Acoustic Feature Compensation Based on Decomposition of Speech and Noise for ASR in Noisy Environments

Hong Kook Kim, Richard C. Rose, and Hong-Goo Kang

AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ 07932, USA
{hkkim, rose, goo}@research.att.com

Abstract

This paper presents a set of acoustic feature pre–processing techniques that are applied to improving automatic speech recognition (ASR) performance on the Aurora 2 noisy speech recognition task. The principal contribution of this paper is an approach for cepstrum domain feature compensation in ASR which is motivated by techniques for decomposing speech and noise that were originally developed for noisy speech enhancement. This approach is applied in combination with other feature compensation algorithms to compensating ASR features obtained from a mel–filterbank cepstrum coefficient (MFCC) front–end. Performance comparisons are made with respect to the application of the minimum mean squared error log spectral amplitude estimator (MMSE–LSA) based speech enhancement algorithm prior to feature analysis. An experimental study is presented where the feature compensation approaches described in the paper are found to reduce ASR word error rate by as much as 31% relative to uncompensated features under simulated environmental and channel mismatched conditions.

1. Introduction

All techniques for HMM model compensation and feature compensation in automatic speech recognition (ASR) are complicated by the fact that the interaction between speech and acoustic background noise in the cepstrum domain is highly non–linear [1]–[3]. Whereas environmental noise and speech are considered to be additive in the linear spectrum domain, their interaction is considered to be more difficult to characterize in the log spectral amplitude and the cepstrum domains. Consequently, the goal of decomposing a noise corrupted speech signal into clean speech and pure noise components has always been difficult to achieve.

This paper presents an approach for cepstrum domain feature compensation in ASR which exploits noisy speech decomposition techniques that were originally developed for speech enhancement. The approach relies on the combination of a minimum mean squared error log spectral amplitude estimator (MMSE–LSA) and an adaptive limiting procedure for estimating a set of frequency dependent spectral gain factors [4]–[6]. It will be shown that this approach can be applied in combination with other feature compensation techniques to improve speech recognition performance under the noise conditions that are simulated in the Aurora 2 noisy speech recognition task [7]. Furthermore, the proposed approach enables us to efficiently perform model compensation in the cepstrum domain because estimated noise and clean speech are considered to be additive in the cepstrum domain. While many simple model compensation techniques are equivalent to performing acoustic feature compensation, this paper will consider cepstrum decomposition from only a feature compensation point of view.

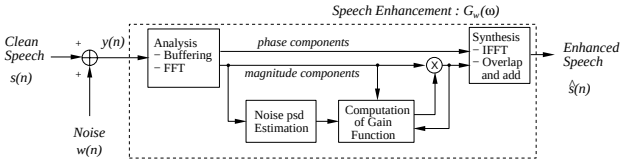

Figure 1: *Block diagram of the speech enhancement in a noise environment.*

Following this introduction, we briefly review the speech enhancement algorithm used in this work and address the speech enhancement based front–end in Section 2. In Section 3, we propose a cepstrum–domain acoustic feature compensation technique that decomposes noisy speech cepstrum into enhanced speech and noise cepstrum from the speech enhancement algorithm. In Section 4, we evaluate the performance of our proposed method on the task of the Aurora 2 database. Finally, we conclude our findings in Section 5.

2. Speech enhancement based front–end

The speech enhancement algorithm shown in Figure 1 operates in the frequency domain. A nonlinear *frequency dependent gain function* is applied to the spectral components of the noisy speech signal in an attempt to obtain estimates of spectral components of corresponding clean speech. A modified MMSE–LSA estimation criterion with a soft–decision based modification is used to derive the gain function [5]. We briefly describe the algorithm in this section to motivate its use as a cepstrum–domain feature compensation method.

