COMPLEX VERSUS REAL-VALUED NEURAL NETWORKS FOR AUDIO SOURCE LOCALISATION USING SIMULATED AND MEASURED DATASETS

Vlad S. Paul **Jacob Hollebon** **Philip A. Nelson**
[1] Institute of Sound and Vibration Research
[2] University of Southampton
[3] Southampton, SO17 1BJ, United Kingdom

ABSTRACT

Complex-valued neural networks can accept complex-valued data as an input and present an alternative to their real-valued counterparts. This can be advantageous for various audio signal processing applications such as for audio source localisation utilising microphone arrays. This paper builds on previous work aimed at comparing the performance of complex and real valued neural networks under equal operating conditions. Furthermore, this work investigates the performance of both types of networks in a 3D source localisation task. In this work an evaluation is made of the performance of networks that are trained using simulated microphone signals but which are then applied to the outputs of real microphone signals. This has advantages due to the simplicity in creating large datasets for the training phase. Both networks are compared to MUSIC, a common classical localisation technique. Results show that both network types can learn from simulated data to localize measured data, although their performance depends on the features with which the networks are trained.

Keywords: Audio source localisation, Complex-valued neural networks, Microphone array, MUSIC

1. INTRODUCTION

Machine learning approaches have shown impressive performance for use in sound source localisation (SSL) compared to classical signal processing techniques, with various neural network architectures developed to solve different localisation tasks. Various surveys [1, 2] and SSL challenges such as *Detection and Classification of Acoustic Scenes and Events (DCASE)* and *Learning 3D Audio Sources (L3DAS)* show the potential of machine learning approaches in this area and the rapid increase in the popularity of their use.

Whilst neural networks are able to learn patterns from a training dataset, one of the main issues is their generalization performance. Datasets which contain training material that is too similar have the risk of overfitting the neural networks, which means that the network can learn the training data well, but cannot generalize to new data, especially if new data is too different from the training material. If the training material is too different, the network can struggle to learn any patterns.

One solution is to build large datasets which contain enough differing material from various scenarios in order to make the network as robust as possible. A large dataset can be easily achieved if one uses simulated data, although it is well known that simulated data is not always representative of the real-world alternative. For example, simulated microphone array data may be easily created under a free field assumption, whereas real-world data usually contains reverberation or scattering effects. In addition to this, especially in audio signal processing, gathering a large set of measurements to build datasets can be difficult and expensive.

Recently, researchers have tried to close the gap between

**Corresponding author*: *vsp1g18@soton.ac.uk.*

synthetic and measured data for applications such as autonomous driving [3], radar [4] or robotics [5]. Whilst the basic idea is to use only simulated data for training and real data for testing the neural networks, one needs to develop an understanding of the ideal training features that extract enough useful information from synthetic data which characterise the measured data. Also, the type of network used can have a significant influence on the performance quality.

This work considers how well a simple multilayer perceptron (MLP) network can learn from only synthetic microphone array signals to localize sources in a 3D space using measured microphone signals as testing material. We focus on comparing a real-valued MLP (rMLP) with a complex-valued MLP (cMLP) using different training features and various network sizes. The localization performance is compared to a classical signal processing algorithm (MUSIC), which is a well-known subspace method for localizing sound sources.

2. REAL- AND COMPLEX-VALUED MULTILAYER PERCEPTRON (MLP)

The architecture of a simple multilayer perceptron (MLP) with one hidden layer is shown in Figure 1. Regardless of the type of MLP network (real- or complex-valued), the forward propagation is given by

$$\mathbf{a}^{(2)} = \mathbf{W}^{(2)}\mathbf{x} + \mathbf{b}^{(2)} \quad (1)$$

$$\mathbf{z}^{(2)} = h(\mathbf{a}^{(2)}) \quad (2)$$

$$\mathbf{a}^{(1)} = \mathbf{W}^{(1)}\mathbf{z}^{(2)} + \mathbf{b}^{(1)} \quad (3)$$

