Audio Declipping

Damoun Ayman
Github repository

December 29, 2020

1 Introduction

Clipping or saturation is common distortions in digital signal processing. Cipping occurs when the signal reaches a maximum threshold and the waveform is truncated. In the literature there are several approaches to answer this issue: Can we get a good estimation of the original signal from the clipped one ?
In this report we address the problem of recovering a signal from clipped measurements. Based on the methodology developed in the article [11].

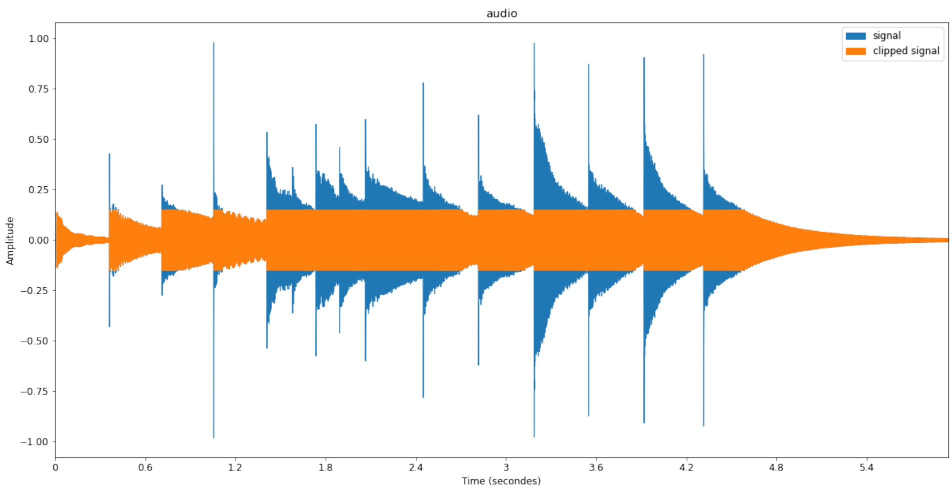

Figure 1: How to estimate the original signal (blue) from the clipped one (orange) ?

2 Audio Declipping with Social Sparsity paper discuss

The authors of the paper [11] presents state of the art approach to recover clipped signal by using iterative thresholding algorithms and the principle of social sparsity.

2.1 Background (Definitions)

Let $s \in \mathbb{C}^N$ be the undistorted signal that we want to recover. Audio declipping can be formulated as :

$$\mathbf{y}^r = \mathbf{M}^r \mathbf{s} \tag{1}$$

$\mathbf{y}^r \in \mathbb{C}^M$ are the reliable sample of the observed signal
$\mathbf{M}^r \in \mathbb{C}^{M \times N}$ is the matrix of the reliable parts of s
we apply a mask M^r to the sample signal to identifey the reliable sample of the observed signal.
we can also define the clipped samples as:

$$\mathbf{y}^m = \mathbf{M}^m \mathbf{s} \tag{2}$$

Both matrices M^r, M^m are based on the skeleton of the identity matrix. they are built by setting the corresponding values of the identity matrix to 0 and do not reduce dimension.

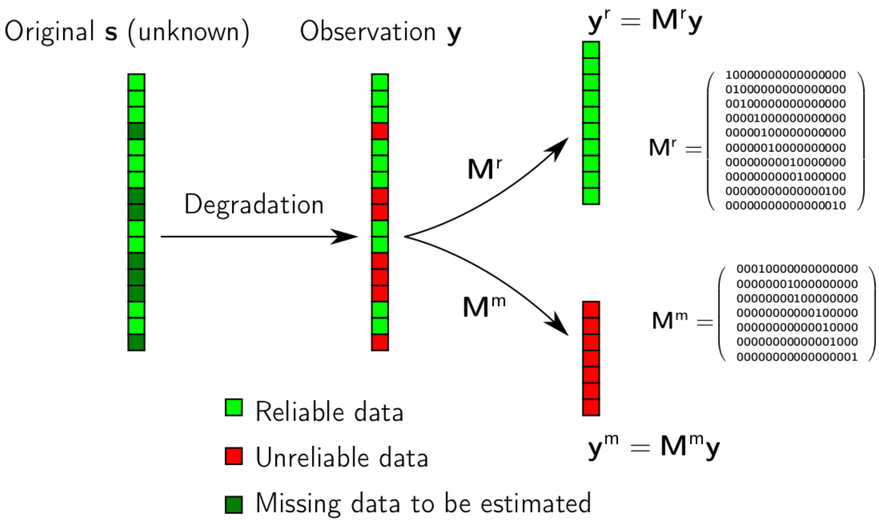

Figure 2: Reliable and unreliable coefficient [8]

