# IMPROVEMENTS ON MEL-FREQUENCY CEPSTRUM MINIMUM-MEAN-SQUARE-ERROR NOISE SUPPRESSOR FOR ROBUST SPEECH RECOGNITION

Dong Yu, Li Deng, Jian Wu, Yifan Gong, and Alex Acero

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
{dongyu; deng; jianwu; ygong; alexac}@microsoft.com

## ABSTRACT

Recently we have developed a non-linear feature-domain noise reduction algorithm based on the minimum mean square error (MMSE) criterion on Mel-frequency cepstra (MFCC) for environment-robust speech recognition. Our novel algorithm operates on the power spectral magnitude of the filter-bank's outputs and outperforms the log-MMSE spectral amplitude noise suppressor proposed by Ephraim and Malah in both recognition accuracy and efficiency as demonstrated on the Aurora-3 corpora. This paper serves two purposes. First, we show that the algorithm is effective on large vocabulary tasks with tri-phone acoustic models. Second, we report improvements on the suppression rule of the original MFCC-MMSE noise suppressor by smoothing the gain over the previous frames to prevent the abrupt change of the gain over frames and adjusting gain function based on the noise power so that the suppression is aggressive when the noise level is high and conservative when the noise level is low. We also propose an efficient and effective parameter tuning algorithm named step-adaptive discriminative learning algorithm (SADLA) to adjust the parameters used by the noise tracker and the suppressor. We observed a 46% relative word error (WER) reduction on an in-house large-vocabulary noisy speech database with a clean trained model, which translates into a 16% relative WER reduction over the original MFCC-MMSE noise suppressor, and 6% relative WER reduction on the Aurora-3 corpora over our original MFCC-MMSE algorithm or 30% relative WER reduction over the CMN baseline.

***Index Terms*** — MMSE Estimator, MFCC, Noise Reduction, Robust ASR, Speech Feature Enhancement, RPROP, SADLA

## 1. INTRODUCTION

Recently we have proposed a non-linear feature-domain noise reduction algorithm based on the minimum mean square error (MMSE) criterion on Mel-frequency cepstra (MFCCs) for environment-robust speech recognition [1]. In [1] we explained that seeking an MMSE estimator on MFCCs can be reduced to seeking a log-MMSE estimator on the Mel-frequency filter bank's (MFFB's) outputs, which in turn can be solved independently for each filter bank channel. We derived the MFCC-MMSE noise suppressor by assigning uniformly distributed random phases to the real-valued filter bank's outputs with the assumption that the artificially generated complex filter bank's outputs follow zero-mean complex normal distributions.

There are two key differences between the MFCC-MMSE noise suppression algorithm and the log-MMSE spectral amplitude estimator proposed by Ephraim and Malah (E&M) [4]. First, the MFCC-MMSE suppression rule is applied to the MFFB's outputs which are better smoothed (lower variance) than the FFT spectral amplitude. Second, the noise variance in the suppression rule used by the MFCC-MMSE suppressor contains an additional term resulting from the phase differences between the clean speech and the noise. We demonstrated on the Aurora-3 corpora [5] that the MFCC-MMSE noise suppressor can achieve better recognition accuracy than the E&M log-MMSE suppressor over all conditions, and the AFE over the well-matched and mid-mismatched conditions combined [1]. The MFCC-MMSE suppressor is also more efficient since the number of the channels in the MFFB is usually much smaller than the number of bins in the FFT domain.