$$\mathbf{z}^{(1)} = h(\mathbf{a}^{(1)}) = \hat{\mathbf{y}}, \quad (4)$$

where $\mathbf{W}^{(2)}, \mathbf{W}^{(1)}$ correspond to the matrices of weights and $\mathbf{b}^{(2)}, \mathbf{b}^{(1)}$ correspond to the bias terms in the hidden and output layer. The input into the hidden layer $\mathbf{a}^{(2)}$ is related to the output of the hidden layer $\mathbf{z}^{(2)}$ by a nonlinear activation function $h()$. Note that the layers are counted from the output backwards in order to enable an easier derivation of the backpropagation algorithm for a MLP network with l number of hidden layers. For a real-valued MLP, all terms in the forward propagation are real, whilst for the complex MLP, all variables, including the activation function $h()$, are complex-valued. The loss function L used during training for this particular work is the mean-squared error, which differs between the rMLP and cMLP (subscript r, c respectively) and is given below

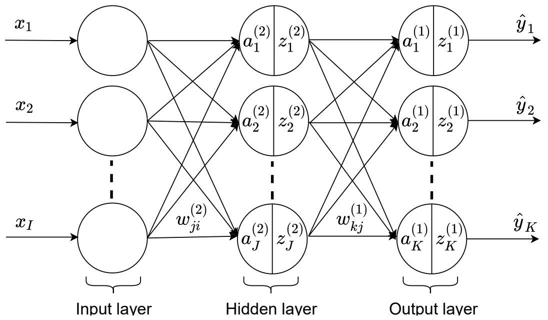

Figure 1: MLP model with one hidden layer

$$L_{\mathrm{r}} = \frac{1}{2K}\sum_{k=1}^{K} d_k^2, \quad \text{where} \quad d_k = \hat{y}_k - y_k \quad (5)$$

$$L_{\mathrm{c}} = \frac{1}{2K}\mathbf{d}^{\mathrm{H}}\mathbf{d} = \sum_{k=1}^{K}\frac{1}{2} \mid d_k \mid^2, \quad \text{where} \quad d_k = \hat{y}_k - y_k. \quad (6)$$

The authors derived a detailed backpropagation algorithm for the real MLP in matrix form in [6] and the gradient equations to update the weights and biases are given below in their final form for a general l-th set of weights.

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(L)}\mathbf{z}^{(l+1)\mathrm{T}} \quad (7)$$

$$\frac{\partial L}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)}, \quad (8)$$

where $\boldsymbol{\delta}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}^{(l-1)\mathrm{T}}\boldsymbol{\delta}^{(l-1)}$ and where $\mathbf{H}^{(l)}$ is the diagonal matrix of derivatives of the activation function $h()$ at the l-th layer. The variable $\mathbf{z}^{(l+1)\mathrm{T}}$ is always the input to the set of weights preceding the l-th hidden layer. If $l = 1$, $\mathbf{z}^{(1)\mathrm{T}} = \mathbf{x}^{\mathrm{T}}$ and $\boldsymbol{\delta}^{(1)} = \mathbf{H}^{(1)\mathrm{T}}\mathbf{d}$, where $\mathbf{d}^{\mathrm{T}} = [d_1, d_2, ..., d_K]$. For the complex-valued MLP, the derivation of the backpropagation was presented in detail in [7] and the gradient equations for the same set of weights and biases at the l-th layer are given by

$$\frac{\partial L}{\partial \mathbf{W}^{(l)*}} = \frac{1}{2}\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\boldsymbol{\delta}^{(l)}\mathbf{z}^{(l+1)\mathrm{H}} \quad (9)$$

$$\frac{\partial L}{\partial \mathbf{b}^{(l)*}} = \frac{1}{2}\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\boldsymbol{\delta}^{(l)}, \quad (10)$$

where $\boldsymbol{\delta}^{(l)} = \widetilde{\mathbf{H}}^{(l)\mathrm{T}}\widetilde{\mathbf{W}}^{(l-1)\mathrm{T}}\boldsymbol{\delta}^{(l-1)}$ and where $\widetilde{\mathbf{H}}^{(l)}$ denotes the composite matrix that contains the diagonal matrices of derivatives of activation functions at the l-th hidden layer and is of form

$$\widetilde{\mathbf{H}}^{(l)} = \begin{bmatrix} \mathbf{H}^{(l)} & \hat{\mathbf{H}}^{(l)} \\ \hat{\mathbf{H}}^{(l)*} & \mathbf{H}^{(l)*} \end{bmatrix}, \tag{11}$$