For example, in dimension N= 4 with samples 2 and 4 distorted

$$\boldsymbol{M}^r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{M}^m = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is clear that it is an undetermined problem. there is an infinity of possible estimation of s that verifies the equation (1) and (2). To find a good solution, we need prior information about s. This information comes in the form the chosen model of the signal, which constrains the values of s. Sparse model can be used to address this problem.

2.1.1 Sparse model

A sparse model assume that a signal s can be represented by summing up few elementary pieces of signal, called atoms.
Formally:

$$\mathbf{s} = \Phi\alpha$$

with $\Phi \in \mathbb{C}^{M*N}$ is called the dictionnary of ϕ_k atoms and $\alpha \in \mathbb{C}^N$ is sparse.

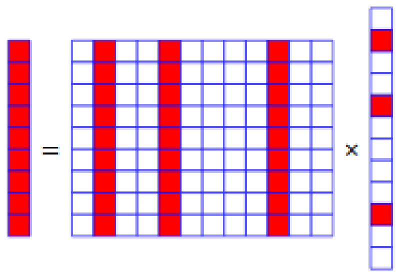

Figure 3: Graphical representation of the sparse synthesis model [5]

Note: A bad choice of dictionary will result in a bad modeling of the signal

2.1.2 Inverse problem framework

Definition 2.1.

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \mathcal{L}(\mathbf{y}, A, \mathbf{s}) + P(\mathbf{s}; \lambda)$$

With $\mathcal{L}(\mathbf{y}, A, \mathbf{s})$ convex loss or data term ,
A regularization term P modeling the assumptions about the sources,
An hyperparameter $\lambda \in \mathbb{R}_+$

2.2 Problem formulation

2.2.1 Constrained and convex inverse problem

Using the dictionnary Φ and sparsity. the audio declipping problem can be formulate as constrained convex optimization problem.

$$\begin{aligned} \hat{\boldsymbol{\alpha}} =& \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{y}^r - \mathbf{M}^r \boldsymbol{\Phi}\boldsymbol{\alpha}\| + \lambda \|\boldsymbol{\alpha}\|_1 \\ \text{s.t. } & \mathbf{M}^{m^+} \boldsymbol{\Phi}\boldsymbol{\alpha} > \theta^{\text{clip}} \\ & \mathbf{M}^{m^-} \boldsymbol{\Phi}\boldsymbol{\alpha} < -\theta^{\text{clip}} \end{aligned} \tag{3}$$

where $\mathbf{M}^{m^+}$ (resp. $\mathbf{M}^{m^-}$) select the positive (resp. negative) missing samples.
θ^{clip} is the clip threshold

2.2.2 Rewrite the constraints

The constrainted convex optimination problem 3 can be rewrited using the well-known function *squared hinge* defined as follows:

$$h^2 : \mathbb{R} \longrightarrow \mathbb{R}_+ \quad z \mapsto h^2(z) = \begin{cases} z^2 & \text{if } z < 0 \\ 0 & \text{if } z \geq 0 \end{cases}$$

By changing the variable $z = x - \theta^{clip}$

$$\begin{cases} \mathcal{L}\left(\theta^{\text{clip}} - x\right) = 0, & \text{if } x \geq \theta^{clip} \\ \mathcal{L}\left(\theta^{\text{clip}} - x\right) = \left(\theta^{\text{clip}} - x\right)^2, & \text{if } x < \theta^{clip} \end{cases}$$

