Interspeech 2018
2-6 September 2018, Hyderabad

Reducing Interference with Phase Recovery in DNN-based Monaural Singing Voice Separation

Paul Magron[1], *Konstantinos Drossos*[1], *Stylianos Ioannis Mimilakis*[2], *Tuomas Virtanen*[1]

[1]Laboratory of Signal Processing, Tampere University of Technology, Finland
[2]Fraunhofer IDMT, Ilmenau, Germany

paul.magron@tut.fi, konstantinos.drossos@tut.fi, mis@idmt.fhg.de, tuomas.virtanen@tut.fi

Abstract

State-of-the-art methods for monaural singing voice separation consist in estimating the magnitude spectrum of the voice in the short-time Fourier transform (STFT) domain by means of deep neural networks (DNNs). The resulting magnitude estimate is then combined with the mixture's phase to retrieve the complex-valued STFT of the voice, which is further synthesized into a time-domain signal. However, when the sources overlap in time and frequency, the STFT phase of the voice differs from the mixture's phase, which results in interference and artifacts in the estimated signals. In this paper, we investigate on recent phase recovery algorithms that tackle this issue and can further enhance the separation quality. These algorithms exploit phase constraints that originate from a sinusoidal model or from consistency, a property that is a direct consequence of the STFT redundancy. Experiments conducted on real music songs show that those algorithms are efficient for reducing interference in the estimated voice compared to the baseline approach.

Index Terms: Monaural singing voice separation, phase recovery, deep neural networks, MaD TwinNet, Wiener filtering

1. Introduction

Audio source separation [1] consists in extracting the underlying *sources* that add up to form an observable audio *mixture*. In particular, monaural singing voice separation aims at predicting the singing voice from a single channel music mixture signal. To address this issue, it is common to act on a time-frequency (TF) representation of the data, such as the short-time Fourier transform (STFT), since the structure of music is more prominent in that domain.

A typical source separation work flow is depicted in Fig. 1. First, from the complex-valued STFT of the mixture $\mathbf{X}$, one extract a nonnegative-valued representation $\mathbf{V}_x$, such as a magnitude or power spectrogram. Then, the magnitude (or power) spectrum of the singing voice is predicted using e.g., nonnegative matrix factorization (NMF) [2, 3], kernel additive models [4] or deep neural networks (DNNs) [5]. Finally, a *phase recovery* technique is used in order to retrieve the complex-valued STFT of the singing voice.

Much research in audio has focused on the processing of nonnegative-valued data. Phase recovery is usually performed by combining the mixture's phase with the estimated voice spectrogram, or by means of a Wiener-like filter [3, 6]. Those approaches result in assigning the mixture's phase to the STFT voice estimate. However, even if the latter leads to quite satisfactory results in practice [2, 3], it has been pointed out that when sources overlap in the TF domain, the assignment of the mixture's phase to the STFT voice estimate is responsible for residual interference and artifacts in the separated signals [7].

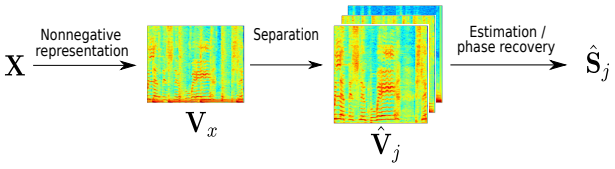

Figure 1: *A typical source separation system in the TF domain.*

In the recent years, some efforts have been made to improve phase recovery in audio source separation. Phase recovery algorithms exploit phase constraints that originate from consistency [8], a property of the STFT that arises from its redundant signal representation, or from a signal model that approximates time-domain signals as a sum of sinusoids [7]. The above mentioned phase constraints have been applied to a source separation task [9, 7] and combined with magnitude estimation techniques in order to design full and phase-aware separation systems [10, 11]. However, these systems are based on variants of NMF methods, which provides fairly good separation results in scenarios where the sources are well represented with stationary spectral atoms (over time) and uniform temporal activations (over frequencies).

