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, USA

Carnegie Mellon

Electrical & Computer ENGINEERING

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.

Audio Clipping

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

Audio clipping typically occurs in one of three ways:

1. Upon recording, as a result of exceeding the dynamic range limitations of the A/D converter.
2. As a result of writing improperly-amplitude-normalized data to a file.
3. On purpose, to achieve a desirable perceptual characteristic.

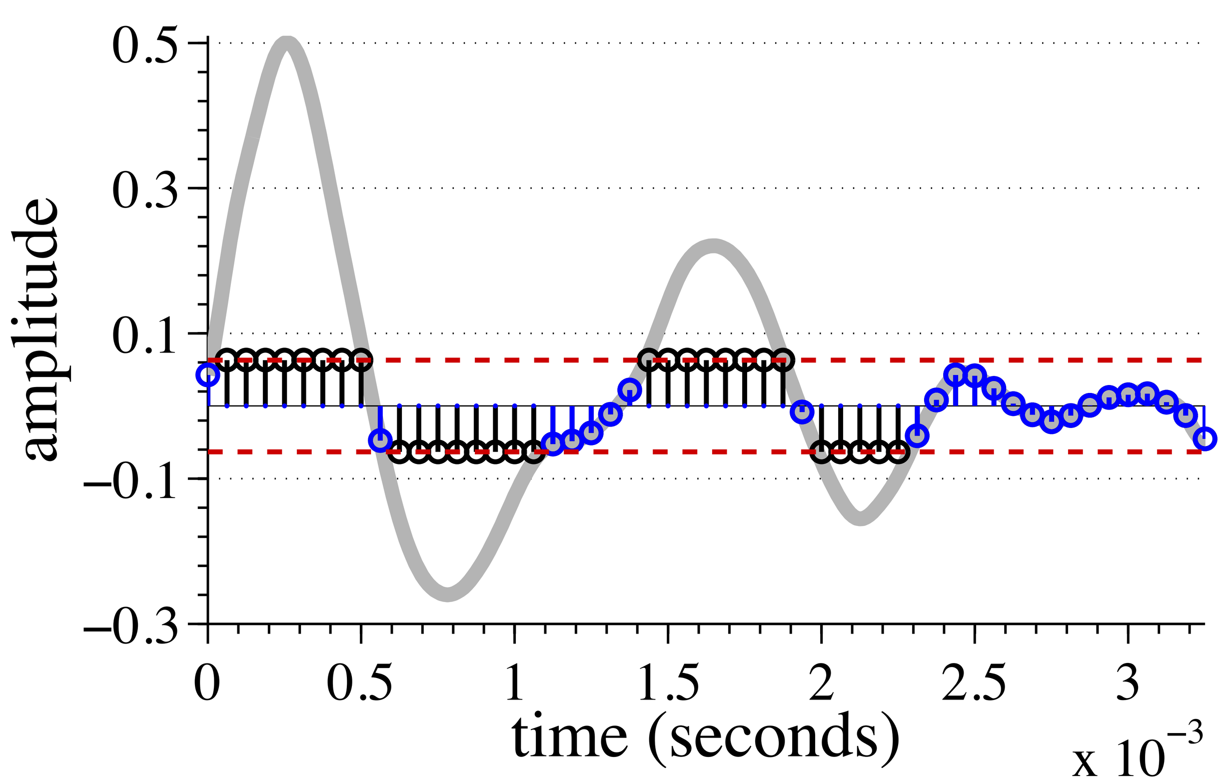

Figure 1: 16-kHz speech signal before and after clipping. The reliable samples after clipping are shown in blue, and the clipped samples are shown in black. The original unclipped signal is in gray.

Sparsity-based declipping (Kitic *et al.*)

- Using the iterative hard thresholding (IHT) algorithm, the Kitic-IHT algorithm learns a sparse representation of incoming clipped speech in terms of Gabor basis vectors.
- The learned sparse representation is then used to reconstruct clipped speech on a frame-by-frame basis.