This paper serves two purposes. First, we show that the algorithm is effective on large vocabulary tasks with tri-phone acoustic models. Second, we report improvements on the suppression rule of the original MFCC-MMSE noise suppressor by smoothing the gain over the previous frames to prevent the abrupt change of the gain over frames and adjusting gain function based on the noise power so that the suppression is aggressive when the noise level is high and conservative when the noise level is low. We also propose an efficient and effective parameter tuning algorithm named step-adaptive discriminative learning algorithm (SADLA) to adjust the parameters used by the noise tracker and the suppressor. We observed a 45.84% relative word error (WER) reduction on an in-house large-vocabulary noisy speech database with a clean trained model, which translates into a 15.75% relative WER reduction over the original MFCC-MMSE noise suppressor, and 6.35% relative WER reduction on the Aurora-3 corpora over our original MFCC-MMSE algorithm or 30.35% relative WER reduction over the CMN baseline. If cepstral mean and variance normalization (CMVN) is applied, the WER is reduced to 10.23%

The rest of the paper is organized as follows. In Section 2, we review the MFCC-MMSE noise suppressor with the focus on the gain function and the statistics used. In Section 3, we describe the improvements on the gain function, including the gain smoothing and noise-level adjusted gain function. In Section 4, we introduce the SADLA parameter tuning algorithm and the trade-offs made in the algorithm. We report the experimental results in Section 4 and conclude the paper in Section 5.

## 2. MFCC-MMSE NOISE SUPPRESSOR

The MFCC-MMSE noise suppressor aims to estimate the clean speech MFCC $\hat{c}_x(k)$ from the noisy speech $\boldsymbol{y}$ for each cepstrum dimension $k$ by minimizing the mean square error between the estimated MFCC $\hat{c}_x(k)$ and the true MFCC $c_x(k)$ with the assumption that noises are additive. We have shown in [1] that the solution to this problem is

$$\begin{aligned}\hat{c}_x(k) &= E\{c_x(k)|\boldsymbol{m}_y\} \\ &= E\left\{\sum_b a_{k,b}\log m_x(b)\middle|\boldsymbol{m}_y\right\}\end{aligned}$$

$$
\begin{aligned}
&= \sum_b a_{k,b} E\{\log m_x(b)\,|\boldsymbol{m}_y\} \\
&\cong \sum_b a_{k,b} E\{\log m_x(b)\,|m_y(b)\}. \\
&= \sum_b a_{k,b} \log\Big(G\big(\xi(b), \nu(b)\big) m_y(b)\Big), \qquad (1)
\end{aligned}
$$

where $a_{k,b}$ are the discrete cosine transform (DCT) coefficients,

$$
m_x(b) = \sum_f w_b(f)|X(f)|^2, and
$$

$$
m_y(b) = \sum_f w_b(f)|Y(f)|^2 \qquad (2)
$$

are the Mel-frequency filter bank's output in power for the clean and noisy speech respectively, $b$ is the filter bank channel id, and

$$
G\big(\xi(b), \nu(b)\big) = \frac{\xi(b)}{1+\xi(b)} exp\left\{\frac{1}{2}\int_{\nu(b)}^{\infty} \frac{e^{-t}}{t} dt\right\} \qquad (3)
$$

is the gain function for each filter-bank output. In (3), the quantity

$$
\nu(b) = \frac{\xi(b)}{1+\xi(b)} \gamma(b) \qquad (4)
$$

is defined by the adjusted *a- priori* SNR

$$
\xi(b) \overset{\text{def}}{=} \frac{\sigma_x^2(b)}{\sigma_d^2(b)} \cong \frac{\sigma_x^2(b)}{\sigma_n^2(b) + \sigma_\varphi^2(b)}, \qquad (5)
$$

and the adjusted *a-posteriori* SNR

$$
\gamma(b) \overset{\text{def}}{=} \frac{m_y^2(b)}{\sigma_d^2(b)} \cong \frac{m_y^2(b)}{\sigma_n^2(b) + \sigma_\varphi^2(b)}. \qquad (6)
$$

for each filter bank channel $b$.

