MFCC, PLP, PNCC, and ETSI-AFE front-ends are 35.6%, 41.0%, 11.3%, and 66.1%, respectively. From table 5 it is evident that the ETSI-AFE front-end has the lowest word accuracy compared to all other feature extractors. In a reverberant environment the ETSI-AFE front-end is not as effective as its performance in additive background noise distortions, which is consistent with the studies of [10]. The proposed RCGCC method performs well both in additive noise distortions and reverberation on the TIdigits task. The recognition accuracies (%WAcc) obtained by all the feature extractors used in our experiments on the AURORA-4 LVCSR task are presented in table 6. The average relative improvements achieved by the proposed method are 33.8% and 30.6% over the MFCC and PLP features, respectively. The proposed feature extractor provided comparable recognition accuracy to the PNCC and ETSI-AFE front-ends on the LVCSR task. In matched conditions (training the recognizer using clean training features and decoding on the clean test features) of all the three corpora, the proposed method performed well compared to the other two robust front-ends.
Figure 5: Auditory spectra of a clean speech signal, (a) before applying the weighting rule, (b) after applying the weighting rule.

Figure 6: Auditory spectra of noisy speech signal degraded with train-station noise with SNR = 5 dB, (a) before applying the weighting rule, (b) after applying the weighting rule.
Figure 7: Auditory spectra of noisy speech signal degraded with babble noise with SNR = 5 dB, (a) before applying the weighting rule, (b) after applying the weighting rule.

Under mismatched conditions RCGCC provided comparable results to the PNCC and ETSI-AFE front-ends. Performances of all the front-ends, including the RCGCC and RGFCC front-ends, are also presented in tables 7 and 8 under various noise scenarios (car, babble, restaurant, street, airport, and train-station) on the test sets B and D, respectively, of the AURORA-4 corpus. Computationally, measured in terms of the time taken to extract features from all AURORA-4 data, the proposed RCGCC method was found less expensive compared to the PNCC and ETSI-AFE. In order to extract features from 11758 speech recordings (AURORA-4 corpus, average duration of utterances = 7.0 seconds) on an AMD machine (clock speed 2300 MHz, 12 core, 128 GB RAM) MFCC, RCGCC, PNCC, and ETSI-AFE (C implementation) front-ends took 867, 1694, 24842 and 2950 seconds, respectively. All the feature extractors except ETSI-AFE were implemented in MATLAB. The execution time mentioned above for each method is the average of 4 execution times.
Table 2: Word accuracy (%) obtained using different versions of MFCC front-ends on the (a) AURORA-2 and (b) AURORA-5 corpora. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface.

(a)

<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>20dB</th>
<th>15dB</th>
<th>10dB</th>
<th>5dB</th>
<th>0dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>99.03</td>
<td>96.76</td>
<td>93.31</td>
<td>85.22</td>
<td>66.10</td>
<td>33.01</td>
<td>78.91</td>
</tr>
<tr>
<td>MFCC_I</td>
<td>99.06</td>
<td>97.54</td>
<td>94.91</td>
<td>87.84</td>
<td>66.72</td>
<td>32.02</td>
<td>79.68</td>
</tr>
<tr>
<td>MFCC_II</td>
<td>99.09</td>
<td>97.65</td>
<td>95.86</td>
<td>90.23</td>
<td>73.78</td>
<td>40.08</td>
<td>82.77</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th></th>
<th>Mic 6</th>
<th>Mic 7</th>
<th>Mic E</th>
<th>Mic F</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>81.41</td>
<td>76.12</td>
<td>75.22</td>
<td>79.33</td>
<td>78.02</td>
</tr>
<tr>
<td>MFCC_I</td>
<td>86.39</td>
<td>80.67</td>
<td>79.52</td>
<td>83.65</td>
<td>82.56</td>
</tr>
<tr>
<td>MFCC_II</td>
<td>89.08</td>
<td>84.49</td>
<td>82.28</td>
<td>85.82</td>
<td>85.42</td>
</tr>
</tbody>
</table>

Table 3: Word accuracy (%) obtained on the AURORA-4 LVCSR task by the RGFCC, proposed feature extractor when short-time cepstral mean and scale normalization (RCGCC) and cepstral mean normalization (RCGCC_I) method is used as post-processing scheme to normalize the features. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCGCC_I</td>
<td>88.91</td>
<td>62.96</td>
<td>78.97</td>
<td>51.93</td>
<td>70.69</td>
</tr>
<tr>
<td>RCGCC</td>
<td>88.90</td>
<td>68.87</td>
<td>80.94</td>
<td>59.25</td>
<td>74.49</td>
</tr>
<tr>
<td>RGFCC</td>
<td>88.94</td>
<td>68.47</td>
<td>80.70</td>
<td>59.39</td>
<td>74.37</td>
</tr>
</tbody>
</table>
Table 4: Word accuracy (%) obtained using different front-ends on the AURORA-2 corpus. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface. Last column provides the relative improvements obtained by the proposed RCGCC over the MFCC, PLP, PNCC and ETSI-AFE methods.