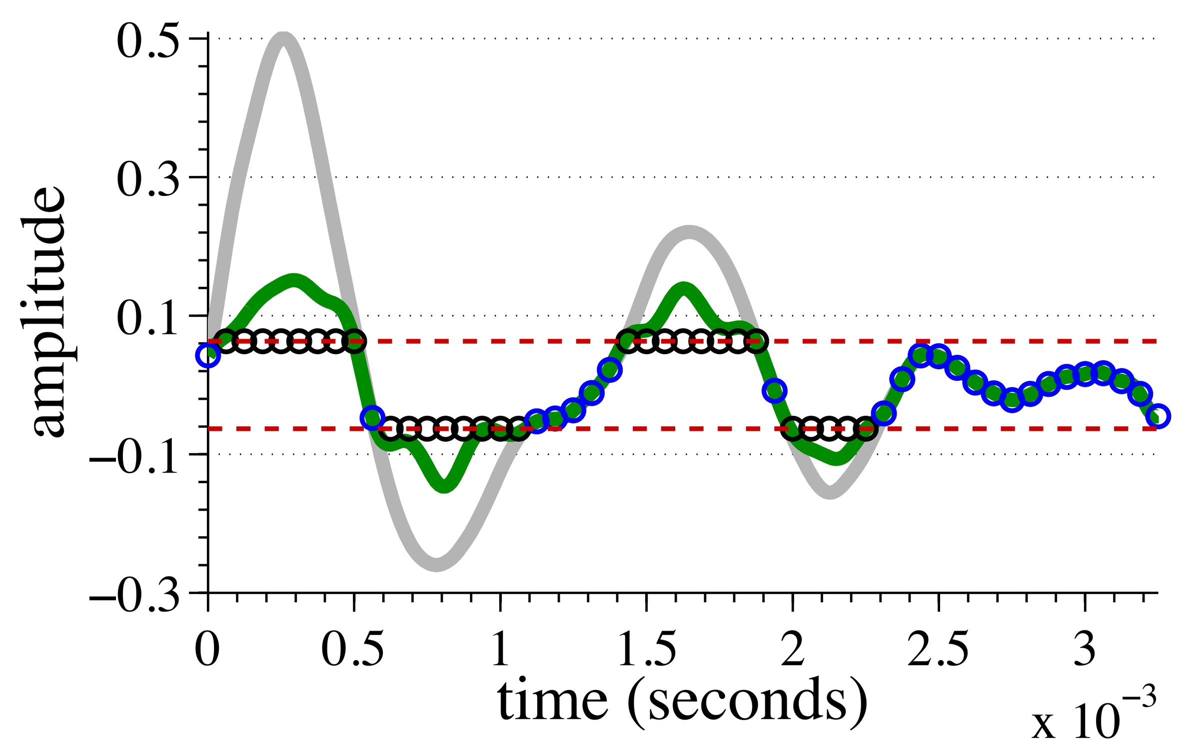

Figure 2: Illustration of the Kitic-IHT reconstruction (green).

Constrained Blind Amplitude Reconstruction (CBAR)

- CBAR declips each speech frame by solving the following nonlinear constrained optimization problem:

$$\underset{\boldsymbol{x}_c}{\text{minimize}} \quad \left|\left|\boldsymbol{D}_2(\boldsymbol{S}_r^T\boldsymbol{x}_r + \boldsymbol{S}_c^T\boldsymbol{x}_c)\right|\right|_2^2$$

$$\text{subject to} \quad \boldsymbol{x}_c \circ \operatorname{sgn} \boldsymbol{S}_c\boldsymbol{x} \geq +\tau\boldsymbol{1}$$

- In the above, $\boldsymbol{D}_2$ is the 2nd-derivative operator, $\boldsymbol{S}_r$ and $\boldsymbol{S}_c$ are masking matrices that separate unclipped and clipped samples, respectively; $\boldsymbol{x}$ contains all samples of the speech frame, $\boldsymbol{x}_r$ and $\boldsymbol{x}_c$ contain unclipped and clipped samples, respectively; τ is the clipping threshold, and $\circ$ represents the element-wise product.

Regularized Blind Amplitude Reconstruction (RBAR)

- Because the objective function in CBAR contains a hard constraint, the required iterative solution can render the algorithm prohibitively slow.
- RBAR circumvents this issue by employing a "soft" constraint in the form of regularization terms on the objective function.

