



Article Machine-Learning-Based Framework for Coding Digital Receiving Array with Few RF Channels

Lei Xiao D, Yubing Han * and Zuxin Weng D

School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

* Correspondence: hanyb@njust.edu.cn

Abstract: A novel framework for a low-cost coding digital receiving array based on machine learning (ML-CDRA) is proposed in this paper. The received full-array signals are encoded into a few radio frequency (RF) channels, and decoded by an artificial neural network in real-time. The encoding and decoding networks are studied in detail, including the implementation of the encoding network, the loss function and the complexity of the decoding network. A generalized form of loss function is presented by constraint with maximum likelihood, signal sparsity, and noise. Moreover, a feasible loss function is given as an example and the derivations for back propagation are successively derived. In addition, a real-time processing implementation architecture for ML-CDRA is presented based on the commercial chips. It is possible to implement by adding an additional FPGA on the hardware basis of full-channel DRA. ML-CDRA requires fewer RF channels than the traditional full-channel array, while maintaining a similar digital beamforming (DBF) performance. This provides a practical solution to the typical problems in the existing low-cost DBF systems, such as synchronization, moving target compensation, and being disabled at a low signal-to-noise ratio. The performance of ML-CDRA is evaluated in simulations.

Keywords: coding digital receiving array; machine learning; low-cost DBF system; few-RF-channel; encoding and decoding

1. Introduction

Digital beamforming (DBF) is a key requirement in modern wireless systems. The cost and power consumption are the main challenges for the wide application of DBF [1], due to the higher number of radio frequency (RF) channels and analog-to-digital converters (ADC) that are required. Therefore, it is critical to develop a novel, low-cost, power-efficient DBF system with few RF channels.

To reduce RF channels, the received array signals have to be compressed/encoded at the beginning of the front-end. However, the decrease in sampled signal dimension will lead to an unacceptable DBF performance. Thus, it is necessary to recover/decode the original received full-array signals to ensure that the signal dimension for DBF is consistent with the traditional full-channel array. Considerable efforts [2–11] have been made to reduce the digital receiving array (DRA) cost based on the above encoding and decoding architecture. According to the difference in codec, these approaches can be classified into the compressed-sensing-based coding DRA (CS-CDRA) [2–6] and the orthogonal-coding-based coding DRA (OC-CDRA) [7–12], as shown in Table 1.

For a typical CS-CDRA, the received full array signals are encoded by a measurement matrix that complies with the restricted isometry property (RIP) [13], such as random Gaussian matrix, random Bernoulli matrix, partial Hadamar matrix, and partial Fourier matrix. Since the received array signal is compressible, it could be reconstructed by solving a convex optimization problem [14].

However, it is impossible to recover the received array signal in real-time due to the iteration requirements of the existing sparse reconstruction algorithms [15–17]. In



Citation: Xiao, L.; Han, Y.; Weng, Z. Machine-Learning-Based Framework for Coding Digital Receiving Array with Few RF Channels. *Remote Sens.* 2022, 14, 5086. https://doi.org/ 10.3390/rs14205086

Academic Editor: Alin Achim

Received: 17 August 2022 Accepted: 9 October 2022 Published: 12 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). addition, the sparse reconstruction algorithms cannot work stably with a low signal-tonoise ratio (SNR).

	Туре	RF Channels	ADCs	Cost ¹	Real-Time Processing	Low SNR Available	Without Data Loss	Without Moving Target Compensation	Without Gate Lobe
	DRA Subarray	Massive Few	Massive Few	Expensive Middle	$\sqrt[]{}$	$\sqrt[n]{\sqrt{1}}$		$\sqrt[]{}$	$\stackrel{\checkmark}{\times}$
Coding- DRA	CS-based OC-based TSPW OC-based OSC ML-based	Few Single Massive Few	Few Single Single Few	Middle Cheap Expensive Middle	$\stackrel{\times}{\checkmark}$	$\stackrel{\times}{\checkmark}$ \checkmark \checkmark	$ \sqrt{\begin{array}{c}2\\\times\\\times\\\checkmark\end{array}} $	$ \sqrt{\frac{2}{x}} $ $ \times $ $ $	\checkmark \checkmark \checkmark

Table 1. Different architectures for low-cost DBF systems.

¹ System cost is mainly determined by the requirement of RF channels. ² Data loss will occur if the reconstruction is based on multi-snapshots. In the meantime, moving target compensation is necessary.

The OC-CDRA introduces a code division multiplexing (CDM) [18] technique to identify each signal path associated with every unique antenna element. Specifically, the received signal from each antenna element is mixed with a high-speed spread spectrum code, such as the Walsh–Hadamard and Gold codes, and decorrelated with the same spread spectrum code after digitization. According to the mixer position, the OC-CDRA can be further divided into on-site coding (OSC) architecture [7–9] and time sequence phase weighting (TSPW) architecture [10-12]. The advantage of OSC architecture is that the received full array signals are encoded at an intermediate frequency (IF) after down-conversion, which is friendly to the implementation of the encoding network. The requirements of ADCs and parallel input/output (I/O) channels at the digital back-end are significantly reduced, but the RF channels are not. Therefore, the OSC architecture is not efficient in terms of cost reduction. However, the TSPW architecture encodes the received array signal at RF before down-conversion. The encoded signals are sampled by few RF channels and ADCs, which significantly reduce the system budget. However, both TSPW and OSC face problems in code synchronization (in both encoding and decoding) and moving target compensation [19]. More importantly, each information bit is decoded from L chips, where L is the code length. the recovered signal in OC-based architecture is only 1/L data volume of the full-channel array, which inevitably leads to data loss after decoding. The system-accumulated gain will be decreased 10log(L) dB.

Different from the above coding DRA (CDRA), there is a generalized coding array, called antenna selection (AS) architecture [20–22]. The fundamental idea behind AS is to reduce the number of RF chains by judiciously activating a subset of antennas. This can realize a reconfigurable array beamforming through element selection (not combination) without decoding. However, meanwhile, the array accumulated gain will be decreased according to the number of discarded array elements, which is also a form of data loss.

Artificial intelligence has shown great potential in various fields, and beamforming systems are no exception. A beamforming neural network (BFNN) is proposed in [23] to optimize the beamformer to maximize the spectral efficiency with hardware limitations. It has strong robustness to imperfect channel state information (CSI). Ref. [24] proposes a beamforming prediction network (BFNet) to jointly optimize the power allocation and virtual uplink beamforming of Multiple-Input and Multiple-Output (MIMO) systems. This can get rid of the complexity caused by excessive iteration to realize real-time calculation. For the local scattering caused by sea surface, [25] introduce a convolutional neural network (CNN) framework to estimate the transmitter's incident angle.

This paper proposes a novel, low-cost, power-efficient DBF system framework called machine-learning-based coding DRA (ML-CDRA). The received full-array signals are encoded into a few RF channels, and decoded by an artificial neural network (ANN) in real-time without any data volume loss. The proposed ML-CDRA can work stably at a low

SNR, as verified in simulations at SNR = -60 dB. Since the recovered signals are decoded by a single snapshot, the moving target compensation problem will not bother ML-CDRA. Moreover, we present a generalized loss function of the decoding network, which is carried out from three directions: maximum likelihood, signal sparsity, and noise (including noise power suppression, equal power constraint, and noise whitening). A feasible loss function is given as an example and the derivations for back propagation are successively derived. At the end, the implementation of ML-CDRA is also discussed based on the existing technologies and devices. A real-time processing architecture for ML-CDRA, with the decoding network implemented by a field-programmable gate array (FPGA), is presented.

It should be emphasized that the proposed ML-CDRA is a novel, low-cost, DBF system framework. The form of encoding and decoding networks are not fixed, which leads to a systematic trade-off between cost, resources, and performance. The loss function of decoding network should also be modified according to the applications.

2. Materials and Methods

As shown in Figure 1, the proposed ML-CDRA is composed of two networks: encoding and decoding networks. The signals received from each antenna are selected and combined into a few RF channels through an encoding network. After digitization, the sampled signals are decoded by a decoding network based on machine learning to recover the originally received full-array signals.



Figure 1. The block diagram of the proposed ML-CDRA.

