FAST RECONSTRUCTION OF SPARSE RELATIVE IMPULSE RESPONSES VIA SECOND-ORDER CONE PROGRAMMING

Pavel Rajmic,[1*] *Zbyněk Koldovský,*[2 †] *Marie Daňková*[1]

[1] Signal Processing Laboratory, Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic, rajmic@vutbr.cz

[2] Acoustic Signal Analysis and Processing Group, Technical University of Liberec, Studentská 1402/2, 461 17 Liberec, Czech Republic, zbynek.koldovsky@tul.cz

ABSTRACT

The paper addresses the estimation of the relative transfer function (RTF) using incomplete information. For example, an RTF estimate might be recognized as too inaccurate in a number of frequency bins. When these values are dropped, an incomplete RTF is obtained. The goal is then to reconstruct a complete RTF estimate, based on (1) the remaining values, and (2) the sparsity of the relative impulse response, which is the time-domain counterpart of the RTF. We propose two fast algorithms for the RTF reconstruction that solve a second-order cone program (SOCP), and show their advantages over the LASSO formulation previously proposed in the literature. Simulations with speech signals show that in terms of speed and accuracy, the proposed algorithms are comparable with the LASSO solution and considerably faster compared to the generic ECOS solver. The new algorithms are, moreover, easier to control through their parameters, which brings their improved stability when the number of reliable frequency bins is very low (less than 10%).

***Index Terms*—** Beamforming, Relative Transfer Function, Relative Impulse Response, Sparsity, Proximal Algorithms, Convex Programming

1. INTRODUCTION

A multi-microphone noisy recording of a directional source can be modeled in the short-term Fourier transform (STFT) domain as

$$\mathbf{X}(k,\ell) = \mathbf{H}(k)\, S(k,\ell) + \mathbf{Y}(k,\ell), \tag{1}$$

where k and ℓ are the frequency and the time-frame indexes, respectively. While $\mathbf{X}$, $\mathbf{H}$ and $\mathbf{Y}$ can be considered three-way arrays, for fixed k and ℓ, the vector $\mathbf{X}(k,\ell)$ of size $M \times 1$ denotes the STFT coefficients of signals acquired with the M microphones, and the scalar $S(k,\ell)$ represents the STFT of the target signal from the first (reference) microphone. Elements of $\mathbf{H}(k)$ correspond to the individual relative transfer functions (RTFs) with respect to the first microphone. For a fixed k, this implies that $\mathbf{H}(k)$ is of size $M \times 1$ and its first element is unity. Its other elements depend on the acoustic transfer functions between the source and the microphones, and, therefore, $\mathbf{H}(k)$ is mainly determined by the environment and by the position of the source and microphones. Notice that $\mathbf{H}(k)$ does not depend on ℓ, meaning that the source is stationary (or at least approximately stationary for a short time interval). Finally, $\mathbf{Y}(k,\ell)$ involves noise and interferences in the mixture.

The estimation of $\mathbf{H}(k)$ is a central issue in multi-microphone noise reduction and signal enhancement. Once these RTFs are known, efficient processors such as the Minimum Variance Distortionless Responses (MVDR) beamformer [1] can be applied in order to obtain a noise-free estimate of $S(k,\ell)$. Many methods have been proposed to estimate $\mathbf{H}(k)$ directly from $\mathbf{X}(k,\ell)$. In particular, an unbiased estimator exploiting the nonstationarity of $S(k,\ell)$ and assuming the stationarity of $\mathbf{Y}(k,\ell)$ has been proposed in [2] and improved for speech signals in [3]. These estimators are computationally simple, but their applicability is limited due to the assumption imposed on $\mathbf{Y}(k,\ell)$. For more general conditions, methods based on Blind Source Separation (BSS) were proposed; see, e.g., [4, 5, 6]. They are computationally more expensive, nevertheless, their performance stays limited [7].

