PROCEEDINGS of the
23rd International Congress on Acoustics
9 to 13 September 2019 in Aachen, Germany

A neural network approach to broadband beamforming

Yugo M. KUNO[1] *, Bruno MASIERO[1] †, Nilesh MADHU[2] ‡

[1]University of Campinas – DECOM, Brazil
[2]Ghent University – imec, Belgium

ABSTRACT

Beamforming techniques are commonly applied to signals captured by sensor arrays to enhance signals received from desired directions while reducing background noise and localized interference. Where the directions of the desired and interfering sources are known, this knowledge, combined with assumptions on the background noise characteristics, is used to derive the beamformer coefficients for each sensor. Usually, this is done by optimizing the second-order statistics of the beamformer response, e.g., minimizing the energy of the output signal while preserving the signals from desired directions. The beamformer coefficients are *independently* derived for each discrete frequency, as an approximation to the true broadband response. Hereby, the complex inter-frequency interactions, e.g., due to windowing and spectral aliasing, are not modelled, leading to sub-optimal filter characteristics. Furthermore, for standard designs, the mainlobe also narrows with frequency, leading to a non-uniform beamwidth. These shortcomings can be overcome by data-driven approaches. As a first attempt towards such approaches, we focus on the problem of *uniform* beamwidth and propose a beamforming neural-network architecture. We compare the spatial characteristics of such an architecture to standard (second-order) beamformers.

Keywords: Constant beamwidth beamforming, neural network, speech enhancement

1 INTRODUCTION

Consumer devices, nowadays, are equipped with a host of sensors that allow them to communicate with each other and with the user. Since speech is the most natural medium of communication for humans, using this as a mode of human-machine interaction is increasingly gaining prominence. For a machine to be able to accurately interpret the underlying meaning in the speech, the first step is the high-quality acquisition of the signal. However, signals acquired through device-mounted microphones are often corrupted due to factors such as presence of interfering audio sources in the background, room reverberation, etc.

One way to obtain an improved target (desired) signal in the presence of interference is by simultaneously sampling the sound field at diverse locations. The spatial diversity in the captured signals can then be exploited to localize the different sources contributing to the sound field and, thereby, to enhance signals coming from the spatial region containing the desired source while attenuating signals from other locations. This process is generally termed beamforming [1, 2], and is accomplished by the use of microphone arrays (with known geometry and synchronized audio capture).

The simplest approach to beamforming is the *delay-and-sum* beamformer (DSB) which simply applies different delays to the signals received at the different microphones, to temporally align all signals coming from a desired direction (while misaligning signals coming from other directions), before averaging across all channels. Effectively, this leads to the signal from the desired location being coherently summed across the different microphones (thus preserving it), whereas the signals from other locations are attenuated due to the temporal misalignment across the different microphones [3]. Usually, beamforming is applied to digital, and hence discrete-time, signals. Therefore, as the delays are also discrete, it becomes impossible to create a perfect alignment between the channels, introducing quantization errors. Signal interpolation or fractional sample shifts [4] could be used to reduce these errors, but this solution is computationally intensive and introduces a new source of errors – according to the interpolation method chosen. A simpler, more practical solution, especially for *compact* microphone arrays, is to apply the same general method, but process the signal in the *frequency* domain instead.

The frequency domain counterpart to the DSB is the Bartlett or conventional beamformer [5]. The conventional beamformer solves the quantization problem since shifting the signal in time is no longer restricted to discrete steps. Nevertheless, this beamformer still suffers from spatial aliasing, caused by the fact that, at certain frequencies, signals from different directions have the same phase differences between microphones, making it impossible for

*yugo4k@decom.fee.unicamp.br
†masiero@unicamp.br
‡nilesh.madhu@ugent.be

the algorithm to align the signals from the target direction without also aligning the signals from such "alias" directions. Note, however, that spatial aliasing is a physical constraint introduced by the array geometry and is a limitation we have to live with. More importantly, for low frequencies, signals arriving from neighboring directions have very small phase differences across the microphones and are, therefore, hard to filter out. Thus, at lower frequencies the DSB has poor angle discrimination. However, this improves as the frequency increases, making the mainlobe narrower and sharper.

