

Blind Adaptive Beamforming of Narrowband Signals using an Uncalibrated Antenna-Array by Machine Learning

1st Soeren Schoenbrod

Julia Computing Inc.

soeren.schoenbrod@juliacomputing.com

2nd Elliot Saba

Julia Computing Inc.

elliott.saba@juliacomputing.com

3rd Miguel Bazdresch

Rochester Institute of Technology

mxbiee@rit.edu

4th Steve Kelly

Julia Computing Inc.

steve.kelly@juliacomputing.com

5rd Tim Besard

Julia Computing Inc.

tim@juliacomputing.com

6th Keno Fischer

Julia Computing Inc.

keno@juliacomputing.com

Abstract—Multiple wireless communication systems compete within the same frequency range. The spatial domain adds an extra degree of freedom to the time and frequency domain. In this paper we introduce a machine learning model in the spatial domain designed to enhance the signal of interest and mitigate interfering signals. The method described does not require a calibrated antenna-array-RF-chain. It is based on a neural network that learns the structure of the signal of interest and separates it from other signals in the time, frequency and spatial domains. It does so without the need of any additional information and can very easily be used as a ballast to increase the Signal to Interference and Noise Ratio (SINR).

Index Terms—spatial signal processing, beamforming, interference mitigation

I. INTRODUCTION

Wireless communications have become the de facto standard for most consumer communication and networking systems. The diversity of wireless communication systems has increased with applications such as smart home, wearables, and various automation systems increasing in ubiquity. The number of wireless communication systems is expected to grow in the foreseeable future [1]. However, the frequency spectrum for wireless transmission is a limited resource. Usually, frequency ranges are allocated by government regulatory agencies. As a result, regional regulatory differences and other limitations have caused overlap between frequency band allocations. A prominent example is the 2.4 GHz band, where systems like Wi-Fi, Bluetooth, ZigBee and multiple Audio-Video (AV) devices coexist. These circumstances lead to difficulties in error-free transmission, particularly in crowded locations.

In addition, Software Defined Radios (SDRs) and other technological developments are making wireless transmission more accessible to hobbyists and other non-specialist actors. For example, recently so called Personal Privacy Devices (PPDs) have become a problem for Global Navigation Satellite

System (GNSS) users [2]. PPDs are low-cost jammers used to mask GNSS signals in order to prevent location tracking by third parties. Illicit use of such PPDs can interfere with the use of location information in law enforcement and emergency situations. Additionally, unintentional jammers, such as microwave ovens, can cause high power interference in already crowded frequency bands, such as the popular 2.4 GHz band.

Most conventional wireless communication systems only utilize the time and frequency domain for signal transmission. The spatial domain (i.e. utilizing multiple, spatially separated antenna elements), however, adds an extra degree of freedom that may be used to overcome the limitations of these traditional systems. Historically the spatial domain is rarely used for signal transmission, due to the high cost involved or the inflexible direction of the beam towards the receiver or transmitter. However, as signal processing has shifted from hardware to software and the cost of multiple antennas has decreased, spatial signal processing has become more accessible. The use of antenna arrays allows adaptive change in the direction of the signal beam.

Adaptive signal beamforming is achieved by shifting the signals from each antenna element in such a way that the signal of interest sum constructively and interfering signals sum destructively. If the narrowband assumption¹ holds, these shifts can be performed by phase shifts. There are multiple methods available to calculate these phase shifts. The most prominent are the Minimum Variance Distortionless Response (MVDR) and the Linearly Constrained Minimum Variance (LCMV) beamformers [3]. Both methods require information about the relative phase of the impinging interferer at each antenna element. In theory, this is a deterministic calculation by the Direction of Arrival (DOA) and the displacement of the antenna elements, or it can be calculated by the eigendecomposition of the covariance matrix [4]. However, the former

¹Narrowband assumption is valid if $\pi B \frac{d}{c} \ll 1$, where B is the Bandwidth of the signal, d the largest distance between antenna elements and c the speed of light.