Let

$$\left[\boldsymbol{\theta}^{clip} - \mathbf{x}\right]_+^2 = \sum_{k:\theta_k^{clip} > 0} \left(\theta_k^{clip} - x_k\right)_+^2 + \sum_{k:\theta_k^{clip} < 0} \left(-\theta_k^{clip} + x_k\right)_+^2$$

That leads to the following unconstrained convex problem:

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{y}^r - \mathbf{M}^r \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \frac{1}{2} \left[\boldsymbol{\theta}^{clip} - \mathbf{M}^m \boldsymbol{\Phi}\boldsymbol{\alpha}\right]_+^2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{4}$$

which is under the form

$$f_1(\boldsymbol{\alpha}) + f_2(\boldsymbol{\alpha})$$

with f_1 Lipschitz-differentiable and f_2 semi-convex.
The iterative shrinkage-thresholding algorithm (ISTA) [2] can be applied to solve the (4).

2.3 Solution

The social sparsity procedure allows shrinkage of a coefficient based on the values of coefficients in its neighborhood. The authors of the paper suggests using four types of social shrinkage of Time-frequency (TF) coefficients to approximate a solution to (4). Let $N(t)$ be the set of indices forming the neighborhood of the index t for the time-frequency coefficients $\alpha = \{\alpha_{tf}\}$.

- **Lasso:**

$$\tilde{\alpha}_{tf} = \mathbb{S}^{L}_{\lambda}\left(\alpha_{tf}\right) = \alpha_{tf}\left(1 - \frac{\lambda}{|\alpha_{tf}|}\right)^{+}$$

- **WGL:** Windowed Group Lasso

$$\tilde{\alpha}_{tf} = \mathbb{S}^{WGL}_{\lambda}\left(\alpha_{tf}\right) = \alpha_{tf}\left(1 - \frac{\lambda}{\sqrt{\sum_{t'\in\mathcal{N}(t)} |\alpha_{t'f}|^{2}}}\right)^{+}$$

- **EW:** Empirical Wiener

$$\tilde{\alpha}_{tf} = \mathbb{S}^{EW}_{\lambda}\left(\alpha_{tf}\right) = \alpha_{tf}\left(1 - \frac{\lambda^{2}}{|\alpha_{tf}|^{2}}\right)^{+}$$

- **PEW:** Persistent Empirical Wiener

$$\tilde{\alpha}_{tf} = \mathbb{S}^{PEW}_{\lambda}\left(\alpha_{tf}\right) = \alpha_{tf}\left(1 - \frac{\lambda^{2}}{\sum_{t'\in\mathcal{N}(t)} |\alpha_{t'f}|^{2}}\right)^{+}$$

The ISTA Social sparsity declipper algorithm used in [11] is presented below: For the choice of hyperpa-

ALGORITHM 1: ISTA-type Social sparsity declipper [11]

Input : y – the observed signal
$\delta = \|\mathbf{\Phi}\mathbf{\Phi}^{*}\|$
$\boldsymbol{\alpha}^{(0)} \in \mathbb{C}^{N}$, $\lambda > 0$

$\mathbf{z}^{0} \leftarrow \boldsymbol{\alpha}^{(0)}$ `/* Initialization */`
for $i = 1, \ldots$ ***until*** *convergence* **do**
  $\mathrm{g}_{1} \leftarrow -\mathbf{\Phi}^{*}\mathbf{M}^{r^{T}}\left(\mathbf{y}^{r} - \mathbf{M}^{r}\mathbf{\Phi}\mathbf{z}^{(i-1)}\right)$ `/* gradients */`
  $\mathrm{g}_{2} \leftarrow -\mathbf{\Phi}^{*}\mathbf{M}^{c^{T}}\left[\boldsymbol{\theta}^{clip} - \mathbf{M}^{c}\mathbf{\Phi}\mathbf{z}^{(i-1)}\right]_{+}$
  $\alpha^{i} \leftarrow \mathbb{S}_{\lambda/\delta}\left(\mathbf{z}^{(i-1)} - \frac{1}{\delta}(\mathbf{g}1 + \mathbf{g}2)\right)$ `/* step, shrink */`
  $\mathbf{z}^{i} \leftarrow \boldsymbol{\alpha}^{(i)} + \gamma\left(\boldsymbol{\alpha}^{(i)} - \boldsymbol{\alpha}^{(i-1)}\right)$ `/* extrapolate */`
end
return $\Phi\alpha^{(i)}$

rameter λ a large value is chosen for a hundreds of iterations, then λ is decreased until the target one.