As an alternative to DSB we can use superdirective beamformers [6]. The minimum variance distortionless response beamformer (MVDR) or Capon beamformer [5] instead of maximizing the signal from the target direction, uses the statistics of the noise field to minimize the energy from directions outside of the target direction; it becomes thus necessary to approximate or model the background noise information. This approach is able to reduce the levels of the sidelobes in comparison to DSB, while also narrowing down the mainlobe. Nevertheless, the MVDR still exhibits a broader mainlobe at low frequencies, which narrows with increasing frequency. There have been attempts to guarantee a constant mainlobe aperture using, e.g., eigenfilter design techniques [7] or Slepian functions [8, 9]. These attempts, however, are still based on second-order statistics and trade-off interference suppression for maintaining mainlobe width.

While the MVDR beamformer requires and uses information about the background noise, the neural network beamformer (NN) proposed in this article goes one step further: it uses the complete information available from the microphone array channels as input to generate a beamforming vector that changes according to the mixed signal characteristics. Because the beamforming vector is flexible, it is able to use information about the target and interfering directions that is encoded in the recorded data. The data-centric model of machine learning (ML) also makes it simple to train the neural network to filter signals that come from a range of directions instead of a single target, resulting in a beampattern that has *uniform* width across almost the entire useful frequency range while, at the same time, having a sharper and stronger attenuation outside of the target range.

In the following, after presenting the signal model and the classical beamforming methods, the details of the proposed network architecture and the training method are discussed. This is followed by a comparison with the classical beamformers, highlighting the benefits of the proposed approach. Note that the discussion in this paper is purely focused on the narrowband signal model in the frequency domain and represents a first attempt to combine data-centric approaches with classical signal processing techniques for beamforming. In the last section, we present directions for further work wherein we shall consider other aspects (such as spectral leakage and windowing effects) that affect narrowband beamforming.

2 SIGNAL MODEL AND CLASSICAL BEAMFORMING METHODS

We consider an array of M microphones at positions $\mathbf{r}_m = (x_m, y_m, z_m)$ sensing the sound field, which is modelled as the superposition of the wave fields generated by N acoustic sources located at $\mathbf{q}_n = (x_n, y_n, z_n)$.

As described in [10, 11, 12], the time-domain samples of each microphone are segmented into frames of K samples, and each frame is converted to the frequency domain using the discrete Fourier transform (DFT). In the presence of additive noise, the $M \times 1$ array output vector for a single frequency ω_k ($0 < k < K/2$) on a single frame can be modelled, according to [13], as

$$\mathbf{x}(\omega_k) = \mathbf{V}(\omega_k)\,\mathbf{y}(\omega_k) + \eta(\omega_k), \tag{1}$$

where $\mathbf{y}(\omega_k) = [y_0(\omega_k)\ \ y_1(\omega_k)\ \cdots\ y_{N-1}(\omega_k)]^T$ represents the source signals in the frequency domain, and $\eta(\omega_k)$ represents additive noise in frequency-domain. The array manifold matrix

$$\mathbf{V}(\omega_k) = [\mathbf{v}(\mathbf{q}_0, \omega_k) \quad \mathbf{v}(\mathbf{q}_1, \omega_k) \quad \cdots \quad \mathbf{v}(\mathbf{q}_{N-1}, \omega_k)], \tag{2}$$

of size $M \times N$, describes the transfer function between source n and sensor m at frequency ω_k. The information on the propagation medium and the characteristics of the propagating wave (i.e., plane or spherical waves, free-field or reverberant conditions, etc.) is encoded in the so-called *steering vector* $\mathbf{v}(\mathbf{q}_n, \omega_k)$ [14]. Furthermore, if the sources are in the far-field, then the sound field can be modelled as a superposition of plane waves and the dependency on source position $\mathbf{q}_n$ can be replaced by its location in *angular* co-ordinates of azimuth and elevation only. We shall subsequently consider linear arrays, whereby the source location for the far-field model is completely specified by the angle θ_n formed between the array axis and the direction of propagation of the wave. As all calculations are done for each frequency independently, we omit the frequency variable ω_k from this point on for the sake of brevity.

2.1 Delay-and-Sum Beamformer (DSB)

Classical methods estimate the source signal from the microphone signals based on spatial filtering, which is implemented as a weighted sum of the signals captured by the sensors [2]. With $\mathbf{w}$ being the complex weight vector, the spatial filtering output can be obtained as:

$$\hat{\mathbf{y}} = \mathbf{w}^H \mathbf{x}. \tag{3}$$

If $\mathbf{w}$ assumes the form of the normalized steering vector $\mathbf{v}(\theta)$, we obtain the frequency domain analogue of the DSB:

$$\mathbf{w}_{\text{BF}} = \frac{\mathbf{v}(\theta)}{\|\mathbf{v}(\theta)\|}. \tag{4}$$

This can can be interpreted as the result of maximizing the signal-to-noise ratio when the array is excited by a plane wave arriving from a desired direction θ and where the background noise is white and uncorrelated across the different microphones [2]. Note that the weights, in this case, do not depend on the incoming signal but only on the geometry of the problem (described in $\mathbf{v}(\theta)$).