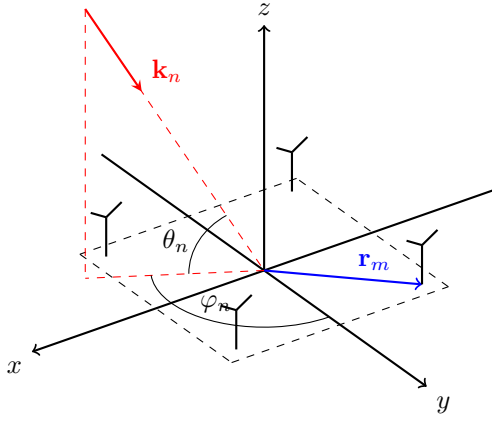


Fig. 1. Exemplary antenna array configuration.

method requires the DOA of the interfering signal, which might be unknown, and the latter requires information about the power of the interfering signal with respect to the signal of interest. The former method also requires a calibrated multi-antenna-RF-chain. There are essentially two types of gain and phase mismatches between the antenna channels that need to be compensated for, if the phase shifts are calculated based on the DOA:

- 1) Gain and phase mismatches that are introduced by the different reception / transmission characteristics of each antenna element.
- 2) Gain and phase mismatches that are introduced by the RF-chain, like down-converters, amplifiers, various filters, unequal cable lengths, etc.

Mismatches between antenna channels occur due to varying manufacturing tolerances, vibrations or temperature. The former is dependent on the DOA, but known to be quite stable over time and can, therefore, be measured in an anechoic chamber and saved for later processing [5]. The latter, however, changes over time and, therefore, needs to be calibrated during run time.

In this work, we present a neural network-based method that is able to enhance the signal of interest and mitigate interfering signals without the need for calibration. It is similar to the eigendecomposition of the covariance matrix. The neural network is trained to find the signal of interest based on its structure. Other signals will be mitigated spatially. The implementation and evaluation are done with the Julia programming language and the Flux machine learning framework [6]. Both provide a fast computation platform, and Flux provides mechanisms to include custom layers into the neural network without compromised performance.

II. SIGNAL MODEL

Consider an M -element antenna array with an arbitrary configuration (see Fig. 1). For an impinging signal onto the

antenna array, the wave vector $\mathbf{k}(\varphi_i, \theta_i)$ is defined as

$$\mathbf{k}(\varphi_i, \theta_i) = -\frac{2\pi}{\lambda_c} \begin{pmatrix} \cos(\theta_i) \sin(\varphi_i) \\ \cos(\theta_i) \cos(\varphi_i) \\ \sin(\theta_i) \end{pmatrix}, \quad (1)$$

where φ_i and θ_i are the azimuth and elevation angle of the i -th impinging signal respectively and λ_c is the carrier wavelength. With the antenna position vector $\mathbf{r}_m$ the steering vector is determined by

$$\mathbf{a}(\varphi_i, \theta_i) = \begin{pmatrix} e^{-j\mathbf{k}(\varphi_i, \theta_i)^T \mathbf{r}_1} \\ \vdots \\ e^{-j\mathbf{k}(\varphi_i, \theta_i)^T \mathbf{r}_M} \end{pmatrix}. \quad (2)$$

This vector represents the phase shifts of an impinging signal from a given DOA described by azimuth and elevation. Prior to signal digitalization, a signal may be influenced by multiple gain and phase distortions, like reception characteristics of each antenna element, amplifiers, downconverters, and filters. These mismatches are expressed by the vector $\mathbf{c} \in \mathbb{C}^M$. Note that all antenna channels must be downconverted with almost the exact same frequency to guarantee a stable phase relationship within a small period of time. This is essential for the estimation of the covariance matrix that is part of the neural network (see next section for more details). The antenna array output for the i -th signal finally yields

$$\mathbf{x}_i(t) = \text{diag}(\mathbf{c}) [\mathbf{a}(\varphi_i, \theta_i) s_i(t) + \mathbf{n}(t)], \quad (3)$$

where $\mathbf{n}(t) \in \mathbb{C}^M$ represents Gaussian white noise of variance σ_n^2 and $s_i(t)$ the incoming signal, which could be either the signal of interest or an interfering signal. The gain and phase mismatch also affects the noise term, since the dominant noise is generated at the antenna array or at the Low Noise Amplifier (LNA). The varying gain and phase mismatch follows down the chain.