<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>20dB</th>
<th>15dB</th>
<th>10dB</th>
<th>5dB</th>
<th>0dB</th>
<th>Avg.</th>
<th>RI (by the RCGCC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>99.03</td>
<td>96.76</td>
<td>93.31</td>
<td>85.22</td>
<td>66.10</td>
<td>33.01</td>
<td>78.91</td>
<td>40.8</td>
</tr>
<tr>
<td>PLP</td>
<td>99.07</td>
<td>96.68</td>
<td>93.41</td>
<td>85.57</td>
<td>67.48</td>
<td>36.32</td>
<td>79.76</td>
<td>38.3</td>
</tr>
<tr>
<td>PNCC</td>
<td>99.20</td>
<td>98.48</td>
<td>97.39</td>
<td>93.34</td>
<td>82.26</td>
<td>54.33</td>
<td>87.50</td>
<td>0.16</td>
</tr>
<tr>
<td>ETSI-AFE</td>
<td>99.23</td>
<td>98.06</td>
<td>96.56</td>
<td>93.02</td>
<td>83.32</td>
<td>60.10</td>
<td>88.38</td>
<td>-7.4</td>
</tr>
<tr>
<td>RCGCC</td>
<td>99.34</td>
<td>98.39</td>
<td>97.39</td>
<td>93.36</td>
<td>82.67</td>
<td>53.97</td>
<td>87.52</td>
<td>-</td>
</tr>
<tr>
<td>RGFCC</td>
<td>99.37</td>
<td>98.31</td>
<td>96.93</td>
<td>92.88</td>
<td>80.00</td>
<td>47.55</td>
<td>85.84</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5: Word accuracy (%) obtained using different front-ends on the AURORA-5 corpus. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface. Last column provides the relative improvements obtained by the proposed RCGCC over the MFCC, PLP, PNCC and ETSI-AFE methods.

<table>
<thead>
<tr>
<th></th>
<th>Mic 6</th>
<th>Mic 7</th>
<th>Mic E</th>
<th>Mic F</th>
<th>Avg.</th>
<th>RI (by the RCGCC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>81.41</td>
<td>76.12</td>
<td>75.22</td>
<td>79.33</td>
<td>78.02</td>
<td>35.6</td>
</tr>
<tr>
<td>PLP</td>
<td>79.55</td>
<td>73.98</td>
<td>73.10</td>
<td>77.51</td>
<td>76.04</td>
<td>41.0</td>
</tr>
<tr>
<td>PNCC</td>
<td>88.12</td>
<td>83.40</td>
<td>80.20</td>
<td>84.51</td>
<td>84.06</td>
<td>11.3</td>
</tr>
<tr>
<td>ETSI-AFE</td>
<td>64.32</td>
<td>47.66</td>
<td>58.12</td>
<td>62.72</td>
<td>58.21</td>
<td>66.1</td>
</tr>
<tr>
<td>RCGCC</td>
<td>89.16</td>
<td>85.27</td>
<td>82.82</td>
<td>86.21</td>
<td>85.87</td>
<td>-</td>
</tr>
<tr>
<td>RGFCC</td>
<td>87.08</td>
<td>81.06</td>
<td>79.80</td>
<td>83.78</td>
<td>82.93</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 6: Word accuracy (%) obtained using different front-ends on the AURORA-4 LVCSR task. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface. Last column provides the relative improvements obtained by the proposed RCGCC over the MFCC, PLP, PNCC and ETSI-AFE methods.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg.</th>
<th>RI (by the RCGCC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>90.02</td>
<td>49.19</td>
<td>71.12</td>
<td>35.45</td>
<td>61.45</td>
<td>33.8</td>
</tr>
<tr>
<td>PLP</td>
<td>89.72</td>
<td>50.41</td>
<td>73.10</td>
<td>39.64</td>
<td>63.22</td>
<td>30.6</td>
</tr>
<tr>
<td>PNCC</td>
<td>88.64</td>
<td>69.85</td>
<td>81.07</td>
<td>60.00</td>
<td>74.89</td>
<td>-1.5</td>
</tr>
<tr>
<td>ETSI-AFE</td>
<td>88.59</td>
<td>69.58</td>
<td>79.52</td>
<td>61.51</td>
<td>74.80</td>
<td>-1.2</td>
</tr>
<tr>
<td>RCGCC</td>
<td>88.90</td>
<td>68.87</td>
<td>80.94</td>
<td>59.25</td>
<td>74.49</td>
<td>-</td>
</tr>
<tr>
<td>RGFCC</td>
<td>88.94</td>
<td>68.47</td>
<td>80.70</td>
<td>59.39</td>
<td>74.37</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7: Word accuracy (%) obtained using different front-ends on the test set B of the AURORA-4 LVCSR corpus under six additive noise scenarios. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface.