Recently, a general approach that aims at improving any RTF estimator has been proposed in [8]. It is based on the assumption that the RTF can be well approximated in the time-domain by a sparse relative impulse response (ReIR). As this sparse ReIR poses a low-rank representation of the RTF, it is possible to estimate it from an incomplete observation of RTF (iRTF), i.e., from the RTF values known only on a constrained set of frequencies. In the following, let this set of frequencies be denoted $\mathcal{S}$.

The idea of using iRTF is advantageous from several viewpoints. Some signals (especially speech signals) are sparse in the (short-time) frequency domain, and therefore the signal-to-noise ratio is highly variable over the frequencies. In particular, there are frequency bins in which the target signal activity is negligible, thus making any estimate of the RTF meaningless (i.e. too inaccurate). Avoiding estimation within these "unreliable" frequencies conveniently corresponds to the concept of iRTF. If any additional information about the reliability/accuracy of the RTF estimate across frequencies is available, it is possible to decide whether a given frequency should or should not be included in $\mathcal{S}$. For example, such information could be obtained through a theoretical error prediction [9] or using a model-based or trained voice activity (speech mask) detectors [10, 11].

The paper addresses the key step, which is the completion of the iRTF. In [8], it has been suggested that convex programming formulations should be exploited, specifically the weighted LASSO [12], where the weights help controlling the sparsity of the solution (the ReIR estimate). However, selecting the weights is a difficult problem as the effect of the weights on the solution cannot be determined

*The work of P. Rajmic and M. Daňková was supported by the joint project of the FWF and the Czech Science Foundation (GAČR): numbers I 3067-N30 and 17-33798L, respectively. Research described in this paper was financed by the National Sustainability Program under grant LO1401. Infrastructure of the SIX Center was used.

†The work of Z. Koldovský was supported by The Czech Science Foundation through Project No. 17-00902S and partly by the California Community Foundation through Project No. DA-15-114599.

analytically. Therefore, [9] later proposed to exploit second-order cone programming (SOCP), whose parametrization has a straightforward interpretation.

However, the computational complexity of the general embedded conic solver (ECOS) from [13], used for solving SOCP in [9], is very high compared to the SpaRIR algorithm proposed in [8] for weighted LASSO. In the present paper, we therefore derive two proximal algorithms solving the same SOCP problem as ECOS, having a complexity comparable to that of SpaRIR. Moreover, the computational complexity of the new algorithms is virtually independent of the number of elements in $\mathcal{S}$. It is thus possible to recast the problem of selecting $\mathcal{S}$, which is a combinatorial problem, as the selection of the SOCP parameters, which is a continuous problem (the details will be presented below Eq. (3)).

The following section presents the original SpaRIR and SOCP formulations from [9]. Section 3 is devoted to the solution of the SOCP program, using fast proximal algorithms. Section 4 presents experiments with speech signals.

2. PROBLEM FORMULATION

For microphone m, let the respective relative impulse responses, i.e. the time-domain counterpart of $\mathbf{H}(k)$, be denoted $\mathbf{h}_m$. It is a real-valued vector; suppose that its length is N and that N is even. Since $\mathbf{h}_m$ can be estimated independently of the other ReIRs, we drop the index without loss of generality and consider a single $\mathbf{h}$ in the following.

The estimate of $\mathbf{H}(k)$, denoted $\widehat{\mathbf{H}}(k)$, which alone is available in practice, is not equally reliable over the frequencies k. Sufficiently accurate elements of $\widehat{\mathbf{H}}(k)$ correspond to the choice of $\mathcal{S}$. This set of frequencies is restricted to $\mathcal{S} \subseteq \{0, \dots, N/2\}$ due to the spectrum conjugate symmetry. The operator of the Discrete Fourier Transform (DFT), denoted F, is assumed to be unitary. The operator $F_{\mathcal{S}} : \mathbb{C}^N \to \mathbb{C}^{|\mathcal{S}|}$ is a subsampled DFT, and it outputs spectral values belonging solely to the indexes $\mathcal{S}$. $F_{\mathcal{S}}$ could be seen as a DFT matrix where rows not indexed in $\mathcal{S}$ are omitted. L^* is the notation for the adjoint to a linear operator L.

