PROCEEDINGS of the
23rd International Congress on Acoustics

9 to 13 September 2019 in Aachen, Germany

Spatial soundfield reproduction using deep neural networks

Hanchi Chen(1), Thushara D. Abhayapala(1)

(1)Audio & Acoustic Signal Processing Group, College of Engineering and Computer Science, Australian National University

Abstract

Sparsity-based sound field reproduction algorithms often result in improved localization and larger reproduction region, but also lead to high computational cost. In this work, we present a novel approach for sparse reproduction, where a deep neural network (DNN) is trained to determine the optimal driving signals for a loudspeaker array, given the desired sound field coefficients as the input. We show that when trained using the proposed method, the DNN-based algorithm can outperform existing Lasso-based algorithms in terms of noise sensitivity and computation speed.

Keywords: Soundfield reproduction, deep neural networks

1 INTRODUCTION

Sound field reproduction systems aim to reproduce a desired sound scene so that listeners in a certain reproduction region can perceive the reproduced sound as if they were located in the original sound field where the recording was made. A loudspeaker array is often placed close to the reproduction region, and various approaches have been developed to determine the placement of these loudspeakers as well as the driving signals for each loudspeaker. One main approach for sound field reproduction is Wave-Field Synthesis [1, 2], which is based on the Kirchhoff-Helmholtz integral, this method delivers high reproduction accuracy over a reasonably large region, however the number of loudspeakers increases significantly for high frequencies. Another approach is Ambisonics / Higher Order Ambisonics [3, 4, 5, 6, 7], where the sound field is represented using first order or higher order spherical harmonics. This approach typically produces a “sweet spot” or a spherical reproduction region which scales with the frequency as well as the number of loudspeakers available, for higher frequencies and large reproduction region, it is necessary to use an unrealistic number of loudspeakers.

Sound field reproduction techniques which exploit the sparsity of the sound field present a new approach to solving the problem. The 1-norm regularized LMS algorithm, or Lasso, can be utilized to effectively reduce the number of active loudspeakers [8, 9, 10, 11, 12]. In [8], a Lasso algorithm is used first to determine a subset of loudspeakers for producing a certain sound field, then the driving signal for each active loudspeaker is calculated using a pressure matching approach. This method is shown to use a smaller number of loudspeakers to produce satisfactory reproduction quality. In [13], a Lasso-based algorithm is developed to optimize the sound field reproduction for under-sampled sound fields and judicious loudspeaker placements. A latent group Lasso algorithm is proposed in [14] to improve the accuracy of Lasso-based reproduction algorithms.

The solution of Lasso problems are not linear, and they need to be calculated in an iterative manner, thus resulting in high computational cost. On the other hand, Deep Neuron Networks (DNN) have predictable computational cost, and dedicated hardware such as GPUs can further accelerate the computation of DNN outputs. More importantly, DNNs are well know for their ability in predicting non-linear transfer functions. Neural Networks have already seen wide use in the audio signal processing area, including speech recognition [15], source separation [16], beamforming [17] and DOA estimation [18]. However, to the best of our knowledge, its application in sound field reproduction has yet to be developed. In this work, we present a DNN-based sound field reproduction algorithm, including its training method. We show that the DNN can be trained to generate sparse loudspeaker driving signals to reproduce plane wave sound fields. The solution is compared with Lasso-based algorithm, and we show that the DNN algorithm is able to generate more stable driving signals in the presence of noise and reverberation.

2 PROBLEM FORMULATION

It has been shown that Least-Mean-Square based Pressure-Matching methods for spatial sound reproduction can lead to degraded localization performance, due to the fact that all the loudspeakers in the reproduction array been activated [8]. Therefore, in this work, we aim to derive a sparse reproduction algorithm which minimizes the number of active loudspeakers for a given sound field. For this purpose, it is necessary that the desired sound field exhibit some form of sparsity in nature. Thus, we make the following assumptions regarding the sound field model:

1: At each frequency k, there exists one dominant acoustic source, the sound field due to which is to be reproduced by a loudspeaker array.

2: The dominant source is sufficiently far, such that its sound field at the reproduction region can be seen as a plane wave.

3: The reverberations, noise and other interferences are weaker than the direct-path signal from the dominant source, and are uncorrelated with the direct-path signal.

