EFFICIENT AUDIO DECLIPPING USING REGULARIZED LEAST SQUARES

Mark J. Harvilla and Richard M. Stern

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213 USA
{mharvill, rms}@cs.cmu.edu

ABSTRACT

While many recently-proposed audio declipping algorithms are highly effective in their ability to restore clipped speech, the algorithms' computational complexities inhibit their use in many practical situations. Real-time or nearly real-time performance is impossible using a typical laptop computer, with some algorithms taking as long as 400 times the actual duration of the input to complete restoration. This paper introduces a novel declipping algorithm, referred to as Regularized Blind Amplitude Reconstruction, which is capable of restoring clipped audio at rates much faster than real time and at restoration qualities comparable to existing algorithms. The quality of declipping is evaluated in terms of automatic speech recognition performance on declipped speech, as well as the degree to which each declipping algorithm improves the audio's signal-to-noise ratio.

Index Terms— Nonlinear distortion, declipping, robust speech recognition, speech enhancement, least squares

1. INTRODUCTION

The fields of speech enhancement, noise reduction, and robust speech recognition (i.e., the design of automatic speech recognition systems that perform well independent of variable deployment conditions) have a long and diverse history. Historically, the focus of many speech enhancement and robust feature extraction algorithms has been to vitiate the impact of "standard" noise types, such as additive noise (*e.g.* [1, 2, 3, 4]), interfering speech (*e.g.* [5, 6]), and reverberation (*e.g.* [7, 8]).

Another type of relevant but largely ignored form of distortion is amplitude clipping. Amplitude clipping is a special case of non-invertible dynamic range compression (DRC) which completely eliminates the positive and negative peaks of an audio waveform beyond a particular amplitude value, τ. In practice, clipping often occurs either (1) when recording an input audio signal that exceeds the dynamic range limitations of the A/D converter, or (2) when writing audio data to a file without first properly normalizing it. Clipping can severely degrade the accuracy of automatic speech recognition (ASR) [9] and is generally regarded as perceptually undesirable [10].

Various audio declipping algorithms have been developed over the past few decades. Autoregressive (AR) modeling for speech declipping has been utilized in work by Janssen *et al.* [11], Dahimene *et al.* [12], and indirectly by Fong and Godsill as the foundation for a particle filter [13]. Other prevailing techniques include recursive vector projection [14], reconstructions based on sparse representations of speech [10, 15], compressed sensing [16], and least squares interpolation [9, 17].

This paper introduces a novel declipping algorithm based on regularized least squares minimization. The concept for the algorithm is motivated by the authors' previously-developed *Constrained Blind Amplitude Reconstruction* (CBAR) algorithm, which uses constrained least squares minimization for audio declipping [9]. As will be shown, the newly-developed *Regularized Blind Amplitude Reconstruction* (RBAR) algorithm greatly reduces the computational complexity of declipping with respect to CBAR while maintaining comparable declipping performance in terms of both signal-to-noise ratio (SNR) and ASR word error rate (WER).

2. AUDIO CLIPPING

A mathematical definition of clipping that will be utilized in this paper is as follows:

$$x_c[n] = \begin{cases} x[n] & \text{if } |x[n]| < \tau \\ \tau \cdot \text{sgn}(x[n]) & \text{if } |x[n]| \geq \tau \end{cases} \quad (1)$$

In Eq. 1, $x[n]$ is an unadulterated speech signal, $x_c[n]$ is a clipped speech signal, and τ is the *clipping threshold*, i.e., the absolute amplitude value beyond which input signal samples are lost. In this paper, the threshold value will be expressed in terms of percentiles of the absolute value of the input speech. We use the designation $\tau = \text{P}_r$, which indicates that r percent of the speech data lies in $(-\tau, +\tau)$ and $(100 - r)$ percent of the data is clipped. Computing τ in this fashion causes the effect of clipping to be independent of arbitrary scaling of the waveform, allowing for more controlled experiments. In this paper, it is assumed that τ is known *a priori* and that clipped samples can be precisely identified.[1]

3. EXISTING APPROACHES AND MOTIVATION

As described in Sec. 1, a wealth of creative techniques have been applied to the problem of declipping. Unfortunately, as illustrated in [9], most of the algorithms provide no improvement in quality (or contribute to further degradation) of clipped signals in all cases except for the most benign clipping thresholds (e.g., $\tau \geq \text{P}_{95}$). For this reason, only the two most effective, state-of-the-art algorithms are described more thoroughly here.