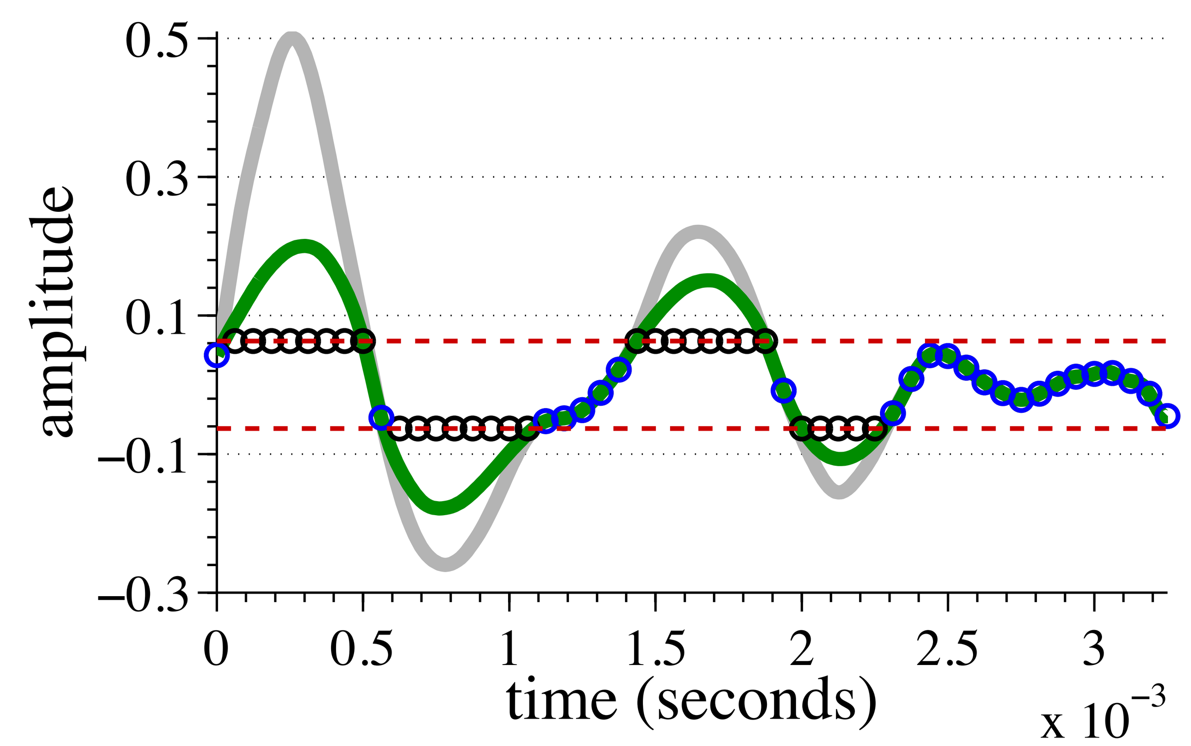

Figure 3: Illustration of the CBAR reconstruction (green); note the smoothness of the interpolation.