We further assume that a higher order microphone array is placed at the observation point to record the original sound field, which is capable of decomposing the captured sound field into Nth order spherical harmonic coefficients [19, 20]

$$P(r,\theta,\phi,k) \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} C_{nm}(k) j_n(kr) Y_{nm}(\theta,\phi), \tag{1}$$

where $P(r,\theta,\phi,k)$ denotes the sound pressure at position (r,θ,ϕ), k is the wave number, $j_n(kr)$ is the spherical Bessel function;, $Y_{nm}(\theta,\phi)$ is the spherical harmonic coefficient and C_{nm} are the spherical harmonic coefficients. Using Assumptions 1 and 3, the spherical harmonic coefficients can be further decomposed as

$$C_{nm}(k) = D_{nm}(k) + R_{nm}(k) + U_{nm}(k), \tag{2}$$

where $D_{nm}(k)$, $R_{nm}(k)$ and $U_{nm}(k)$ are the spherical harmonic coefficients of the direct-path signal, reverberation/interference, and measurement noise, respectively.

Assuming that a loudspeaker array is used to reproduce the sound field due to the dominant acoustic source, with the spherical harmonic based channel vector of the lth loudspeaker expressed as $\boldsymbol{h}_\ell = [H_{00}^{\ell}, H_{11}^{\ell} ... H_{NN}^{\ell}]^T$, the goal of this paper is to find a set of loudspeaker driving signals W_l, such that the resulting sound field best approximates the desired direct-path signal, while minimizing the total number of active loudspeakers. The corresponding cost function can be written as

$$\arg\min_{\boldsymbol{W}} \|\boldsymbol{W}\boldsymbol{H} - \boldsymbol{D}\|_2^2 + \lambda \|\boldsymbol{W}\|_1, \tag{3}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote vector 2-norm and 1-norm, respectively, $\boldsymbol{W} = [W_1, W_2, ... W_L]$ is the vector of driving signals for all the L speakers in the array, $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ... \boldsymbol{h}_L]$ is the channel matrix between the loudspeaker array and the reproduction region, and $\boldsymbol{D} = [D_{00}, D_{11}, ... D_{NN}]$ is the vector of desired spherical harmonic coefficients.

We note that instead of attempting to reproduce the complete sound field represented by C_{nm}, the cost function (3) aims to minimize the reproduction error for the direct-path signal only, and ignore the noise and interference components that may be present in the recorded sound field, which differs from existing sparse reproduction algorithms in the literature. The motivation is that, in a non-anechoic reproduction environment, the recorded reverberation will add to the reverberation of the reproduction room, hence leading to excessive reverberation energy which degrades the clarity of the desired signal; furthermore, the reverberations may activate additional loudspeakers, which further degrades the localization of the system.

3 DNN-based sound field reproduction

3.1 Sparse driving signal calculation using complex Lasso

The cost function (3) has the standard Lasso problem formulation, where the penalty parameter λ can be adjusted to change the balance between the reproduction accuracy and the sparsity of the loudspeaker driving signals. Various algorithms have been proposed for solving the Lasso optimization problems, including Coordinate Descent and Alternating Direction Method of Multipliers (ADMM). In general, these algorithms operate on real-valued inputs. However, the problem (3) is complex-valued. It is thus necessary to transform the problem into a real-valued equivalent problem so that it can be solved using existing methods.

To perform the transform, we first define $\hat{\boldsymbol{W}}_\ell = \left[\mathscr{R}(W_\ell)\,\mathscr{I}(W_\ell)\right]$, $\hat{\boldsymbol{D}}_{nm} = \left[\mathscr{R}(D_{nm})\,\mathscr{I}(D_{nm})\right]$

$$\hat{\boldsymbol{H}}^\ell_{nm} \quad = \quad \begin{bmatrix} \mathscr{R}(H^\ell_{nm}) & -\mathscr{I}(H^\ell_{nm}) \\ \mathscr{I}(H^\ell_{nm}) & \mathscr{R}(H^\ell_{nm}) \end{bmatrix}. \tag{4}$$