Assuming a single signal of interest and Z interfering signals, the digitized signal yields

$$\mathbf{x}(t) = \text{diag}(\mathbf{c}) \left[\mathbf{a}(\varphi, \theta) s(t) + \sum_{i=1}^Z \mathbf{a}(\varphi_{z,i}, \theta_{z,i}) s_{z,i}(t) + \mathbf{n}(t) \right], \quad (4)$$

where $s(t) \in \mathbb{C}$ is the signal of interest and $s_{z,i}(t) \in \mathbb{C}$ the interfering signal. This signal model is used for training of the neural network.

III. NEURAL NETWORK

Consider the neural network shown in Fig. 2. The first layer filters the signal of interest from interfering signals in the time and frequency domain. This first layer is not an ordinary convolutional filter, but rather two convolutional filters that are repeated over the antenna channels. Therefore, the same filters are applied to all antenna channels. This is crucial to preserve the phase relationship among the antenna channels. The activation function of this layer is the identity, the signal is not padded before applying the filter and the filter length

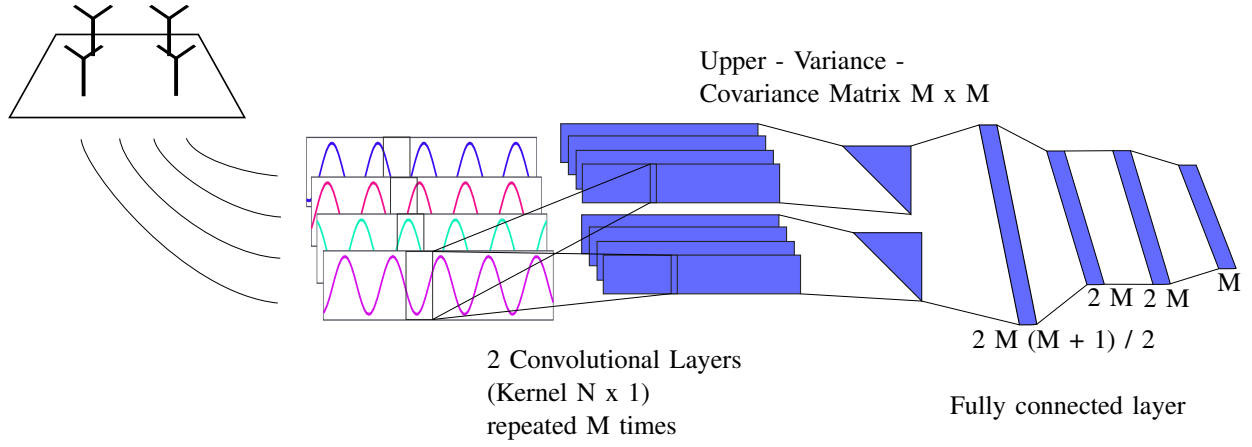


Fig. 2. Neural network

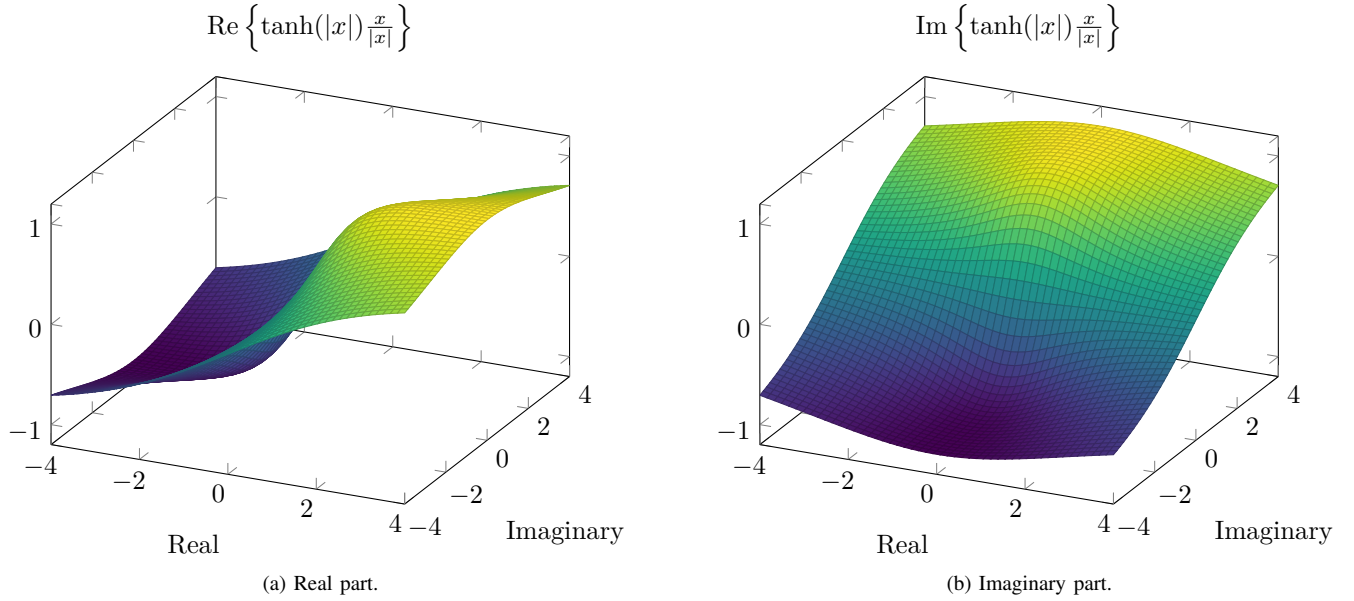


Fig. 3. Activation function ctanh .

has been set to 40. The next layer is the calculation of the upper triangular block of the covariance matrix, including the variance of each channel. This is inspired by the state of the art beamformers MVDR, LCMV and eigen-beamformer, all of which use the covariance matrix for beamforming. Here only the upper triangular matrix is calculated, because the covariance matrix is Hermitian and the lower triangular matrix, therefore, holds the same but conjugated information. From there on, multiple dense layers follow to estimate a beamformer that enhances the signal of interest and mitigates interfering signals. All of these dense layers use the activation function

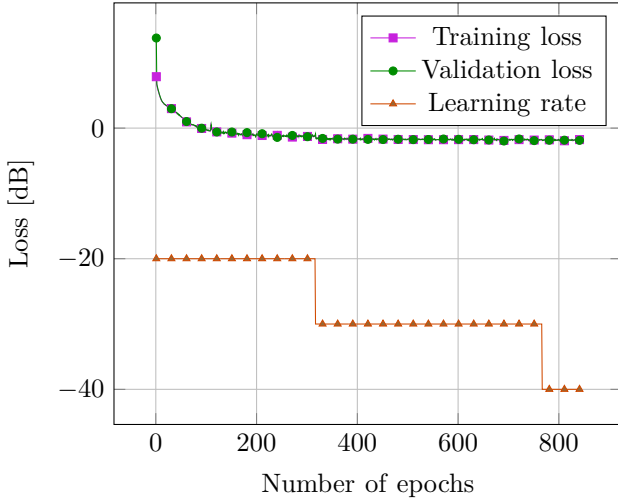
$$\text{ctanh}(x) = \tanh(|x|) \frac{x}{|x|}, \quad (5)$$

which was introduced by Pfeifenberger in [7]. The real and imaginary part of this function are given in Fig. 3a and Fig. 3b