Let $S(\omega) = A(\omega)e^{j\varphi(\omega)}$, $W(\omega)$, and $Y(\omega) = R(\omega)e^{j\vartheta(\omega)}$ be the Fourier expansions of clean speech $s(n)$, additive noise $w(n)$, and noisy speech $y(n)$, respectively. The objective of the MMSE–LSA is to find the estimator $\hat{A}(\omega)$ that minimizes the distortion measure $E\{(\log A(\omega) - \log \hat{A}(\omega))^2\}$ for a given noisy observation spectrum $Y(\omega)$. The modified MMSE–LSA gives an estimate of clean speech spectrum that has the form

$$\hat{A}(\omega) = G_M(\omega)G_{LSA}(\omega)R(\omega) = G_w(\omega)R(\omega) \qquad (1)$$

where $G_M(\omega)$ is the gain modification function and $G_{LSA}(\omega)$ is the gain function [5]. $G_M(\omega)$ represents the probablities of speech being present in frequency ω and is referred to as the soft–decision modification of the optimal estimator [5]. $G_{LSA}(\omega)$ is derived in [5] as

$$G_{LSA}(\omega) = \frac{\xi(\omega)}{1+\xi(\omega)}\exp\left(\frac{1}{2}\int_{v(\omega)}^{\infty}\frac{e^{-t}}{t}dt\right) \qquad (2)$$

where $v(\omega) = \frac{\xi(\omega)}{1+\xi(\omega)}\gamma(\omega)$, $\gamma(\omega) = \frac{R^2(\omega)}{\lambda_w(\omega)}$, $\xi(\omega) = \frac{\eta(\omega)}{1-q(\omega)}$, $\eta(\omega) = \frac{\lambda_s(\omega)}{\lambda_w(\omega)}$, $\lambda_s(\omega) = E\{|S(\omega)|^2\} = E\{A^2(\omega)\}$, and

$\lambda_w(\omega) = E\{|W(\omega)|^2\}$. $\gamma(\omega)$ is called the *a posteriori* signal–to-noise ratio (SNR), $\eta(\omega)$ is called the *a priori* SNR, and $q(\omega)$ is the prior probability that there is no speech presence in frequency ω. In addition, the $\lambda_s(\omega)$ and $\lambda_w(\omega)$ denote the power spectral densities (psd's) of speech and noise signals, respectively.

The estimation of the noise psd, $\lambda_w(\omega)$, is a very delicate component of the enhancement system of Equations (1) and (2), especially for non–stationary noise conditions. We use a spectral minimum tracking approach for estimating $\lambda_w(\omega)$ [8]. In contrast to voice activity detection oriented approaches, the minimum tracking method does not require explicit thresholds for identifying speech and noise–only intervals. The method determines the minimum of the short–time psd estimate within a finite window length and assumes that the bias compensated minimum[1] is the noise psd of the analysis frame. This approach works very well in real communication environments where the channel conditions are slowly varying with respect to the analysis frame length [9].

Equation (2) also shows that the amount of noise reduction is determined by how aggressively the *a priori* SNR is applied. The amount of noise reduction can be reduced by overestimating $\eta(\omega)$ and increased by underestimating $\eta(\omega)$. An aggressive scheme reduces the amount of noise; however, it may be harmful for ASR because it distorts the feature vectors in speech regions. There are no unique optimum parameter settings because these parameters are also dependent on the characteristics of input noise and the efficiency of the noise psd estimation. It has been found that a more aggressive scheme is optimal for car noise signals but a less aggressive scheme is optimal for clean and babble noise signals [6]. The settings we have chosen in this work are somewhat biased to car noise signals. Use of this speech enhancement algorithm as a preprocessor to feature extraction will be referred to in Section 4 as the *speech enhancement based front–end* (SE). The use of the spectral gain function as part of a cepstrum compensation method is described in Section 3.

3. Cepstrum–domain feature compensation

3.1. Cepstrum subtraction method (CSM)

The speech enhancement algorithm works by multiplying the frequency dependent gain function, $G_w(\omega)$, by the noisy magnitude spectrum as described in Equation (1). Figure 1 describes how the speech enhancement algorithm can be represented as a nonlinear filter whose frequency response is defined by $G_w(\omega)$. If the inverse Fourier transform of $G_w(\omega)$ is $g_w(n)$, then the enhanced signal, $\hat{s}(n)$, is given by

$$\hat{s}(n) = y(n) * g_w(n) = (s(n) + w(n)) * g_w(n) \qquad (3)$$

where $y(n)$ is noisy speech and $w(n)$ is additive noise. Assuming that the enhanced speech signal is an estimate of the clean speech signal, the cepstrum for clean speech, $c_{\hat{s}}$, is approximated as