Then, we construct the real-valued matrices $\bar{\boldsymbol{W}} = \left[\hat{\boldsymbol{W}}_1\,\hat{\boldsymbol{W}}_2 \ldots \hat{\boldsymbol{W}}_L\right]$, $\bar{\boldsymbol{D}} = \left[\hat{\boldsymbol{D}}_{00}\,\hat{\boldsymbol{D}}_{11} \ldots \hat{\boldsymbol{D}}_{NN}\right]$

$$\bar{\boldsymbol{H}} \quad = \quad \begin{bmatrix} \hat{\boldsymbol{H}}^1_{00} & \hat{\boldsymbol{H}}^1_{11} & \ldots & \hat{\boldsymbol{H}}^1_{NN} \\ \hat{\boldsymbol{H}}^2_{00} & \hat{\boldsymbol{H}}^2_{11} & \ldots & \hat{\boldsymbol{H}}^2_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{H}}^L_{00} & \hat{\boldsymbol{H}}^L_{11} & \ldots & \hat{\boldsymbol{H}}^L_{NN} \end{bmatrix}. \tag{5}$$

Then the real-valued equivalent of (3) can be written as [21]

$$\underset{\bar{\boldsymbol{W}}}{\arg\min} \|\bar{\boldsymbol{W}}\bar{\boldsymbol{H}} - \bar{\boldsymbol{D}}\|_2^2 + \lambda \sum_{\ell=1}^{L} \|\hat{\boldsymbol{W}}_\ell\|_2. \tag{6}$$

Eq. (6) represents a group Lasso problem with a constant group size of 2, which can be solved using the ADMM method [22].

3.2 Training dataset for the DNN

The Lasso solution in Section 3.1 cannot be used directly to compute the desired driving signal, because D_{nm} is combined with the unknown interfere and noise components in the received coefficients C_{nm}. In this section, we propose a method to train a deep neural network, such that the DNN learns to estimate the desired plane wave component, and produce the desired sparse driving signals in the presence of noise and interference.

In the ideal case where R_{nm} and N_{nm} equal to zero, for a given loudspeaker channel matrix $\boldsymbol{H}$ and a given impinging direction (ϑ_j, φ_j), we can use the complex Lasso algorithm to solve for the optimal driving signal vector $\boldsymbol{W}_\ell$ where based on Assumption 2 in Section 2, the spherical harmonic coefficients due to the dominant plane wave can be expressed as [3]

$$D^j_{nm} = i^n \cdot Y^*_{nm}(\vartheta_j, \varphi_j). \tag{7}$$

Considering a total of J impinging directions which are uniformly sampled over the sphere, we can then solve for the corresponding J number of driving vectors. The total set of $\{\boldsymbol{D}_j\} = \{[D^j_{00}, D^j_{11}, \ldots D^j_{NN}]\}$ and $\{\boldsymbol{W}_j\}$ form the basis of the training dataset for the DNN.

In reality, the phase of D^j_{nm} will vary, and thus phase of the optimal driving signal will change accordingly. To model this behavior, we augment the training dataset by adding a random phase shift to both D^j_{nm} and $\boldsymbol{W}_j$, such that

$$\boldsymbol{D}_{j,z} = \boldsymbol{D}_j \cdot \rho_{j,z}, \tag{8}$$

$$\boldsymbol{W}_{j,z} = \boldsymbol{W}_j \cdot \rho_{j,z}, \tag{9}$$

where $\rho_{j,z}$ is a complex random variable with unity amplitude and a uniform phase distribution over $[0,2\pi)$.

Our goal is to train the neural network such that its output is not affected by the presence of the reverberations and noises that may be present in the recorded coefficients C_{nm}. To model this behavior, we further augment the training data by adding a complex gaussian noise σ to each plane wave coefficients, such that

$$\boldsymbol{D}_{j,z,v} = \boldsymbol{D}_{j,z} + \boldsymbol{\sigma}_{j,z,v}, \tag{10}$$

where $\boldsymbol{\sigma}_{j,z,v} = [\sigma_{00}^{j,z,v}, \sigma_{11}^{j,z,v}, ...\sigma_{NN}^{j,z,v}]$ is a vector of instances of the random variable σ, corresponding to each spherical harmonic coefficient. Since the DNN is expected to produce the same output regardless of the value of $\boldsymbol{\sigma}_{j,z,v}$, the desired output signal $\boldsymbol{W}_{j,z}$ is not further augmented, thus all of $\boldsymbol{D}_{j,z,v}$ are paired with the same desired output vector $\boldsymbol{W}_{j,z}$. This augmentation procedure results in a total of $J \times Z \times V$ training samples.