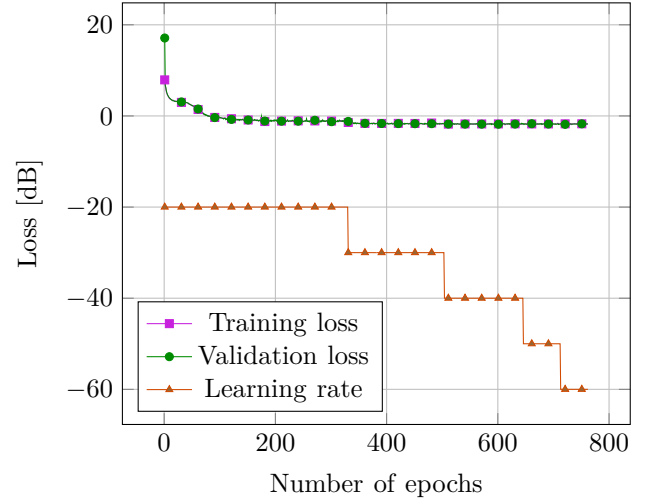
respectively. In contrast to other activation functions preserves this activation function the complex valued information.

The loss function is the Mean Squared Error (MSE) between the beamformed measurement and the true signal of interest. The beamformer is normalized by its first element before it is applied to the measurement.

The model is trained by simulations, as there are no large real-world datasets available. Creating such a dataset would require a tremendous amount of work of creating physical measurements with a signal of interest and an interfering signal with varying power, different DOAs, varying gain and phase mismatches, etc. and appears infeasible. The simulation is carried out as follows: For each training data for the neural network the measurement is simulated according to equation (4). For the purpose of this evaluation a 2 by 2 Uniform Rectangular Array (URA) is simulated, where the mutual



(a) Signal of interest: Continuous Wave (CW).



(b) Signal of interest: Quadrature Modulation (QM).

Fig. 4. Training loss, validation loss and learning rate over number of epochs (first stage).

distance of the antenna elements is half the center wavelength. The phase mismatch for each antenna channel is randomly taken from a uniform distribution from 0 to 2π . The gain mismatch is taken from a normal distribution of $\mathcal{N}(1.0, 0.1)$. Both gain and phase mismatch are fixed throughout a single simulation of the measurement, but vary with every taken measurement. This is due to the assumption, that the gain and phase mismatch only slowly vary over time. The noise $\mathbf{n}(t)$ is simulated as white Gaussian noise. In the following evaluation only a single interfering signal is evaluated next to the signal of interest. Note that only $M - 1$ signals can be separated spatially, when using an M -element antenna array. The DOA of both signals is randomly chosen in three-dimensional space. They are not restricted to be distinct from each other.

Two types of training have been carried out to evaluate the performance of the model with different types of signals:

- 1) • Signal of interest: Continuous Wave (CW) signal at the center frequency plus a Doppler with a standard deviation of 100 Hz.
- Interfering signal: CW signal with an arbitrarily chosen frequency between -2.5 MHz and 2.5 MHz around the center frequency.
- 2) • Signal of interest: Quadrature Modulation (QM) signal at the center frequency plus a Doppler with a standard deviation of 100 Hz. The code frequency is set to 2 MHz and the QM signal consists of 4 symbols.
- Interfering signal: CW signal with an arbitrarily chosen frequency between -2.5 MHz and 2.5 MHz around the center frequency.

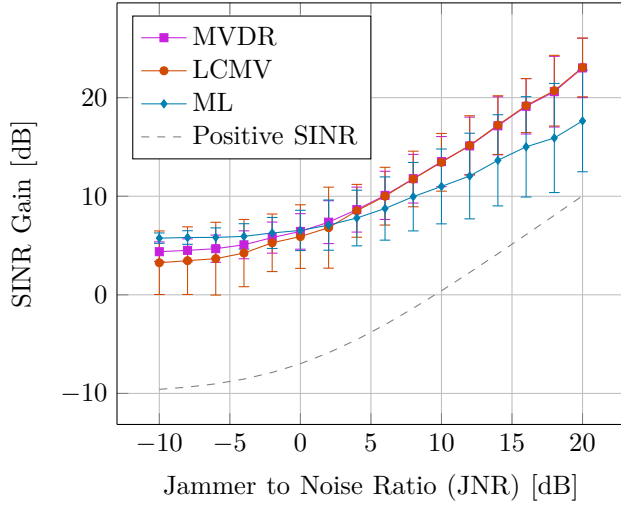
The power of the signal of interest is set to 10 dB. The symbols within the QM signal are randomized for each training set. The model is trained in a two-stage process. In the first stage, the model is trained with an interfering signal that has the same power as the signal of interest to emphasize that the

structure of the signal is important instead of the signal power. In the second stage, the model is trained with an interfering signal that has a varying power from -10 dB to 20 dB. The center frequency is set to 2.4 GHz and the sampling frequency is set to 5 MHz. Each simulated measurement contains 200 samples, that correspond to time span of $40 \mu\text{s}$. The batch size of each training set is 100. Each epoch includes the evaluation of 100 batches. For each epoch new batches are generated by the simulator explained above. Adam has been chosen as the optimization algorithm [8].

