UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

FAST ADAPTIVE DIGITAL BEAMFORMING WITH MACHINE LEARNING

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By

YOON-SL KIM
Norman, Oklahoma
2022

FAST ADAPTIVE DIGITAL BEAMFORMING WITH MACHINE LEARNING

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. David Schvartzman Cohenca, Chair

Dr. Robert Palmer

Dr. Tian-You Yu

Dedicated to Acorn, Bee, and Cream

Acknowledgments

One of the best blessing is to have a great teacher in your life. First I would like to thank my advisor, Dr. David Schvartzman. He taught and guided me and opened up a new world of becoming a researcher. I enjoyed our discussion to explore many different topics related to radars. I was inspired and motivated by his passion for the research. Also, I appreciate that he always puts students as the top priority and is ready to spend many hours tackling research problems with me. I would also like to thank my committee members, Dr. Robert Palmer and Dr. Tian-You Yu. I appreciate how they have taught me from my first day at the Advanced Radar Research Center until the day I defended my Master thesis. Their support and encouragement had made this journey more exciting.

There were ups and downs during this journey, and I was able to overcome challenging times and overwhelming moments because of my best friends, Acorn, Bee, and Cream. They complete my life.

Table of Contents

Acknowledgments **v**

Table of Contents **vii**

List of Tables **viii**

List of Figures **xiv**

Abstract **xv**

1 Introduction **1**

1.1 Background 1

1.2 Motivation 4

1.3 Contributions and Outline 4

2 Review of Artificial Intelligence Concepts **7**

2.1 Machine Learning 7

2.2 Deep Neural Networks 10

2.2.1 Activation functions 13

2.2.2 Data Normalization 17

3 Simulation of Radar Signals **22**

3.1 Radar Fundamentals 22

3.2 Phased Array Radar 29

3.3 Simulation of Digital Phased Array Radar 33

3.3.1 Simulation of Point Target Signals 34

3.3.2 Simulation of Weather-like Signals 35

3.3.3 Digital Beamforming 41

3.4 Computational Complexity of Digital Beamforming Methods 55

4 Deep Learning for Adaptive Digital Beamforming 57

4.1 Training Data for DLAF 58

4.2 Architecture of the Neural Network 60

4.3 Methodology 65

4.3.1 Complexity of the Neural Network 68

5 Application and Evaluation of DLAF 70

5.1 Qualitative Analysis 70

5.2 Quantitative Analysis 84

6 Conclusion and Future Work 92

6.1 Summary and Conclusion 92

6.2 Future Work 95

References 96

Appendix A Acronyms 107

List of Tables

3.1 Point target simulation parameters based on TEP specification 34

3.2 Weather Simulation Parameters: 1D Horus 43

List of Figures

2.1 Ridge classification example 9

2.2 Saliency example 10

2.3 Classification of ML 10

2.4 Illustration of DNN 12

2.5 Beam pattern generated by ML DBF method in biomedical field 13

2.6 Synthetic aperture images by ML DBF in biomedical field 14

2.7 Behavior of several activation functions 15

2.8 Test image for evaluating behavior of different activation functions 17

2.9 Output images from activation functions 18

2.10 Evaluation of batch and layer normalization methods 21

3.1 Illustration of Phase Array Radar (PAR) 29

3.2 Fundamental concept of PAR 30

3.3 The National Weather Radar Testbed (NWRT) 31

3.4 Image of the Turbulent Eddy Profiler (TEP) 32

3.5 Multifunction Phased Array Radar (MPAR) 33

3.6 Polarimetric Atmospheric Imaging Radar (PAIR) 34

3.7 The mobile fully digital Horus radar 35

3.8 Geometry of TEP 36

3.9 Location of a simulated moving point target 36

3.10 Radar variables of tornadic supercell storm case from NEXRAD 37

3.11 Radar variables of hail storm case from NEXRAD 37

3.12 Radar variables of thunderstorm case from NEXRAD 38

3.13 Output from robust filling and de-aliasing for the tornadic supercell storm case 39