To improve the training speed as well as outcome, the training data for the neural network should be normalized. We define the normalized training input and output data as $\tilde{\boldsymbol{D}}_{j,z,v}^{\text{norm}}$ and $\tilde{\boldsymbol{W}}_{j,z}^{\text{norm}}$, where

$$\tilde{\boldsymbol{D}}_{j,z,v}^{\text{norm}} = \frac{\tilde{\boldsymbol{D}}_{j,z,k}}{\|\tilde{\boldsymbol{D}}_{j,z,k}\|_2}, \tilde{\boldsymbol{W}}_{j,z}^{\text{norm}} = \frac{\tilde{\boldsymbol{W}}_{j,z}}{\|\tilde{\boldsymbol{D}}_{j,z,k}\|_2}. \tag{11}$$

Thus, it is necessary to also normalize the input data when using the trained DNN to calculate driving signal, and consequently the DNN output needs to be multiplied by the corresponding input coefficient norm to obtain the correct loudspeaker driving signals.

3.3 Structure of the DNN

In general, DNNs operate on real-valued data. Thus it is necessary to convert the complex valued input signals into real-valued vectors for the DNN. A natural choice is to use the real and imaginary parts of $\boldsymbol{D}_{j,z,k}$ and $\boldsymbol{W}_{j,z}$ directly, however, it was found that the complex and non-linear relationship between the real and imaginary part results in poor learning outcome for the neural network.

Therefore, we propose to instead compute the absolute value and phase of the training data, so that

$$\tilde{\boldsymbol{D}}_{j,z,k} = \left[|\boldsymbol{D}_{j,z,k}^{\text{norm}}|, \quad \angle \boldsymbol{D}_{j,z,k}^{\text{norm}}\right], \tag{12}$$

$$\tilde{\boldsymbol{W}}_{j,z} = \left[|\boldsymbol{W}_{j,z}^{\text{norm}}|, \quad \angle \boldsymbol{W}_{j,z}^{\text{norm}}]\right]. \tag{13}$$

We propose to use a neural network with two fully connected hidden layers, each consisting of 1,000 neurons and using the rectified linear unit (ReLU) activation function. The input layer is of size $2(N+1)^2$, corresponding to the size of $\tilde{\boldsymbol{D}}_{j,z,k}$; while the output layer consists of $(N+1)^2$ neurons, corresponding to the size of $\tilde{\boldsymbol{W}}_{j,z}$. The linear activation function is used for the output layer.

4 PERFORMANCE ANALYSIS

4.1 System setup

To validate the performance of the proposed DNN-based sound field reproduction system, we train the proposed DNN using a realistic system setup, consisting of 30 loudspeakers surrounding the reproduction region. The loudspeakers are arranged in a dodecahedron geometry, with each loudspeaker placed on the middle point of every edge of the dodecahedron. This placement allows sound field reproduction up to 3rd order via near-uniform sampling of the sphere[23].

For simplicity, we model the loudspeakers as plane wave sources, hence each element in the channel matrix can be written as

$$H_{nm}^{\ell} = i^n \cdot Y_{nm}^{*}(\theta_\ell, \phi_\ell), \tag{14}$$

where (θ_ℓ, ϕ_ℓ) represent the impinging direction of the ℓth speaker in the array. We uniformly sample $J = 1348$ directions over the sphere, and generate the corresponding plane wave coefficients up to order $N = 1$ using group Lasso with $\lambda = 0.02$. The plane wave coefficient vectors from each impinging direction is then augmented with $Z = 50$ random phase angles. Finally, we add $V = 10$ complex random noise vectors to each augmented coefficient vector at $\text{SNR} = 10$, resulting in a total of $674,000$ training samples. The DNN is then trained using the MatLab Deep Learning Toolbox.

4.2 Behavior analysis

4.2.1 Driving signal sparsity

To analyze the behavior of the trained neural network, we use both the Lasso algorithm and the DNN to calculate the loudspeaker driving signals for plane waves of various directions. For comparison, we also the results using the Least-Mean-Square (LMS) Mode Matching method, which is calculated by [6]

$$\boldsymbol{W} = \boldsymbol{D}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H, \tag{15}$$

where $(\cdot)^H$ denotes Hermitian transpose. The loudspeaker driving signals for impinging directions $(\vartheta = 36^\circ, \varphi = -31^\circ)$ and $(\vartheta = 104^\circ, \varphi = 140^\circ)$ are shown in Fig. 1. We note that neither of these two impinging directions are included in the training dataset.