2.4 Numerical results

The autors diclipped same audio signal (speech and music). The figure 4 shows the results of a signal with a clipping level $\theta^{clip} = 0.2$ using the different estimators. In the time domain, it turns out that the operators (P)EW , HT and OMP give much better estimates than the (WG)L. OMP appears to produce too many oscillations on high-frequency, while HT occasionally exceeds the original amplitude values.

In the figure 5, all operators are improving the SNR_{m}. However, Lasso and WGL seem to be the weakest overall. This confirms the results of Figure 4. The experiments in 4 5 shows that only EW and PEW are well-performing out of the four choices, and they outperform the OMP approache.

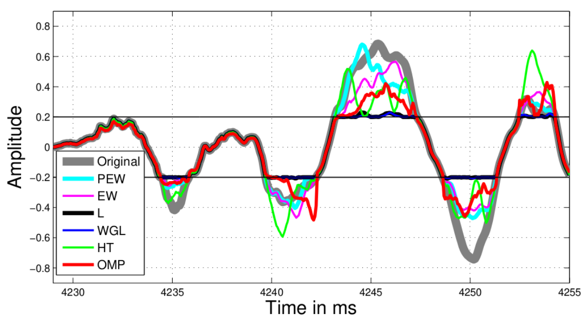

Figure 4: Declipped signal ($\theta^{clip} = 0.2$) using the Lasso, WGL, EW, PEW, HT, and OMP operators [5]

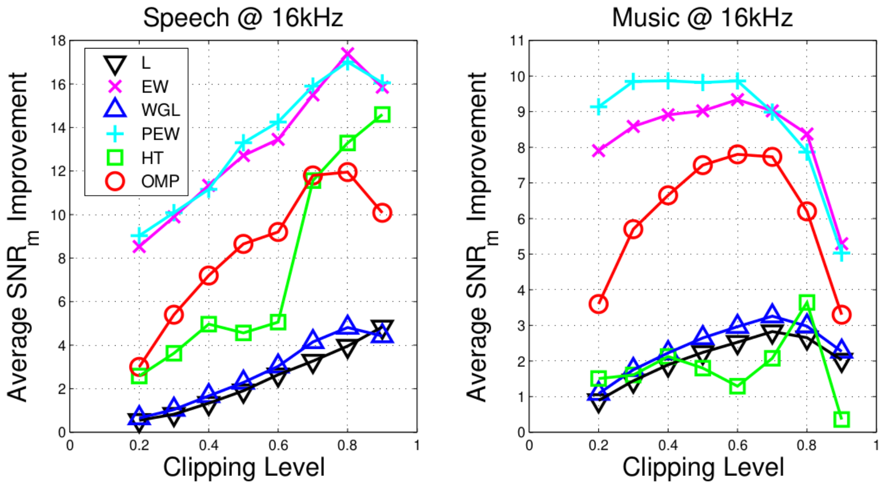

Figure 5: The improvement of SNR_m as a function of clipping level. [5]

3 Implementation

In the literature there are several toolbox that implement the state of the art declipping techniques. The authors of the paper [15] carried out an in-depth survey to study the different approaches of audio declipping, as well as a comparison between different methods. the authors of [15] provide a matlab declipping toolbox that contains the following methods :