where $\mathbf{H}^{(l)} = diag\left(\frac{\partial h(z_1^{(l)})}{\partial a_1^{(l)}} \dots \frac{\partial h(z_K^{(l)})}{\partial a_K^{(l)}}\right)$ and $\hat{\mathbf{H}}^{(l)} = diag\left(\frac{\partial h(z_1^{(l)})}{\partial a_1^{(l)*}} \dots \frac{\partial h(z_K^{(l)})}{\partial a_K^{(l)*}}\right)$. In Equation (9), $\mathbf{I}$ corresponds to an identity matrix and $\mathbf{0}$ is a matrix of zeros. $\widetilde{\mathbf{W}}^{(l-1)}$ denotes the composite matrix of the weight matrix at the $(l-1)$-th layer and is of form

$$\widetilde{\mathbf{W}}^{(l-1)} = \begin{bmatrix} \mathbf{W}^{(l-1)} & 0 \\ 0 & \mathbf{W}^{(l-1)^*} \end{bmatrix}. \tag{12}$$

Similarly to the real MLP, the vector $\boldsymbol{\delta}^{(l)}$ contains the gradient terms up to the input into the l-th layer. In other words, for $l = 1$, the vector becomes $\boldsymbol{\delta}^{(1)} = \widetilde{\mathbf{H}}^{(1)\mathrm{T}}\tilde{\mathbf{d}}$ and $\tilde{\mathbf{d}} = \begin{bmatrix}\mathbf{d}^* & \mathbf{d}\end{bmatrix}$, where $\mathbf{d} = \hat{\mathbf{y}} - \mathbf{y}$.

It is important to note that the backpropagation equations presented here were derived to work for any type of complex-valued activation functions, holomorphic or non-holomorphic, so that one is not limited by the choice of activation function. A detailed discussion of the differences between complex-valued functions can be found for example in [8].

3. EXPERIMENTS

3.1 Dataset generation

Two main datasets were used. One was based on simulations of sound interacting with a rigid sphere. The other was based on measurements of the sound fields generated in a semi-anechoic laboratory space provided by the Audiolab at the Institute of Sound and Vibration Research in Southampton, shown in Figure 2. The measured signals were generated by loudspeakers surrounding an Eigenmike having 32 capsules [9] that was placed in the middle of the semi-anechoic room. The total number of loudspeakers available was 39 but only 10 of these were used to evaluate the ability of the neural networks to localise sources. The positions of the sources used for localisation by the neural networks were reasonably evenly spread in angular location and these are shown in Figure 3. The positions of all 39 sources were used in evaluating the effectiveness of the MUSIC algorithm, as described in more detail below. In constructing the datasets for training and evaluation of both neural networks and the MUSIC algorithm, speech, guitar and noise signals were used. Out of the 32 channels available from the Eigenmike, only 4 of these were used to create the dataset. The channels were chosen to produce a tetrahedral sensor array with microphones having the following spherical coordinates: $(45°, 35°, 4.2\text{ cm})$, $(-45°, -35°, 4.2\text{ cm})$, $(135°, -35°, 4.2\text{ cm})$ and $(-135°, 35°, 4.2\text{ cm})$, where 4.2 cm corresponds to the radius of the sphere. The chosen microphones correspond to channels 6, 10, 26 and 22 of the Eigenmike. The use of these channels is motivated by the fact that the multichannel recording consists of time

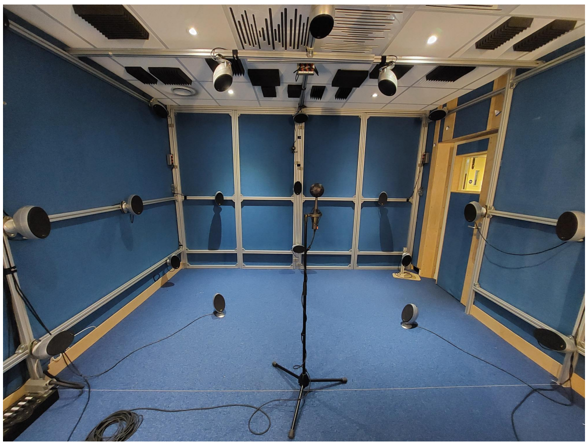

Figure 2: Room, microphone array and loudspeaker array used for the dataset generation.