An entry within a vector, say $\mathbf{x}$, at position $n = 0, \dots, N-1$ will be referred to as x_n or $[\mathbf{x}]_n$. The complex conjugate of $x \in \mathbb{C}$ will be denoted $\overline{x}$.

2.1. Weighted LASSO formulation

The method in [8] aims to find the sparsest representation of the incomplete RTF in the time domain using weighted LASSO [12, 14]. The reconstructed ReIR is sought as the solution to

$$\arg\min_{\mathbf{h}} \|F_{\mathcal{S}}\mathbf{h} - \boldsymbol{\mu}\|_2^2 + \|\mathbf{w} \odot \mathbf{h}\|_1, \tag{2}$$

where $\boldsymbol{\mu}$ is an $|\mathcal{S}| \times 1$ complex vector representing the iRTF, with elements $\mu_k = \hat{\mathbf{H}}(k)$, $k \in \mathcal{S}$; $\mathbf{w}$ is a vector of nonnegative weights, multiplied elementwise by the unknown $\mathbf{h}$. This problem can be interpreted as finding a sparse impulse response, which is penalized when its Fourier transform moves away from the prescribed spectral values. Moreover, the time-domain sparsity is affected by weighting. As a surrogate of the true sparsity measure we use the ℓ_1-norm, see [15], for example.

There is a number of fast algorithms for solving such a problem, for example proximal algorithms, which are discussed in Sec. 3. A disadvantage of LASSO formulation lies in the difficulty of describing the impact of the weights $\mathbf{w}$ on the solution.

2.2. SOCP formulation

This formulation is derived from a different point of view. A sparse vector $\mathbf{h}$ is sought such that at the frequencies in $\mathcal{S}$, the spectrum of $\mathbf{h}$ is not far away from the prescribed complex values μ_k, $k \in \mathcal{S}$:

$$\arg\min_{\mathbf{h}} \|\mathbf{h}\|_1 \quad \text{s.t. } \left|[F\mathbf{h}]_k - \mu_k\right| \le \epsilon_k,\ k \in \mathcal{S}, \text{ and } \mathbf{h} \in \mathbb{R}^N. \tag{3}$$

The parameters ϵ_n determine maximal errors of the spectral coefficients of $\mathbf{h}$. Both moduli and phases are taken into account. From a certain point of view, it is also possible to say $k \in \{0, \dots, N/2\}$ and to set ϵ_k for all unreliable frequencies $k \notin \mathcal{S}$ very high.

Recall that N is assumed to be even. Then, due to the properties of the DFT, the constraint $\mathbf{h} \in \mathbb{R}^N$ is equivalent to saying that

$$[F\mathbf{h}]_k = \overline{[F\mathbf{h}]_{N-k}} \quad \text{for } k = 1, \dots, N/2 - 1, \tag{4}$$

$$[F\mathbf{h}]_0 \in \mathbb{R},\ [F\mathbf{h}]_{N/2} \in \mathbb{R}, \tag{5}$$

the bar denoting the complex conjugate. Therefore, problem (3) is equivalent to the following unconstrained one:

$$\arg\min_{\mathbf{h}} \|\mathbf{h}\|_1 + \chi_C(\mathbf{h}), \tag{6}$$

where χ_C is the indicator function of the convex set C, defined as

$$C = \{\mathbf{z} \mid \left|[F\mathbf{z}]_k - \mu_k\right| \le \epsilon_k, k \in \mathcal{S}, \text{ and (4), (5) hold}\}. \tag{7}$$

In order to be able to find efficient algorithms, it will be convenient to develop a new form of the same problem, which reads

$$\arg\min_{\mathbf{h}} \|\mathbf{h}\|_1 + \chi_{C'}(F_{\mathcal{S}}\mathbf{h}), \tag{8}$$

with C' defined as

$$\begin{aligned} C' = \{\mathbf{x} \mid\ & |x_k - \mu_k| \le \epsilon_k,\ k \in \mathcal{S}, \\ & x_k = \overline{x_{N-k}} \text{ for } k = 1, \dots, N/2 - 1,\ x_0, x_{\frac{N}{2}} \in \mathbb{R}\}. \end{aligned} \tag{9}$$

Note that problem (3) is a special case of a large class of second-order cone programs. In fact, it falls into a subclass termed "convex quadratically constrained linear programs".