<table>
<thead>
<tr>
<th></th>
<th>Car (2)</th>
<th>Babble (3)</th>
<th>Restaurant (4)</th>
<th>Street (5)</th>
<th>Airport (6)</th>
<th>Train (7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>65.45</td>
<td>47.29</td>
<td>49.50</td>
<td>44.01</td>
<td>50.13</td>
<td>38.75</td>
</tr>
<tr>
<td>PLP</td>
<td>64.86</td>
<td>47.44</td>
<td>50.28</td>
<td>47.00</td>
<td>50.64</td>
<td>42.21</td>
</tr>
<tr>
<td>PNCC</td>
<td>80.96</td>
<td>68.29</td>
<td>66.48</td>
<td>67.22</td>
<td>67.88</td>
<td>68.25</td>
</tr>
<tr>
<td>ETSI-AFE</td>
<td>81.33</td>
<td>69.87</td>
<td>63.61</td>
<td>68.69</td>
<td>64.75</td>
<td>69.21</td>
</tr>
<tr>
<td>RCGCC</td>
<td>79.38</td>
<td>67.65</td>
<td>66.43</td>
<td>67.21</td>
<td>67.18</td>
<td>65.36</td>
</tr>
<tr>
<td>RGFCC</td>
<td>80.52</td>
<td>67.73</td>
<td>66.11</td>
<td>65.45</td>
<td>65.82</td>
<td>65.19</td>
</tr>
</tbody>
</table>
Table 8: Word accuracy (%) obtained using different front-ends on the test set D of the AURORA-4 LVCSR corpus under six additive noise scenarios. The higher the word accuracy the better is the performance of the front-end. For each column the best results are in boldface.

<table>
<thead>
<tr>
<th></th>
<th>Car (9)</th>
<th>Babble (10)</th>
<th>Restaurant (11)</th>
<th>Street (12)</th>
<th>Airport (13)</th>
<th>Train (14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>51.31</td>
<td>32.23</td>
<td>36.32</td>
<td>27.37</td>
<td>36.17</td>
<td>29.28</td>
</tr>
<tr>
<td>PLP</td>
<td>52.12</td>
<td>38.97</td>
<td>39.59</td>
<td>32.93</td>
<td>39.04</td>
<td>35.17</td>
</tr>
<tr>
<td>PNCC</td>
<td>70.61</td>
<td>59.96</td>
<td>55.40</td>
<td>56.94</td>
<td>58.49</td>
<td>58.60</td>
</tr>
<tr>
<td>ETSI-AFE</td>
<td>72.82</td>
<td>62.76</td>
<td>54.62</td>
<td>57.09</td>
<td>59.78</td>
<td>61.99</td>
</tr>
<tr>
<td>RCGCC</td>
<td>70.10</td>
<td>58.75</td>
<td>55.95</td>
<td>55.51</td>
<td>59.67</td>
<td>55.55</td>
</tr>
<tr>
<td>RGFCC</td>
<td>70.24</td>
<td>58.49</td>
<td>56.17</td>
<td>54.73</td>
<td>60.85</td>
<td>55.84</td>
</tr>
</tbody>
</table>

4. Conclusions

In this paper we presented a robust feature extractor, denoted in this paper as robust compressive gammachirp filterbank cepstral coefficients (RCGCC), that includes the use of an asymmetric level-dependent compressive gammachirp filterbank (cGCFB) for mapping the speech spectrum onto the auditory frequency axis, a simple auditory domain spectrum enhancement technique based a sigmoid type suppression rule, an MMSE - soft speech presence probability (SPP)-based noise spectrum estimator method [17], and a short-time cepstral mean and scale normalization (STCMSN) method [7], for speech recognition tasks. Speech recognition experiments are performed on the AURORA-2 and AURORA-5 TIdigits corpora and the AURORA-4 large vocabulary continuous speech recognition (LVCSR) corpus to cover matched as well as various mismatch conditions due to additive background noise, channel distortion and real time reverberation. It is evident from the reported results that the proposed method provided comparable results to that of the ETSI-AFE and PNCC front-ends under additive noise distortions (AURORA-2 and AURORA-4 tasks) and performed better under reverberant environments (on the AURORA-5 task). The proposed robust feature extractor performed
well both in matched and mismatched training/test environments and was also found computationally, measured in terms of execution time, less expensive

Acknowledgement

The authors would like to thank the reviewers for their valuable comments which has enabled us to significantly improve the quality of the paper.

References


[4] ETSI ES 202 050, Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; advanced front-end feature extraction algorithm; Compression algorithms; 2003.


[42] Li Deng, Alex Acero, L. Jiang, Jasha Droppo, and Xuedong Huang, "High-Performance Robust Speech Recognition Using Stereo Training Data," in Proc. ICASSP, Institute of Electrical and Electronics Engineers, Inc., Salt Lake City, Utah, May 2001.