2.1. Signal Model

The encoded signals sampled by ADCs are:

$$y = G\Phi(x + n_1) + n_2 \tag{1}$$

where $\boldsymbol{y} \in \mathbb{C}^M$, *G* is the gain of receiver module, $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$ is the encoding matrix (M < N, M is the number of few RF channels, N is the number of antenna elements), $\boldsymbol{x} \in \mathbb{C}^N$ is the received array signal, $\boldsymbol{n}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{n_1})$ is the independent and identically distributed Gaussian noise in wireless channels, and $\boldsymbol{n}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{n_2})$ is the random thermal noise in RF channels.

The traditional array signal model supposes that the noise in each channel is independent and identically distributed. However, this is totally different in CDRA, which is correlated due to the encoding. The same wireless channel noise may be mixed into two different RF channels. Thus, it is necessary to clarify the composition of the noise in sampled signals.

$$\boldsymbol{\eta} = \boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{n}_1 + \boldsymbol{n}_2 \tag{2}$$

where η is the noise in the sampled signal.

The power of n_2 is determined by the noise figure *F* of the digital receiver module. For example, considering the typical value F = 3 dB, the introduced thermal noise n_2 has equal power with the input noise after being amplified. In addition, the power of n_2 in each RF channel may be different, which is controlled by the encoding matrix Φ .

The covariance matrix of noise η is

$$\Sigma_{\eta} = G^{2} \boldsymbol{\Phi} \Sigma_{n_{1}} \boldsymbol{\Phi}^{\mathrm{H}} + \Sigma_{n_{2}}$$

= $G^{2} \boldsymbol{\Phi} \Sigma_{n_{1}} \boldsymbol{\Phi}^{\mathrm{H}} + \gamma G^{2} (\boldsymbol{\Phi} \Sigma_{n_{1}} \boldsymbol{\Phi}^{\mathrm{H}}) \circ \boldsymbol{I}$ (3)

where $\gamma = 10^{(F/10)} - 1$, $(\cdot)^{H}$ denotes the conjugate transpose, \circ denotes the Hadamard product, and *I* is the identity matrix.

Obviously, the noise in the sampled signal is correlated.

2.2. Encoding Network

The encoding network can significantly reduce the requirement of RF channels. The received array signals are compressed into a few RF channels according to the encoding matrix $\boldsymbol{\Phi}$, which describes the connection between antenna elements and RF channels. The form of encoding matrix $\boldsymbol{\Phi}$ is diverse. Different encoding matrices will bring a different spatial sensitivity to CDRA.

Consider the single far-field target case; the received array signal *x* can be expressed as

$$= a(\theta)s \tag{4}$$

where $a(\theta) \in \mathbb{C}^N$ is the steering vector, and θ is the direction of arrival.

x

The encoded signal of m^{th} channel is

$$y_m = G\boldsymbol{\varphi}_m^{\mathrm{H}}(\boldsymbol{x} + \boldsymbol{n}_1) + n_{2,m}$$

= $G\boldsymbol{\varphi}_m^{\mathrm{H}}\boldsymbol{a}(\theta)\boldsymbol{s} + (G\boldsymbol{\varphi}_m^{\mathrm{H}}\boldsymbol{n}_1 + n_{2,m})$ (5)

where φ_m^{H} is the m^{th} row vector of Φ , and $n_{2,m}$ is the thermal noise of the m^{th} channel. Therefore, the SNR of m^{th} channel is

$$SNR_m = \frac{P_s}{P_n} = \frac{G^2 |\boldsymbol{\varphi}_m^H \boldsymbol{a}(\theta)|^2 \sigma_s^2}{D(G \boldsymbol{\varphi}_m^H \boldsymbol{n}_1) + \sigma_{n_2,m}^2}$$
(6)

where σ_s^2 is the signal power, $\sigma_{n_2,m}^2$ is the thermal noise power of *m*th channel, $|\cdot|$ denotes the magnitude, and D(·) denotes the variance.

It is easy to find that the signal power P_s is a function of the direction. This means that the signal from different directions may suffer losses after the encoding network compared with the phased coherent combination. This loss is determined by the array arrangement, direction of arrival, and encoding matrix. It will eventually be reflected in the maximum SNR after DBF according to the weight of the corresponding direction, namely, spatial sensitivity. Thus, the encoding network should be carefully designed in different applications.

According to the difference in implementation, the encoding network can be divided into the fixed encoding network and the tunable encoding network. As shown in Figure 2a, the fixed encoding network is implemented by a multiple input–multiple output (MIMO) feed network without any additional equipment. The tunable encoding network introduces additional RF switches or phase shifters to the feed network, as shown in Figure 2b,c. An RF switch or phase shifter can provide more freedoms for signal processing. Therefore, the received array signals can be encoded in various schemes to obtain optimal performance in different applications by changing the encoding matrix.



Figure 2. Different encoding networks. (a) Fixed encoding network. (b) Tunable encoding network with RF switches. (c) Tunable encoding network with phase shifters.

The main difficulty of the encoding network is the topological structure overlapping of the feed network. The overlapping wiring can be realized by a multi-layer printed circuit board (PCB) with through holes. Nevertheless, the design of an overlapping feed network is still a challenge when the topological structure is too complex. The RF microelectromechanical systems (RF-MEMS) switches matrix [26] is one possible scheme to realize the overlapping feed network. However, the RF-MEMS switch matrix still needs further development, especially regarding large-scale and RF performance loss.

The most common feed network without overlap is the adjacent subarray structure. A typical passive array antenna structure is given in Figure 3a. Each group of four adjacent elements is combined into a single RF channel by a fixed feed network. The subarray pattern of this structure is immutable, and the main lobe is aligned to the broadside of the array. Figure 3b is a typical phased array antenna with subarray structure, and each group of four adjacent elements is combined into a single RF channel after the phase shifter. However, its subarray pattern can be changed by switching the phase shifter.

2.3. Decoding Network

Due to (1) is an under-determined equation, which has an infinite number of solutions. It is impossible to recover the original received array signal from the few-channel signal by a linear transformation with a single snapshot. Therefore, non-linear processing is an inevitable choice for the decoding network.

Considering the computational complexity and the real-time processing requirements in engineering, the existing sparse reconstruction algorithms are unsuitable for CDRA [15–17]. ANN, which has been rapidly developing in this decade, has the ability to cope with these difficulties.

The decoding network of the proposed ML-CDRA is carried out by an ANN in this paper. The forward propagation of ANN is implemented by limited multipliers and adders with a low latency. The pipeline architecture can achieve single-snapshot, real-time decoding in a pressureless manner. *It should be noted that the specific network structure of ANN is open for ML-CDRA. This is a bargain between resources and performance.*



Figure 3. The encoding networks with subarray structure. (**a**) Subarray without phase shifter. (**b**) Subarray with phase shifter.

Here, a generalized loss function of the decoding network based on the maximum a posteriori estimation (MAP) is given as

$$J = J_1 + \alpha J_2 + \beta J_3 \tag{7}$$

where J_1 is derived from the maximum likelihood estimation. J_2 and J_3 are the constraints for signal and noise, respectively. α and β are hyperparameters. The details of the generalized loss function are presented in the next section.

The decoding network is trained based on back propagation, which is carried out by gradient descent as

$$\mathbf{Z}_{i+1} = \mathbf{Z}_i + \mu \frac{\partial}{\partial \mathbf{Z}^*} J(\mathbf{Z}, \mathbf{Z}^*)$$
(8)

where Z_i is the complex network parameter Z in i^{th} iteration, μ is the iteration step-size, $\frac{\partial}{\partial Z^*} J(Z, Z^*)$ is the partial derivative of $J(Z, Z^*)$ to Z^* , $J(Z, Z^*)$ is the loss function, and $(\cdot)^*$ denotes the conjugate.

Figure 4 describes the training of a decoding network based on the proposed generalized loss function. The forward propagation of the network is performed by single snapshot. Therefore, the reconstructed noise is independent in the time dimension. This ensures that the time accumulation gains of ML-CDRA will not deteriorate compared with the full-channel DRA. The back propagation of the network is performed by batch processing. According to the authors' experience, it is sufficient to accomplish the training of the decoding network within 1000 batches. Moreover, the training time can be shorter if an offline pre-training is performed based on a simulation or limited actual data. Considering that the non-cooperative objects are more common, the online training ability is essential for the decoding network.