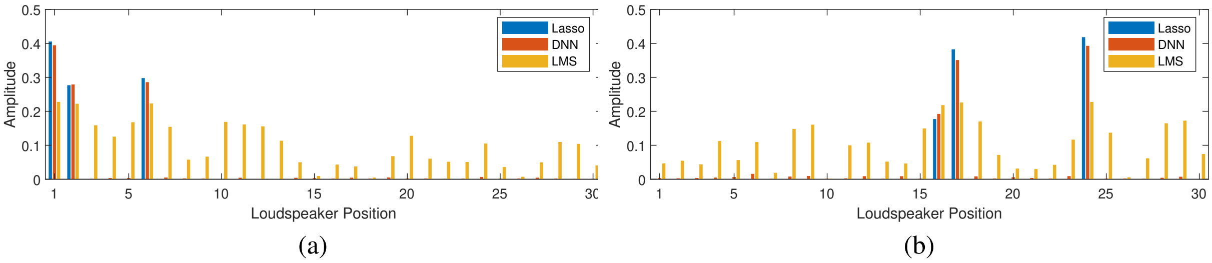

Figure 1. Comparison of loudspeaker driving signal amplitudes calculated using Lasso, DNN and LMS mode matching, for impinging direction (a) $\vartheta = 36^\circ, \varphi = -31^\circ$, and (b) $\vartheta = 104^\circ, \varphi = 140^\circ$.

From Fig. 1, we can observe that the complex Lasso algorithm is able to generate very sparse solutions to reproduce the plane waves, with only 3 loudspeakers active in each case, and the activated loudspeakers are the nearest ones to the desired impinging direction. The DNN is able to produce driving signals that are nearly identical to the Lasso solution, indicating that the network has successfully learned the behavior of the Lasso algorithm. On the other hand, the LMS Mode Matching algorithm results in solutions where nearly all the loudspeakers are activated with similar driving amplitude. It can be expected that both the Lasso and DNN algorithm can produce better localization performance than the LMS algorithm.

We also notice in Fig. 1 that the DNN tends to produce very low amplitude driving signals for many unnecessary speakers. The amplitude of these normalized outputs are typically less than 0.015, which is more than 10 times lower than the dominant driving signals. This artefact can be rectified by applying a simple threshold to the normalized DNN output which forces small driving signals to 0.

4.2.2 Noise resistance

To verify the DNN-based algorithm's ability to reject noise / interference, we compare the variation of its output under noisy input conditions. First we define the performance parameter

$$\eta \triangleq \frac{\sum_{\ell=1}^{L} E\{|W_\ell^{\text{true}} - W_l^{\text{noise}}|\}}{L}, \tag{16}$$

where $E\{\cdot\}$ denotes expectation, W_ℓ^{true} and W_ℓ^{noise} denote the calculated driving signal due to noise-free input coefficients and noisy input coefficients, respectively. Essentially (16) calculates the average variation of driving

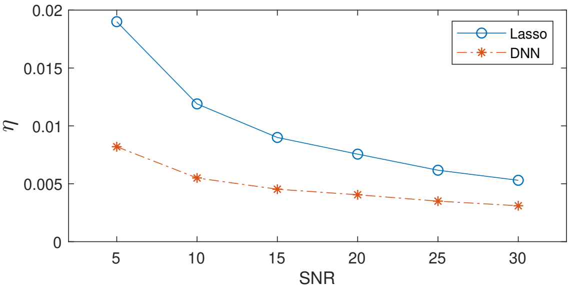

Figure 2. Mean variation of driving signal amplitude at various SNR.

Table 1. Computation time for rendering an audio stream using ADMM Lasso, and DNN using both CPU and GPU.

Audio Length	ADMM Lasso	DNN-CPU	DNN-GPU
0.1 second	1.3s	0.61s	0.67s
1 second	19s	4.3s	1.0s
10 second	161s	48s	5.4s

signal magnitude due to noise; a low value of η indicates that the output is able to remain relatively constant in spite of noise and interference.

The value of η at SNR $= 5 - 30$ is calculated for both the Lasso algorithm and the DNN algorithm. For each SNR, η is estimated by averaging over $1{,}000$ iterations. The results are shown in Fig. 2.