The noise variance $\sigma_n^2(b) = E\{m_n^2(b)\}$ is estimated using a minimum controlled recursive moving-average noise tracker similar to the one described in [2], $\sigma_x^2(b)$ is estimated using the decision-directed approach documented in [3][4], and the variance $\sigma_\varphi^2(b)$ resulted from instantaneous phase differences between the clean speech and the mixing noise is estimated as

$$
\begin{aligned}
\sigma_\varphi^2(b) &= E\left\{\left(\sum_f 2|X(f)||N(f)| \cos\varphi(f)\, w_b(f)\right)^2\right\} \\
&= 2\sum_f w_b^2(f) E\{|X(f)|\}^2\, E\{|N(f)|\}^2 \\
&\cong 2\frac{\sum_f w_b^2(f)}{\left(\sum_f w_b(f)\right)^2} \sqrt{\frac{\sigma_x^2(b)}{\sigma_n^2(b)}}\, \sigma_n^2(b). \qquad (7)
\end{aligned}
$$

The complete feature extraction pipeline is shown in Fig. 1.

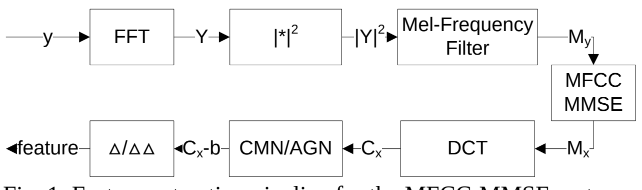

Fig. 1: Feature extraction pipeline for the MFCC-MMSE system.

## 3. SUPPRESSION RULE IMPROVEMENTS

In our original MFCC-MMSE algorithm, the gain $G$ in (3) is a function of the *a-priori* SNR $\xi(b)$ and *a-posteriori* SNR $\gamma(b)$. In this section, we describe two improvements we have made on the gain functions.

### 3.1 Noise-level dependent gain function

It is well known that most speech enhancement algorithms improve the ASR recognition accuracy for the noisy speech at the cost of the degraded performance for the clean speech. Theoretically, this should not happen. For example, when the SNR is very high, the gain $G\big(\xi(b), \nu(b)\big)$ based on (3) is close to 1 (no suppression) in theory. However, in reality it's very difficult to accurately estimate the noise and the SNR. Distortions are inevitably introduced in the enhanced speech and can outweigh the noise reduction for the clean speech. We have observed this behavior using the model trained with clean speech as reported in Section 5.

To prevent the recognition accuracy degradation for the clean speech, we revised the gain function to be

$$
G\big(\xi(b), \nu(b), \sigma_n^2(b)\big) = \begin{cases} 1 & if\ \sigma_n^2(b) < \theta_l \\ G\big(\xi(b), \nu(b)\big)^{\frac{(\sigma_n^2(b)-\theta_l)}{(\theta_h-\theta_l)}} & if\ \theta_l \le \sigma_n^2(b) \le \theta_h \\ G\big(\xi(b), \nu(b)\big) & if\ \sigma_n^2(b) > \theta_h \end{cases} \qquad (8)
$$

where $\theta_l$ and $\theta_h$ are thresholds, so that it depends not only on instantaneous SNRs, but also on the noise power. The gain function (8) indicates that no suppression is applied if the noise power is below the threshold $\theta_l$, and the full suppression is applied if the noise power is above the threshold $\theta_h$. If the noise power is within $[\theta_l\ \theta_h\ ]$, the gain is reduced based on the noise power level. We make the gain function dependent on the noise power instead of the SNR due to the fact that the noise power is usually more stable than the instantaneous SNR and can be estimated without introducing latency as compared to the utterance SNR.

### 3.2 Gain Smoothing

In our original MFCC-MMSE noise suppressor, the gain $G$ in (3) depends only on the *a-priori* SNR $\xi(b)$ and *a-posteriori* SNR $\gamma(b)$ of the current frame. Sometimes the instantaneous SNR changes drastically which in turn causes abrupt change of the gain. To prevent this from happening, we smooth the gain with the previous frame so that

$$
\begin{aligned}
&G\big(\xi(b), \nu(b), \sigma_n^2(b)\big)_t \\
&= \alpha G\big(\xi(b), \nu(b), \sigma_n^2(b)\big)_t + (1-\alpha) G\big(\xi(b), \nu(b), \sigma_n^2(b)\big)_{t-1} \qquad (9)
\end{aligned}
$$

where $\alpha$ is the smooth factor. Note that while the optimal $\alpha$ should be SNR dependent, we have chosen to use one $\alpha$ for all conditions.