Abbreviation	Full name	Reference
C-OMP	Constrained Orthogonal Matching Pursuit	[1]
A-SPADE	Analysis SParse Audio DEclipper	[7]
S-SPADE	Synthesis SParse Audio DEclipper	[14]
$l1$ CP	$l1$-minimization using Chambolle–Pock (analysis)	[15]
$l1$ DR	$l1$-minimization using Douglas–Rachford (synthesis)	[9]
R$l1$CC CP	Reweighted $l1$-min. with Clipping Constraints using Chambolle–Pock (analysis)	[15]
R$l1$CC DR	Reweighted $l1$-min. with Clipping Constraints using Douglas–Rachford (synthesis)	[12]
SS EW	Social Sparsity with Empirical Wiener	[5]
SS PEW	Social Sparsity with Persistent Empirical Wiener	[5]
CSL1	Compressed Sensing method minimizing $l1$-norm	[4]
PWCSL1	Perceptual Compressed Sensing method minimizing $l1$-norm	[4]
PWCSL1	Parabola-Weighted Compressed Sensing method minimizing $l1$-norm	[15]
PW$l1$ CP	Parabola-Weighted $l1$-minimization using Chambolle–Pock (analysis)	[15]
PW$l1$ DR	Parabola-Weighted $l1$-minimization using Douglas–Rachford (synthesis)	[13]
DL	Dictionary Learning approach	[10]
NMF	Nonnegative Matrix Factorization	[3]
Janssen	Janssen method for inpainting	[6]

In the paper the authors compare this methods according to several metrics, for example Signal-to-Distortion Ratio (SDR). the SDR evaluate the physical quality of restoration.

Definition 3.1. *The SDR for two signals u and v is defined as:*

$$\mathrm{SDR}(\mathbf{u},\mathbf{v}) = 20\log_{10}\frac{\|\mathbf{u}\|_2}{\|\mathbf{u}-\mathbf{v}\|_2}$$

Warning: The evaluation of the SDR on the whole signal may penalize the approaches that produce signals inconsistent in the reliable part.

To bypass this problem, in [5] the SDR is computed on the clipped part only. So for a clipped signal y and its estimation $\hat{y}$, SDR is computed as:

$$\mathrm{SDR}_c(\mathbf{y}, \hat{\mathbf{y}}) = 20 \log_{10} \frac{\|\mathrm{M}^c y\|_2}{\|\mathrm{M}^c (y - \hat{y})\|_2}$$

Then the difference between the SDR of the restored and the clipped signal is defined as:

$$\Delta \mathrm{SDR_c} = \mathrm{SDR_c(y, \hat{y})} - \mathrm{SDR_c(y, s)}$$

According to the bar graphs 6 the Social Sparsity with Persistent Empirical Wiener method is the most efficient in terms of SDR. For a more detailed comparison check out [15].

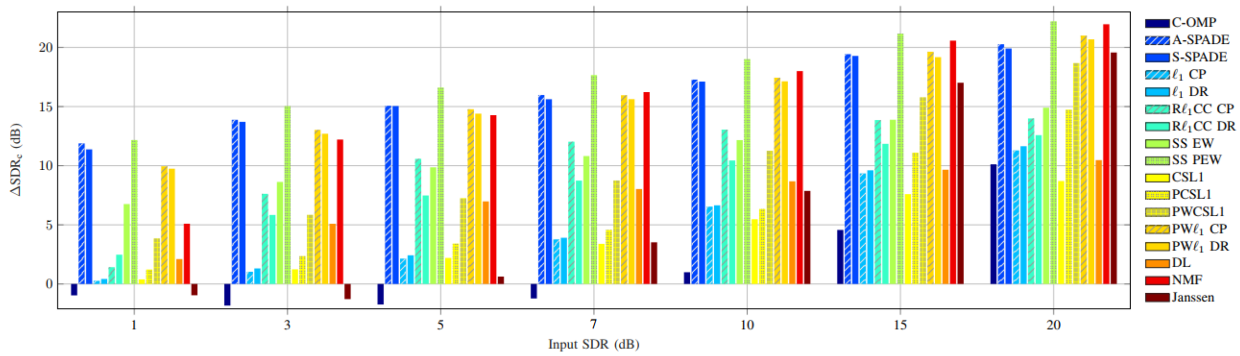

Figure 6: Average ΔSDR_c results. [15]

The audio declipping toolbox is used to declipped some audio:

Social Sparsity with Empirical Wiener method:

The Social Sparsity with Empirical Wiener method is tested on a "a08_violin" audio with the following settings (1). the figure 7 show the result of the declipping.