3.1. Consistent iterative hard thresholding

Kitic *et al.* recently proposed a highly-effective sparsity-based algorithm for declipping [10], which will be referred to as *Kitic-IHT*. Each incoming frame of clipped speech is represented using a sparse

[1] In the absence of noise, the identification of clipped samples is trivial. When additive noise is present, however, the problem becomes more difficult.

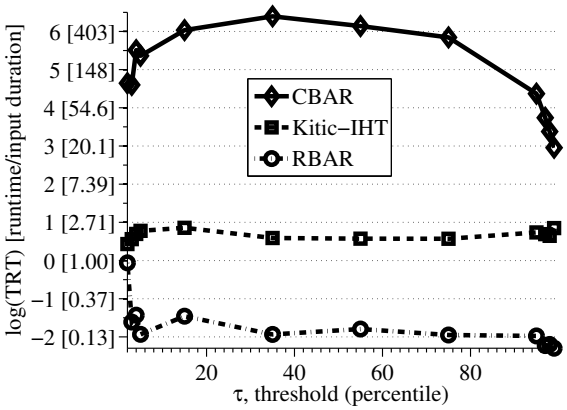


Fig. 1. Average runtime of declipping algorithms over 50 independent trials when used to repair a voiced speech segment. The plotted data depicts the natural logarithm of the TRT value; the actual TRT value is indicated in brackets.

linear combination of Gabor basis vectors. The weights of the linear combination are learned using a modified form of Iterative Hard Thresholding [18]. The algorithm is deemed "consistent" as it requires the interpolated sample values to be greater than or equal to τ in the absolute sense, and carry the same sign as the corresponding clipped signal samples. As will be shown, Kitic-IHT gives rise to substantial improvements in SNR and WER.

3.2. Constrained blind amplitude reconstruction

The recently proposed *Constrained Blind Amplitude Reconstruction* (CBAR) algorithm interpolates clipped segments by minimizing the energy of the second derivative of the reconstructed signal. To understand the algorithm more formally, consider the following definitions.

Let $\boldsymbol{x}$ be a column vector of length L, which contains all the samples of a frame of clipped speech. Suppose there are $R \leq L$ reliable samples contained in the vector $\boldsymbol{x}_r$ and $C = L - R$ clipped samples contained in the vector $\boldsymbol{x}_c$. Let $\boldsymbol{S}_r$ be the R x L matrix obtained from the L x L identity matrix by removing all rows corresponding to a clipped sample. Similarly, let $\boldsymbol{S}_c$ be the C x L matrix obtained from the L x L identity matrix by removing all rows corresponding to reliable samples. Finally, let $\boldsymbol{D}_i$ represent the i^{th} derivative, a linear operator. Note that:

$$\boldsymbol{x} = \boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \boldsymbol{x}_c \tag{2}$$

The CBAR declipping algorithm restores clipped signal samples by solving the following constrained minimization problem:

$$\begin{aligned} \underset{\boldsymbol{x}_c}{\text{minimize}} \quad & ||\boldsymbol{D}_2 \left(\boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \boldsymbol{x}_c \right)||_2^2 \\ \text{subject to} \quad & \boldsymbol{x}_c \circ \operatorname{sgn} \boldsymbol{S}_c \boldsymbol{x} \geq +\tau \mathbf{1} \end{aligned} \tag{3}$$