.

## 4. PARAMETER TUNING

Conventionally the parameters used in the noise tracking and speech enhancement algorithms are determined by trial-and-error or by an expert who knows the approximate range of the best values. In our original experiments [1], we set the parameters based on the suggestions from [2] and [4]. In this section, we cast the parameter tuning problem as a multi-objective minimum word error rate optimization problem and propose an efficient and effective way to train the parameters.

### 4.1 Multi-objective optimization problem

To optimize the parameters, we need to have a reference and a judgment function. In the human-to-human communication scenario, the reference for the speech enhancement algorithms is usually the clean speech, and the judgment function is usually the 5-level mean opinion score (MOS) provided by the human listeners or its approximation perceptual evaluation of speech quality (PESQ) score. Since the goal of our speech enhancement algorithm is to improve the ASR recognition accuracy by making the noisy speech closer to the clean speech we use the clean-trained ASR model as the reference and the word error rate as the judgment function.

There are two objectives in our optimization process. First, we want to optimize the parameters to minimize the average WER $\varepsilon_a$, i.e.,

$$\hat{\rho}_a = \underset{\rho}{\operatorname{argmin}}\, \varepsilon_a. \tag{10}$$

Second, we want to optimize the parameters to minimize the WER $\varepsilon_c$ on the clean speech, i.e.,

$$\hat{\rho}_c = \underset{\rho}{\operatorname{argmin}}\, \varepsilon_c. \tag{11}$$

Note that these objectives may conflict with each other. For example, a more aggressive suppression would reduce the average WER but may increase the WER on the clean speech. This two-objective optimization problem can be reduced to a single-objective optimization problem by choosing an operating point $\beta \in [0\ 1]$ such that

$$\hat{\rho} = \underset{\rho}{\operatorname{argmin}}\, \varepsilon = \underset{\rho}{\operatorname{argmin}} \beta\varepsilon_c + (1-\beta)\varepsilon_a. \tag{12}$$

For example, if we seek to have no degradation or little degradation on the clean speech, we can choose $\beta = 0.9$, which means we are willing to sacrifice 1% of the WER on the clean speech only if the reduction on the average WER is 9% or more. Different operating points can be used based on the specific usage condition.

### 4.2 Optimization algorithm

The optimization of the objective function (12) has two intrinsic difficulties. First, many parameters used in our noise suppressor are thresholds and smoothing factors. It is very difficult (if not impossible) to get the closed form formula of the derivatives of the WER against the parameters. In other words, the algorithm cannot depend on the closed-form derivatives. Second, there are many local minimums and plateaus in the search space since there are many parameters to lean and the relationship between the parameters are very complicated. The algorithm needs to have some ability to jump out of the local minimums.

With these requirements and constraints in mind we have developed an efficient and effective optimization algorithm. The algorithm optimizes the parameters one by one using approaches similar to the RPROP algorithm [6] with three key differences. First, our algorithm does not require the derivative information and can walk through the plateaus quickly. Second, our algorithm randomly chooses the equally good values and has better chance to walk down the hill instead of being locked at a local minimum. Third, our algorithm splits the training set into several parts and tunes the parameters with one additional part included iteration by iteration until the whole training set is used. We have observed that by doing this the algorithm has better chance to walk out of the local minimum.

Table 1 and 2 summarize the detailed steps in the algorithm. Note that although the algorithm works well practically, it does not guarantee a global optimal solution. In fact, it is a good compromise between efficiency and the possibility of finding the optimal solution. Some algorithms such as the particle swarm optimization can have better chance to find the optimal solution but turned out to be much less efficient compared to the algorithm proposed here. Also note that our algorithm is generic enough that it can be used to solve other optimization problems.