2.4. An Example for Generalized Loss Function

It should first be emphasized that the form of loss function for the decoding network is diverse. The basic idea of the proposed generalized loss function is to focus on the signal and noise simultaneously, especially the noise.



Figure 4. The training of decoding network.

A feasible loss function is given as an example based on the proposed generalized loss function, as follows. The back propagation of the decoding network is carried out by gradient descent, which is derived in Appendix A.

$$I = J_1 + \alpha J_2 + \beta J_3 \tag{9}$$

where

$$J_1 = \sum_{t=1}^{L} \|\boldsymbol{y}(t) - \boldsymbol{G}\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t)\|_{\Sigma^{-1}}^2$$
$$J_2 = K - \sum_{n=1}^{N} \exp\left(-\frac{\lambda_n^2(\boldsymbol{R}_{\hat{\boldsymbol{x}}})}{2\delta_s^2}\right)$$
$$J_3 = \left\|\boldsymbol{R}_{\hat{\boldsymbol{\eta}}} - \delta_n^2 \boldsymbol{I}\right\|_{\mathrm{F}}^2$$

where $\|\mathbf{Z}\|_{\Sigma^{-1}}^2 = \mathbf{Z}^{\mathrm{H}} \boldsymbol{\Sigma}_{\eta}^{-1} \mathbf{Z}$, $(\cdot)^{-1}$ denotes the inverse of matrix, *L* is the number of snapshots, $\mathbf{R}_{\hat{\mathbf{x}}} = \frac{1}{L} \sum_{t=1}^{L} \hat{\mathbf{x}}(t) \hat{\mathbf{x}}^{\mathrm{H}}(t)$ is the covariance of the decoded signal $\hat{\mathbf{x}}$, $\|\cdot\|_{\mathrm{F}}$ denotes the l_{F} -norm, $\lambda_n(\mathbf{R}_{\hat{\mathbf{x}}})$ is the n^{th} eigenvalue of $\mathbf{R}_{\hat{\mathbf{x}}}$, *K* is the sparsity, δ_s is the shape parameter, $\mathbf{R}_{\hat{\eta}}$ is the covariance of the decoded noise, and δ_n is the noise power-suppression coefficient.

2.4.1. Maximum Likelihood Estimation

According to Section 2.1, the sampled signals are

$$y = G\Phi x + \eta \tag{10}$$

where noise η obeys the normal distribution with mean **0** and covariance matrix Σ_{η} .

The maximum likelihood estimation is obtained by maximizing the Gaussian density function

$$f(\widehat{\boldsymbol{x}};\boldsymbol{x}(t)) = \left(\frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}_{\eta}|_{\text{det}}}}\right)^L \exp\left(-\frac{1}{2} \times \sum_{t=1}^L (\boldsymbol{y}(t) - G\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(t))^H \boldsymbol{\Sigma}_{\eta}^{-1} (\boldsymbol{y}(t) - G\boldsymbol{\Phi}\widehat{\boldsymbol{x}}(t))\right)$$
(11)

where $|\cdot|_{det}$ denotes the determinant.

Therefore, the maximum likelihood estimation can be regarded as

$$J_{1} = \sum_{t=1}^{L} \|\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t)\|_{\Sigma^{-1}}^{2}$$
(12)

2.4.2. Sparsity Constraint for Signal

The eigenvalues of the received array signal covariance are sparse, since the targets are limited. Therefore, the sparsity of signal can be constrained as

$$J_{2} = K - \sum_{n=1}^{N} \|\lambda_{n}(\boldsymbol{R}_{\hat{x}})\|_{0}$$
(13)

where $\|\cdot\|_0$ denotes the l_0 -norm.

As the l_0 -norm minimization is an NP-hard problem, it should be replaced by other models, such as l_1 -norm and smoothed l_0 -norm (SL0) [27]. The SL0 approximates l_0 -norm by a smooth Gaussian function, which is differentiable during the back propagation of the decoding network. (13) can be rewritten by SL0 as

$$J_2 = K - \sum_{n=1}^{N} \exp\left(-\frac{\lambda_n^2(\boldsymbol{R}_{\hat{\boldsymbol{x}}})}{2\delta_s^2}\right)$$
(14)

2.4.3. Noise Reduction and Whitening

Suppose the received array signals are accurately reconstructed by the constraints of J_1 and J_2 . As we know, the SNR is determined by signal and noise. Thus, the noise constraint can be designed as

$$J_3 = \left\| \boldsymbol{R}_{\widehat{\eta}} - \delta_n^2 \boldsymbol{I} \right\|_{\mathrm{F}}^2 \tag{15}$$

Noise reduction and whitening are carried out by δ_n and I, respectively. In addition, the $\delta_n^2 I$ also implies an equal-power constraint for different channels.

The existing reconstruction algorithms [15–17] often neglect the statistical characteristics of the reconstructed noise, which is crucial to signal accumulation (including array signal processing and time-domain accumulation). The reconstructed noise power and correlation directly determine the SNR after DBF. We can clearly point out that the upper limit of the array signal reconstruction problem is determined by the noise constraint.

3. Results

To evaluate the performance of the proposed ML-CDRA, a 48-element 12-channel ML-CDRA is studied and compared with different DBF systems as shown in Figure 5, including a 48-element, traditional, full-channel DRA (48-DRA, which has the same antenna elements), a 12-element, traditional, full-channel DRA (12-DRA, which has the same RF channels), and a 48-element, single-channel TSPW array [10] with code length L = 48.



Figure 5. Different DBF systems studied in simulations.

As the receiving antenna of each scheme is a uniform, linear array antenna with halfwavelength inter-sensor spacing, each antenna element is omnidirectional. The encoding network of ML-CDRA is the 4-in-1 subarray structure without a phase shifter, as given in Figure 3a. The decoding network is a three-layer, fully connected, multi-layer perceptron (MLP) with 12 neurons in the input layer, 1024 neurons in the hidden layer, and 48 neurons in the output layer. The loss function is given in Section 2.4. The other details of the decoding network are given in Appendix A, including the activation function and the gradient descent.

3.1. SNR after DBF

Assume that a far-field target at $\theta_1 = 5^\circ$ with single frequency. Both wireless channel noise n_1 and RF channel noise n_2 are Gaussian noise. The noise figure of the digital receiver module F = 3 dB. The beamforming weight $w = (1, \exp(-j2\pi \frac{d \sin(\theta_1)}{\lambda}), \cdots, \exp(-j2\pi \frac{(N-1)d \sin(\theta_1)}{\lambda})^T)$, where $d = \lambda/2$.

As shown in Figure 6, the SNR after the DBF of ML-CDRA is almost consistent with 48-DRA, but only 1/4 RF channels are required. This is nearly 6 dB higher than 12-DRA, which has the same RF channels. In addition, the simulation results also reflect that the system-accumulated gain decreases in TSPW caused by data loss are 10log(L) = 16.8 dB. More importantly, the ML-CDRA works stably even if the input SNR is as low as -60 dB. According to the simulation results, it can be reasonably speculated that the proposed ML-CDRA will still work effectively with a lower SNR.



Figure 6. The SNR after DBF with different input SNR.

Furthermore, the spectrum of the decoded signal after DBF is given in Figure 7 to prove that the proposed ML-CDRA can recover the received array signal perfectly. In the normalized spectrum diagram, the noise power of ML-CDRA is basically the same as 48-DRA, and lower than 12-DRA. It should be noted that the noise of ML-CDRA is approximately uniformly distributed across the spectrum.

3.2. Beamforming Performance

The normalized array pattern of different DBF systems are given in Figure 8. The proposed ML-CDRA has the same beamforming performance compared with 48-DRA, and a narrower beamwidth compared with 12-DRA, which means a better directivity. The traditional subarray, which has the same 4-in-1 structure without phase shifter as shown in Figure 3a, can also achieve a similar narrow beam compared with ML-CDRA. However, the spatial filtering cannot be effectively realized when the undesired targets appear at the



gate lobe. This problem was solved in the proposed ML-CDRA by recovering the received array signal.

Figure 7. The normalized spectrum diagram of the decoded signal after DBF with input SNR = -20 dB.