In the constraint term of Eq. 3, the $\circ$ represents the Hadamard (elementwise) product of two vectors or matrices. The product $\boldsymbol{S}_c \boldsymbol{x}$ is a C x 1 vector containing the clipped samples from the original signal frame, $\boldsymbol{x}$, but with the reliable samples removed. Where the observed clipped sample is equal to $+\tau$, the underlying unclipped sample (the value of which is to be estimated) must be greater than or equal to $+\tau$. Inversely, where the observed clipped sample is equal to $-\tau$, the underlying unclipped sample must be less than or equal to $-\tau$. Requiring (each element of) the elementwise product of $\boldsymbol{x}_c$ and the sign of the corresponding observed clipped samples to be greater than τ incorporates this knowledge.

3.3. Motivation for current work

While both Kitic-IHT and CBAR can be highly effective for clipped signal repair, their respective computational complexities may cause the algorithms to be impractical for many real-world use cases and certainly eliminate the possibility of real-time, "online" processing. The efficiency of an algorithm can be measured in terms of its *times real-time* (TRT) value, defined as the ratio of the time it takes to process audio data to the actual duration of the audio data. For example, if an algorithm takes 5 seconds to process a 4-second duration utterance, its TRT value would be 5/4 = 1.25. The TRT values as a function of τ for Kitic-IHT and CBAR are shown in Fig. 1. It can be seen that both Kitic-IHT and CBAR process data much slower than real time, with CBAR running at over 400 times real-time in the worst case. The motivation for the work in this paper is to develop an algorithm that exhibits declipping performance comparable to CBAR and Kitic-IHT but with the ability to process data at a much faster rate.

4. REGULARIZED BLIND AMPLITUDE RECONSTRUCTION

The principal culprit underlying the slow processing speed of CBAR is the imposition of a hard constraint on the minimization of the energy of the second derivative. By modifying the form of the CBAR objective function to include a regularization term, the hard constraint becomes unnecessary, and a closed-form solution is possible.

4.1. Regularization

A standard least squares problem can be stated as follows:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 \tag{4}$$

Regularization is often used to modify the standard least squares problem statement such that the solution vector, $\hat{\boldsymbol{w}}$, likely has more desirable characteristics. For example, rather than solving Eq. 4, one may be interested in finding a solution with relatively low energy. This can be achieved by solving the following problem:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + \lambda ||\boldsymbol{H}\boldsymbol{w}||_2^2 \tag{5}$$

If $\boldsymbol{H} = \boldsymbol{I}$, the energy of $\boldsymbol{w}$ is minimized in the original space, otherwise its energy is minimized in the space defined by the linear operator $\boldsymbol{H}$. Naturally, λ is an adjustable real-valued parameter that quantifies the relative importance of the regularizing term in the minimization. Note that any number of linear regularizing terms can be added to the objective function and a closed-form solution is still possible. The form of regularization most relevant to this discussion is as follows:

$$\begin{aligned} \hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + \lambda_0 ||\boldsymbol{t}_0 - \boldsymbol{H}_0 \boldsymbol{w}||_2^2 \\ + \lambda_1 ||\boldsymbol{t}_1 - \boldsymbol{H}_1 \boldsymbol{w}||_2^2 \end{aligned} \tag{6}$$

In Eq. 6, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ are target vectors used to guide the solution in the spaces defined by $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$, respectively. For declipping,

these terms will be used to guide the solution toward values greater than $+\tau$, where the signal is clipped at $+\tau$, and less than $-\tau$, where the signal is clipped at $-\tau$. Using $J(\boldsymbol{w})$ to denote the objective function as a whole, the solution vector, $\hat{\boldsymbol{w}}$, is obtained by finding the matrix derivative of $J(\boldsymbol{w})$ and setting it equal to $\mathbf{0}$, as follows.

$$\frac{\partial}{\partial \boldsymbol{w}} J(\boldsymbol{w}) = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}) + 2\lambda_0 \boldsymbol{H}_0^T(\boldsymbol{H}_0\boldsymbol{w} - \boldsymbol{t}_0) + 2\lambda_1 \boldsymbol{H}_1^T(\boldsymbol{H}_1\boldsymbol{w} - \boldsymbol{t}_1) \tag{7}$$