2.2 Superdirectional Beamformer

According to [6], "the term *superdirectivity* describes the ability of a beamformer to suppress noise coming from all directions without affecting a desired signal from one principal direction." Several methods have been proposed to achieve superdirectionality. Probably the most popular method is the *minimum variance distortionless response*, which minimizes the power of the filter's output while maintaining a unity gain in direction θ. In [15] it is shown that this restriction leads to

$$\mathbf{w}_{\text{MVDR}} = \frac{\mathbf{R}_n^{-1}\mathbf{v}(\theta)}{\mathbf{v}(\theta)^H \mathbf{R}_n^{-1}\mathbf{v}(\theta)}, \tag{5}$$

where $\mathbf{R}_n$ is the autocorrelation matrix of the noise component in $\mathbf{x}$.

As opposed to the conventional beamformer, the weights are now linearly dependent on the noise statistics, which is often difficult to estimate. In many practical application the weights $\mathbf{w}$ are, therefore, calculated for an idealized noise field [6].

3 NEURAL NETWORK BEAMFORMING

3.1 Neural network architecture

The architecture of the neural network can be summarized as a group of multilayer perceptrons (MLP) [16] coupled with the usual beamforming filter block; each MLP on the network uses information from a single frequency and a single frame of the microphone channels (in the frequency domain) as input, and outputs a beamforming vector element for that frequency, as shown in figure 1. The group of MLPs, then, generates a beamforming vector that changes for each frame of the input information; the beamforming filter block, of course, outputs the target signal approximation.

The MLP for each frequency ω_k has two sets of weight parameters (the matrices $\mathbf{P}_{W_1}, \mathbf{P}_{W_2}$) and bias parameters (the vectors $\mathbf{p}_{B_1}, \mathbf{p}_{B_2}$); between the first parameter layer, composed by $\mathbf{P}_{W_1}$ and $\mathbf{p}_{B_1}$, and the second parameter layer, composed by $\mathbf{P}_{W_2}$ and $\mathbf{p}_{B_2}$, a nonlinear function F is applied on the middle neuron layer. Again, as all calculations are done for each frequency independently, we omit ω_k from this point on.

For each frequency, the input of the MLP is the the vector $\overline{\mathbf{x}}$, which is the input channels normalized by the average energy of all channels (separately for each frame), while the output is the weight vector $\mathbf{w}_{\text{NN}}$. The hermitian transpose of the weight vector is then multiplied by the signal vector $\mathbf{x}$, as usual, to obtain the scalar $\hat{y}$, the approximation of the target signal. The network architecture can be described mathematically for each frequency ω_k as follows:

$$\begin{aligned}
\overline{\mathbf{x}} &= \frac{\mathbf{x}}{\|\mathbf{x}\|}, \\
\mathbf{u} &= \mathbf{P}_{W_1} \cdot \overline{\mathbf{x}} + \mathbf{p}_{B_1} \\
\mathbf{z} &= F(\mathbf{u}), \\
\mathbf{w}_{\text{NN}} &= \mathbf{P}_{W_2} \cdot \mathbf{z} + \mathbf{p}_{B_2}, \\
\hat{y} &= (\mathbf{w}_{\text{NN}})^H \cdot \mathbf{x}.
\end{aligned} \tag{6}$$

This is a very straightforward neural network for each frequency, albeit with one significant point of departure, namely, even though the output of each MLP is the beamforming vector, what the NN actually generates is the target signal approximation. This allows training not with the unknown beamforming weight vectors for each