Figure 8. The normalized array pattern of different DBF systems.

3.3. Spatial Sensitivity

Different from the antenna pattern, the spatial sensitivity describes the relationship between the maximum SNR after DBF and the directions. Signals from different directions with the same input SNR are fed into ML-CDRA in turn. After decoding, the recovered array signals are combined according to the DBF weight of the corresponding direction, e.g., $w = (1, \exp(-j2\pi \frac{d\sin(\theta)}{\lambda}), ..., \exp(-j2\pi \frac{(N-1)d\sin(\theta)}{\lambda})^{T})$, where θ is the direction of arrival.

Figure 9 shows several subarray structure-encoding networks, and the spatial sensitivities of these schemes are given in Figure 10a. Wherever the signal comes from, the full-channel DRA can achieve the same maximum DBF gain by phased coherent combination. Therefore, both 48-DRA and 12-DRA appear as a flat line. It is easy to see that the subarray structure ML-CDRA has a better performance than 12-DRA around the broadside direction. The SNR after the DBF of the 4-in-1 structure is profitable in about $(-20^\circ, 20^\circ)$ compared with 12-DRA. Moreover, if the phase-shifters are introduced into the encoding

network, as shown in Figure 3b, the ML-CDRA can obtain the maximum SNR after DBF in any direction. As shown in Figure 10b, the maximum gain of ML-CDRA is achieved at 15° by adjusting the phase shifters.



Figure 9. Different subarray structure studied for spatial sensitivity.



Figure 10. The spatial sensitivity of subarray structure with input SNR = -20 dB. (a) Without phase shifter. (b) With phase shifter and pointing at 15° .

For the ML-CDRA with random encoding structure, as in Figure 2a, the received array signals are combined into a few RF channels according to a random Bernoulli matrix. Figure 11 shows that the SNR of this structure after DBF is similar to the 48-DRA in the broadside direction, and fluctuates around the 12-DRA in other directions while obtaining a narrower beamwidth, as discussed in Section 3.2. The random encoding structure ML-CDRA can take care of the entire space simultaneously, and the SNR fluctuation in different directions is more robust than the subarray structure.

A different encoding structure means a different spatial sensitivity, which may be an advantage in some scenes. For example, in the case of target tracking, the subarray structure ML-CDRA can obtain the maximum gain in the desired direction using phase shifters, as in Figure 10b. The signal from other directions will be attenuated due to the spatial sensitivity. Therefore, the ML-CDRA can naturally realize spatial anti-jamming.





4. Implementation

To further prove the "low-cost" property, the implementation of the proposed ML-CDRA is discussed based on the commercial chips in this section. The complexity of the decoding network is also studied.

A real-time processing implementation architecture for ML-CDRA is presented in Figure 12. The forward propagation of the decoding network is performed by multipliers and adders in FPGA. The back propagation is carried out by gradient descent in a digital signal processor (DSP) and updated with a coherent processing cycle (CPI). For the full-channel DRA, the DBF weight calculation is also performed in DSP based on the array signals, as shown in Figure 12a. Therefore, the gradient descent of ML-CDRA can be integrated into the same DSP chip without any additional data-transfer overhead.



Figure 12. The implementation architectures for the full-channel DRA and the ML-CDRA. (**a**) A typical architecture for full-channel DRA. (**b**) A real-time processing architecture for ML-CDRA with the decoding network implemented by FPGA.

To realize the real-time decoding, each multiplication and addition operation of the decoding network needs to be configured with independent resources. The pipeline architecture shown in Figure 13, which can execute all the operations of each layer of the decoding network in each clock cycle, is more suitable to realize the real-time decoding compared with the instruction set architecture.

To be more specific, the resources consumed to implement the decoding network can be divided into registers, adders, and multipliers. For a scale-limited network, the logic cells and configurable logic blocks (CLB) in FPGA can supply the registers and adders required in the decoding network. The activation functions of the hidden layer neurons (such as Sigmoid, ReUL, sech, etc.) can be implemented by the look-up tables (LUT). In addition, the multiplication ability can be evaluated by multiply accumulate (MAC). Many operations (such as convolution, dot product, and matrix operations) can be converted to multiple MAC operations. Therefore, the networks requiring convolution operation, such as convolutional neural network (CNN), also have the possibility to be used in the proposed ML-CDRA.



Figure 13. Types of processing architecture. (a) The pipeline architecture. (b) The instruction set architecture.

Consider a *N*-element *M*-channel ML-CDRA, realized by a three-layer, fully connected MLP with *M* neurons in the input layer, *K* neurons in the hidden layer, and *N* neurons in the output layer. The system clock is f_c . For the super-heterodyne receiver, the received signal is divided into I&Q branches after down-conversion. Therefore, the number of multiply accumulate operations performed per second is

$$2(MK + KN) \times f_c \tag{16}$$

Take a 48-element 12-channel ML-CDRA as an example, which is studied in the simulation. The decoding network is realized by a three-layer, fully connected MLP with 12 neurons in the input layer, 1024 neurons in the hidden layer, and 48 neurons in the output layer. Suppose the system clock is 100 MHz. Therefore, the number of multiply accumulate operations performed per second is 2 * (12 * 1024 + 1024 * 48) * 100 MHz = 12,288 GMAC/s.

Table 2 gives several available commercial FPGAs [28,29]. The Virtex UltraScale+ FPGA VU11P and VU13P are powerful enough to implement the above decoding network using only one chip. The Virtex-7 XC7VX690T can also be used to implement the decoding network with lower complexity. In addition, these FPGAs have enough I/O to assign a dedicated pin to each ADC without time division multiplexing. It should be noted that the decoding network mainly consumes the DSP Slice resources. The remaining resources in FPGA can still be used for other functions.

Hence, the proposed ML-CDRA can be implemented by an additional FPGA on the basis of full-channel DRA. As shown in Figure 12, the required RF channels, variable gain amplifiers (VGA), ADCs and I/Os of ML-CDRA are significantly reduced compared with the full-channel DRA. Moreover, the space requirement, heat dissipation and crosstalk of RF channels can also be alleviated. Considering that the additional cost of the decoding network is lower than that of the reduced RF channel overhead, the proposed ML-CDRA is attractive.

Furthermore, for the large-scale decoding network, adding one chip cannot implement the ML-CDRA. A multiple FPGA cascade is a possible means to solve this problem. In addition, the sparsely connected networks may be a potential direction to reduce the redundant computing and resource consumption. Resource reuse is also a good choice in scenes with a low sampling rate. Some new hardware architectures/devices designed for artificial intelligence, such as the Adaptive Compute Acceleration Platform (ACAP) of Xilinx, may solve the real-time decoding challenge from the hardware level.

Table 2. Several available commercial FPGAs for ML-CDRA.

		Logic Cells	CLB			DSP			RAM		
	Device		Slices	Flip-Flops	LUTs	DSP Slices	Fmax /MHz	DSP Performance GMAC/s ¹	Distributed RAM/Mb	Block RAM /Mb	I/O
Virtex-7	XC7VX 690T	693.12 K	108.3 K	866.4 K	433.2 K	3600	741	5335	10.8	52.92	1000
Virtex UltraScale+	VU11P VU13P	2835 K 3780 K	324 K 432 K	2592 K 3456 K	1296 K 1728 K	9216 12,288	891	16,422 21,897	36.2 48.3	70.9 94.5	624 832

¹ Peak DSP performance numbers are based on symmetrical filter implementation, DSP performance = DSP Slices × Fmax × 2.

5. Discussion

As shown in Figure 10, the maximum SNR of the ML-CDRA with 6-in-1 overlapping is higher than 48-DRA in the broadside direction. This is an interesting result of the proposed encoding and decoding architecture. We have two explanations for this result. The first key point is the broadside direction, while the signal comes from the broadside of the array antenna, there is no phase difference between the received array signals. Hence, the array signal is lossless during the encoding. Another key point is the loss function of the decoding network. The proposed generalized loss function includes the maximum likelihood estimation, the signal sparsity constraint, and the noise constraint, where the noise constraint has noise suppression and whitening abilities. Therefore, the proposed ML-CDRA architecture should be considered to have denoising abilities. In conclusion, it is possible to exceed the full channel array in theory.