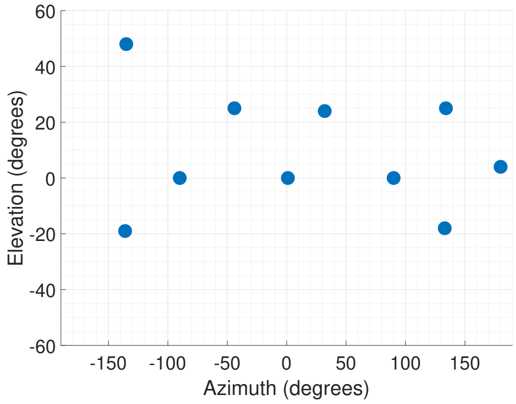

Figure 3: Chosen loudspeaker positions for the dataset generation.

differences due to the spacing between the microphones and level differences due to the sound scattering generated by the rigid sphere [10].

The simulated dataset was created by convolving the mono input signal (speech, guitar or noise) with simulated impulse responses from the 10 specified source positions to the four microphone positions on the surface of the microphone array. Each source was assumed to produce a plane wave incident on a rigid sphere of radius a in free field, assuming the microphones have a perfect omnidirectional response with negligible capsule size situated on the surface of the rigid sphere. In the frequency domain, the acoustic pressure response p given by this model is [11]

$$p(\Theta, k) = \frac{1}{(ka)^2} \sum_{n=0}^{\infty} \frac{(2n+1) j^{n+1} P_n(\cos(\Theta)}{h_n'(ka)}, \qquad (13)$$

where Θ is the angle between the position on the rigid sphere surface and the incident source position (given by the dot product between the two unit norm vectors pointing towards the microphone and source position). The wavenumber is denoted by k whilst j is the imaginary unit, P_n is the n-th order Legendre polynomial and h_n' is the derivative of the n-th order spherical Hankel function of the second kind. In practise, the infinite summation must be truncated and in this work a maximal order $N = 80$ was used. This corresponds to accurate representation of the acoustic pressure across the entire audible spectrum as per the $N = ka$ rule of thumb [12]. A radius of $a = 0.042$ m was used to match that of the Eigenmike. Following this, the final training and testing datasets for each signal type consisted of 4-channel recordings (simulated or measured) of 3 types of signals from 10 different source positions. For the training stage using simulated data, each signal was split into frames of 128 samples, which corresponds to a frame duration of 2.7 ms at a sampling frequency of 48 kHz. The frame was long enough to capture the relative delays between the 4 microphones. The total number of frames was split randomly into 80% used for training and 20% used for validation. This helps to avoid issues during training such as overfitting. The measured data was only used to test the performance of the networks after they were trained on the simulated recordings. Both network types were trained in turn by using all 10 source positions to localise either the speech, guitar or noise signals.

3.2 Feature generation

Two different input features were used separately to train the networks and compare their performance. The first chosen feature is the basic Fast Fourier transform (FFT) of the 128 samples time frame, where the 4 channels were concatenated into a 1-dimensional complex-valued vector. For the rMLP, the real and imaginary parts of the 1-dimensional vector were concatenated in order to create a single real-valued input vector. In the case of the cMLP, the 1-dimensional complex-valued vector was used directly as an input feature. The second input feature that was used is based on the well-known generalized cross-correlation with phase transform (GCC-PHAT), which has been used extensively for training machine learning algorithms to localise sound sources. In this work, only the normalized cross-power spectrum (CPS) is used and can be defined as

$$CPS(k) = \frac{X_i(k) X_j(k)^*}{\mid X_i(k) X_j(k)^* \mid}, \qquad (14)$$

where X_i, X_j are the N-point FFT spectra of any two different microphones ($i \neq j$) and k corresponds to the frequency bin, which should not be confused with the wavenumber k used above in Equation (13).