3. PROBLEM SOLUTION

Problem (8) is convex since C' is a convex set, and both $\|\cdot\|_1$ and $\chi_{C'}$ are convex functions. For finding the minimizer of a sum of convex functions, proximal algorithms [16, 17] are a popular choice. We first present some necessary ingredients that will be needed to adapt proximal algorithms to our problem.

3.1. Proximal operators

The proximal algorithms are iterative. They rely on evaluating the so-called proximal operator of individual convex functions in each iteration. The proximal operator of f is a mapping $\mathrm{prox}_f : \mathbb{R}^N \to \mathbb{R}^N$ with many essential properties [18, 16]. In particular, we will later make use of two cases, namely

$$\mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{x}) = \mathrm{soft}_\lambda(\mathbf{x}) \tag{10}$$

i.e. the soft thresholding [19, 14, 20] with threshold λ, mapping elementwise $x_n \to \mathrm{sgn}(x_n) \cdot \max(|x_n| - \lambda, 0)$, and

$$\mathrm{prox}_{\chi_C}(\mathbf{x}) = \mathrm{proj}_C(\mathbf{x}), \tag{11}$$

i.e. the orthogonal projection onto a convex set C.

Furthermore, there is a special rule if f is a composite function involving a linear operator L with the property that LL^* is just a multiple of the identity:

Lemma 1 (see [16], Table 10.1.x for the real case)**.** *If $f = g \circ L$ is a composition of a convex function g with a linear operator L such that $LL^* = \nu \cdot Id$ with $\nu > 0$, it holds*

$$\mathrm{prox}_f(\mathbf{x}) = \mathbf{x} + \nu^{-1} L^* \left(\mathrm{prox}_{\nu g}(L\mathbf{x}) - L\mathbf{x}\right). \quad (12)$$

Now, denote $B_2(\mu, \epsilon)$ the ball of radius $\epsilon > 0$ centered at $\mu \in \mathbb{C}$, i.e. $B_2(\mu, \epsilon)$ is the set of complex points at most ϵ far from μ. It is not too difficult to see that

Lemma 2. *Projection of $x \in \mathbb{C}$ onto the ball $B_2(\mu, \epsilon)$ is*

$$\mathrm{proj}_{B_2(\mu,\epsilon)}(x) = \frac{\epsilon(x-\mu)}{\max(|x-\mu|, \epsilon)} + \mu. \quad (13)$$

As the last ingredient, let the proximal operator $\mathrm{prox}_{\chi_{C'}}$ be identified, where C' has been defined in (9). According to (11), it is the projection of an $\mathbf{x} \in \mathbb{C}$, formally

$$\mathrm{proj}_{C'}(\mathbf{x}) = \underset{\mathbf{w} \in \mathbb{C}^N}{\arg\min} \|\mathbf{w} - \mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{w} \in C'. \quad (14)$$

As will become clear later, the input of the projection onto C' will always be a complex conjugate vector, i.e. a vector $\mathbf{x}$ that satisfies the conditions set out on the second line of (9). Together with the fact that the quadratic functional in (14) is separable (up to pairs of indexes $[k, N-k]$ for $k = 1, \ldots, N/2-1$) this results in the fact that the projection is implemented elementwise:

$$[\mathrm{proj}_{C'}(\mathbf{x})]_k = \begin{cases} \mathrm{proj}_{B_2(\mu_k,\epsilon_k)}(x_k) & \text{for } k \in \mathcal{S} \\ x_k & \text{for } k \notin \mathcal{S}. \end{cases} \quad (15)$$

3.2. Proximal algorithms

We utilize two numerical solvers, namely the Douglas-Rachford (DR) algorithm and the Chambolle-Pock (CP) algorithm.