Setting Eq. 7 equal to $\mathbf{0}$, as noted, yields the following solution:

$$\hat{\boldsymbol{w}} = (\boldsymbol{A}^T\boldsymbol{A} + \lambda_0 \boldsymbol{H}_0^T\boldsymbol{H}_0 + \lambda_1 \boldsymbol{H}_1^T\boldsymbol{H}_1)^{-1} \times (\boldsymbol{A}^T\boldsymbol{y} + \lambda_0 \boldsymbol{H}_0^T\boldsymbol{t}_0 + \lambda_1 \boldsymbol{H}_1^T\boldsymbol{t}_1) \tag{8}$$

4.2. Applying regularization to declipping

Of the $C = L - R$ clipped samples in the vector $\boldsymbol{x}$, suppose there are C_p positively-clipped samples (i.e., samples clipped at $+\tau$) and C_n negatively-clipped samples (i.e., samples clipped at $-\tau$). Expanding on the notation developed in Sec. 3.2, define $\boldsymbol{S}_c^+$ to be the C_p x C matrix obtained from the C x C identity matrix by removing all rows corresponding to a negatively-clipped sample. Similarly, let $\boldsymbol{S}_c^-$ be the C_n x C matrix obtained from the C x C identity matrix by removing all rows corresponding to positively-clipped samples. Note the following relationship is true:

$$\boldsymbol{x}_c = (\boldsymbol{S}_c^+)^T\boldsymbol{x}_c^+ + (\boldsymbol{S}_c^-)^T\boldsymbol{x}_c^- \tag{9}$$

Given the signal decomposition of Eq. 9, the regularized objective function for declipping can be framed as follows:

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} ||\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r + \boldsymbol{D}_2\boldsymbol{S}_c^T\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_0 - \boldsymbol{S}_c^+\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_1 - \boldsymbol{S}_c^-\boldsymbol{x}_c||_2^2 \tag{10}$$

The first term in Eq. 10 represents the energy of the 2nd derivative of the reconstructed signal; the second and third terms represent the squared-error between target vectors, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, and the positively-clipped and negatively-clipped sample sets, respectively. Equation 8 can be used to solve Eq. 10 by making the following associations to Eq. 6, and noting that $\boldsymbol{x}_c$ replaces $\boldsymbol{w}$.

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} ||\underbrace{\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r}_{\boldsymbol{y}} + \underbrace{\boldsymbol{D}_2\boldsymbol{S}_c^T}_{-\boldsymbol{A}}\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_0 - \underbrace{\boldsymbol{S}_c^+}_{\boldsymbol{H}_0}\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_1 - \underbrace{\boldsymbol{S}_c^-}_{\boldsymbol{H}_1}\boldsymbol{x}_c||_2^2 \tag{11}$$

Therefore,

$$\hat{\boldsymbol{x}}_c = -(\boldsymbol{S}_c\boldsymbol{D}_2^T\boldsymbol{D}_2\boldsymbol{S}_c^T + \lambda((\boldsymbol{S}_c^+)^T\boldsymbol{S}_c^+ + (\boldsymbol{S}_c^-)^T\boldsymbol{S}_c^-))^{-1} \times (\boldsymbol{S}_c\boldsymbol{D}_2^T\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r - \lambda((\boldsymbol{S}_c^+)^T\boldsymbol{t}_0 - (\boldsymbol{S}_c^-)^T\boldsymbol{t}_1)) \tag{12}$$