In this paper, we propose to rather investigate on improved phase recovery algorithms in DNN-based source separation. Indeed, state-of-the-art results for source separation are obtained with deep learning methods in both monaural [12, 13] and multichannel [14, 15] scenarios. This goes for the particular case of monaural singing voice separation [16, 17, 18]. The most recent approach, which is called MaD TwinNet [18], predicts a voice magnitude spectrogram that is further combined with the mixture's phase. We propose to assess the potential of recent phase recovery algorithms as alternative methods to this baseline in order to enhance the separation quality. We test the proposed techniques on realistic music songs used in the signal separation evaluation campaign (SiSEC) [19], and we observe that these algorithms are interesting alternatives to the baseline since they allow to reduce interference at the cost of very few additional artifacts.

The rest of this paper is structured as follows. Section 2 presents the MaD TwinNet system used for magnitude spectrum prediction. Section 3 introduces the most recent phase recovery algorithms. Experiments are conducted in Section 4, and Section 5 draws some concluding remarks.

2. MaD TwinNet

The most up-to-date deep learning system for monaural singing voice separation is the Masker Denoiser (MaD) architecture with Twin Networks regularization (MaD TwinNet) [18].

10.21437/Interspeech.2018-1845

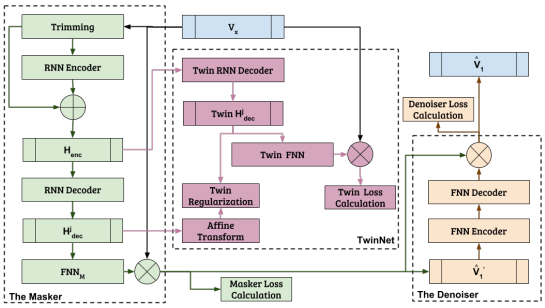

Figure 2: *Illustration of the Mad TwinNet system (adapted from [18]). With green color is the Masker, with magenta the TwinNet, and with light brown the Denoiser.*

Therefore, we will use it as a core system in our separation framework. We briefly present its architecture hereafter, and more details on it can be found in [17, 18].

MaD TwinNet consists of the Masker, the Denoiser, and the TwinNet, and it is illustrated in Fig. 2. The Masker consists of a bi-directional recurrent neural network (Bi-RNN), the RNN encoder (RNN_{enc}), an RNN decoder (RNN_{dec}), a sparsifying transform that is implemented by a feed-forward neural network (FNN), with shared weights through time, followed by a rectified linear unit (ReLU), and the skip-filtering connections [16]. The input to the Masker is a $\mathbf{V}_x$ and the output of the skip-filtering connections is a first estimate of the singing voice spectrogram denoted $\hat{\mathbf{V}}_1'$. Prior to the encoding of $\mathbf{V}_x$, a trimming operation is applied to $\mathbf{V}_x$. That operation preserves information only up to 8 kHz, and is used to decrease the amount of trainable parameters of the Masker. Then, the RNN_{enc} is used to encode the temporal information of $\mathbf{V}_x$, and its output is used as an input to RNN_{dec}, which produces the latent representation of the target source TF mask. The latent representation is then transformed to a TF mask by the sparsifying transform. The output of the sparsifying transform along with the $\mathbf{V}_x$, are used as an input to a skip-filtering connection, which outputs $\hat{\mathbf{V}}_1'$.

Since $\hat{\mathbf{V}}_1'$ is expected to contain interference from other music sources [16, 17], the Denoiser aims at further enhancing the estimate of the Masker. A denoising filter is learned and applied to the estimate of the Masker, $\hat{\mathbf{V}}_1'$. More specifically, $\hat{\mathbf{V}}_1'$ is propagated to an encoding and a decoding stage. Each stage is implemented by a FNN, with shared weights through time. Each FNN is followed by a ReLU. Then, the output of the decoder and $\hat{\mathbf{V}}_1'$ are used as an input to the skip-filtering connections. This yields the final voice magnitude estimate $\hat{\mathbf{V}}_1$.