The DR algorithm [16] was designed for optimization problems that do not involve a linear operator. Therefore, formulation (6) is the right basepoint for DR; nevertheless, writing $\chi_C = \chi_{C'} \circ F_{\mathcal{S}}$ and using the fact that $F_{\mathcal{S}} F_{\mathcal{S}}^* = Id$, we find that Lemma 1 with $\nu = 1$ can be applied to obtain

$$\mathrm{prox}_{\chi_C}(\mathbf{x}) = \mathrm{proj}_C(\mathbf{x}) = \mathbf{x} + F_{\mathcal{S}}^* \left(\mathrm{proj}_{C'}(F_{\mathcal{S}}\mathbf{x}) - F_{\mathcal{S}}\mathbf{x}\right). \quad (16)$$

The DR algorithm is given in Alg. 1. The DR algorithm always converges; the single parameter γ is responsible for the convergence speed. In our scenario, γ plays the role of the threshold for soft thresholding. The iterations are terminated if a convergence criterion is met. If convergence is not fully achieved, an additional projection step can be appended right after the last loop of the algorithm: The soft thresholding is responsible for sparsifying the vector, which can lead to violating the spectral restriction. The natural final step is thus projecting the frequencies in $\mathcal{S}$ onto the constraints and applying the inverse DFT.

The CP algorithm [21] is a primal-dual algorithm developed to solve problems where one of the functions is composed with a linear operator. This perfectly corresponds to the formulation in (8). The steps of the CP algorithm are formally given in Alg. 2. The convergence is guaranteed whenever $\zeta\sigma\|F_{\mathcal{S}}\| = \zeta\sigma < 1$, while the actual values of the respective parameters influence the speed of convergence.

Algorithm 1: Douglas-Rachford algorithm solving (8)

Input: Starting point $\mathbf{y}^{(0)} \in \mathbb{R}^N$, $\lambda = 1$ and $\gamma > 0$
for $i = 0, 1, \ldots$ **do**
$\mathbf{u}^{(n)} = F_{\mathcal{S}}\, \mathbf{y}^{(n)}$
$\mathbf{x}^{(n)} = \mathbf{y}^{(n)} + F_{\mathcal{S}}^* \left(\mathrm{proj}_{C'}(\mathbf{u}^{(n)}) - \mathbf{u}^{(n)}\right)$
$\mathbf{y}^{(n+1)} = \mathbf{y}^{(n)} + \lambda \left(\mathrm{soft}_\gamma(2\mathbf{x}^{(n)} - \mathbf{y}^{(n)}) - \mathbf{x}^{(n)}\right)$
return $\mathbf{y}^{(n+1)}$

Algorithm 2: Chambolle-Pock algorithm solving (8)

Input: Starting primal point $\mathbf{p}^{(0)} \in \mathbb{R}^N$ and dual point $\mathbf{q}^{(0)} \in \mathbb{C}^{|\mathcal{S}|}$, parameters $\zeta, \sigma > 0$ and $\theta \in [0, 1]$
Set $\dot{\mathbf{p}}^{(0)} = \mathbf{p}^{(0)}$
for $i = 0, 1, \ldots$ **do**
$\mathbf{u}^{(n)} = \mathbf{q}^{(n)} + \sigma F_{\mathcal{S}} \dot{\mathbf{p}}^{(n)}$
$\mathbf{q}^{(n+1)} = \mathbf{u}^{(n)} - \mathrm{proj}_{C'}(\mathbf{u}^{(n)}/\sigma)$
$\mathbf{p}^{(n+1)} = \mathrm{soft}_\zeta(\mathbf{p}^{(n)} - \zeta F_{\mathcal{S}}^* \mathbf{q}^{(n+1)})$
$\dot{\mathbf{p}}^{(n+1)} = \mathbf{p}^{(n+1)} + \theta(\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)})$
return $\dot{\mathbf{p}}^{(n)}$