$$c_{\hat{s}} = c_y + c_{g_w}. \qquad (4)$$

This equation implies that the noisy speech cepstrum, c_y, can be decomposed into a linear combination of the estimated clean speech cepstrum, $c_{\hat{s}}$, and noise cepstrum, c_{g_w}. We call this approach *cepstrum subtraction method* (CSM). Hence, by exploiting several well–behaved noise estimation algorithms that were described in Section 2, we are able to obtain a linear decomposition of speech and noise in the noisy speech cepstrum domain. We can construct a noise–robust front–end as shown in Figure 2. In the figure, the speech enhancement algorithm acts as a generator of a nonlinear filter from the noisy speech. After that, the noise mel–filterbank cepstrum coefficients (MFCC's) are obtained by applying an inverse discrete cosine transform (IDCT) to the transfer function of the nonlinear filter. The MFCC's corresponding to the noisy speech are obtained from a conventional filterbank analysis that is described in Section 4.1. Finally, we add the noisy speech cepstrum and the estimate of the noise cepstrum to obtain an estimate of the clean speech MFCC. The computation of $G_w(\omega)$ and c_{g_w} is performed at the same frame rate as the conventional filterbank analysis.

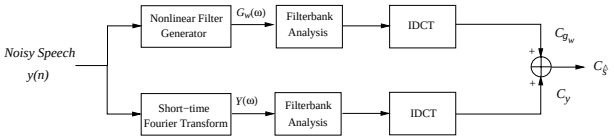

Figure 2: *A noise–robust front–end by using the cepstrum subtraction method.*

3.2. Properties of CSM

The CSM has two major advantages over traditional acoustic noise compensation approaches. The first is its ability to make a "soft–decision" about whether a given frequency bin within an input frame corresponds to speech or noise. This allows the method to continually update noise spectral estimates in those regions of the spectrum where speech energy is low, but not update estimates of the noise spectrum for frequency bins corresponding to spectral peaks where the noise signal is masked by speech. This is seen to be important when compared to common implementation of cepstrum mean subtraction (CMS) which is used to compensate for linear channel distortions. Most implementations of CMS estimate separate cepstrum averages in speech and noise regions by performing a hard classification of input frames into speech and noise frames. The algorithm in Section 2 provides a more principled approach for this decomposition. The second advantage of CSM is that it provides estimates of $G_w(\omega)$ that are updated for each analysis frame. As a result, there is no need to introduce the algorithmic delay associated with buffering observation frames that is typically required for CMS.

In order to illustrate the effects of CSM and SE in the cepstrum domain when applied to compensating for noise corrupted speech, a mel–cepstral distance (MCD) was computed. This distance is plotted for an example speech waveform in Figure 3. Figures 3(a) and (b) are the clean speech waveform and the waveform corrupted by car noise at 10 dB SNR for a digit string "34126" spoken by a male, which are obtained from the TI digit database [11] and the noisy TI digit database [7]. The MCD was defined by $\text{MCD} = D_b \cdot \sqrt{0.1 \cdot d^2(0) + 2\sum_{i=1}^{12} d^2(i)}$, where $D_b(= 0.1)$ was the constant for matching the distance value and the dB value, and $d(i) = c_{clean}(i) - c_{noisy}(i)$ for $0 \le i \le 12$ when $c_{clean}(i)$ and $c_{noisy}(i)$ were the i-th MFCC vector components obtained from clean speech and noisy speech, respectively. The scale factor of 0.1 was introduced to reproduce the weighting applied to energy in speech recognition. From the figure, it is clear that SE and CSM have visibly reduced MCD with respect to the baseline uncompensated front–end. This is true for all but the first 200 msec of the utterance in Figure 3 because the speech enhancement algorithm needs those initial frames to track the

[1] Since the minimum value of a set of random variables is smaller than their mean, the minimum noise estimation should be biased.