RNNs appear to be a suitable choice for modeling the long term temporal patterns (e.g., melody and rhythm) that govern music signals like the singing voice. However, such signals can be dominated by local structures, shorter than the long temporal patterns [18], making it harder to model the longer term structure. To deal with this issue, the authors in [20] proposed to use the hidden states of a backward RNN for regularizing the hidden states of a forward RNN. This regularization results in enforcing the forward RNN to model longer temporal structures and dependencies. The backward RNN, and the replication of the process used to optimize the backward RNN, is called Twin Network, or TwinNet.

More specifically, TwinNet is used in the MaD TwinNet architecture [18] to regularize the output of RNN_{dec} in the Masker. Additionally to the forward RNN of the RNN_{dec} and the subsequent sparsifying transform, the authors in [18] use the output of the RNN_{enc} as an input to a backward RNN, which is then followed by a sparsifying transform. The backward RNN and the associated sparsifying transform are used in the TwinNet regularization scheme.

3. Phase recovery

3.1. Baseline approach

Once the voice magnitude spectrum $\hat{\mathbf{V}}_1$ is estimated, the baseline approach used in [18] consists in using the mixture's phase to retrieve the STFT of the voice:

$$\hat{\mathbf{S}}_1 = \hat{\mathbf{V}}_1 \odot e^{\odot i \angle \mathbf{X}}, \tag{1}$$

where $\odot$ and $.^{\odot}$ respectively denote the element-wise matrix multiplication and power, and $\angle$ denotes the complex argument. Retrieving the complex-valued STFTs by using the mixture's phase is justified in TF bins where only one source is active. Indeed, in such a scenario, the mixture is equal to the active source. However, this is not the case in TF bins where sources overlap, which is common in music signals. This motivates improving phase recovery for addressing this issue.

3.2. Phase constraints

Improved phase recovery can be achieved by exploiting several phase constraints, that either arise from a property of the STFT or from the signal model itself.

3.2.1. Consistency

Consistency [8] is a direct consequence of the overlapping nature of the STFT. Indeed, the STFT is usually computed with overlapping analysis windows, which introduces dependencies between adjacent time frames and frequency channels. Consequently, not every complex-valued matrices $\mathbf{Y} \in \mathbb{C}^{F \times T}$ are the STFT of an actual time-domain signal. To measure this mismatch, the authors in [8] proposed an objective function called *inconsistency* defined as:

$$\mathcal{I}(\mathbf{Y}) = ||\mathbf{Y} - \mathcal{G}(\mathbf{Y})||_{\text{F}}^2, \tag{2}$$

where $\mathcal{G}(\mathbf{Y}) = \text{STFT} \circ \text{STFT}^{-1}(\mathbf{Y})$, STFT^{-1} denotes the inverse STFT and $||.||_{\text{F}}$ is the Frobenius norm. It is illustrated in Fig. 3. Minimizing this criterion results in computing a complex-valued matrix that is as close as possible to the STFT of a time signal. The authors in [21] proposed an iterative procedure, called the Griffin Lim algorithm, that updates the phase of $\mathbf{Y}$ while its magnitude is kept equal to the target value. This technique was used in the original MaD system [17] to retrieve the phase of the singing voice, but it was later replaced in [18] by simply using the mixture's phase, since it was observed to perform better.