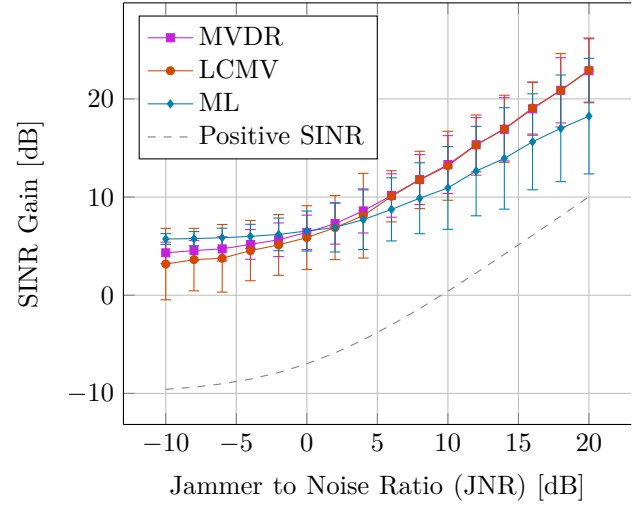
IV. RESULTS

Fig. 4 shows the results of the training loss and validation loss over the number of epochs for both signal types within the first stage. Fig. 7 shows the corresponding results for the second stage. The training loss is a metric used to assess how a deep learning model fits the training data, whereas validation loss is a metric used to assess the performance of a deep learning model on the validation set. The validation set is performed by the same simulation generator as explained above. The validation loss is monitored over the number of epochs. If the validation loss has not improved over the period of 50 epochs, the learning rate η is dropped by a tenth. If the validation loss has not improved over the period of 100 epochs, the training is aborted.

The performance of the proposed neural network is evaluated using a Monte-Carlo simulation. The randomly varied parameters are the DOA for the signal of interest, the interfering signal, the gain and phase mismatch, the Doppler of the signal of interest, the frequency of the interfering signal and their starting phases like in the training sequences. The power of the interfering signal is varied on the x-axis with 500 Monte-Carlo samples taken for each power level of the interfering signal. The proposed machine learning beamformer is compared to MVDR and LCMV (see Fig. 5a), representing state-of-the-

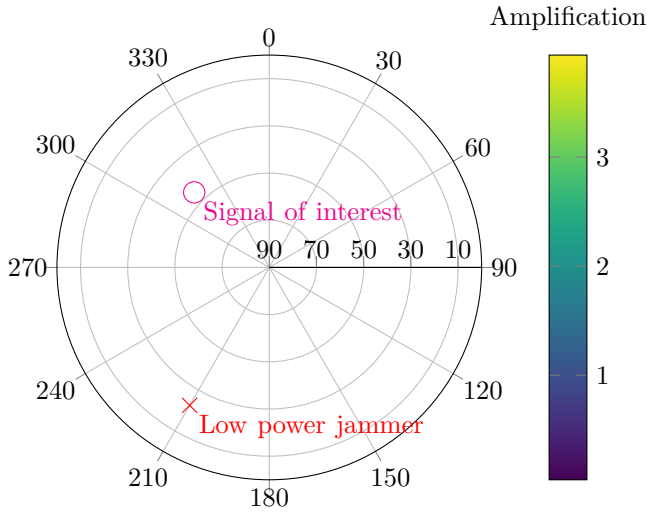


(a) Signal of interest: CW.