Table 1: The top-level function of the SADLA parameter tuning algorithm

```
Run $n$ iterations {
  Add a new part from the training set;
  For each parameter {
    Tune the parameter to minimize $\varepsilon$;
  }
}
```

Table 2: The function to learn one parameter $p$ in the SADLA parameter tuning algorithm

```
Initialize current value $v$, step size $s$, current WER $\epsilon$, and
  current best WER $\hat{\varepsilon}$;
Initialize last decision $d_{t-1}$ to be correct;
Set current best values $\hat{v} \leftarrow \{v\}$;

Run $m$ iterations or till $|s|$< minimum step $s_p$ {
  $v \leftarrow v + s$;
  Get the new WER $\epsilon_m$ on the training set;
  if ($\hat{\varepsilon} > \epsilon_m$) { $\hat{\varepsilon} \leftarrow \epsilon_m$; $\hat{v} \leftarrow \{v\}$; }
  else if ($|\hat{\varepsilon} - \epsilon_m| \le \theta$) { $\hat{v} \leftarrow \hat{v} \cup \{v\}$; }

  if ($\epsilon_m < \epsilon$) {
    if ($d_{t-1} = true$) $s \leftarrow s \times 1.2$;
    else $s \leftarrow s \times 0.5$;
    $d_{t-1} \leftarrow true$;
  }
  else { $s \leftarrow -s \times 0.5$;  $d_{t-1} \leftarrow false$; }

  $\epsilon \leftarrow \epsilon_n$;
}

Return a randomly selected value from $\hat{v}$.
```

## 5. PERFORMANCE EVALUATION

We have evaluated the improved MFCC-MMSE noise suppressor on an in-house large-vocabulary speech recognition task and the standard Aurora-3 task [5].

The in-house dataset consists of a 1078-word noise suppressor parameter tuning set, a 399-word development set, and a 12351-word test set, each of which is equally separated into three categories: recorded under quiet environment (QE), under mild noise condition (MN), and under high noise condition (HN), all with far-end microphones. The clean-trained (using a large collection of other datasets) tri-phone acoustic model (AM) with five Gaussian mixtures per state is used in the experiments. The 13-dimention mel-frequency cepstrum coeffient (MFCC), its delta, delta-delta, and triple-delta features are transformed into a 39-dimention feature using the HLDA algorithm. The detailed feature extraction pipeline is depicted in Figure 1. We have run five iterations of the SADLA algorithm to adjust the noise suppressor parameters and selected the parameter set that achieved the highest accuracy on the weighted average of the training set and the development set.

Fig. 2 and Tables 3 and 4 compare the recognition performance on the in-house dataset under the conditions of without noise suppressor (baseline), with the original MFCC-MMSE noise suppressor and with the improved MFCC-MMSE noise suppressor. We can see that the original MFCC-MMSE noise suppressor achieved 35.71% relative WER reduction over the baseline on average. However, it increases the WER under the quiet environment by 24.28% relatively since the distortions introduced outweigh the gain under that condition. The improved MFCC-MMSE noise suppressor solved that problem as indicated by the 1.23% relative WER reduction under QE condition by using the noise-power-dependent gain function. The noise-power-dependent gain function, together with the gain smoothing technique also allows for more aggressive suppression under high noise condition and thus leads to higher WER reduction on average. Tables 3 and 4 show that the improved noise suppressor achieved 45.84% relative WER reduction against the baseline system, and 15.75% relative WER reduction over the original noise suppressor.

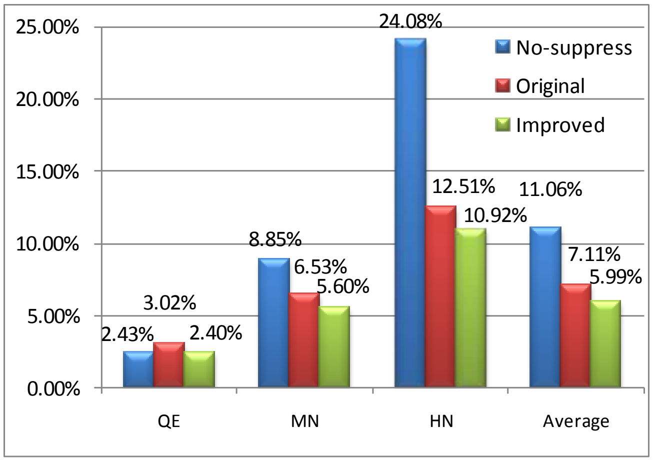

Fig. 2: Absolute WER on the in-house dataset

Table 3: Relative WER reduction against the system without noise suppression on the in-house dataset

| | QE | MN | HN | Average |
|---|---|---|---|---|
| **Original** | -24.28% | 26.21% | 48.05% | 35.71% |
| **Improved** | 1.23% | 36.72% | 54.65% | 45.84% |