Since each recording consisted of 4 channels, the CPS at each frequency bin k had 6 values, corresponding to all possible combinations of pairs of channels. For the cMLP, the CPS of all pairs of microphones were concatenated into a 1-dimensional complex-valued vector, whilst for the rMLP, the real and imaginary parts were again concatenated to create a real-valued input vector. The estimated output of the networks $\hat{\mathbf{y}}$ was compared to a target output $\mathbf{y}$ which consisted of the source locations expressed in terms of azimuth $\phi \in [-180°, 180°]$ and elevation angles $\theta \in [-90°, 90°]$ using Eulers formula $e^{j\theta} = \cos(\theta) + j\sin(\theta)$. For example, if a source is positioned at $(\phi, \theta) = (60°, 30°) = (\pi/3, \pi/6)$, the target output $\mathbf{y}$ is $(e^{j\frac{\pi}{3}}, e^{j\frac{\pi}{6}})$. This was done in order to enable a fair comparison between the real- and complex-valued networks, where for the rMLP the target output layer consisted of the real and imaginary values of the target location concatenated into a two element vector, whilst for the cMLP the output of the Euler formula was used directly as the target output.

3.3 Benchmark technique

The MUSIC algorithm [13] was implemented as a benchmark technique. The MUSIC algorithm is a subspace method that computes a pseudospectrum, the peaks of which occur at the angular locations of the sources. The algorithm uses a basic peak finding method to detect the angular locations of the highest peaks. Assuming a multichannel mixture of sources $\mathbf{x}(k)$ at the k-th frequency bin, where the size of the vector $\mathbf{x}$ corresponds to the number of microphones, the first step is to compute the cross-correlation matrix $\mathbf{S}_{xx} = E[\mathbf{x}\mathbf{x}^{\mathrm{H}}]$. The matrix $\mathbf{S}_{xx}$ is square, its dimensions given by the number of microphones. The next step is to compute the eigenvalue decomposition of $\mathbf{S}_{xx}$ such that the singular values are placed in a descending order based on their magnitude. The matrix of eigenvectors is divided into a signal subspace $\mathbf{U}_s$ corresponding to the first few singular values, and a noise subspace $\mathbf{U}_n$ corresponding to the remaining singular values. The number of singular values corresponding to the signal subspace are equal to the number of sources to be localised, whilst the rest of the singular values correspond to the noise subspace. Based on the fact that the signal subspace of the sources to be localised is orthogonal to the noise subspace, a MUSIC pseudospectrum can be computed from

$$P_{MUSIC}(\theta, \phi) = \frac{1}{\mathbf{a}(\theta, \phi)^{\mathrm{H}}\mathbf{U}_n\mathbf{U}_n^{\mathrm{H}}\mathbf{a}(\theta, \phi)}, \tag{15}$$

where $\mathbf{a}$ corresponds to the steering vectors from all microphones to a direction determined by (θ, ϕ). The pseudospectrum can be computed for a set of values of angular source positions distributed over a spherical surface surrounding the microphones. It follows from the orthogonality property of signal and noise subspaces that the pseudospectrum should show a peak at the angular location from which sound is arriving. The steering vectors are given by the frequency response functions relating the pressures at the microphone array to the source output. These are in turn related to the Fourier transforms of the equivalent impulse responses. Based on the array geometry and the direction of arrival of the plane wave source, steering vectors can be computed, as described for example in [14]. In the case considered here, in order to create a grid of angular locations, steering vectors were generated for all 39 available loudspeaker positions in the Audiolab in Figure 2 . Within these 39 synthesised steering vectors, only 10 corresponding to the chosen loudspeakers were used for the localisation experiment. The synthesised steering vectors were used to evaluate the performance of MUSIC on both simulated and measured data.

It is worth noting that in the case of measured data, MUSIC can use measured impulse responses that are transformed into the frequency domain to give steering vectors for the computation of the pseudospectrum. However, in the case dealt with here, only the synthesised steering vectors were used to evaluate how well the MUSIC algorithm can make use of simulated information to localise real recordings.

3.4 Network parameters

Both rMLP and cMLP networks were implemented in MATLAB using two hidden layers of 100 neurons each. The network architecture was kept simple using two hidden layers, since the main aim of the paper is to compare the performance of the real-and complex-valued networks using a like-for-like comparison, rather than necessarily finding the best network design to solve this task.