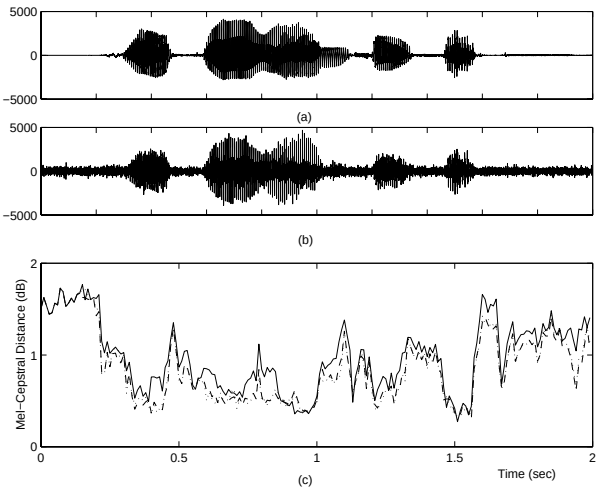

Figure 3: *Speech waveforms and the mel–cepstral distances after applying each robust technique; (a) clean speech waveform of a connected digit string "34126" spoken by a male; (b) noisy speech waveform corrupted by car noise at 10 dB SNR; (c) mel–cepstral distances between (a) and (b) by the baseline (—), SE (· · ·), and CSM (– –).*

noise statistics.

3.3. Combination of feature and channel compensation

It is often the case that, in addition to environmental acoustic noise, there also exists linear channel distortion which may be caused by transducer mismatch. In this case, a more accurate model of the speech corruption process would be given by

$$y(n) = (s(n) + w_1(n)) * h(n) + w_2(n) \tag{5}$$

where $h(n)$ refers to an impulse response associated with channel distortion, and $w_1(n)$ and $w_2(n)$ are environmental acoustic noise and additive channel noise, respectively. The right–hand side of Equation (5) is decomposed into two components: signal–dependent component, $s(n) * h(n)$, and noise component, $w_1(n)*h(n)+w_2(n)$. The speech enhancement algorithm in Section 2 was designed to remove the noise component. Following the same notation as used in Equation (3), the enhanced speech obtained after applying the signal distortion model given in Equation (5) can be written as

$$\widehat{s * h}(n) = \hat{s}_h(n) = y(n) * g_{w_1 * h + w_2}(n) \tag{6}$$

where $g_{w_1 * h + w_2}(n)$ denotes the time–domain nonlinear frequency dependent gain function described in Section 3.1. Similarly, from Equation (4), the cepstrum for the channel corrupted clean speech can be written as

$$c_{\hat{s}_h} = c_y + c_{g_{w_1 * h + w_2}} \tag{7}$$

where c_y and $c_{g_{w_1 * h + w_2}}$ are the noisy speech cepstrum in Equation (5) and the noise cepstrum corresponding to $g_{w_1 * h + w_2}(n)$, respectively. However, the estimated clean cepstrum has the channel distortion which is convolved with actual clean speech. To obtain the clean speech cepstrum, $c_{\hat{s}}$, from the channel corrupted speech cepstrum, $c_{\hat{s}_h}$, an estimate of the cepstrum domain representation for channel distortion, $h(n)$, is needed. A good approximate estimate of this distortion can be obtained by a long–term average [12], $\bar{c}_{\hat{s}_h}$, which is simply subtracted from the channel corrupted cepstrum:

$$c_{\hat{s}} = c_{\hat{s}_h} - \bar{c}_{\hat{s}_h}. \tag{8}$$

4. Speech recognition experiments

In this section, we evaluated the ASR performance of SE and CSM on a connected digit task as defined by the Aurora 2 database under two different paradigms. First, the standard procedure that has been specified by the Aurora group for standardizing a distributed speech recognition (DSR) front–end has been used. We followed the procedures for HMM training under multiple conditions for recognition using the HTK software tools [7]. Second, SE and CSM were evaluated in a highly mismatched training–testing condition. Speaker–independent HMM's were trained with the telephone–line speech data and tested on the Aurora 2 database. Head–body–tail models were trained for the connected digit task [12]. The WATSON speech recognition engine was used for ASR [13]. We begin this section by explaining our baseline front–end.

4.1. Baseline front–end

In order to extract MFCC's, speech signals were blocked into speech segments of 20 ms with a frame rate of 100 Hz. A Hamming window was applied to each speech segment and a 512–point FFT was computed over the windowed speech segment. A preemphasis filter with a factor of 0.95 was applied in the frequency domain. A set of 24 filterbank log–magnitudes were obtained by applying a set of triangular weighting functions over a 4 kHz bandwidth in the spectral magnitude domain. The characteristics of these filterbanks were similar but not identical to those used in [14]. An IDCT was applied to obtain 13 MFCC's. First and second difference MFCC's were also computed over five and three frame windows, respectively.