Output of SS EW

- Result obtained in 1357.491 seconds.
- SDR of the clipped signal is 7.000 dB.
- SDR of the reconstructed signal is 14.867 dB.
- SDR improvement is 7.867 dB.

Listing 1: settings

```
% input SDR of the clipped signal
inputSDR = 7;      % set the input SDR value

% DGT parameters
wtype = 'hann';   % window type
w = 8192;         % window length
a = w / 4;        % window shift
M = 2*8192;       % number of frequency channels

% set shrinkage operator
shrinkage = 'EW'; % 'L', 'WGL', 'EW', 'PEW';
number_lambdas = 20;
inner_iterations = 500;
```

Social Sparsity with Persistent Empirical Wiener method:

the figure 8 show the result of the declipping.

Output of SS PEW

- Result obtained in 1581.076 seconds.

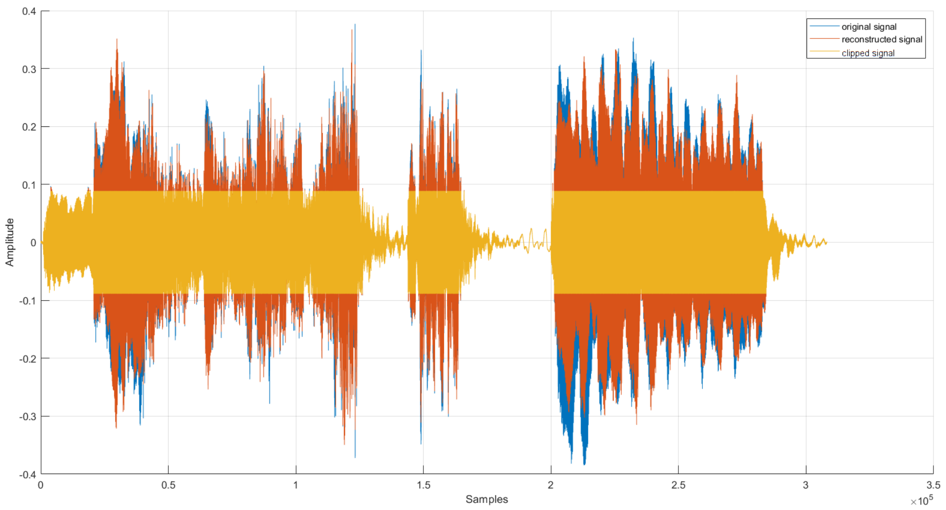

Figure 7: Audio declipping Social according to Social Sparsity with Empirical Wiener

- SDR of the clipped signal is 7.000 dB.
- SDR of the reconstructed signal is 19.702 dB.
- SDR improvement is 12.702 dB.

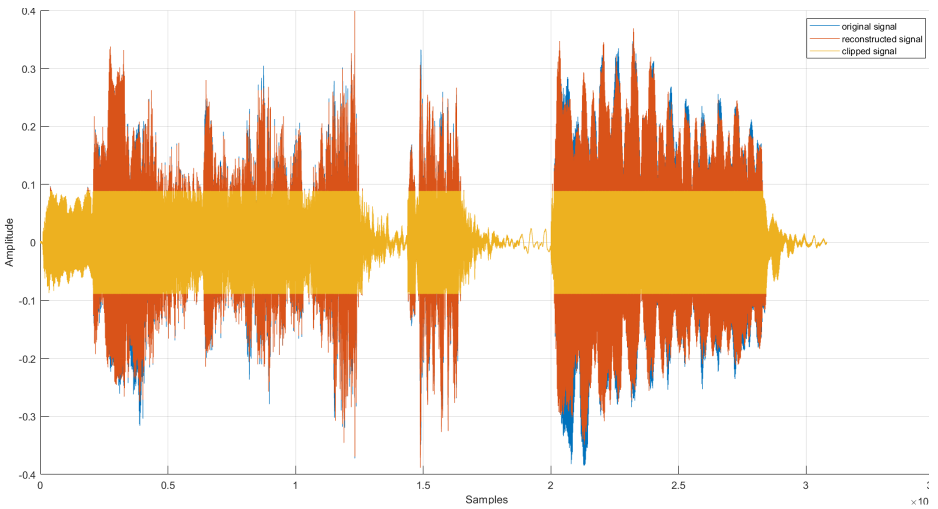

Figure 8: Audio declipping Social according to Social Sparsity with Empirical Wiener