For the case where the FFT spectrum is used as an input feature, using a 128 point FFT, the input layer for the cMLP network was 260 values long, which correspond to the 65 FFT bins of the 4 microphone channels concatenated into a complex-valued vector. For the rMLP, the real and imaginary values of the cMLP input layer were concatenated to form a 520 samples long real-valued vector. When the CPS was used as input feature, the cMLP network had an input layer of 390 values and the rMLP network had an input layer of 780 values. Due to the fact that the dimensions of both input and output layers are doubled in the rMLP case, the size of the hidden layers was also doubled compared to the cMLP case. The activation function used in the hidden layers of the cMLP was the complex-cardioid function, which was introduced in [15] and is a complex-valued extension of the ReLU function used in the rMLP case. Both rMLP and cMLP had the $tanh()$ activation function in the output layer. This was chosen because the localisation estimates can have values between -1 and 1 and the $tanh()$ function is suitable for this case. Both networks were trained using the stochastic gradient descent algorithm and the training was stopped either after 200 iterations, or if the validation error started to increase, suggesting the occurrence of overfitting.

4. RESULTS

The localisation performance will be evaluated using two scenarios. For both scenarios the networks are trained on simulated data, however the evaluation content is varied between simulated and measured data for the two scenarios. Whilst it is expected that the performance of the networks with measured data will be lower than with simulated data, the question is whether the performance is suf-

Table 1: Localisation performance in degrees using 2 hidden layers with 100 neurons and an FFT length of 128 to create the input features. The angular error ϵ is averaged over 5 different network training trials.

	Noise		Speech		Guitar	
	FFT	CPS	FFT	CPS	FFT	CPS
rMLP	91°	0.8°	92.6°	1°	89.2°	1.4°
cMLP	**82°**	**0.7°**	**82.4°**	**0.9°**	**82.4°**	**1.04°**

Table 2: Localisation performance in degrees using 2 hidden layers with 100 neurons and an FFT length of 128 to create the CPS input feature. The angular error ϵ is averaged over 5 different network training trials.

	Noise	Speech	Guitar
rMLP	27.6°	43.6°	**67°**
cMLP	**21.1°**	**39.9°**	71.7°

ficiently satisfactory considering the ease in generating the dataset. An obvious reason for this expectation is that the simulated data is reflection free, while the measured data was recorded in a semi-anechoic environment.

Table 1 shows a comparison between the rMLP and cMLP when trained and tested on synthesised data using 80% of the simulated data for training and 20% for testing for both input features. The results are generated from localising short bursts of noise, a speech signal or a guitar recording. To assess the error in the localisation estimates from each network, the great circle distance was used to define the angular error between the estimated and target source positions. The angular separation ϵ between the target and estimated positions, $\mathbf{n}_T$ and $\mathbf{n}_E$, is given by the dot product [16]

$$\epsilon = \frac{1}{F}\sum_{f=1}^{F} \arccos(\mathbf{n}_{f_T} \cdot \mathbf{n}_{f_E}), \tag{16}$$

where F corresponds to the total number of estimates from all locations and the error ϵ is averaged over all F estimates. Note here $\mathbf{n}_{T,E}$ are unitary vectors in Cartesian coordinates pointing in the direction of the target/estimated source position. This metric is useful as it combines the error in both azimuth and elevation into one single value that is consistent for source positions over the sphere, and not skewed by positions approaching the sphere poles. The angular error was calculated in turn for each evaluation signal. It can be seen that the localisation estimates of the MLP models are quite close to each other, but the cMLP has a slightly lower angular error than the rMLP overall. The network models are not able to perform well when trained directly on the FFT spectrum. It appeared during the training stage that the networks using the current architectures became trapped in local minima, and were therefore not able to perform well during the testing stage. A possible reason for the poor performance could be the fact that the FFT length was very small (128 points) therefore the input layers contained too little useful information and the networks failed to find the right patterns. The effect of the FFT length on the networks performance will be further investigated. Even so, the cMLP was able to perform slightly better, perhaps becoming trapped in a lower local minimum. When using the CPS as an input feature, the cMLP performs slightly better than the rMLP, however the difference in performance in all scenarios is below 1°. As expected, the best localisation performance happens when the networks are trained on bursts of noise, since this type of signal includes energy at all frequencies, which can be helpful for the network models.