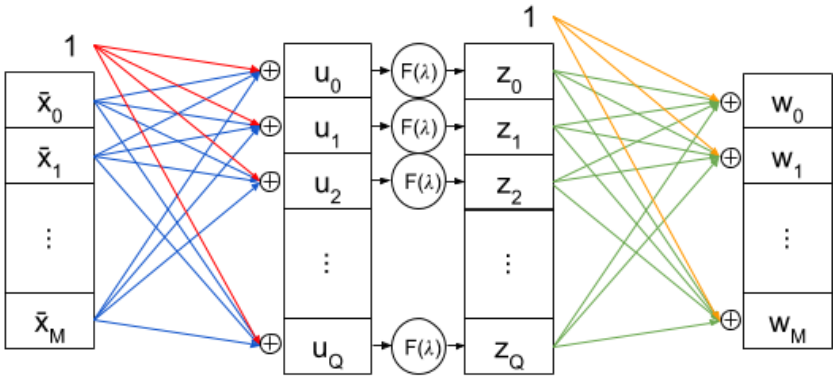

Figure 1. MLP diagram for each frequency; each colored arrow represents one of the MLP parameters (the arrows multiplying the "1" value being the bias parameters) with a total of two parameter layers; blue and green representing the weight parameters $\mathbf{P}_{W_1}$ and $\mathbf{P}_{W_2}$, red and orange representing the bias parameters $\mathbf{p}_{B_1}$ and $\mathbf{p}_{B_2}$.

frame as a target, but with the desired signal $\hat{y}$ as the output target. Therefore, this training resembles a form of autoencoder [17], but in which the decoded data is the desired signal $\hat{y}$ instead of the input signal $\bar{\mathbf{x}}$, and instead of compressing the signal as in the original use of autoencoders, these layers are used to analyze the mixed signal information. A simple representation of the architecture, for a single frequency, is shown in figure 1. The network uses ten middle layer neurons for each microphone channel. We consider an eight microphone array, thus the middle layer of each MLP has 80 neurons with one such MLP for each value of the frequency range.

The normalization of the input of the network is a staple procedure in neural network training, but this is especially important in this context since the output of the network is generated by using the input information twice; the normalization for the generation of the beamforming weights is necessary so that these weights don't change just by varying the gain of the microphone channels signal $\mathbf{x}$.

The network nonlinear activation function chosen for this MLP was the following variation of the sigmoid function for complex values

$$F(u) = \frac{u}{\sqrt{\lambda^2 + |u|^2}}. \tag{7}$$

The constant λ has a different value for each frequency, and is also a trained parameter of the neural network. It helps to change the slope of the sigmoid with reduced impact in the simultaneous adjustments to the parameters of $\mathbf{P}_{W_1}$ and $\mathbf{p}_{B_1}$ during training. This improvement essentially creates another set of network parameters $\lambda(\omega_k)$. This choice of nonlinear activation function has the benefits of preserving the phase of the result of the previous layer operations, while introducing a bounded nonlinearity that allows the network operation to approximate any continuous function in a compact subset of inputs $\bar{\mathbf{x}}$ [18].

3.2 Training dataset

As mentioned previously, instead of defining a set of beamforming weights as the training target, the desired signal approximation is the final output in the proposed network architecture. Since this is a beamforming technique that relies strictly on the available geometric information on the channels, with each frequency treated separately, it is possible to use training source signals that are completely simulated.

Training was done with samples composed of one desired signal direction, one interfering signal direction, random amplitudes and phases for both signals, and random uncorrelated additive noise. Since the beamformer coefficients are dependent on the configuration of the array we consider, here, a uniform linear array (ULA) of 8 microphones, each 8 cm apart, and we trained the system for the case where the sources are set to be 1 m away from the center of the ULA. The reference signal for the training was the signal coming from the target direction and arriving at the microphone closest to the geometrical center of the array. The desired angle ranges were defined as the *broadside direction*, $(-10°, 10°)$, and the *lateral direction*, $(30°, 50°)$, with the interfering angle ranges set to be the remaining angles in the range $(-90°, 90°)$ for each case.

The purpose of the uncorrelated additive noise is to make the network learn to discriminate against variations in the channel information that do not arise from the geometric model of the dataset. The introduction of the uncorrelated noise in the input of the network allows learning this behavior, which would be different from usual approaches such as introducing a constraint on the norm of the beamforming vector (as done for the MVDR beamformer in [6]).

3.3 Training method

The dataset described previously clearly has no defined number of training samples considering the, in principle, near infinite number of variations of source signals, desired and interfering specific directions, and uncorrelated noise for each sample. The only limitation on the number of these variations is the period of the chosen random number generator period, which is exceedingly large.

As such, the training batch size must only be large enough to account for the training criteria of a large range of angle combinations for each batch, while the number of samples used in each epoch becomes more of a progress analysis parameter than a crucial aspect of training. The loss function used for the gradient descent was the mean squared error of the desired signal approximation, which is particularly interesting considering it is the same as the energy of the desired signal approximation noise. Since the epochs have a negligible chance of repeating itself during the whole training, training doesn't show signs of overfitting, and the stopping criteria is a inconsequential loss reduction over a number of epochs.