The encoding and decoding network optimizations may be a key piece of research for the ML-CDRA in the future. This paper does not require the specific network structure of the decoding network, but its importance is indisputable. The system performance may be further improved by using the network structures proposed in recent years. As explained in Section 2.2, the spatial sensitivity of the ML-CDRA is determined by the encoding network.

6. Conclusions

This paper proposes a novel, low-cost DBF system framework called ML-CDRA. The received full array signals are encoded into a few RF channels, and decoded by an ANN in real-time without any data volume loss. The low-cost power-efficient ML-CDRA framework has a similar digital beamforming performance compared with the traditional full-channel receiving array. The encoding and decoding networks can be flexibly designed according

to their requirements in different applications. Moreover, they can be implemented based on the existing commercial chips.

Author Contributions: Conceptualization, L.X. and Y.H.; Data curation, L.X.; Formal analysis, L.X.; Funding acquisition, Y.H.; Investigation, L.X.; Methodology, L.X. and Y.H.; Project administration, L.X.; Resources, Y.H.; Supervision, L.X. and Y.H.; Validation, L.X. and Z.W.; Visualization, L.X. and Y.H.; Writing—original draft, L.X.; Writing—review & editing, L.X. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivatives of the Loss Function

The decoding network is implemented by a three-layer, fully connected MLP network, as shown in Figure A1, which is composed of an input layer, a hidden layer, and an output layer. There are *K* neurons in the hidden layer, and the activation function of the k^{th} neuron is $g_k(t) = \operatorname{sech}[v_k^T \times (y(t) - c_k)]$. The decoded signal $\hat{x}(t) = Wg(t)$, where $g(t) = [g_1(t), g_2(t), ..., g_K(t)]^T$.





The loss function given in Section 2.4 is

$$J = J_1 + \alpha J_2 + \beta J_3 \tag{A1}$$

where

$$J_{1} = \sum_{t=1}^{L} \|\boldsymbol{y}(t) - \boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{\hat{x}}(t)\|_{\Sigma^{-1}}^{2}$$
$$J_{2} = K - \sum_{n=1}^{N} \exp\left(-\frac{\lambda_{n}^{2}(\boldsymbol{R}_{\boldsymbol{\hat{x}}})}{2\delta_{s}^{2}}\right)$$
$$J_{3} = \left\|\boldsymbol{R}_{\boldsymbol{\hat{\eta}}} - \delta_{n}^{2}\boldsymbol{I}\right\|_{\mathrm{F}}^{2}$$

The derivatives of this loss function to W^* , v_k^* , and c_k^* are derived as follows.

Appendix A.1. The Preliminary

The derivatives $D_Z J(Z, Z^*)$ and $D_{Z^*} J(Z, Z^*)$ are defined by the following differential expression

$$d J(\mathbf{Z}, \mathbf{Z}^*) = D_{\mathbf{Z}} J(\mathbf{Z}, \mathbf{Z}^*) d \operatorname{vec}(\mathbf{Z}) + D_{\mathbf{Z}^*} J(\mathbf{Z}, \mathbf{Z}^*) d \operatorname{vec}(\mathbf{Z}^*)$$
(A2)

where

$$D_{\mathbf{Z}}J(\mathbf{Z},\mathbf{Z}^{*}) = \operatorname{vec}^{\mathrm{T}}\left(\frac{\partial}{\partial \mathbf{Z}}J(\mathbf{Z},\mathbf{Z}^{*})\right)$$
$$D_{\mathbf{Z}^{*}}J(\mathbf{Z},\mathbf{Z}^{*}) = \operatorname{vec}^{\mathrm{T}}\left(\frac{\partial}{\partial \mathbf{Z}^{*}}J(\mathbf{Z},\mathbf{Z}^{*})\right)$$

where $\text{vec}(\cdot)$ is the vectorization operator, $(\cdot)^T$ is the transpose operator.

Appendix A.2. The Derivatives of J_1

$$J_{1} = \sum_{t=1}^{L} \|\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t)\|_{\Sigma^{-1}}^{2}$$

$$= \sum_{t=1}^{L} (\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t))^{\mathrm{H}}\boldsymbol{\Sigma}_{\eta}^{-1}(\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t))$$
(A3)

(1) The derivative $D_{W^*}J_1$ is

$$D_{\mathbf{W}^*}J_1 = D_{\widehat{\mathbf{x}}}J_1 \times D_{\mathbf{W}^*}\widehat{\mathbf{x}} + D_{\widehat{\mathbf{x}}^*}J_1 \times D_{\mathbf{W}^*}\widehat{\mathbf{x}}^*$$
(A4)

where

$$D_{W^*}\widehat{x} = \mathbf{0}$$

Therefore,

$$D_{\mathbf{W}^*}J_1 = D_{\widehat{\mathbf{x}}^*}J_1 \times D_{\mathbf{W}^*}\widehat{\mathbf{x}}^*$$

Deriving the derivatives $D_{\hat{x}^*}J_1$ and $D_{W^*}\hat{x}^*$,, respectively. The total differential of (A3) is

$$d(J_1) = \sum_{t=1}^{L} (\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t))^{\mathrm{H}}\boldsymbol{\Sigma}_{\eta}^{-1}(-G\boldsymbol{\Phi})d(\hat{\boldsymbol{x}}) + \sum_{t=1}^{L} (\boldsymbol{y}(t) - G\boldsymbol{\Phi}\hat{\boldsymbol{x}}(t))^{\mathrm{T}}(\boldsymbol{\Sigma}_{\eta}^{-1})^{\mathrm{T}}(-G\boldsymbol{\Phi}^*)d(\hat{\boldsymbol{x}}^*)$$

Therefore, the derivative $D_{\widehat{x}^*} J_1$ is

$$D_{\widehat{\mathbf{x}}^*}J_1 = \sum_{t=1}^{L} (\mathbf{y}(t) - G\boldsymbol{\Phi}\widehat{\mathbf{x}}(t))^{\mathrm{T}} (\boldsymbol{\Sigma}_{\eta}^{-1})^{\mathrm{T}} (-G\boldsymbol{\Phi}^*)$$

The derivative $D_{W^*} \hat{x}^*$ is

$$D_{\mathbf{W}^*}\widehat{\mathbf{x}}^* = \mathbf{g}^{\mathrm{H}}(t) \otimes \mathbf{I}_N$$

where \otimes is the Kronecker product; I_N is the *N*-order identity matrix.

Therefore, the derivative of J_1 to W^* is

$$D_{\mathbf{W}^*} J_1 = \sum_{t=1}^{L} (\mathbf{y}(t) - G \boldsymbol{\Phi} \hat{\mathbf{x}}(t))^{\mathrm{T}} (\boldsymbol{\Sigma}_{\eta}^{-1})^{\mathrm{T}} (-G \boldsymbol{\Phi}^*) (\mathbf{g}^{\mathrm{H}}(t) \otimes \mathbf{I}_N)$$
(A5)

(2) The derivative $D_{v_k^*} J_1$ is

$$D_{v_k^*} J_1 = D_{\widehat{x}} J_1 \times D_{v_k^*} \widehat{x} + D_{\widehat{x}^*} J_1 \times D_{v_k^*} \widehat{x}^*$$
(A6)

where

$$D_{\boldsymbol{v}_k^*} \widehat{\boldsymbol{x}} = D_{g_k} \widehat{\boldsymbol{x}} \times D_{\boldsymbol{v}_k^*} g_k + D_{g_k^*} \widehat{\boldsymbol{x}} \times D_{\boldsymbol{v}_k^*} g_k^*$$
$$D_{\boldsymbol{v}_k^*} \widehat{\boldsymbol{x}}^* = D_{g_k} \widehat{\boldsymbol{x}}^* \times D_{\boldsymbol{v}_k^*} g_k + D_{g_k^*} \widehat{\boldsymbol{x}}^* \times D_{\boldsymbol{v}_k^*} g_k^*$$

Due to

$$D_{g_k} \widehat{\boldsymbol{x}} = \boldsymbol{w}_k$$
$$D_{g_k^*} \widehat{\boldsymbol{x}} = \boldsymbol{0}$$
$$D_{g_k} \widehat{\boldsymbol{x}}^* = \boldsymbol{0}$$
$$D_{g_k^*} \widehat{\boldsymbol{x}}^* = \boldsymbol{w}_k^*$$

D

where $W = [w_1, w_2, ..., w_K]$.