3.2.2. Sinusoidal model

Alternatively, one can extract phase constraints from the sinusoidal model, which is widely used for representing audio signals [11, 22]. It can be shown [23] that the STFT phase $\boldsymbol{\mu}$ of a signal modeled as a sum of sinusoids in the time domain follows the *phase unwrapping* (PU) equation:

$$\mu_{ft} \approx \mu_{ft-1} + 2\pi l \nu_{ft}, \tag{3}$$

where l is the hop size of the STFT and ν_{ft} is the normalized frequency in channel f and time frame t. This relationship between adjacent TF bins ensures a form of temporal coherence of

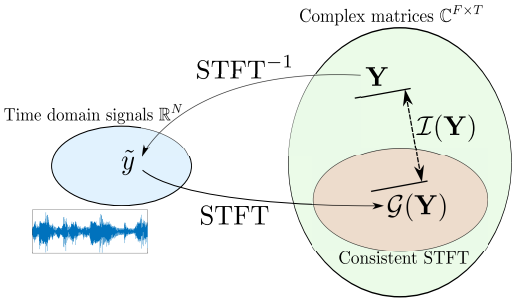

Figure 3: *Illustration of the concept of inconsistency.*

the signal. It has been used in many audio applications, including time stretching [23], speech enhancement [22] and source separation [7, 11, 24].

3.3. Wiener filters

One way to incorporate those phase constraints in a separation system is to apply a Wiener-like filter to the mixture. The classical Wiener filter [3] consists in multiplying the mixture by a nonnegative-valued gain matrix (or *mask*):

$$\hat{\mathbf{S}}_j = \mathbf{G}_j \odot \mathbf{X}, \tag{4}$$

where $j \in \{1, 2\}$ is the source index, and the gain is:

$$\mathbf{G}_j = \frac{\hat{\mathbf{V}}_j^{\odot 2}}{\hat{\mathbf{V}}_1^{\odot 2} + \hat{\mathbf{V}}_2^{\odot 2}}, \tag{5}$$

where the fraction bar denotes the element-wise matrix division. Since this filter simply assigns the mixture's phase to each source, more sophisticated versions of it have been designed[1]:

- *Consistent* Wiener filtering [9] exploits the consistency constraint (2) through a soft penalty that is added to a cost function measuring the mixing error;
- *Anisotropic* Wiener filtering [24] builds on a probabilistic model with non-uniform phases. This enables one to favor a phase value that is given by (3);
- *Consistent anisotropic* Wiener filtering (CAW) [25] is a combination of the previous approaches, where both phase constraints can be accounted for.

For generality, we consider here the CAW filter. It depends on two parameters κ and δ, which respectively promote anisotropy (and therefore the phase model given by (3)) and consistency, i.e., the constraint (2). CAW has been shown to perform better than the other filters that use only one phase constraint [25].

3.4. Iterative procedure

Another phase retrieval algorithm has been introduced in [7]. This approach aims at minimizing the mixing error:

$$\mathcal{C}(\hat{\mathbf{S}}) = \sum_{ft} |x_{ft} - \sum_j \hat{s}_{j,ft}|^2, \tag{6}$$

subject to $|\hat{\mathbf{S}}_j| = \hat{\mathbf{V}}_j \ \forall j$. An iterative scheme is obtained by using the auxiliary function method which provides updates on $\hat{s}_{j,ft}$. In a nutshell, it consists in computing the mixing error at one given iteration, distributing this error onto the estimated sources, and then normalizing the obtained variables so that their magnitude is equal to the target magnitude values $\hat{\mathbf{V}}_j$ (this differs from Wiener filters where the masking process modifies the target magnitude value).

[1] Due to space constraints, we cannot provide the mathematical derivation of those filters, but the interested reader will find more technical details in the corresponding referenced papers.