4 RESULTS

First and foremost, it is important to note that while the results shown in this section are the beampattern features of the NN beamformer and its comparison with classical beamformers, the training objective described in the previous section is not directly the improvement of the beampattern. Its actual objective is minimizing the energy of the target signal approximation noise; the beampattern evaluation only uses one signal source per sample, without uncorrelated noise addition, sweeping all angles to evaluate the beamformer on the desired and interference ranges. Figures 2, 3, and 4, show the beampattern comparison between the results for the DSB, MVDR, and the proposed NN approaches. Figure 5 shows the WNG for the NN approach.

One of the major differences between these methods is that the classical approaches are insensitive to gain and phase variations in the source signals and, as such, there is no variation in the beampattern gain or phase difference; their beampatterns are exact. The NN method, on the other hand, has a changing beamforming vector that is derived using the input signal, so that there are variations in their beampattern gain and phase difference that depend on the relative gains and phases of the desired and interference signals, as well as the uncorrelated noise. Because of these variations we also show, alongside gain and phase difference, their standard deviation on figures 2 and 3. Another important difference is that the current approach, because of its stochastic training method, shows slight angle asymmetry, while the classical methods are exactly symmetric for the broadside target.

4.1 Broadside DSB, MVDR, and NN beamformers comparison

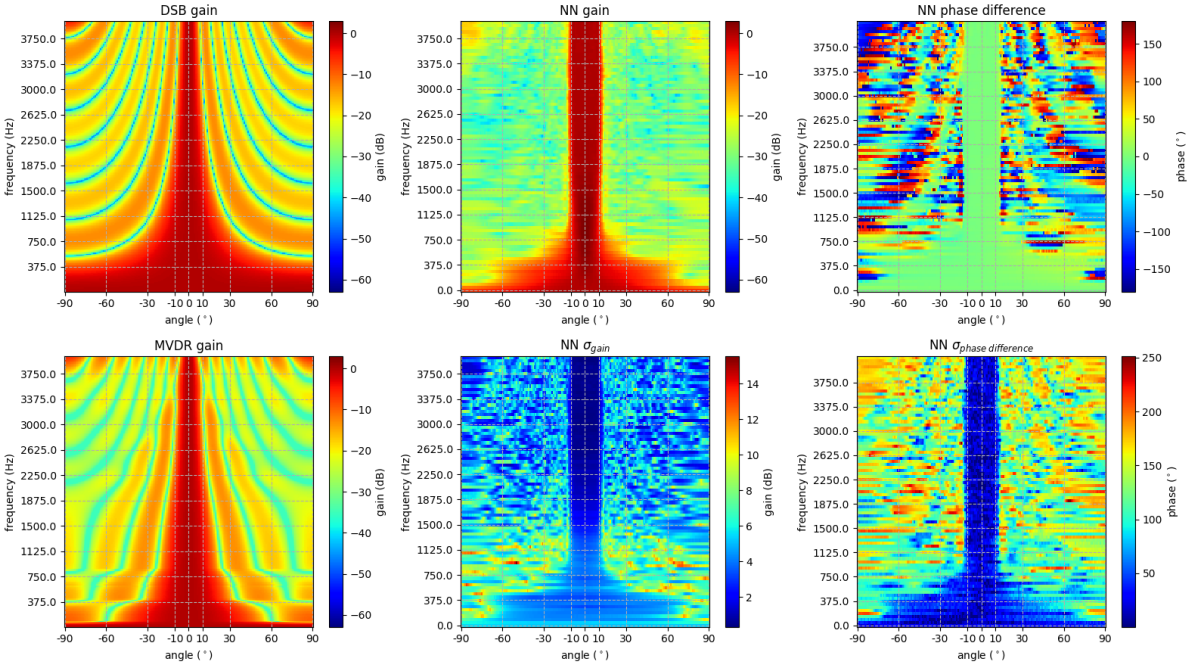

Figure 2. Beampattern results for the broadside range $(-10°, 10°)$, showing classical beamformer results, the NN gain, phase difference, and their standard deviations.

Figures 2 and 4 present NN results with attenuation outside of the target angle range that is at least 10 dB and 5 dB lower than the sidelobe tips of, respectively, the DSB and MVDR beampatterns.