Therefore,

$$D_{\boldsymbol{v}_k^*}J_1 = D_{\boldsymbol{\hat{x}}}J_1 \times D_{g_k}\boldsymbol{\hat{x}} \times D_{\boldsymbol{v}_k^*}g_k + D_{\boldsymbol{\hat{x}}^*}J_1 \times D_{g_k^*}\boldsymbol{\hat{x}}^* \times D_{\boldsymbol{v}_k^*}g_k^*$$

Deriving the derivatives $D_{v_k^*}g_k$ and $D_{v_k^*}g_{k''}^*$ respectively. The total differential of the activation function is

$$d(g_{k}(t)) = \operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})](\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{T}}d(\boldsymbol{v}_{k})$$

+sech'[$\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})](-\boldsymbol{v}_{k}^{\mathrm{T}})d(\boldsymbol{c}_{k})$
$$d(g_{k}^{*}(t)) = (\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{H}}d(\boldsymbol{v}_{k}^{*})$$

+sech'[$\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})](-\boldsymbol{v}_{k}^{\mathrm{H}})d(\boldsymbol{c}_{k}^{*})$

where $\operatorname{sech}'(\cdot)$ is the derivative of $\operatorname{sech}(\cdot)$.

Therefore,

$$D_{\boldsymbol{v}_{k}^{*}}g_{k} = \boldsymbol{0}$$

$$D_{\boldsymbol{v}_{k}^{*}}g_{k}^{*} = (\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{H}}$$

$$D_{\boldsymbol{c}_{k}^{*}}g_{k} = \boldsymbol{0}$$

$$D_{\boldsymbol{c}_{k}^{*}}g_{k}^{*} = -(\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*}\boldsymbol{v}_{k}^{\mathrm{H}}$$

Therefore, the derivative of J_1 to v_k^* is

$$D_{\boldsymbol{v}_{k}^{*}} J_{1} = D_{\boldsymbol{\hat{x}}^{*}} J_{1} \times D_{\boldsymbol{g}_{k}^{*}} \boldsymbol{\hat{x}}^{*} \times D_{\boldsymbol{v}_{k}^{*}} \boldsymbol{g}_{k}^{*}$$

$$= \sum_{t=1}^{L} (\boldsymbol{y}(t) - \boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{\hat{x}}(t))^{\mathrm{T}} (\boldsymbol{\Sigma}_{\eta}^{-1})^{\mathrm{T}} (-\boldsymbol{G}\boldsymbol{\Phi}^{*}) \boldsymbol{w}_{k}^{*} \qquad (A7)$$

$$(\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*} (\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{H}}$$

(3) The derivative $D_{c_k^*} J_1$ is

$$D_{c_k^*} J_1 = D_{\widehat{\mathbf{x}}^*} J_1 \times D_{g_k^*} \widehat{\mathbf{x}}^* \times D_{c_k^*} g_k^*$$

= $\sum_{t=1}^{L} (\mathbf{y}(t) - G \mathbf{\Phi} \widehat{\mathbf{x}}(t))^{\mathrm{T}} (\mathbf{\Sigma}_{\eta}^{-1})^{\mathrm{T}} (-G \mathbf{\Phi}^*) \mathbf{w}_k^*$ (A8)
(sech' $[\mathbf{v}_k^{\mathrm{T}} (\mathbf{y}(t) - \mathbf{c}_k)])^* (-\mathbf{v}_k^{\mathrm{H}})$

The derivation of derivative $D_{c_k^*} J_1$ is the same as $D_{v_k^*} J_1$.

Appendix A.3. The Derivatives of J_2

$$J_2 = K - \sum_{n=1}^{N} \exp\left(-\frac{\lambda_n^2(\boldsymbol{R}_{\hat{\boldsymbol{x}}})}{2\delta_s^2}\right)$$
(A9)

where $\mathbf{R}_{\hat{\mathbf{x}}} = \frac{1}{L} \sum_{t=1}^{L} \hat{\mathbf{x}}(t) \hat{\mathbf{x}}^{H}(t)$. (1) The derivative $D_{W^*} J_2$ is

$$D_{W^*}J_2 = D_{\hat{x}}J_2 \times D_{W^*}\hat{x} + D_{\hat{x}^*}J_2 \times D_{W^*}\hat{x}^*$$
(A10)

18 of 21

According to the derivatives of J_1

$$D_{\mathbf{W}^*} \widehat{\mathbf{x}} = \mathbf{0}$$

 $D_{\mathbf{W}^*} \widehat{\mathbf{x}}^* = \mathbf{g}^{\mathrm{H}}(t) \otimes \mathbf{I}_{\mathrm{N}}$

Therefore,

where

$$D_{\widehat{\mathbf{x}}^*}J_2 = D_{\lambda_n}J_2 \times D_{R_{\widehat{\mathbf{x}}}}\lambda_n \times D_{\widehat{\mathbf{x}}^*}R_{\widehat{\mathbf{x}}}$$

 $D_{W^*}J_2 = D_{\widehat{x}^*}J_2 \times D_{W^*}\widehat{x}^*$

Deriving the derivatives $D_{\lambda_n} J_2$, $D_{R_{\hat{x}}} \lambda_n$ and $D_{\hat{x}^*} R_{\hat{x}}$, respectively.

$$D_{\lambda_n} J_2 = \sum_{n=1}^N \frac{\lambda_n(\mathbf{R}_{\hat{\mathbf{x}}})}{\delta_s^2} \exp\left(-\frac{\lambda_n^2(\mathbf{R}_{\hat{\mathbf{x}}})}{2\delta_s^2}\right)$$

According to [30], the derivative $D_{R_{\hat{x}}}\lambda_n$ is

$$D_{\boldsymbol{R}_{\hat{\boldsymbol{x}}}}\lambda_n = \operatorname{vec}^{\mathrm{T}}(\frac{\boldsymbol{v}_n^*\boldsymbol{u}_n^{\mathrm{T}}}{\boldsymbol{v}_n^{\mathrm{H}}\boldsymbol{u}_n})$$

where $v_n^{\rm H}$ and u_n are the left and right eigenvector of $R_{\hat{x}}$, respectively.

$$R_{\hat{x}}u_n = \lambda_n u_n$$
$$v_n^{\rm H} R_{\hat{x}} = v_n^{\rm H} \lambda_n$$

The derivative $D_{\widehat{x}^*} R_{\widehat{x}}$ is

$$D_{\widehat{\mathbf{x}}^*} \mathbf{R}_{\widehat{\mathbf{x}}} = \frac{1}{L} \sum_{t=1}^{L} (\mathbf{I}_N \otimes \widehat{\mathbf{x}}(t))$$

Therefore,

$$D_{\mathbf{W}^{*}}J_{2} = \sum_{n=1}^{N} \frac{\lambda_{n}(\mathbf{R}_{\widehat{\mathbf{x}}})}{\delta_{s}^{2}} \exp\left(-\frac{\lambda_{n}^{2}(\mathbf{R}_{\widehat{\mathbf{x}}})}{2\delta_{s}^{2}}\right) \operatorname{vec}^{\mathrm{T}}\left(\frac{\boldsymbol{v}_{n}^{*}\boldsymbol{u}_{n}^{\mathrm{T}}}{\boldsymbol{v}_{n}^{\mathrm{H}}\boldsymbol{u}_{n}}\right)$$

$$\frac{1}{L}\sum_{t=1}^{L} (\mathbf{I}_{N} \otimes \widehat{\mathbf{x}}(t))(\boldsymbol{g}^{\mathrm{H}}(t) \otimes \mathbf{I}_{N})$$
(A11)

(2) The derivative $D_{v_k^*} J_2$ is

$$D_{\boldsymbol{v}_{k}^{*}}J_{2} = D_{\boldsymbol{\widehat{x}}^{*}}J_{2} \times D_{\boldsymbol{v}_{k}^{*}}\boldsymbol{\widehat{x}}^{*}$$

$$= \sum_{n=1}^{N} \frac{\lambda_{n}(\boldsymbol{R}_{\boldsymbol{\widehat{x}}})}{\delta_{s}^{2}} \exp\left(-\frac{\lambda_{n}^{2}(\boldsymbol{R}_{\boldsymbol{\widehat{x}}})}{2\delta_{s}^{2}}\right) \operatorname{vec}^{\mathrm{T}}\left(\frac{\boldsymbol{v}_{n}^{*}\boldsymbol{u}_{n}^{\mathrm{T}}}{\boldsymbol{v}_{n}^{\mathrm{H}}\boldsymbol{u}_{n}}\right)$$

$$\frac{1}{L} \sum_{t=1}^{L} (\boldsymbol{I}_{N} \otimes \boldsymbol{\widehat{x}}(t)) \boldsymbol{w}_{k}^{*} (\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*} (\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{H}}$$
(A12)

The derivation of derivative $D_{v_k^*} J_2$ is the same as $D_{v_k^*} J_1$.