Algorithm 1: PU-Iter

Inputs: Mixture $\mathbf{X}$, magnitudes $\hat{\mathbf{V}}_j$ and frequencies $\boldsymbol{\nu}_j$
Compute gains $\mathbf{G}_j$ according to (5)
for $t = 1$ **to** $T - 1$ **do**
  $\forall j, f$:
  $\phi_{j,ft} = \angle \hat{s}_{j,ft-1} + 2\pi l \nu_{j,ft}$
  $\hat{s}_{j,ft} = v_{j,ft} e^{i\phi_{j,ft}}$
  for $it = 1$ **to** max_iter **do**
    $y_{j,ft} = \hat{s}_{j,ft} + g_{j,ft}(x_{ft} - \sum_j \hat{s}_{j,ft})$
    $\hat{s}_{j,ft} = v_{j,ft} \frac{y_{j,ft}}{|y_{j,ft}|}$
  end
end
Output: Estimated sources $\hat{\mathbf{S}}_j$

The key idea of the algorithm is to initialize the phase of the estimates $\hat{\mathbf{S}}_j$ with the values provided by the sinusoidal model (3). This results in a fast procedure (initial estimates are expected to be close to a local minimum) and the output estimates benefit from the temporal continuity property of the sinusoidal phase model. This procedure, called PU-Iter, is summarized in Algorithm 1. It does not exploit the consistency constraint, but it was proven to perform better than consistent Wiener filtering in scenarios where magnitude spectrograms are reliably estimated [7].

4. Experimental evaluation

4.1. Setup

We consider 100 music songs from the Demixing Secrets Database, a semi-professionally mixed set of music song used for the SiSEC 2016 campaign [19]. The database is split into two sets of 50 songs (training and test sets). Each song is made up of $J = 2$ sources: the singing voice track and the musical accompaniment track. The signals are sampled at 44100 Hz and the STFT is computed with a 46 ms long Hamming window, with a padding factor of 2 and a hop size of 384 samples.

For the MaD TwinNet, we used the pre-trained parameters that are available through the Zenodo on-line repository [26] and correspond to the results presented in [18]. The frequencies $\boldsymbol{\nu}_j$ used for applying PU (3) are estimated by means of a quadratic interpolated FFT (QIFFT) [27] on the log-spectra of the magnitude estimates $\hat{\mathbf{V}}_j$. PU-Iter uses 50 iterations, and the CAW filter uses the same stopping criterion as in [9, 25] (i.e., a relative error threshold of 10^{-6}).

Source separation quality is measured with the signal-to-distortion, signal-to-interference, and signal-to-artifact ratios (SDR, SIR, and SAR) [28] expressed in dB, which are computed on sliding windows of 30 seconds with 15 second overlap. These metrics are calculated using the `mir_eval` toolbox [29]. Online are available a demo of the separated audio sequences[2] as well as the code of this experimental study[3].

[2] `http://arg.cs.tut.fi/demo/phase-madtwinnet`
[3] `https://github.com/magronp/phase-madtwinnet`

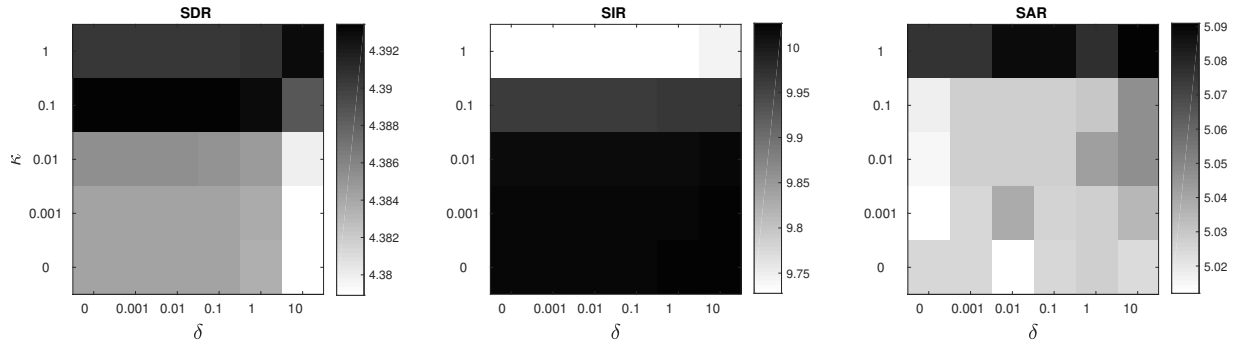

Figure 4: *Separation performance (SDR, SIR and SAR in dB) of the CAW filtering for various phase parameters. Darker is better.*