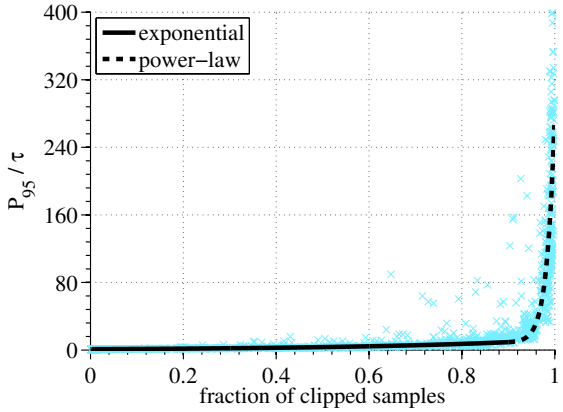


Fig. 2. Scatter plot showing the relationship between the ratio $\frac{P_{95}}{\tau}$ and the fraction of clipped samples in a frame of clipped speech. The plot also shows a piecewise least squares fit to the data, which is used to assign the target vectors in Eq. 12.

The overall signal frame is then resynthesized using Eq. 2. In practice, audio data are processed in 50 ms frames extracted every 12.5 ms. Once declipped, the audio is reconstructed using overlap-add (OLA) [19]. Because RBAR has the tendency of smoothing fricatives, only voiced frames are processed. Voiced frames are detected using cepstral analysis [20].

4.3. Amplitude prediction

In order to compute $\hat{\boldsymbol{x}}_c$ in Eq. 12, values must be assigned to the target vectors, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$. They should be assigned such that the interpolation tends toward a legitimate solution in which the interpolating samples fall above τ in positively-clipped segments, and below $-\tau$ in negatively-clipped segments.

Because the first term of Eq. 10 enforces a smooth reconstruction, it is reasonable to assign dynamically the target vectors to a constant value equal to some robust measure of the peak amplitude in a given clipped frame. Figure 2 shows a scatter plot of the ratio of the 95th percentile of a frame of speech before clipping, to the clipping threshold, τ (i.e., $\frac{P_{95}}{\tau}$) as a function of the fraction of clipped samples in each frame of speech. The points on the scatter plot were obtained by artificially clipping a clean database of speech (independent of the testing data) at five different thresholds:[2] P_{15}, P_{35}, P_{55}, P_{75}, and P_{95}, and using the pre-clipped clean data to determine the ratio, $\frac{P_{95}}{\tau}$.

Nonlinear least squares [21] can be used to fit a regression function to the data in Fig. 2. The optimal fit was found to be a piecewise combination of exponential and power-law functions. Denoting the ratio as $\phi = \frac{P_{95}}{\tau}$ and the fraction of clipped samples as ρ, the resulting regression function is given by:

$$\phi(\rho) = \begin{cases} e^{2.481\rho} & \text{for } \rho \leq 0.9 \\ 271.7493\rho^{59.9519} + 8.8361 & \text{for } \rho > 0.9 \end{cases} \tag{13}$$

Given the value of ρ for an incoming frame of clipped speech

[2]The clipping thresholds for setting τ and artificially clipping the speech are determined from the percentiles over an entire utterance; the threshold used in the ratio, $\frac{P_{95}}{\tau}$, is associated with an individual short-duration frame.

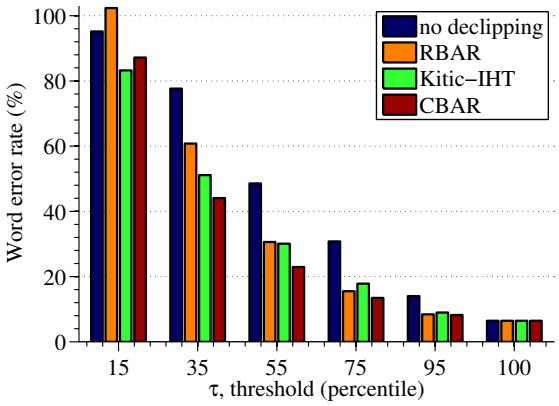


Fig. 3. Results of ASR experiments on speech clipped at various thresholds and then declipped using the indicated algorithm. The ASR system was trained on features extracted from clean, unclipped speech.