4.2. Multi–training condition

The experiments described in Table 1 were performed under the paradigm specified by the Aurora group in [7]. Tables 1(a) and (b) show the word accuracies obtained using several different front–ends for clean speech and for noisy speech, respectively. For the noisy speech results, we averaged the word accuracies between 0 dB and 20 dB SNR. In the table, *Set A, B,* and *C* refer to matched noise condition, mismatched noise condition, and mismatched noise and channel condition, respectively. The first three rows in the tables show that the speech enhancement algorithm reduced the word error rates (WER's) in both clean and noisy environments. It can also be shown that SE outperforms CSM when the techniques are applied without any explicit mechanism for compensation with respect to linear channel distortion.

The last three rows of Tables 1(a) and (b) display the word accuracy obtained when SE and CSM were combined with CMS and energy normalization. There are several observations that can be made from these results. First, CMS, when applied to the baseline front–end, significantly reduced WER on clean and noisy speech by 7% and 13%, respectively. We also investigated the effect of each front–end under different noise types and SNR's. It was found that CMS improved the recognition performance for all noise types and SNR's with respect to the baseline performance. This is because most of the noises were reasonably stationary. Using SE and CSM with CMS gave a 5% reduction in WER compared to those using SE and CSM independently. Second, it was found that CSM+CMS provided slightly more consistent performance increases across different noise types than SE+CMS. Finally, it was found that CSM+CMS outperformed other methods under conditions of linear channel mismatch.

Table 1: Comparison of word accuracies (%) and word error rate reduction between several different front–ends on the Aurora 2 database under the multi–training condition.

(a) Word accuracy for clean speech.

Front–end	*Set A*	*Set C*	Avg. (Impr.)
Baseline	98.55	98.34	98.48
SE	98.60	98.59	98.60 (7.7%)
CSM	98.61	98.54	98.59 (7.0%)
Baseline+CMS	98.89	98.81	98.86
SE+CMS	98.86	98.80	98.84 (-2.1%)
CSM+CMS	98.83	98.83	98.83 (-2.9%)

(b) Word accuracy averaged over between 0 dB to 20 dB SNR.

Front–end	*Set A*	*Set B*	*Set C*	Avg. (Impr.)
Baseline	86.93	86.27	84.58	86.20
SE	89.65	88.35	86.79	88.56 (17.1%)
CSM	89.21	88.00	85.94	88.07 (13.6%)
Baseline+CMS	89.05	88.61	89.67	89.00
SE+CMS	89.84	89.14	89.97	89.59 (5.3%)
CSM+CMS	89.78	89.12	90.39	89.64 (5.8%)

4.3. Mismatched transducer condition

In this condition, each digit was modeled by a set of left–to–right continuous density HMM's. We used a total of 274 context–dependent subword models, which were trained by maximum likelihood estimation. Subword models contained a head–body–tail structure. The head and tail models were represented with three states, and the body models were represented with four states. Each state had eight Gaussian mixtures. Silence was modeled by a single state with 32 Gaussian mixtures. As a result, the recognition system had 274 subword HMM's, 831 states, and 6,672 mixtures. The training set consisted of 9,766 digit strings recorded over the public switched telephone network (PSTN).

Tables 2(a) and (b) show the word accuracy under clean and noisy test conditions. Similar to the results shown in Table 1, SE and CSM provided much better performance than the baseline. When no CMS was used, SE performed better than CSM. However, CSM was significantly better than SE when CMS was applied. Importantly, CSM+CMS reduced a WER by 31.6%, which was much higher than the WER reduction obtained for the multi–training condition shown in Table 1. This is because one of the dominant sources of variabilities between training and testing conditions was transducer variability, which can be interpreted as channel distortion. The training database was recorded by using a vast array of transducers through the PSTN, but the testing database was not. All the test datasets in Table 2 can be considered to include significant channel distortion, while the *Set C* in Table 1 only has a single simulated channel mismatch. As we mentioned in the previous section, CSM+CMS could greatly improve the performance under channel distortion condition.