In higher frequencies, both DSB and MVDR results present pronounced sidelobes near the endfire angles, with nearly the same energy as the mainlobe; the NN results in the same range still show general attenuation of 20 dB or more, with small regions of 10 dB attenuation. Also, for frequencies above 1 kHz, the network beampattern shows a very sharp change between the mainlobe and the interference range, with the interface between desired and interference angles displaying a rate of change that is larger than 20 dB in less than 5°.

For lower frequencies, the desired and interfering signals have very small phase difference, so that classical beamformers display reduced interference attenuation in this range. Since the NN beamformer is a data centric approach and the neural network is being trained with target and interfering signals, it's to be expected that the network will show increased gain standard deviation for lower frequencies in the beampattern, exactly because the NN is attempting to predict how to vary the gain to remove an interference that isn't present during the beampattern evaluation.

4.2 Lateral DSB, MVDR, and NN beamformers comparison

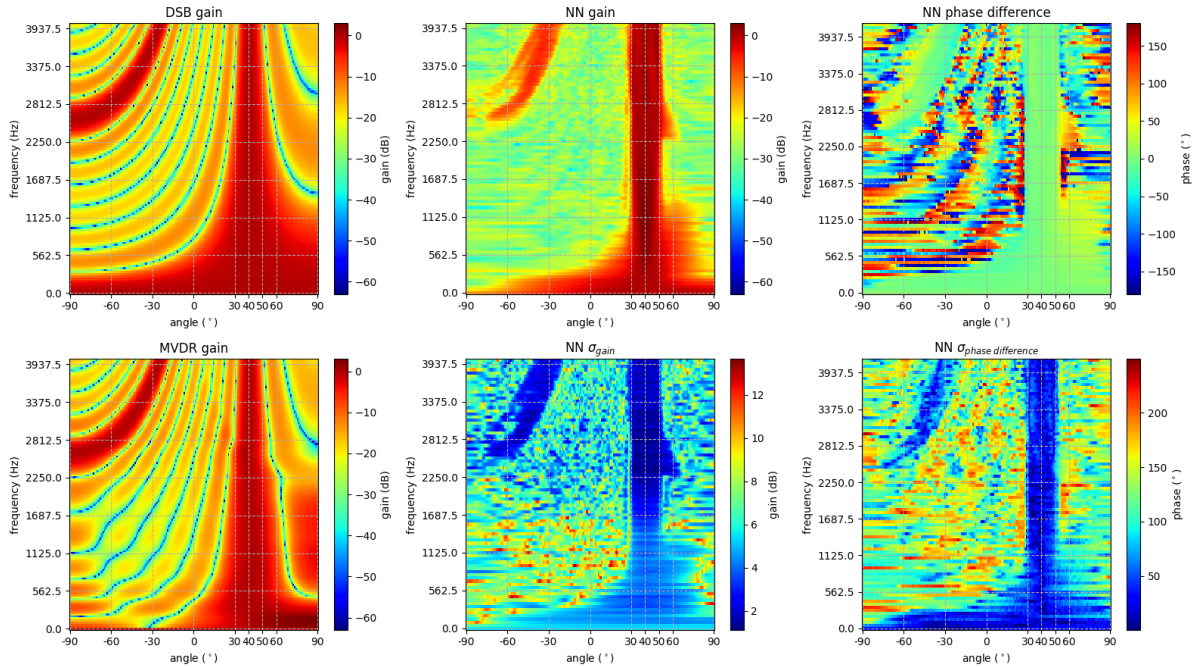

Figure 3. Beampattern results for the lateral range $(30°, 50°)$, showing classical beamformer results, the NN gain, phase difference, and their standard deviations.

The results shown in figures 3 and 4 for beamformers with a main target angle of 40° show that the classical approaches lose their benefit of symmetric beamforming around the intended target. The NN beamformer for the $(30°, 50°)$ range, on the other hand, exhibits an appreciably more constant beamwidth along the frequency axis; however, it still displays asymmetry for frequencies near the range of the high sidelobe that appears in the negative angles for frequencies above 2.5 kHz. This slight asymmetry of the mainlobe is the trade off for the attenuation of the sidelobe: this result, outside of stochastic effects, is caused by the coupled training targets of generating the optimal approximation for signals in the target range and the maximal attenuation for signals outside the same range.

We can also say that the sidelobe is better attenuated with the NN approach, appearing on a more constrained region of the graph and with a slightly lower maximum (at least 5 dB attenuation). The trade off for the reduced sidelobe is a slight deformation in the mainlobe for high frequencies, but this deviation is below 2 dB.