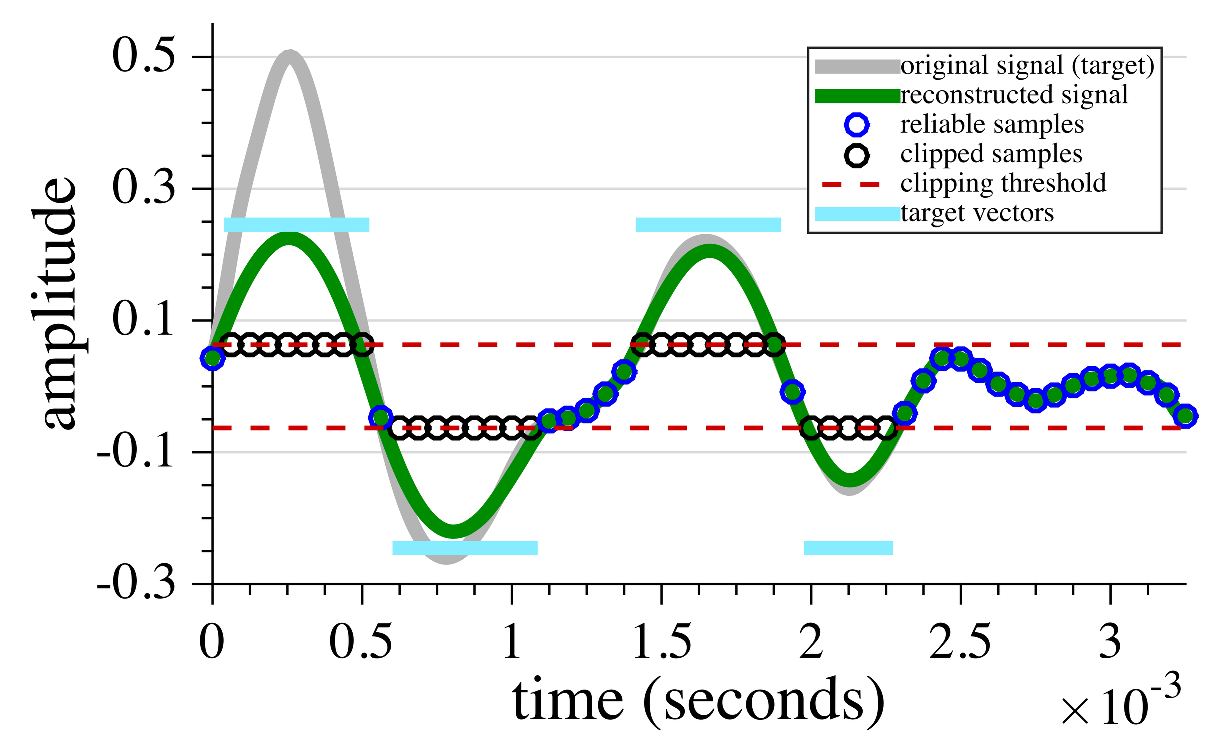

Figure 3: Illustration of the RBAR reconstruction; can be obtained up to 2000 times faster than CBAR.

- RBAR solves the following equation to estimate the clipped sample values for each frame:

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

Experimental Results

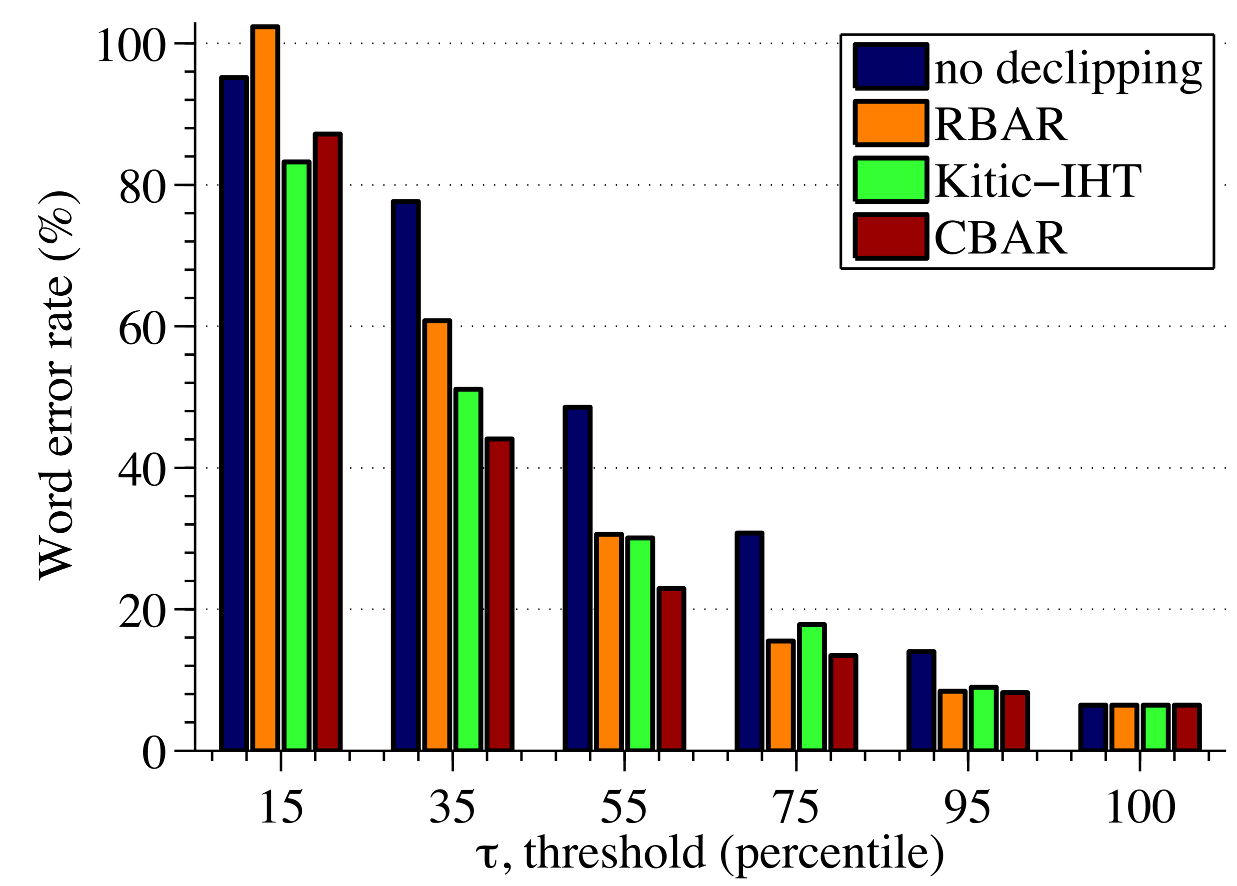

Figure 5: Word error rates of the CMU Sphinx-III automatic speech recognition system using the DARPA RM1 database.