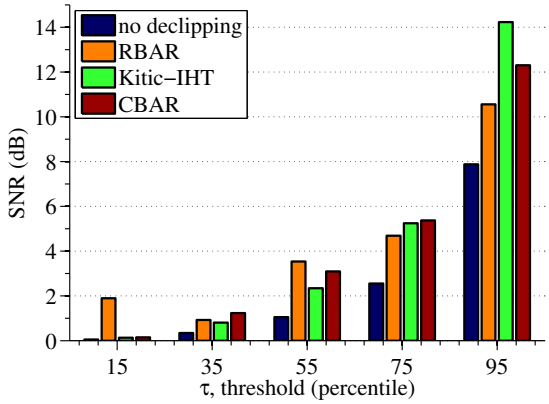


Fig. 4. Median SNR of the RM1 speech database clipped at various thresholds and then declipped using the indicated algorithm. The noise signal in the SNR computation is taken to be the difference between the (de-)clipped signal and the original, unadulterated speech. In some cases, e.g., $\tau = \mathrm{P}_{35}$ and $\tau = \mathrm{P}_{55}$, RBAR yields the largest improvement in SNR over baseline.

(which can be computed trivially with the knowledge of which samples are clipped), the target values are then set as follows:

$$\boldsymbol{t}_0\phi(\rho)\tau\mathbf{1}; \quad \boldsymbol{t}_1 - \phi(\rho)\tau\mathbf{1} \tag{14}$$

5. EXPERIMENTAL RESULTS

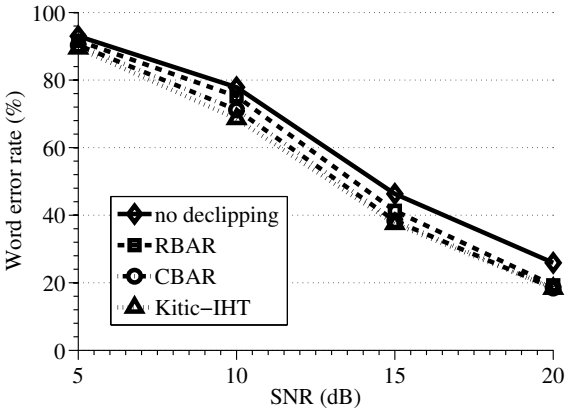
Speech recognition experiments are run using CMU Sphinx-III [22], trained on the clean RM1 database [23], with Mel-Frequency Cepstral Coefficient (MFCC) features. The RM1 database is sampled at 16 kHz and contains 1600 training utterances and 600 test utterances. A standard bigram language model and 8-component GMM-based acoustic model were used. Further experimental details are provided in [9].

Figure 3 compares the word error rates (WERs) obtained using RBAR with error rates with no processing, CBAR, and the Kitic-IHT algorithm. It can be seen that RBAR provides a significant reduction in WER that matches or exceeds the performance of Kitic-IHT for the important range of thresholds τ between P_{55} and P_{95}, and provides a very substantial benefit compared to no processing at all but the worst clipping thresholds. Figure 4, demonstrates that RBAR provides competitive improvements in SNR as well.

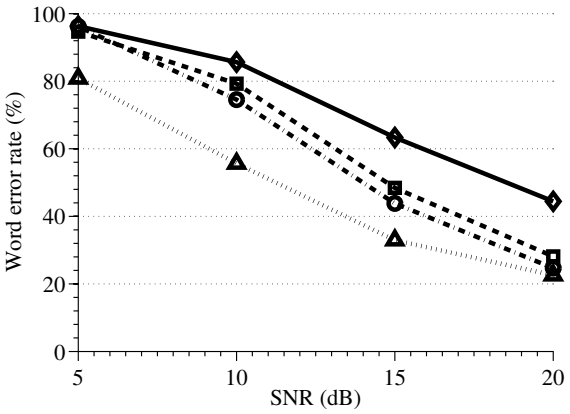
Most importably, Fig. 1 confirms that dramatically reduced runtimes that are possible with RBAR relative to Kitic-IHT and CBAR. RBAR is between 110 and 4,149 times faster than CBAR (for $\tau = \mathrm{P}_2$ and $\tau = \mathrm{P}_{35}$, respectively), and between 1.6 times and 23 times faster than Kitic-IHT (for $\tau = \mathrm{P}_2$ and $\tau = \mathrm{P}_{99}$, respectively). Figure 5 compares the performance of the various algorithms in additive white noise. While RBAR is not quite as robust to additive noise as Kitic-IHT (Fig. 5) it still leads to significant reductions in WER, close to what had been observed with the CBAR algorithm.