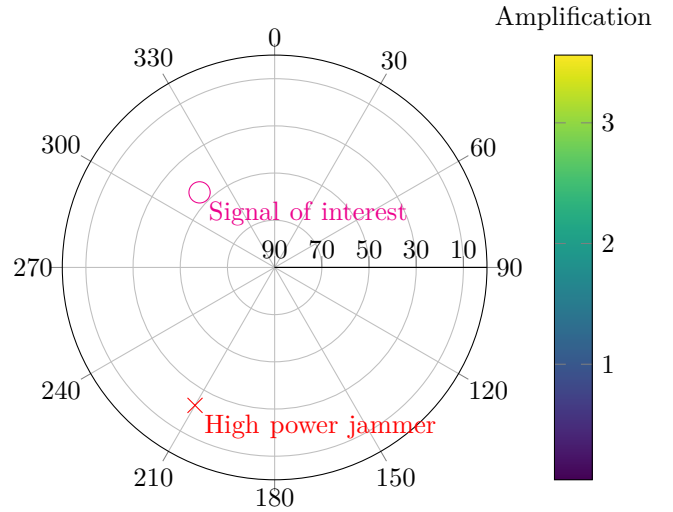


(b) Signal of interest: QM.

Fig. 5. Monte-Carlo simulation to evaluate the Machine Learning (ML) beamformer against state of the art beamformer. Note that the state of the art beamformers have access to the spatial information of the impinging signals, e.g. the exact steering vector of the interfering signal or the steering vector of the signal of interest. In contrast, the ML calculates the beamformer only based on the incoming measurement.



(a) Low power jammer with Jammer to Noise Ratio (JNR) of -10 dB.



(b) High power jammer with JNR of 20 dB.

Fig. 6. Exemplary beamform pattern.

art beamforming methods. The figure of merit is SINR gain due to application of the beamformer. Since beamforming is a linear process, it is applied to the signal of interest and the interfering signal separately to calculate the SINR. The SINR gain is the ratio of the SINR before and after the beamformer is applied.

The machine learning beamformer outperforms the state of the art beamformer for small JNRs. Moreover, it achieves the optimal gain of $M = 4$ (ca. 6 dB) in the case of low-powered interference, whereas the state of the art beamformers fall behind by a few decibels. This is due to the fact that the interfering signal is always nullified by the state of the art beamformer. This projection reduces the signal space by one

dimension. However, this is an unreasonable harsh measure against a low-powered interfering signal. This could be accommodated by a threshold on the power of the interfering signal. However, the great advantage of the ML beamformer is that such thresholds are not required. This advantage is illustrated in Fig. 6a and Fig. 6b. Both figures demonstrate the pattern of the amplification, that is gained by applying the ML beamformer. Fig. 6a shows the effect of a low power jammer and Fig. 6b shows the effect of a high power jammer. In the case of a low power jammer a hard null is not required for the DOA of the jammer. Instead, the amplification for the signal of interest is maximized. In the case of a high power jammer it makes sense to mitigate the interfering signal to a

larger extent. Due to this projection the signal of interest can not be maximized to its optimal value.

A disadvantage of the proposed ML beamformer is that for high power interference, the machine learning falls behind the state of the art beamformers, but it is still well above the positive SINR. Nonetheless, high power interference should be the rare case. Therefore, it is better to optimize the case without loud jammers. Note that the state of the art beamformers used for comparison have access to the spatial information of the impinging signals, e.g. the exact steering vector of the interfering signal or the steering vector of the signal of interest. Such vectors can be difficult to estimate in a dynamic scenario, especially if gain and phase mismatches between antenna channels are considered within the RF-chain. In contrast, the ML calculates the beamformer only based on the incoming measurement. It does so by incorporating the time, frequency and spatial domain.