(3) The derivative $D_{c_k^*} J_2$ is

$$D_{c_{k}^{*}}J_{2} = D_{\widehat{x}^{*}}J_{2} \times D_{c_{k}^{*}}\widehat{x}^{*}$$

$$= \sum_{n=1}^{N} \frac{\lambda_{n}(\boldsymbol{R}_{\widehat{x}})}{\delta_{s}^{2}} \exp\left(-\frac{\lambda_{n}^{2}(\boldsymbol{R}_{\widehat{x}})}{2\delta_{s}^{2}}\right) \operatorname{vec}^{\mathrm{T}}\left(\frac{\boldsymbol{v}_{n}^{*}\boldsymbol{u}_{n}^{\mathrm{T}}}{\boldsymbol{v}_{n}^{\mathrm{H}}\boldsymbol{u}_{n}}\right)$$

$$\frac{1}{L} \sum_{t=1}^{L} (\boldsymbol{I}_{N} \otimes \widehat{x}(t)) \boldsymbol{w}_{k}^{*} (\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*} (-\boldsymbol{v}_{k}^{\mathrm{H}})$$
(A13)

The derivation of derivative $D_{c_k^*}J_2$ is the same as $D_{c_k^*}J_1$.

Appendix A.4. The Derivatives of J_3

$$J_3 = \left\| \boldsymbol{R}_{\widehat{\boldsymbol{\eta}}} - \delta_n^2 \boldsymbol{I}_N \right\|_F^2 \tag{A14}$$

where $R_{\hat{\eta}} = \frac{1}{L} \sum_{t=1}^{L} \hat{\eta}(t) \hat{\eta}^{H}(t)$. (1) The derivative $D_{W^*} J_3$ is

$$D_{W^*}J_3 = D_{\widehat{\eta}}J_3 \times D_{W^*}\widehat{\eta} + D_{\widehat{\eta}^*}J_3 \times D_{W^*}\widehat{\eta}^*$$
(A15)

According to the derivatives of J_1

$$D_{W^*} \widehat{\boldsymbol{\eta}} = \boldsymbol{0}$$

 $D_{W^*} \widehat{\boldsymbol{\eta}}^* = \boldsymbol{g}^{\mathrm{H}}(t) \otimes \boldsymbol{I}_{\mathrm{N}}$

Therefore,

$$D_{\mathbf{W}^*}J_3 = D_{\widehat{\boldsymbol{\eta}}^*}J_3 \times D_{\mathbf{W}^*}\widehat{\boldsymbol{\eta}}^*$$

where

$$D_{\widehat{\eta}^*}J_3 = D_{R_{\widehat{\eta}}}J_3 \times D_{\widehat{\eta}^*}R_{\widehat{\eta}} + D_{R_{\widehat{\eta}}^*}J_3 \times D_{\widehat{\eta}^*}R_{\widehat{\eta}}^*$$

Deriving the derivatives $D_{R_{\hat{\eta}}}J_3$, $D_{\hat{\eta}^*}R_{\hat{\eta}}$, $D_{R_{\hat{\eta}}^*}J_3$ and $D_{\hat{\eta}^*}R_{\hat{\eta}}^*$, respectively.

$$D_{R_{\hat{\eta}}}J_{3} = \operatorname{vec}^{H}(R_{\hat{\eta}} - \delta_{n}^{2}I_{N})$$
$$D_{R_{\hat{\eta}}^{*}}J_{3} = \operatorname{vec}^{T}(R_{\hat{\eta}} - \delta_{n}^{2}I_{N})$$
$$D_{\hat{\eta}^{*}}R_{\hat{\eta}} = \frac{1}{L}\sum_{t=1}^{L}(I_{N} \otimes \hat{\eta}(t))$$
$$D_{\hat{\eta}^{*}}R_{\hat{\eta}}^{*} = \frac{1}{L}\sum_{t=1}^{L}(\hat{\eta}(t) \otimes I_{N})$$

Therefore,

$$D_{\mathbf{W}^*} J_3 = \left[\operatorname{vec}^{\mathrm{H}}(\mathbf{R}_{\widehat{\eta}} - \delta_n^2 \mathbf{I}_N) \frac{1}{L} \sum_{t=1}^{L} (\mathbf{I}_N \otimes \widehat{\eta}(t)) + \operatorname{vec}^{\mathrm{T}}(\mathbf{R}_{\widehat{\eta}} - \delta_n^2 \mathbf{I}_N) \frac{1}{L} \sum_{t=1}^{L} (\widehat{\eta}(t) \otimes \mathbf{I}_N) \right] (\mathbf{g}^{\mathrm{H}}(t) \otimes \mathbf{I}_N)$$
(A16)

(2) The derivative $D_{v_{\mu}^{*}}J_{3}$ is

$$D_{\boldsymbol{v}_{k}^{*}}J_{3} = \left[\operatorname{vec}^{\mathrm{H}}(\boldsymbol{R}_{\widehat{\boldsymbol{\eta}}} - \delta_{n}^{2}\boldsymbol{I}_{N})\frac{1}{L}\sum_{t=1}^{L}(\boldsymbol{I}_{N}\otimes\widehat{\boldsymbol{\eta}}(t)) + \operatorname{vec}^{\mathrm{T}}(\boldsymbol{R}_{\widehat{\boldsymbol{\eta}}} - \delta_{n}^{2}\boldsymbol{I}_{N})\frac{1}{L}\sum_{t=1}^{L}(\widehat{\boldsymbol{\eta}}(t)\otimes\boldsymbol{I}_{N})\right] \times \boldsymbol{w}_{k}^{*}(\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})^{\mathrm{H}}$$
(A17)

The derivation of derivative $D_{v_k^*} J_3$ is the same as $D_{v_k^*} J_1$. (3) The derivative $D_{c_k^*} J_3$ is

$$D_{\boldsymbol{c}_{k}^{*}}J_{3} = \left[\operatorname{vec}^{\mathrm{H}}(\boldsymbol{R}_{\widehat{\boldsymbol{\eta}}} - \delta_{n}^{2}\boldsymbol{I}_{N})\frac{1}{L}\sum_{t=1}^{L}(\boldsymbol{I}_{N}\otimes\widehat{\boldsymbol{\eta}}(t)) + \operatorname{vec}^{\mathrm{T}}(\boldsymbol{R}_{\widehat{\boldsymbol{\eta}}} - \delta_{n}^{2}\boldsymbol{I}_{N})\frac{1}{L}\sum_{t=1}^{L}(\widehat{\boldsymbol{\eta}}(t)\otimes\boldsymbol{I}_{N})\right]$$

$$\times \boldsymbol{w}_{k}^{*}(\operatorname{sech}'[\boldsymbol{v}_{k}^{\mathrm{T}}(\boldsymbol{y}(t) - \boldsymbol{c}_{k})])^{*}(-\boldsymbol{v}_{k}^{\mathrm{H}})$$
(A18)

The derivation of derivative $D_{c_k^*} J_3$ is the same as $D_{c_k^*} J_1$.