Table 4: Relative WER reduction against the original noise suppressor on the in-house dataset

| | QE | MN | HN | Average |
|---|---|---|---|---|
| **Improved** | 20.53% | 14.24% | 12.71% | 15.75% |

The Aurora-3 [5] task consists of noisy digit recognition sub-tasks under realistic automobile environments. In the Aurora-3 corpus, each utterance was labeled as coming from either a high, low, or quiet noise environment, and as being recoded using a close-talk microphone or a hands-free, far-field microphone. All AMs in Aurora-3 were trained with multi-style training scripts came with the corpora and covers well-match, mid-mismatch, and high-mismatch conditions.

By applying the same techniques to the Aurora-3, we get 11.36% absolute WER on average, which translates to 6.35% and 30.35% relative WER reduction over the original MFCC-MMSE algorithm and the system without the noise suppressor (but with CMN) respectively as indicated in Table 5. With CMVN applied, the average absolute WER is reduced to 10.23%, or 37.28% relative WER reduction over the no-suppress baseline.

Table 5: Experimental results on Aurora-3 corpus

| | No-suppress | Original | Improved | +CMVN |
|---|---|---|---|---|
| **Absolute WER** | 16.31% | 12.13% | 11.36% | 10.23% |
| **Relative WER reduction** | baseline | 25.63% | 30.35% | 37.28% |
| **Relative WER reduction** | N/A | baseline | 6.35% | 15.67% |

## 5. SUMMARY AND CONCLUSIONS

In this paper, we presented new improvements over the MFCC-MMSE noise suppressor we proposed recently [1]. Specifically, we introduced the noise-power-dependent gain function and gain smoothing technique to the noise suppressor, and described an efficient and effective step-adaptive discriminative learning algorithm to learn the parameters used in the noise tracker and suppressor. These improvements allow us to suppress aggressively under the high noise condition and conservatively under the quiet environment. The effectiveness of the improvements is demonstrated on an in-house large-vocabulary dataset and the Aurora-3 corpus. On the in-house dataset, the improved noise suppressor achieved 45.84% and 15.75% relative WER reduction on average against the system with no noise suppression and with the original MFCC-MMSE noise suppressor respectively using the clean-trained AM. The higher average WER reduction was achieved with improved accuracy under the quiet condition. On the Aurora-3 corpora, we observed 6.35% relative WER reduction over the original MFCC-MMSE noise suppressor, or 30.35% relative WER reduction over the CMN baseline.

## REFERENCES

[1] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, A. Acero, "a Minimum-Mean-Square-Error Noise Reduction Algorithm on Mel-Frequency Cepstra for Robust Speech Recognition", ICASSP 2008, Las Vegas, USA.

[2] G I. Cohen and B. Berdugo. “Noise estimation by minima controlled recursive averaging for robust speech enhancement,” IEEE Signal Proc. Letters, Vol. 9, 2002, pp. 12-15.

[3] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator,” IEEE Trans. Acoustics, Speech and Signal Proc, Vol. ASSP-32, pp. 1109-1121, 1984.

[4] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” IEEE Trans. Acoustics, Speech and Signal Proc, vol. ASSP-33, pp. 443–445, 1985.

[5] H. G. Hirsch and D. Pearce, “The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” Proc. ISCA ITRW ASR, 2000.

[6] M. Riedmiller, and H. Braun, “RPROP - A fast adaptive learning algorithm”, Technical Report (Also Proc. of ISCIS VII), Universitat Karlsruhe, 1992.