3.14 Output from robust filling and de-aliasing for the hail storm case 39

3.15 Output from robust filling and de-aliasing for the thunderstorm Case 40

3.16 One way of transmit pattern for Horus 41

3.17 Illustration of PAR simulation 42

3.18 Signal-to-Noise Ratio (SNR) of simulated spoiled beams for weather cases 42

3.19 Illustration of Digital BeamForming (DBF) 43

3.20 Illustration of non-adaptive and adaptive DBF methods 44

3.21 Antenna patterns generated by the Fourier and the Capon methods based on TEP specification 45

3.22 Power plot of the Fourier method with simulated point target data 48

3.23 Power plot of the Capon method with simulated point target data 48

3.24 The movement of a bird observed by TEP 49

3.25 Power plot of the Fourier method with real TEP data 49

3.26 Power plot of the Capon method with real TEP data 50

3.27 Power and reflectivity field of the Fourier and the Capon methods: Tornadic supercell storm case 51

3.28 Antenna pattern generated by the Fourier and the Capon methods around a stationary point target: Tornadic supercell storm case 51

3.29 Antenna pattern generated by the Fourier and the Capon methods for a tornadic supercell storm case 52

3.30 Power and reflectivity field of the Fourier and the Capon methods for a hail storm case 52

3.31 Antenna pattern generated by the Fourier and the Capon methods around a stationary point target for the hail storm case 53

3.32 Antenna pattern generated by the Fourier and the Capon methods for the hail storm case 53

3.33 Power and reflectivity field of the Fourier and the Capon methods for the thunderstorm case 54

3.34 Antenna pattern generated by the Fourier and the Capon methods around a stationary point target for the thunderstorm case 54

3.35 Antenna pattern generated by the Fourier and the Capon methods in weather area for the thunderstorm case 55

3.36 Complexity order of the Fourier and the Capon methods 56

4.1 Flowchart of the proposed neural network for adaptive DBF 57

4.2 Structure of input and output data for DLAF Training 59

4.3 Architecture of the neural network. It contains 7 convolution layers followed by normalization layers and activation layers, and a dense layer as an output layer. 60

4.4 Illustration of phase and amplitude of training data for DLAF 61

4.5 Illustration of outputs from 64 convolution filters applied to the phase training data 62

4.6 Illustration of outputs from 64 convolution filters applied to the amplitude training dat 63

4.7 First set of convolution, normalization, and activation layer applied to the phase training data 64

4.8 First set of convolution, normalization, and activation layer applied to the amplitude training data 64

4.9 Last set of convolution, normalization, and activation layer applied to the phase training data 64

4.10 Last set of convolution, normalization, and activation layer applied to the amplitude training data 65

4.11 Output from flatten and dense layers 66

4.12 Training output from DLAF for phase data 66

4.13 Training output from DLAF for amplitude data 67

4.14 Training results of the DLAF 67

4.15 Loss with layer normalization 68

4.16 Loss with batch normalization 68

5.1 Power plot of simulated moving point target by the Capon, the Fourier, and DLAF methods 72

5.2 3D antenna pattern plot of simulated moving point target by the Capon, the Fourier, and DLAF methods 73

5.3 2D antenna pattern plot of simulated moving point target by the Capon, the Fourier, and DLAF methods 73

5.4 Power plot of real point target data by the Capon, the Fourier, and DLAF methods 75

5.5 3D antenna pattern plot of real point target data by the Capon, the Fourier, and DLAF methods 76

5.6 2D antenna pattern plot of real point Target Data by the Capon, the Fourier, and DLAF methods 76