V. CONCLUSION

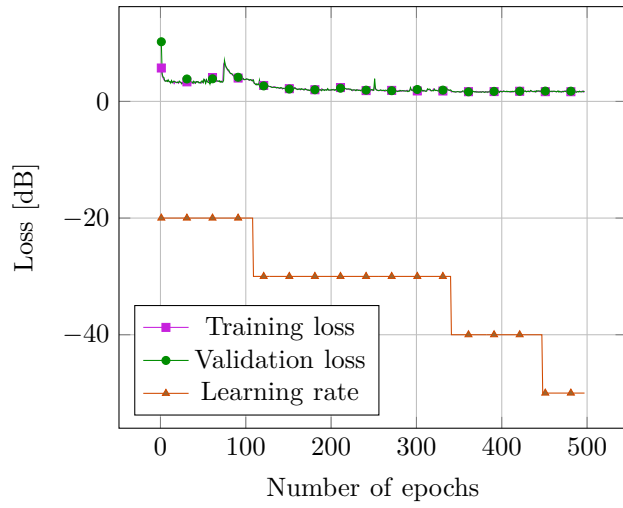
Within the research of this paper a machine learning beamformer has been developed. It adopts some principles of existing beamformers, such as the calculation of covariance matrices, and incorporates them into a neural network architecture. Compared to the state of the art beamformers, the machine learning beamformer does not require the steering vector of the interfering signal nor does it require the steering vector of the signal of interest. The great benefit of the proposed beamformer is that it can find the signal of interest by only evaluating the incoming measurement. The neural network returns a beamformer that enhances the signal of interest and mitigates any interfering signal in such a way that the SINR is increased. Thereby, it performs better than the compared state of the art beamformers for low JNRs. No threshold is required to detect and mitigate an interfering signal. The neural network is self-contained in the sense that it gets the (disturbed) measurement as an input and outputs the clean signal of interest by applying the beamformer.

ACKNOWLEDGMENT

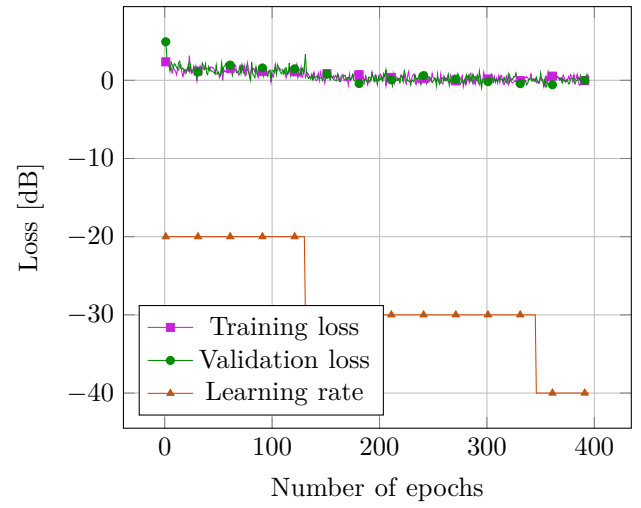
This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112190101. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.”

REFERENCES

- [1] MarketsAndMarkets, “Wireless connectivity market by connectivity technology, type, end-use, and region - global forecast to 2025,” MarketsAndMarkets, Wireless Connectivity Market Report, Oct. 2020.
- [2] S. Pullen and G. X. Gao, “GNSS jamming in the name of privacy: Potential threat to GPS aviation,” in *InsideGNSS*, Apr. 2012.
- [3] M. Souden, J. Benesty, and S. Affes, “A study of the LCMV and MVDR noise reduction filters,” *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4925–4935, 2010.
- [4] A. Weiss and B. Friedlander, ““Almost blind” steering vector estimation using second-order moments,” *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 1024–1027, 1996.
- [5] R. Schmid, M. Rothacher, D. Thaller, and P. Steigenberger, “Absolute phase center corrections of satellite and receiver antennas,” *GPS Solutions*, vol. 9, no. 4, pp. 283–293, Nov 2005. [Online]. Available: <https://doi.org/10.1007/s10291-005-0134-x>
- [6] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, “Fashionable modelling with flux,” *CoRR*, vol. abs/1811.01457, 2018. [Online]. Available: <https://arxiv.org/abs/1811.01457>
- [7] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Deep complex-valued neural beamformers,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2902–2906.
- [8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>



(a) Signal of interest: CW.



(b) Signal of interest: QM.

Fig. 7. Training loss, validation loss and learning rate over number of epochs (second stage).