It can be seen from Fig. 2 that the DNN algorithm consistently outperforms the Lasso solution in terms of noise resistance, this is particularly significant at low SNR. Unsurprisingly, the value of η gradually decreases with increased SNR, for both the Lasso and DNN algorithm. We can thus conclude that although the DNN algorithm cannot completely remove the effect of noise and interference, it is able to generate more consistent and accurate driving signals than the Lasso algorithm, especially at lower SNR.

4.3 Computational Complexity

The computation times for rendering an audio stream using ADMM Lasso and the DNN algorithm are shown in Table. 1. All algorithms are implemented in MatLab, running on a PC with 2.6GHz Core-i7, 16GB RAM and GTX 1060 GPU. We can see that for rendering a very short audio frame (0.1s), the time cost is very similar for all three algorithms; however, as the audio length increase, the time cost of ADMM Lasso increases almost linearly, while the DNN algorithms see a much smaller increase, especially when GPU is utilized. This phenomenon is mainly due to overheads of the MatLab's in-built Deep Learning Toolbox. Overall, it can be seen that the DNN algorithm has a much higher computation speed compared to ADMM Lasso, especially when GPU is utilized, hence the proposed method has great potential for real-time rendering applications.

5 EXPERIMENTAL RESULTS

To validate the performance of the proposed DNN algorithm in real-life scenarios, we use the Eigenmike and a spherical 30-loudspeaker array to record and reproduce sound fields in lab environment. The loudspeaker array is arranged in the same geometry as described in Sec. 4.1, as can be seen in Fig. 3(a). We note that the acoustic lining on the walls only partially reduce reflection.

In the experiment, we first use loudspeaker No.23 (labeled in Fig. 3(b)) to play a wide band music content. The audio is recorded by the Eigenmike placed at the origin (center). The Eigenmike is placed with its coordinate system rotated 45° clockwise from the coordinate system of the loudspeaker array, which is equivalent to shifting loudspeaker No.23 counter-clockwise, to the virtual source location indicated in Fig. 3(b), when reproducing the sound field. This makes the problem of reproducing the recorded sound field no longer trivial, and require more than one loudspeaker.

(a)

(b)

Figure 3. (a) The dodecahedron loudspeaker array used to reproduce sound field. (b) The position of the virtual source direction and the loudspeakers in its proximity.

The recorded music is processed in an overlap-add manner with a frame size of 1024 samples and 75% window overlap, at 48kHz sample rate. For each frequency bin, we calculate the spherical harmonics up to order $N = 1$, and use the trained DNN to generate the driving signal for each loudspeaker. The mean driving signal energy of each loudspeaker is shown in Fig. 4, which is calculated by averaging the frequency-domain signal energy over 200-4000 Hz.

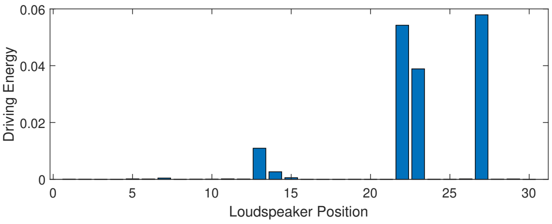

Figure 4. Driving signal energy of the loudspeaker array to produce the sound field recorded by the Eigenmike.

We can see from Fig. 4 that the output energy shows a very sparse pattern, with loudspeaker 22, 23 and 27 having the highest output energy, followed by loudspeaker 13 and 14. Comparing with Fig. 3(b), we see that these are the nearest speakers to the source location, and their driving energy is proportional to the distance between the speaker and the source location. Subjective listening also confirms that the sound imaging is correct over a large area around the origin, further validating that the proposed algorithm is able to reproduce the desired sound field using sparse driving signals.

6 CONCLUSION

We present a novel DNN-based sound field reproduction algorithm, where a DNN is trained to generate a sparse set of loudspeaker driving signals to reproduce a desired plane wave, while suppressing noise and interference added to the plane wave. We show that the proposed algorithm is more noise-invariant than the Lasso solution, with lower computation cost. The algorithm is validated in the lab using wide band recording and a 30-channel loudspeaker array.

ACKNOWLEDGEMENTS

This work is supported by Australian Research Council Grant DP180102375.

REFERENCES

[1] A. J. Berkhout, “A holographic approach to acoustic control,” *J. Audio Eng. Soc*, vol. 36, no. 12, pp. 977–995, 1988.