Appendix A.5. The Derivatives of J

Finally, the derivatives of *J* to W^* , v_k^* , and c_k^* are

$$D_{W^*}J = D_{W^*}J_1 + \alpha D_{W^*}J_2 + \beta D_{W^*}J_3$$

$$D_{v_k^*}J = D_{v_k^*}J_1 + \alpha D_{v_k^*}J_2 + \beta D_{v_k^*}J_3$$

$$D_{c_k^*}J = D_{c_k^*}J_1 + \alpha D_{c_k^*}J_2 + \beta D_{c_k^*}J_3$$
(A19)

References

- Cantrell, B.; de Graaf, J.; Willwerth, F.; Meurer, G.; Leibowitz, L.; Parris, C.; Stapleton, R. Development of a Digital Array Radar (DAR). *IEEE Aerosp. Electron. Syst. Mag.* 2002, 17, 22–27. [CrossRef]
- Zhang, R.; Shim, B.; Wu, W. Direction-of-Arrival Estimation for Large Antenna Arrays with Hybrid Analog and Digital Architectures. *IEEE Trans. Signal Process.* 2022, 70, 72–88. [CrossRef]
- 3. Ibrahim, M.; Ramireddy, V.; Lavrenko, A.; König, J.; Römer, F.; Landmann, M.; Grossmann, M.; Del Galdo, G.; Thomä, R.S. Design and analysis of compressive antenna arrays for direction of arrival estimation. *Signal Process.* **2017**, *138*, 35–47. [CrossRef]
- 4. Shen, Q.; Liu, W.; Cui, W.; Wu, S. Underdetermined DOA Estimation Under the Compressive Sensing Framework: A Review. *IEEE Access* 2016, *4*, 8865–8878. [CrossRef]
- Wang, J.; Sheng, W.; Han, Y.; Ma, X. Adaptive beamforming with compressed sensing for sparse receiving array. *IEEE Trans. Aerosp. Electron. Syst.* 2014, 50, 823–833. [CrossRef]
- Bajor, M.; Haque, T.; Han, G.; Zhang, C.; Wright, J.; Kinget, P.R. A Flexible Phased-Array Architecture for Reception and Rapid Direction-of-Arrival Finding Utilizing Pseudo-Random Antenna Weight Modulation and Compressive Sampling. *IEEE J. Solid-State Circuits* 2019, 54, 1315–1328. [CrossRef]
- Rahman, M.R.; Bojja-Venkatakrishnan, S.; Alwan, E.A.; Volakis, J.L. Spread Spectrum Techniques with Channel Coding for Wideband Secured Communication Links. In Proceedings of the 2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting, Toronto, ON, Canada, 5–10 July 2020; pp. 1783–1784. [CrossRef]
- Alwan, E.A.; Khalil, W.; Volakis, J.L. Ultra-wideband on-site coding receiver (OSCR) for digital beamforming applications. In Proceedings of the 2013 IEEE Antennas and Propagation Society International Symposium (APSURSI), Orlando, FL, USA, 7–13 July 2013; pp. 620–621. [CrossRef]
- Ullah, K.; Venkatakrishnan, S.B.; Volakis, J.L. Low Power and Low Cost Millimeter-Wave Digital Beamformer Using an Orthogonal Coding Scheme. In Proceedings of the 2021 USNC-URSI Radio Science Meeting (USCN-URSI RSM), Honolulu, HI, USA, 9–13 August 2021; pp. 64–68. [CrossRef]
- Zhang, J.; Wu, W.; Fang, D.G. Single RF Channel Digital Beamforming Multibeam Antenna Array Based on Time Sequence Phase Weighting. *IEEE Antennas Wirel. Propag. Lett.* 2011, 10, 514–516. [CrossRef]
- 11. Zhang, D.; Zhang, J.D.; Cui, C.; Wu, W.; Fang, D.G. Single RF Channel Digital Beamforming Array Antenna Based on Compressed Sensing for Large-Scale Antenna Applications. *IEEE Access* **2018**, *6*, 4340–4351. [CrossRef]

- 12. Ma, Y.; Miao, C.; Wu, W.; Li, Y. Hyper Beamforming in Time Modulated Linear Arrays. In Proceedings of the 2020 IEEE Asia-Pacific Microwave Conference (APMC), Hong Kong SAR, China, 8–11 December 2020 ; pp. 448–450. [CrossRef]
- Ding, J.; Chen, L.; Gu, Y. Perturbation Analysis of Orthogonal Matching Pursuit. *IEEE Trans. Signal Process.* 2013, 61, 398–410. [CrossRef]
- 14. Donoho, D.L. Compressed sensing. IEEE Trans. Inf. Theory 2006, 52, 1289–1306. [CrossRef]
- Bian, S.; Zhang, L. Overview of Match Pursuit Algorithms and Application Comparison in Image Reconstruction. In Proceedings of the 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2021; pp. 216–221. [CrossRef]
- Yaghoobi, M.; Wu, D.; Davies, M.E. Fast Non-Negative Orthogonal Matching Pursuit. *IEEE Signal Process. Lett.* 2015, 22, 1229–1233. [CrossRef]
- Han, B.; Jiang, H. An Improved Stagewise Orthogonal Matching Pursuit-Recursive Algorithm for Power Quality Signal Reconstruction. In Proceedings of the 2019 Computing, Communications and IoT Applications (ComComAp), Shenzhen, China, 26–28 October 2019; pp. 293–298. [CrossRef]
- Hanzo, L.L.; Yang, L.L.; Kuan, E.L.; Yen, K. CDMA Overview. In Single and Multi-Carrier DS-CDMA: Multi-User Detection, Space-Time Spreading, Synchronisation, Networking and Standards; IEEE Press: Piscataway, NJ, USA, 2004; pp. 35–80. [CrossRef]
- 19. Chen, Q.; Zhang, J.D.; Wu, W.; Fang, D.G. Enhanced Single-Sideband Time-Modulated Phased Array With Lower Sideband Level and Loss. *IEEE Trans. Antennas Propag.* 2020, *68*, 275–286. [CrossRef]
- Wang, X.; Greco, M.S.; Gini, F. Adaptive Sparse Array Beamformer Design by Regularized Complementary Antenna Switching. IEEE Trans. Signal Process. 2021, 69, 2302–2315. [CrossRef]
- 21. Tabrikian, J.; Isaacs, O.; Bilik, I. Cognitive Antenna Selection for Automotive Radar Using Bobrovsky-Zakai Bound. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 892–903. [CrossRef]
- Garcia-Rodriguez, A.; Masouros, C.; Rulikowski, P. Efficient large scale antenna selection by partial switching connectivity. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 6269–6273. [CrossRef]
- Lin, T.; Zhu, Y. Beamforming Design for Large-Scale Antenna Arrays Using Deep Learning. *IEEE Wirel. Commun. Lett.* 2020, 9, 103–107. [CrossRef]
- 24. Huang, H.; Peng, Y.; Yang, J.; Xia, W.; Gui, G. Fast Beamforming Design via Deep Learning. *IEEE Trans. Veh. Technol.* 2020, 69, 1065–1069. [CrossRef]
- 25. Zhao, X.; Yang, Y.; Chen, K.S. Direction-of-Arrival Estimation over Sea Surface from Radar Scattering Based on Convolutional Neural Network. *Remote Sens.* **2021**, *13*, 2681. [CrossRef]
- 26. Zhou, W.W.; Sheng, W.X.; Cui, J.; Han, Y.B. SR-Crossbar topology for large-scale RF MEMS switch matrices. *IET Microwaves Antennas Propag.* 2019, 13, 231–238. [CrossRef]
- Komolafe, T.E.; Wang, K.; Du, Q.; Hu, T.; Yuan, G.; Zheng, J.; Zhang, C.; Li, M.; Yang, X. Smoothed L0-Constraint Dictionary Learning for Low-Dose X-Ray CT Reconstruction. *IEEE Access* 2020, *8*, 116961–116973. [CrossRef]
- 28. Xilinx. 7 Series FPGAs Data Sheet: Overview; Xilinx: San Jose, CA, USA, 2020; Rev. 2.6.1.
- 29. Xilinx. UltraScale Architecture and Product Data Sheet: Overview; Xilinx: San Jose, CA, USA, 2022; Rev. 4.1.1.
- Hjorungnes, A.; Gesbert, D. Complex-Valued Matrix Differentiation: Techniques and Key Results. *IEEE Trans. Signal Process.* 2007, 55, 2740–2746. [CrossRef]