The benefit of training the network with the addition of small amplitude uncorrelated additive noise is shown on figure 5, where we see that the median of the WNG lies below 0 dB for frequencies of interest in both the broadside and lateral angle ranges. This result indicates that, even though the current architecture has a small number of layers, it is able to achieve a better balance of directivity and uncorrelated noise suppression, even for low frequencies; interestingly, the result for the broadside range is better than for the lateral range. It's important

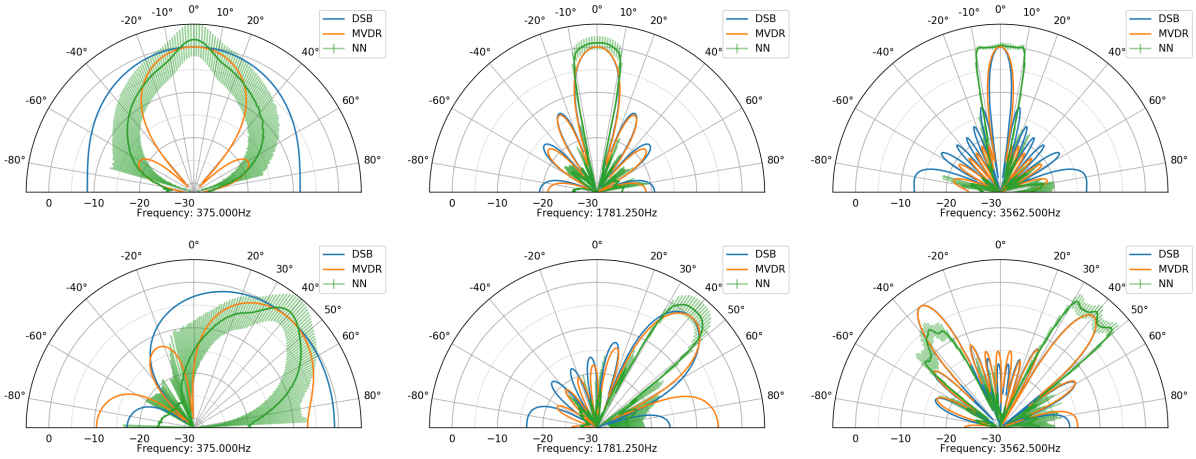

Figure 4. Comparison of beampattern gains for single frequencies along the frequency range for DSB, MVDR and NN. The three upper plots show results for the broadside range $(-10°, 10°)$, while the three lower plots do the same for the lateral range $(30°, 50°)$.

to note that the MVDR beampattern shown in figures 2, 3 and 4 is not optimized for white noise gain suppression, in which case it strongly resembles the DSB for low frequencies.

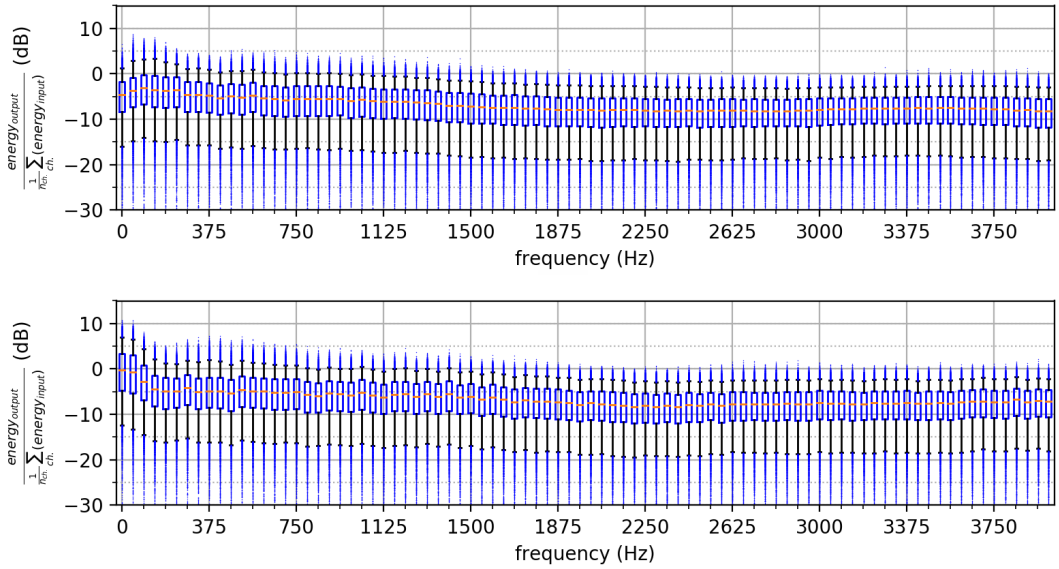

Figure 5. Uncorrelated noise gain for NN beamformer with the target angle ranges $(-10°, 10°)$ on the top, and $(30°, 50°)$ on the bottom; the boxplots show the median, lower to upper quartiles, the 5% to 95% range, and the outliers.