6. ACKNOWLEDGEMENTS

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20024.



(a) $\tau = \mathrm{P}_{95}$



(b) $\tau = \mathrm{P}_{75}$

Fig. 5. Results of ASR experiments on clipped speech added to white Gaussian noise and then declipped using the indicated algorithm. Noise is added after clipping; perfect knowledge of which samples are clipped is assumed to be known.

7. REFERENCES

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, Apr 1979.

[2] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, Apr 1979, vol. 4, pp. 208–211.

[3] P.J. Moreno, B. Raj, and R.M. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, May 1996, vol. 2, pp. 733–736 vol. 2.

[4] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012.

[5] H.-M. Park and R. M. Stern, "Spatial separation of speech signals using amplitude estimation based on interaural comparisons of zero crossings," *Speech Communication*, vol. 51, pp. 15–25, January 2009.

[6] C. Kim, K. Kumar, B. Raj, and R. M. Stern, "Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain," in *Interspeech 2009*, Brighton, UK, September 2009.

[7] D. Gelbart and N. Morgan, "Evaluating long-term spectral subtraction for reverberant ASR," in *Proc. IEEE ASRU Workshop*, 201, pp. 103–106.

[8] K. Kumar, B. Raj, R. Singh, and R.M. Stern, "An iterative least-squares technique for dereverberation," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 5488–5491.

[9] M.J. Harvilla and R.M. Stern, "Least squares signal declipping for robust speech recognition," in *INTERSPEECH*, September 2014.

[10] S. Kitic, L. Jacques, N. Madhu, M. Hopwood, A. Spriet, and C. De Vleeschouwer, "Consistent iterative hard thresholding for signal declipping," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, May 2013.

[11] A. Janssen, R. Veldhuis, and L. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Trans. on Acoust., Speech and Signal Processing*, pp. 317–330, April 1986.

[12] A. Dahimene, M. Noureddine, and A. Azrar, "A simple algorithm for the restoration of clipped speech signal," in *Informatica*, 2008, pp. 183–188.

[13] W. Fong and S. Godsill, "Monte carlo smoothing for nonlinearly distorted signals," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, May 2001.

[14] S. Miura, H. Nakajima, S. Miyabe, S. Makino, T. Yamada, and K. Nakadai, "Restoration of clipped audio signal using recursive vector projection," in *TENCON*, November 2011.

[15] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio Inpainting," *IEEE Trans. on Acoust., Speech and Signal Processing*, pp. 922–932, April 2012.

[16] B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 12, pp. 2627–2637, Dec 2013.

[17] I. Selesnick, "Least squares with examples in signal processing," [Online], March 2013.

[18] T. Blumensath and M.E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[19] A.V. Oppenheim and R.W. Schafer, *Discrete-time Signal Processing*, chapter 8, pp. 669–670, Prentice Hall, 3 edition, 2010.

[20] A.M. Noll and M.R. Schroder, "Short-time "cepstrum" pitch detection," *Journal of the Acoustical Society of America*, vol. 36, no. 5, pp. 1030, 1964.

[21] The MathWorks, Natick, MA, *Constrained Nonlinear Optimization Algorithms*, 2014.

[22] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer, "The 1996 Hub-4 Sphinx-3 System," in *Proceedings of the DARPA Speech Recognition Workshop*, 1997.

[23] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett, "The DARPA 1000-word resource management database for continuous speech recognition," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, April 1988.