5. Conclusion

A procedure for performing cepstrum based feature compensation for noise corrupted speech has been presented. The procedure is based on techniques for decomposing speech and noise that were originally developed for noisy speech enhancement. An experimental study performed on the Aurora 2 database demonstrated several important results. First, both the application of SE as a noise preprocessing approach and CSM as a cepstrum compensation approach resulted in significant reduction in ASR WER over the baseline MFCC front–end. Second,

Table 2: Comparison of word accuracies (%) and word error rate reduction between several different front–ends on the Aurora 2 database under the mismatched transducer condition.

(a) Word accuracy for clean speech.

Front–end	*Set A*	*Set C*	Avg. (Impr.)
Baseline	99.23	99.05	99.17
SE	99.05	99.20	99.10 (-8.4%)
CSM	99.20	98.95	99.12 (-6.0%)
Baseline+CMS	99.25	99.30	99.27
SE+CMS	99.03	98.90	98.99 (-38.4%)
CSM+CMS	99.28	99.15	99.24 (-4.1%)

(b) Word accuracy averaged over between 0 dB to 20 dB SNR.

Front–end	*Set A*	*Set B*	*Set C*	Avg. (Impr.)
Baseline	70.44	75.10	70.66	72.35
SE	79.27	78.84	80.13	79.27 (25.0%)
CSM	74.89	77.36	77.44	76.39 (14.6%)
Baseline+CMS	71.62	75.84	71.49	73.28
SE+CMS	79.12	78.84	79.27	79.04 (21.5%)
CSM+CMS	81.13	82.34	81.67	81.72 (31.6%)

application of standard energy and cepstrum mean normalization procedures further reduced WER's under noise conditions. Finally, it was found when speech was corrupted by a combination of linear channel and additive acoustic distortions, the combination of CSM and CMS resulted in the largest performance improvements among all the algorithms considered in this work.

6. References

[1] Junqua, J.-C. and Haton, J.-P., *Robustness in automatic speech recognition*, Boston, MA: Kluwer Academic, 1996.

[2] Kermorvant, C, *A comparison of noise reduction techniques for robust speech recognition*, IDIAP Research Report, IDIAP–RR 99-10, July 1999.

[3] Gales, M. and Young S. J., "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 352-359, Sept. 1996.

[4] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum mean–square error log–spectral amplitude estimator," *IEEE Trans. ASSP*, vol. 33, no. 2, pp. 443-445, Apr. 1985.

[5] Malah, D., Cox, R. V., and Accardi, A. J., "Tracking speech–presence uncertainty to improve speech enhancement in non–stationary noise environments" in *Proc. ICASSP*, Phoenix, AZ, vol. 2, pp. 789-792, Mar. 1999.

[6] Accardi, A. J. and Cox, R. V., "A modular approach to speech enhancement with an application to speech coding" in *Proc. ICASSP*, Phoenix, AZ, vol. 1, pp. 201-204, Mar. 1999.

[7] Pearce, D. and Hirsch, H., "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP*, Beijing, China, Oct. 2000.

[8] Martin, R., "Spectral subtraction based on minimum statistics," in *Proc. Euro. Signal Process. Conf. (EUSIPCO)*, Edinburgh, UK, pp. 1182-1185, Sept. 1994.

[9] Martin, R. and Cox, R. V., "New speech enhancement techniques for low bit rate speech coding," in *Proc. 1999 IEEE Workshop on Speech Coding*, Porvoo, Finland, pp. 165-167, June 1999.

[10] Chengalvarayan, R., "Look–a–head sequential feature vector normalization for noisy speech recognition," in *Proc. ICSLP*, Beijing, China, Oct. 2000.

[11] Leonard, R. G., "A database for speaker–independent digit recognition," in *Proc. ICASSP*, San Diego, CA, vol. 3, pp. 42.11.1-4, Mar. 1984.

[12] Rahim, M., *et al.*, "Signal conditioning techniques for robust speech recognition," *IEEE Signal Process. Lett.*, vol. 3, no. 4, pp. 107-109, Apr. 1996.

[13] Sharp, R. D., *et al.*, "The WATSON speech recognition engine," in *Proc. ICASSP*, Munich, German, pp. 4065-4068, Apr. 1997.

[14] ETSI ES 201 108 v1.1.2, *Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front–end feature extraction algorithm; compression algorithms*, Apr. 2000.