[2] S. Sascha, R. Rabenstein, and J Ahrens, "The theory of wave field synthesis revisited," in *In 124th Convention of the AES*, 2008.

[3] D.B. Ward and T.D Abhayapala, "Reproduction of a plane-wave sound field using an array of loudspeakers," *IEEE Trans. on Speech and Audio Process.*, vol. 9, no. 6, pp. 697–707, 2001.

[4] T. Betlehem and T.D. Abhayapala, "Theory and design of sound field reproduction in reverberant rooms," *J. Acoust. Soc. Am.*, vol. 117, no. 4, pp. 2100–2111, 2005.

[5] Y.J. Wu and T.D. Abhayapala, "Theory and design of soundfield reproduction using continuous loudspeaker concept," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 17, no. 1, pp. 107–116, 2009.

[6] A. Gupta and T. D. Abhayapala, "Three-dimensional sound field reproduction using multiple circular loudspeaker arrays," *IEEE Trans. on Audio, Speech, and Lang. Process.*, 19(5), pp. 1149–59, July 2011.

[7] Y.J. Wu .and T.D. Abhayapala, "Spatial multizone soundfield reproduction," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Process. (ICASSP).* IEEE, 2009, pp. 93–96.

[8] M. Jia, J. Zhang, Y. Wu, and J. Wang, "Sound field reproduction via the alternating direction method of multipliers based lasso plus regularized least-square," *IEEE Access*, pp. 1–1, 2018.

[9] N. Radmanesh and I.S. Burnett, "Generation of isolated wideband sound fields using a combined two-stage lasso-ls algorithm.," *IEEE Trans. Audio, Speech & Lang. Process.*, vol. 21, no. 2, pp. 378–387, 2013.

[10] H. Khalilian, I.V. Bajić, and R.G. Vaughan, "Towards optimal loudspeaker placement for sound field reproduction," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2013, pp. 321-325.

[11] W. Jin and W.B. Kleijn, "Theory and design of multizone soundfield reproduction using sparse methods," *IEEE/ACM Trans. on Audio, Speech and Lang. Process.*, vol. 23, no. 12, pp. 2343–2355, 2015.

[12] S. Koyama and H. Saruwatari, "Sound field decomposition in reverberant environment using sparse and low-rank signal models," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Process.* 2016, pp. 395–399.

[13] G. N. Lilis, D. Angelosante, and G. B. Giannakis, "Sound field reproduction using the lasso," *IEEE/ACM Trans. on Audio, Speech and Lang. Process.*, vol. 18, no. 8, pp. 1902–1912, Nov 2010.

[14] P.A. Gauthier, P. Grandjean, and A. Berry, "Structured sparsity for sound field reproduction with overlapping groups: Investigation of the latent group lasso," in *2018 AES International Conference on Spatial Reproduction - Aesthetics and Science*, Jul 2018.

[15] G.E. Dahl, D. Yu, L. Deng, A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. on Audio, Speech and Lang. Process.*, 20(1), pp. 30-42, 2012.

[16] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks.," *IEEE/ACM Trans. on Audio, Speech and Lang. Process.*, vol. 24, no. 9, pp. 1652–1664, 2016.

[17] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Process.*. 2016, pp. 196–200.

[18] S. Chakrabarty and Emanuël A.P. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *IEEE WASPAA*, 2017, pp. 136–140.

[19] T. D Abhayapala and D.B. Ward, "Theory and design of high order sound field microphones using spherical microphone array," in *ICASSP*, 2002, vol. 2, pp. 1949–1952.

[20] H. Chen, T. D. Abhayapala, and W. Zhang, "Theory and design of compact hybrid microphone arrays on two-dimensional planes for three-dimensional soundfield analysis," *J. Audio Eng. Soc*,, vol. 138, no. 5, pp. 3081–3092, 2015.

[21] A. Maleki, L. Anitori, Z. Yang, and R.G. Baraniuk, "Asymptotic analysis of complex lasso via complex approximate message passing (camp)," *IEEE Trans. Inf. Theor.*, vol. 59, no. 7, pp. 4290–4308, July 2013.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[23] H. Dickins, G. Chen and W Zhang, "Soundfield control for consumer device testing," in *9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2015, pp. 1–5.