5 FINAL REMARKS

The proposed NN beamformer architecture and training method show several benefits compared to classical beamformers. The improvements seen in the beampattern evaluation are its near-uniform width mainlobe, with very sharp edges in the transition from target to interference angle range. Further, it exhibits better spatial selectivity for low frequencies as well, without an appreciable white noise gain (WNG). Note that the MVDR beamformer presented in comparison is *not* optimized with respect to the WNG, hence the selectivity at low frequencies is rather optimistic. This is a first proof of concept that shows significant results for near field applications, and we intend to do further research in the application of these techniques for the far field case. Also, the current network design is robust to the uncorrelated noise amplification, a feature that is quite beneficial, especially for lower end or *ad hoc* microphone array configurations. For our next steps, we aim to address interfrequency effects in the narrowband model and investigate the use of interfrequency correlations to improve the beamformer performance.

ACKNOWLEDGEMENTS

This work was partially funded by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES) through its Academic Excellence Program (PROEX), and also by the São Paulo Research Foundation (FAPESP), grant #2017/08120-6.

REFERENCES

[1] Brandstein M, Ward DB. Microphone Arrays. 2nd ed. Brandstein M, Ward D, editors. Digital Signal Processing. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001.

[2] Nascimento VH, Masiero BS, Ribeiro FP. Acoustic Imaging Using the Kronecker Array Transform. In: Coelho RF, Nascimento VH, de Queiroz RL, Romano JMT, Cavalcante CC, editors. Signals and Images: Advances and Results in Speech, Estimation, Compression, Recognition, Filtering, and Processing. CRC Press; 2015. p. 153–178.

[3] Dudgeon DE, Mersereau RM. Multidimensional Digital Signal Processing. 2nd ed. Prentice-Hall Signal Processing; 1995.

[4] Laakso TI, Välimäki V, Karjalainen M, Laine UK. Splitting the Unit Delay - Tools for Fractional Delay Filter Design. IEEE Signal Processing Magazine. 1996 Jan;13(1).

[5] Krim H, Viberg M. Two decades of array signal processing research: the parametric approach. IEEE Signal Processing Magazine. 1996 jul;13(4):67–94.

[6] Bitzer J, Simmer KU. Superdirective Microphone Arrays. In: Microphone Arrays. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001. p. 19–38.

[7] Doclo S, Moonen M. Design of far-field and near-field broadband beamformers using eigenfilters. Signal Processing. 2003 dec;83(12):2641–2673.

[8] Slepian D. Prolate spheroidal wave functions, Fourier analysis and uncertainty–IV: Extension to Many Dimensions; Generalized Prolate Spherical Functions. Bell Syst Tech J. 1964;43(6):3009–3057.

[9] Pezeshki A, Van Veen BD, Scharf LL, Cox H, Lundberg Nordenvaad M. Eigenvalue Beamforming Using a Multirank MVDR Beamformer and Subspace Selection. IEEE Transactions on Signal Processing. 2008 may;56(5):1954–1967.

[10] Madhu N. Acoustic source localization: algorithms, applications and extensions to source separation [PhD]. Uni Bochum; 2009.

[11] Madhu N, Gückel A. Multi-channel source separation: Overview and comparison of mask-based and linear separation algorithms. In: Machine Audition: Principles, Algortihms and Systems. IGI Global, USA; 2010. .

[12] Masiero BS, Nascimento VH. Revisiting the Kronecker Array Transform. IEEE Signal Processing Letters. 2017;24(5):525–529.

[13] Johnson DH, Dudgeon DE. Array signal processing concepts and techniques. Prentice Hall, Englewood-Cliffs N.J.; 1993.

[14] Gannot S, Vincent E, Markovich-Golan S, Ozerov A. A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation. IEEE/ACM Transactions on Audio Speech and Language Processing. 2017;25(4):692–730.

[15] Van Trees HL. Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory. vol. 4. New York, USA: John Wiley & Sons, Inc.; 2002.

[16] Haykin S. Neural Networks: A Comprehensive Foundation. 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall; 2007.

[17] Kramer MA. Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal. 1991 feb;37(2):233–243.

[18] Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Networks. 1991 jan;4(2):251–257.