Both algorithms are comparable in terms of the computational cost per iteration. When addition of vectors and multiplication by scalar is neglected, both CP and DR perform one $\mathrm{proj}_{C'}$, one soft, one $F_{\mathcal{S}}$ and one $F_{\mathcal{S}}^*$ in each iteration. Despite the utilization of the FFT in place of the DFT, the application of $F_{\mathcal{S}}$ and $F_{\mathcal{S}}^*$ with $\mathcal{O}(N \log_2 N)$ complexity dominates the computational cost.

One can observe a slight increase in computational time as the size of $\mathcal{S}$ increases: The soft thresholding always operates on a full-length vector, and $F_{\mathcal{S}}$ and $F_{\mathcal{S}}^*$ are implemented using full FFT and IFFT, respectively. Hence, the only factor that influences the speed in the mentioned respect is the projection onto C' — the larger the set $\mathcal{S}$, the more projections with (13) are computed.

In case that the number of reliable frequencies is very low, using the pruned FFT [22] or the (easily paralellizable) Goertzel algorithm [23, 24, 25] may be favorable over the full FFT.

Returning back to problem (2), we can see that the first term there is differentiable. This allows solving the problem using SpaRIR, an algorithm based on the forward-backward proximal scheme [16]. SpaRIR requires application of the operators $F_{\mathcal{S}}$ and $F_{\mathcal{S}}^*$ in the forward step (using the FFT), and a weighted soft thresholding in the backward step. This makes SpaRIR slightly faster than the CP and DR algorithms, as will be shown by the experiment.

4. EXPERIMENTS

We restrict ourselves to the case of $M = 2$ microphones for simplicity, nevertheless, the generalization to a higher number of microphones is straightforward.

As the target signal, we use a 10 seconds long female utterance from SiSEC 2013[1] from the task "Two-channel mixtures of speech and real-world background noise". The spatial image of the

[1] http://sisec.wiki.irisa.fr/

target is generated by convolving the signal with real-world room impulse responses (RIR) from [26] available online[2]. The two microphones are placed such that their mutual distance is 3 cm. Similarly, noise is generated by convolving Gaussian white noise with RIRs corresponding to a different position. This simulates a directional noise such as a fan noise. The reverberation time T_{60} is 610 ms; the source–microphone distance is 2 m. The target and the noise source are located, respectively, in the direction of 0° and 75° on the left-hand side; the sampling frequency is 16 kHz.

The data is generated in 100 trials. In each trial, a random interval of the target signal of T seconds in length is mixed with a newly generated noise at 0 dB signal-to-noise ratio (SNR); the SNR is averaged over both microphones. After transforming the mixed signals into the STFT domain (FFT length 1024 and hop-size 256), the spectra are analyzed, and the set $\mathcal{S}$ is selected. Two methods are considered in this regard: the oracle approach exploiting the known SNR within each frequency and the non-oracle method based on the kurtosis of the frequency components [8]. $\mathcal{S}$ is selected such as to contain p percent of the frequencies with the highest SNR/kurtosis. Both variants aim to select frequencies where the speech is dominant.

Next, the nonstationarity-based RTF estimator from [2] is used to estimate the RTF between the microphones; let the estimate be denoted $\widehat{\mathbf{H}}$. The iRTF is obtained by taking only the subvector of $\widehat{\mathbf{H}}$ whose elements are in $\mathcal{S}$. Then, LASSO and SOCP formulations are used to reconstruct the iRTF. In LASSO, the weights are selected as recommended in [8], while in SOCP the theoretical variance of the estimator is used to select ϵ_k for $k \in \mathcal{S}$ [9]. For any $k \notin \mathcal{S}$, we choose a practical approach to set $\epsilon_k = 100$, which is a sufficiently high value that guarantees that the value $\widehat{\mathbf{H}}(k)$ does not influence the reconstruction of iRTF.