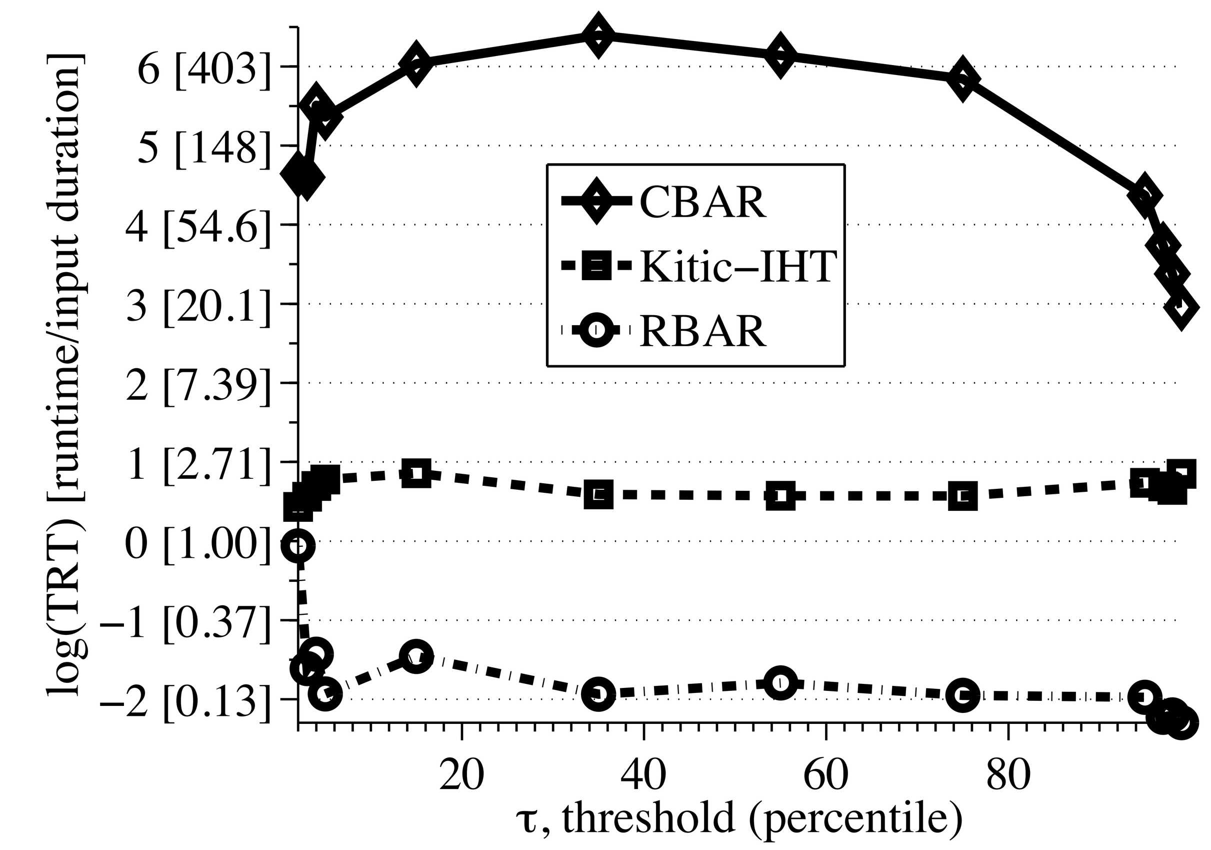

Figure 6: Times real time (TRT) values of individual declipping algorithms. RBAR runs in approximately 1/5 of real time (5 seconds of audio takes 1 second to process). CBAR may take over 400 times real time.

Summary

- The use of a soft constraint dramatically decreases computational complexity.
- The RBAR algorithm can achieve comparable results to CBAR and Kitic-IHT in significantly less time.
- Future work should focus on improving the estimate of the target vectors in the objective function.