Table 1: *Source separation performance (median SDR, SIR and SAR in dB) for various phase recovery approaches.*

	SDR	SIR	SAR
Baseline	**4.57**	8.17	**5.97**
PU-Iter	4.52	8.87	5.52
CAW	4.46	**10.32**	4.97

4.2. Performance of the Wiener filters

We first investigate on the performance of the phase-aware extensions of Wiener filtering presented in Section 3.3. We apply CAW with variable anisotropy and consistency parameters and we present the median results over the dataset in Fig. 4. We observe that increasing κ leads to improve the distortion metric and artifact rejection, but decreases the SIR. The value $\kappa = 0.01$, for which the decrease in SIR is very limited, appears as a good compromise. On the other hand, increasing the consistency weight δ overall increases the SIR and SAR, but reduces the SDR (except for a high value of the anisotropy parameter κ). In particular, $\delta = 0.1$ slightly boosts the SIR compared to $\delta = 0$, without sacrificing the SDR too much.

Note that alternative values of the parameters reach different compromises between those indicators. For instance, if the main objective is the reduction of artifacts, one can choose a higher value for κ. Conversely, if the goal is to reduce interference, then it is suitable to pick a null value for the anisotropy parameter combined with a moderate consistency weight. Finally, note that such filters actually use the power spectrograms (not the magnitudes) to compute a mask (*cf.* (5)). Therefore, better results could be reached by using a network that directly outputs power spectrograms instead of magnitudes.

4.3. Comparison to the baseline

We now compare the baseline technique (*cf.* Section 3.1) with PU-Iter and CAW using the parameters values obtained in the previous experiment. Results are presented in Table 1. The best results in terms of SDR and SAR are obtained with the baseline method, while the CAW filter yields the best results in terms of interference reduction (an improvement of more than 2 dB compared to the baseline). Nonetheless, those results must be nuanced by the fact that these drops in SDR and SAR are limited (compared to the increase in SIR) when going from the baseline to alternative phase recovery techniques. Indeed, PU-Iter improves the SIR by 0.8 dB at the cost of a very limited drop in SDR (-0.05 dB) and quite limited in SAR (-0.45 dB). CAW's drop in SDR and SAR is more important(-0.1 dB and -1 dB), but it yields estimates with significantly less interference ($+2$ dB in SIR).

Consequently, we cannot argue that one method is better than another, but rather that they yield different compromises between the metrics. Thus, the phase recovery technique must be chosen in conformity with the main objective of the separation. If the main goal is the suppression of artifacts then one should use the baseline strategy. If one looks for stronger interference reduction, then CAW is a suitable choice. Finally, PU-Iter is the appropriate choice for applications where the SAR can be slightly sacrificed at the benefit of a 0.7 dB boost in SIR.

Note that in this work, we used the same STFT setting as in [18] for simplicity. However, this is not optimal from a phase recovery perspective. Indeed, the importance of consistency is strongly dependent on the amount of overlap in the transform, and the PU technique's performance is highly impacted by the time and frequency resolutions [7]. Consequently, the STFT parameters (window size, zero-padding, overlap ratio) could be more carefully tuned so one can exploit the full potential of those phase recovery techniques.

5. Conclusions and future work

In this work, we addressed the problem of STFT phase recovery in DNN-based audio source separation. Recent phase retrieval algorithms yield estimates with less interference than the baseline approach using the mixture's phase, at the cost of limited additional distortion and artifacts. Future work will focus on alternative separation scenarios, where the phase recovery issue is more substantial. Indeed, phase recovery has more potential when the sources are more strongly overlapping in the TF domain, such as in harmonic/percussive source separation [30]. Another interesting research direction is the joint estimation of magnitude and phase in a unified framework, rather than in a two-stage approach. For instance, the Bayesian framework introduced in [14] has a great potential for tackling this issue.