The MUSIC algorithm, perhaps unsurprisingly, performs extremely well when simulated data is used both to determine the steering vectors and to evaluate its performance. Thus the subspace technique outperforms both network types, managing to always estimate the correct position of the loudspeaker that played the signal. Of course the MUSIC localization estimates could only be from the synthesised locations, which were quite sparse (39 points distributed over the sphere) and so the algorithm does not have to scan over a large number of angles. Secondly, the synthesised steering vectors used to estimate the pseudospectrum from Equation (15) are exactly the same as those used to simulate the audio signal arriving from any of the 10 chosen locations.

For the scenario where the trained networks are tested using measured data, Table 2 shows a comparison between the two networks now using the CPS input feature only, since the FFT spectrum showed a relatively poor training performance using simulated data. It can be observed that the networks trained on synthesised data are still able to localise sources using measured data, although the performance is reduced. For both noise bursts and speech signals, the cMLP slightly outperforms its counterpart,

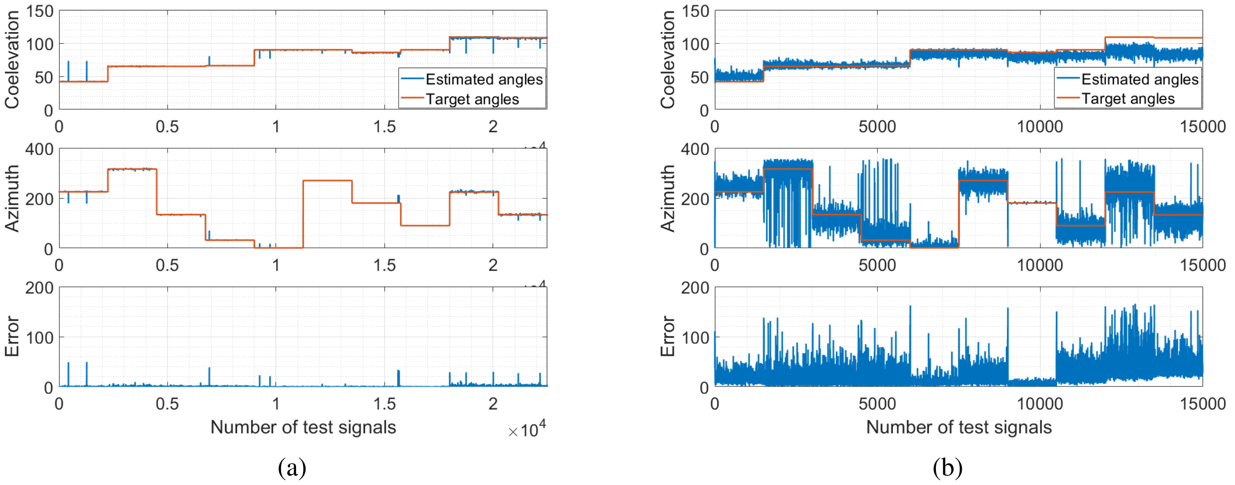

Figure 4: Localisation estimates measured in degrees on the vertical axes of the cMLP using the CPS feature as input when tested on (a) simulated noise signals and (b) measured noise signals.

whilst for the guitar signal, both networks struggle to perform well, with the rMLP being marginally better than the cMLP. Figure 4 illustrates the localisation estimates of the complex-valued MLP for the two testing scenarios (simulated and measured evaluation data) using short bursts of noise as simulated training signal. As expected, the location estimates from the measured data are noisier than those from the simulated data, although the trend of the target angles is followed. The same behaviour is shown in the case of the speech signals, but the networks are not able to localise the measured guitar signals, resulting in an angular error of $67°$ and $71.7°$. This is surprising, since both networks performed well when tested on the simulated dataset. One reason for this could be that the guitar recording is of a higher complexity than the speech and noise signals and the recorded guitar signals were very different to those simulated, compared to the noise and speech signal cases.