4 Conclusion

Several techniques have been developed in order to attempt the reversal of a clipped signal. The article [5] provides a new approach that outperforms other methods in terms of SDR according to [15]. More precisely The Persistent Empirical Wiener (PEW) operator is used for audio declipping using synthesis social sparsity.

References

[1] A. Adler et al. “A constrained matching pursuit approach to audio declipping”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 329–332. DOI: 10.1109/ICASSP.2011.5946407.

[2] A. Beck and M. Teboulle. “A fast Iterative Shrinkage-Thresholding Algorithm with application to wavelet-based image deblurring”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009, pp. 693–696. DOI: 10.1109/ICASSP.2009.4959678.

[3] Ç. Bilen, A. Ozerov, and P. Pérez. “Audio declipping via nonnegative matrix factorization”. In: *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2015, pp. 1–5. DOI: 10.1109/WASPAA.2015.7336948.

[4] B. Defraene et al. “Declipping of Audio Signals Using Perceptual Compressed Sensing”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.12 (2013), pp. 2627–2637. DOI: 10.1109/TASL.2013.2281570.

[5] Clément Gaultier. “Design and evaluation of sparse models and algorithms for audio inverse problems”. Theses. Université Rennes 1, Jan. 2019. URL: https://tel.archives-ouvertes.fr/tel-02148598.

[6] A. Janssen, R. Veldhuis, and L. Vries. “Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.2 (1986), pp. 317–330. DOI: 10.1109/TASSP.1986.1164824.

[7] Srdan Kitic, Nancy Bertin, and Rémi Gribonval. “Sparsity and cosparsity for audio declipping: a flexible non-convex approach”. In: *CoRR* abs/1506.01830 (2015). arXiv: 1506.01830. URL: http://arxiv.org/abs/1506.01830.

[8] Kowalski Matthieu. “Time-frequency frames and applications to audio analysis - Part 2: time-frequency coefficients modelling. CIRM.” In: 2014. ISBN: CIRM. DOI: 10.24350/CIRM.V.18614903.

[9] Pavel Rajmic et al. “A New Generalized Projection and Its Application to Acceleration of Audio Declipping”. In: *Axioms* 8 (Sept. 2019), p. 105. DOI: 10.3390/axioms8030105.

[10] Lucas Rencker et al. “Consistent Dictionary Learning for Signal Declipping”. In: *Latent Variable Analysis and Signal Separation*. Ed. by Yannick Deville et al. Cham: Springer International Publishing, 2018, pp. 446–455. ISBN: 978-3-319-93764-9.

[11] Kai Siedenburg, Matthieu Kowalski, and Monika Doerfler. “Audio declipping with social sparsity”. In: May 2014, pp. 1577–1581. ISBN: 978-1-4799-2893-4. DOI: 10.1109/ICASSP.2014.6853863.

[12] Alejandro J. Weinstein and Michael B. Wakin. “Recovering a Clipped Signal in Sparseland”. In: *CoRR* abs/1110.5063 (2011). arXiv: 1110.5063. URL: http://arxiv.org/abs/1110.5063.

[13] Pavel Zaviska, Pavel Rajmic, and Jiri Schimmel. “Psychoacoustically Motivated Audio Declipping Based on Weighted l1 Minimization”. In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (July 2019). DOI: 10.1109/tsp.2019.8769109. URL: http://dx.doi.org/10.1109/TSP.2019.8769109.

[14] P. Záviška et al. “A Proper Version of Synthesis-based Sparse Audio Declipper”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 591–595. DOI: 10.1109/ICASSP.2019.8682348.

[15] Pavel Záviška et al. *A survey and an extensive evaluation of popular audio declipping methods*. 2020. arXiv: 2007.07663 [eess.AS].