6. Acknowledgments

P. Magron is supported by the Academy of Finland, project no. 290190. S.-I. Mimilakis is supported by the European Unions H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet. P. Magron, K. Drossos and T. Virtanen wish to acknowledge CSC-IT Center for Science, Finland, for computational resources. Part of the computations leading to these results was performed on a TITAN-X GPU donated by NVIDIA to K. Drossos. Part of this research was funded by the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND.

7. References

[1] P. Comon and C. Jutten, *Handbook of blind source separation: independent component analysis and applications*. Academic press, 2010.

[2] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, March 2007.

[3] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, March 2009.

[4] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4298–4310, August 2014.

[5] P.-S. Huang, M. K. M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[6] A. Liutkus and R. Badeau, "Generalized Wiener filtering with fractional power spectrograms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015.

[7] P. Magron, R. Badeau, and B. David, "Model-based STFT phase recovery for audio source separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 6, pp. 1095–1105, June 2018.

[8] J. Le Roux, N. Ono, and S. Sagayama, "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction," in *Proc. ISCA Workshop on Statistical and Perceptual Audition (SAPA)*, September 2008.

[9] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 217–220, March 2013.

[10] J. Le Roux, H. Kameoka, E. Vincent, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF under spectrogram consistency constraints," in *Proc. Acoustical Society of Japan Autumn Meeting*, September 2009.

[11] J. Bronson and P. Depalle, "Phase constrained complex NMF: Separating overlapping partials in mixtures of harmonic musical sources," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[12] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[13] E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Two-stage single-channel audio source separation using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1773–1783, September 2017.

[14] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, September 2016.

[15] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band DenseNets for audio source separation," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, October 2017.

[16] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "A recurrent encoder-decoder approach with skip-filtering connections for monaural singing voice separation," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2017.

[17] S. I. Mimilakis, K. Drossos, J. F. Santos, G. Schuller, T. Virtanen, and Y. Bengio, "Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018.

[18] K. Drossos, S. I. Mimilakis, D. Serdyuk, G. Schuller, T. Virtanen, and Y. Bengio, "MaD TwinNet: Masker-denoiser architecture with twin networks for monaural sound source separation," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*, July 2018.

[19] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 Signal Separation Evaluation Campaign," in *Proc. International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, February 2017.

[20] D. Serdyuk, N.-R. Ke, A. Sordoni, A. Trischler, C. Pal, and Y. Bengio, "Twin Networks: Matching the future for sequence generation," in *Proc. of International Conference on Learning Representations (ICLR)*, April 2018.

[21] D. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236–243, April 1984.

[22] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1931–1940, December 2014.

[23] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, May 1999.

[24] P. Magron, R. Badeau, and B. David, "Phase-dependent anisotropic Gaussian model for audio source separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017.

[25] P. Magron, J. Le Roux, and T. Virtanen, "Consistent anisotropic Wiener filtering for audio source separation," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, October 2017.

[26] K. Drossos, S. I. Mimilakis, D. Serdyuk, G. Schuller, T. Virtanen, and Y. Bengio, "Mad twinnet pre-trained weights," Feb. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1164592

[27] M. Abe and J. O. Smith, "Design criteria for simple sinusoidal parameter estimation based on quadratic interpolation of FFT magnitude peaks," in *Audio Engineering Society Convention 117*, May 2004.

[28] E. Vincent, R. Gribonval, and C. Févotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 4, pp. 1462–1469, July 2006.

[29] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, October 2014.

[30] W. Lim and T. Lee, "Harmonic and percussive source separation using a convolutional auto encoder," in *Proc. European Signal Processing Conference (EUSIPCO)*, August 2017.