The MUSIC algorithm was also employed using simulated steering vectors for the pseudospectrum but with measured evaluation signals. The angular errors averaged over all 10 locations for the noise, speech and guitar signals are $50°, 48°, 51.5°$ respectively. The performance is very similar for the three different signals, which is advantageous compared to the neural networks. However the performance is lower than for the network models for noise and speech localisation. One reason for the lower localisation performance of the MUSIC algorithm could be the sparse search grid, containing only 39 points on the sphere. However, for some of the source locations, the MUSIC pseudospectrum did not contain a clear peak corresponding to one location, but was rather noisy and so the highest peak was chosen as the location estimate.

Based on the results discussed above, two main conclusions can be drawn. First, the complex-valued multilayer perceptron appears to perform at least as well, and often better than, its real-valued counterpart for localising various sources in space using a simulated dataset. Whilst the difference in localisation accuracy between the two models are relatively small, it may be possible to further improve the cMLP using for example different complex-valued activation functions or by training using a complex-valued step size, as discussed for example in [17]. The second main conclusion is that small neural network architectures (as those presented here) are able to learn from simulated data to localise measured data in 2 out of 3 cases, which can be crucial for applications where there is very difficult to have a large measured dataset. The authors plan to investigate more complex network models such as recurrent neural networks (RNN) to see if using complex-valued data can improve the localisation of real-valued networks and if complex networks can provide a better localisation performance using measured data, if trained on simulated signals, especially when the acoustic environment becomes more challenging, such as that for example in a reverberant room.

5. CONCLUSIONS

This paper presented a comparison between real- and complex-valued multilayer perceptrons trained with simulated datasets to localise acoustic sources using both simulated and measured microphone signals. The authors showed that the complex-valued networks performed slightly better than their real counterpart for most scenarios. It was also demonstrated that even simple network architectures are able to learn from simulated data to localise measured data, provided the correct features are used when training the networks.

6. REFERENCES

[1] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," *The Journal of the Acoustical Society of America*, vol. 146, no. 5, pp. 3590–3628, 2019.

[2] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *The Journal of the Acoustical Society of America*, vol. 152, no. 1, pp. 107–151, 2022.

[3] B. Osinski, A. Jakubowski, P. Zikecina, P. Milos, C. Galias, S. Homoceanu, and H. Michalewski, "Simulation-based reinforcement learning for real-world autonomous driving," in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 6411–6418, IEEE, 2020.

[4] N. Inkawhich, M. J. Inkawhich, E. K. Davis, U. K. Majumder, E. Tripp, C. Capraro, and Y. Chen, "Bridging a gap in sar-atr: Training on fully synthetic and testing on measured data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2942–2955, 2021.

[5] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744, IEEE, 2020.

[6] V. S. Paul and P. A. Nelson, "Matrix analysis for fast learning of neural networks with application to the classification of acoustic spectra," *The Journal of the Acoustical Society of America*, vol. 149, no. 6, pp. 4119–4133, 2021.

[7] V. Paul and P. A. Nelson, "Analysis of complex-valued neural networks for audio source localisation," in *Proceedings of the Institute of Acoustics Vol. 44. Pt. 3.*, (Milton Keynes, United Kingdom), 2022.

[8] J. Bassey, L. Qian, and X. Li, "A survey of complex-valued neural networks," *arXiv preprint arXiv:2101.12249*, 2021.

[9] J. Meyer and G. Elko, "A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. II–1781, IEEE, 2002.

[10] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," *arXiv preprint arXiv:2006.01919*, 2020.

[11] P. Morse and K. Ingard, *Theoretical Acoustics*. International series in pure and applied physics, Princeton University Press, 1986.

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

[13] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[14] M. S. Brandstein and D. B. Ward, *Fundamentals of Microphone Arrays*. Springer Science & Business Media, 2001.

[15] P. Virtue, X. Y. Stella, and M. Lustig, "Better than real: Complex-valued neural nets for mri fingerprinting," in *2017 IEEE international conference on image processing (ICIP)*, pp. 3953–3957, IEEE, 2017.

[16] K. Gade, "A non-singular horizontal position representation," *The Journal of Navigation*, vol. 63, no. 3, pp. 395–417, 2010.

[17] H. Zhang and D. P. Mandic, "Is a complex-valued stepsize advantageous in complex-valued gradient learning algorithms?," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2730–2735, 2015.