Let the reconstructed RTF be denoted $\mathbf{G}$. We evaluate $\mathbf{G}$ by measuring SNR when it is used to block the target signal. Specifically, the output of the blocking is

$$V(k,\ell) = \mathbf{G}(k)X_1(k,\ell) - X_2(k,\ell). \tag{17}$$

If $\mathbf{G}$ were the exact RTF between the two channels, $V(k,\ell)$ would contain only the noise signal, but with the spectrum modified by an unknown filter. To avoid the influence of that filter on the output SNR, we apply least squares: The spectrum of $V(k,\ell)$ is modified to be as close to the spectrum of the true noise image on a given microphone as possible. Projections on both microphones are considered. The SNR is afterwards evaluated and averaged.

Fig. 1 shows the results obtained using weighted LASSO (computed by SpaRIR from [8]) and SOCP in combination with the two variants of selecting $\mathcal{S}$; the SNR as a function of p is depicted. For $p = 100\%$, all approaches give the same result (except for small deviations) corresponding to the original RTF estimator. For $p < 100\%$, the results correspond to the RTF completion from the iRTF (using p percent of frequencies).

LASSO and SOCP give approximately the same results for $p > 10\%$ with both selection methods. The results with the oracle selection are obviously better than those with the kurtosis-based method for most values of p. An optimum percentage appears to be within the interval from $p = 15$ to $p = 20$ for both selectors. These values are in good agreement with a typical number of active frequencies of a speech signal at short time intervals. The results here confirm the general claims of [8]: With the reconstructed iRTF

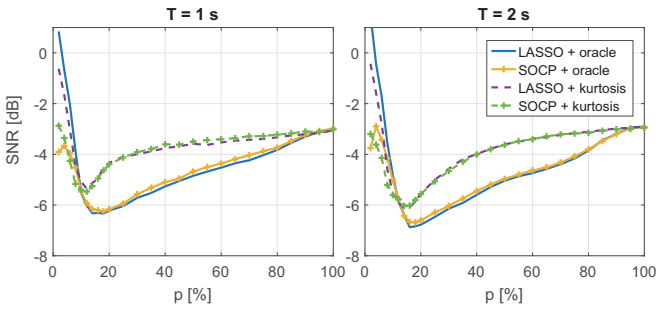

Figure 1: Average SNR at the output of the target signal blocking (smaller values are better) obtained in 100 trials on 1 s (left) and 2 s intervals (right).

(using frequencies with high SNR) it is possible to achieve better target signal blocking than with the complete RTF estimate.

The results for $p < 10\%$ point to an advantage of SOCP over LASSO. While the SNR of LASSO grows when p approaches zero (i.e. target blocking becomes poor), SOCP gives stable performance also for small p. This is achieved thanks to constraining deviations from the selected spectral values in SOCP (while LASSO is unconstrained in this respect).

The solutions obtained using the Chambolle-Pock and Douglas-Rachford algorithms (both with a fixed number of 600 iterations) as well as by using the ECOS package [13] were identical (and are therefore not compared in Fig. 1). The computational time per trial (averaged over all the values of p in consideration) was 0.17 s, 0.13 s, and 20.3 s, respectively, meaning that ECOS is considerably slower for the particular optimization problem solved here than the methods proposed in this paper. The average computational time by SpaRIR was 0.06 s; however, SpaRIR automatically stops after a variable number of iterations. The average time *per iteration* of the CP, DR, and SpaRIR, was 0.14 ms, 0.11 ms, and 0.12 ms, respectively.

Software. The experiments were performed in Matlab R2016b on a PC with 2.6 GHz Intel i7 CPU and 8 GB RAM. An interested reader can download the Matlab files from the GitHub repository.[3] The folder `synthetic` provides a demo showing the CP and DR algorithms recovering a synthetic sparse signal. The demo generates a random problem in each run. The folder `speech` reproduces experiment with the speech signal, as presented in this paper.

5. CONCLUSION

The main message of the paper is that the reconstruction of iRTF through LASSO can be replaced with a reconstruction through SOCP, using fast proximal algorithms such as the Chambolle-Pock or the Douglas-Rachford algorithm. SOCP performs comparably with LASSO. In addition, SOCP is stable as compared to LASSO when only a small number of frequencies (less than 10%) are used for the RTF estimation. The other advantage of the solution proposed here is that, in SOCP, the (discrete) problem of the selection of $\mathcal{S}$ can be recast to the (continuous) problem of selection of ϵ_k for all k. The computational complexity per iteration of the Chambolle-Pock or the Douglas-Rachford algorithms is virtually independent of the size of $\mathcal{S}$ in contrast to the case of the ECOS solver.

[2] `http://www.eng.biu.ac.il/gannot/downloads/`

[3] `https://github.com/rajmic/sparse-ReIR-proximal`

6. REFERENCES

[1] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1614–1626, Aug 2001.

[2] O. Shalvi and E. Weinstein, "System identification using nonstationary signals," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 2055–2063, Aug 1996.

[3] I. Cohen, "Relative transfer function identification using speech signals," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 451–459, Sept 2004.

[4] W. Kellermann, H. Buchner, and R. Aichner, "Separating convolutive mixtures with TRINICON," vol. V, May 2006, pp. 961–964.

[5] F. Nesta, P. Svaizer, and M. Omologo, "Convolutive bss of short mixtures by ICA recursively regularized across frequencies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 624–639, March 2011.

[6] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano, "Blind source separation combining independent component analysis and beamforming," vol. 2003, no. 11, pp. 1135–1146, Nov. 2003.

[7] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 2, pp. 109–116, 2003.

[8] Z. Koldovský, J. Málek, and S. Gannot, "Spatial source subtraction based on incomplete measurements of relative transfer function," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1335–1347, Aug 2015.

[9] Z. Koldovský, J. Málek, and P. Tichavský, *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science. Springer, 2015, ch. Improving relative transfer function estimates using second-order cone programming, pp. 227–234.

[10] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 196–200.

[11] S. Araki, M. Okada, T. Higuchi, A. Ogawa, and T. Nakatani, "Spatial correlation model based observation vector clustering and mvdr beamforming for meeting recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 385–389.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[13] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *2013 European Control Conference (ECC)*, July 2013, pp. 3071–3076.

[14] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[15] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proceedings of The National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[16] P. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212, 2011.

[17] N. Komodakis and J. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, Nov 2015.

[18] J. J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société Mathématique de France*, no. 93, pp. 273–299, 1965.

[19] D. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613 –627, 1995.

[20] P. Rajmic, "Exact risk analysis of wavelet spectrum thresholding rules," in *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on*, vol. 2, 12 2003, pp. 455–458 Vol.2.

[21] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[22] R. Stasinski, "FFT pruning – a new approach," in *EUSIPCO-86 Signal procesing III: Theories and applications*, vol. 1. North-Holland, 1986, pp. 267–270. [Online]. Available: http://www.eurasip.org/Proceedings/Eusipco/Eusipco1986/EUSIPCO86_VolI.pdf

[23] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American Mathematical Monthly*, vol. 65, no. 1, pp. 34–35, 1958.

[24] P. Sysel and P. Rajmic, "Goertzel algorithm generalized to non-integer multiples of fundamental frequency," *EURASIP Journal on Advances in Signal Processing*, vol. 56, 3 2012. [Online]. Available: http://asp.eurasipjournals.com/content/2012/1/56/abstract

[25] P. Rajmic, Z. Prusa, and C. Wiesmeyr, "Computational cost of chirp Z-transform and generalized Goertzel algorithm," in *EUSIPCO 2014 (22nd European Signal Processing Conference 2014) (EUSIPCO 2014)*, Lisbon, Portugal, Sept. 2014.

[26] E. Hadad, F. Heese, P. Vary, and S. Gannot, "Multichannel audio database in various acoustic environments," in *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sept 2014, pp. 313–317.