5.7 Power PPI by the Capon, the Fourier, and DLAF methods of the supercell case 77

5.8 Power by the Capon, the Fourier, and DLAF methods at the range 21.125km for 360° in azimuth of the supercell case 78

5.9 Reflectivity field (dBZ) of the supercell case by the Capon, the Fourier, and DLAF methods 78

5.10 Antenna pattern by the Capon, the Fourier, and DLAF methods of the supercell storm in the point target area 78

5.11 Antenna pattern by the Capon, the Fourier, and DLAF of the supercell case in the weather area 79

5.12 Power PPI by the Capon, the Fourier, and DLAF methods of the hail storm case 80
5.13 Power by the Capon, the Fourier, and DLAF methods at the range 21.125km for 360° in azimuth of the hail storm case 80
5.14 Reflectivity field of the hail case by the Capon, the Fourier, and DLAF methods 81
5.15 Antenna pattern by the Capon, the Fourier, and DLAF methods for the hail storm case in point target area 81
5.16 Antenna pattern by the Capon, the Fourier, and DLAF methods for the hail storm case in the point weather area 81
5.17 Power PPI by the Capon, the Fourier, and DLAF methods of the thunderstorm Case 82
5.18 Power by the Capon, the Fourier, and DLAF at the range 21.125km for360° in azimuth of the thunderstorm case 82
5.19 Reflectivity field of the thunderstorm case by the Capon, the Fourier, and DLAF methods 83
5.20 Antenna pattern by the Capon, the Fourier, and DLAF methods for the thunderstorm case in the point target area 83
5.21 Antenna pattern of the Capon, the Fourier, and DLAF methods for the thunderstorm case in point weather area 83
5.22 Measured computation time of Capon and DLAF DBF 84
5.23 MSE between the Capon and DLAF beamforming weights in real and imaginary parts 86
5.24 MSE between the Capon and DLAF beamforming weights in phase and amplitude 87
5.25 Power by the Capon, the Fourier, and DLAF methods for different SNR . . 89
5.26 Antenna pattern by the Capon, the Fourier, and DLAF methods for the point target area and the weather area with different SNR 90

5.27 Correlation coefficient of generated patterns by the Capon and DLAF methods for the point target area 90

5.28 Correlation coefficient of generated patterns by the Capon and DLAF methods for the weather area 91

Abstract

Digital polarimetric Phased Array Radar (PAR) has the potential to achieve high temporal and spatial resolution needed for next-generation weather observations. It offers flexibility in scanning strategies, including the use of electronic steering, coupled with traditional "pencil" beams, spoiled beams, and/or multiple beams. Digital beamforming (DBF) is one of the unique capabilities of PAR used to simultaneously form multiple beams digitally, a critical capability needed to improve temporal resolution. DBF can be done with non-adaptive or adaptive methods. The Fourier method pre-computes the beamforming weights based on the configuration of the antenna elements, which results in fixed sidelobe levels and angular resolution. In contrast, the Capon method computes the beamforming weights adaptively from receive signals. The Capon method can provide enhanced angular resolution over the Fourier method by lowering antenna sidelobe levels in directions with strong contaminating returns. However, it is challenging to implement the Capon method in real-time as the number of antenna elements grows due to the high computational cost.

This work proposes a computationally efficient Deep Learning Adaptive digital beam-Forming (DLAF) method using a neural network trained with time-series IQ signals as input and the Capon beamforming weights as output. The network is built with a combination of convolution, activation, normalization, and dense layers. The novelty of the proposed network is that it implements an AI-based adaptive beamforming method preserving signal phase information, which improves the efficiency and accuracy of the learning. The proposed DLAF model is evaluated using a simulated point target, an actual point target, and simulated weather signals for a planar digital array architecture. Specifically the performance of DLAF is evaluated and compared to both the Capon and Fourier methods. Two qualitative metrics used for the evaluation are the quality of the generated beam pattern, such as the half-power beamwidth, artifacts from sidelobes, and the capability of placing nulls at locations with strong interference signals. Four quantitative metrics used for the evaluation are computation time, the performance of DLAF as a function of SNR, mean squared error of beamforming weight values, and normalized cross-correlation of generated beam patterns. Preliminary results show that DLAF was trained successfully to perform adaptive DBF and mimic the behavior of the Capon method for point and distributed targets. Furthermore, DLAF can significantly reduce the computational time which allows real-time implementation of adaptive DBF.

Chapter 1

Introduction

1.1 Background

Radar technology has existed since the 1940s [1] and was largely developed during the World War II. After the development of initial defense radars, weather radar systems were produced, with most based on parabolic-reflector antennas. In the late 1980s, dual-polarization technology research for meteorological observations started developing [2], whereby a radar can transmit and receive horizontally (H) and vertically (V) polarized electric fields simultaneously. These two orthogonally polarized fields can provide information of the shape, orientation, and size of the scatters in the radar resolution volume [3]. The U.S. operational weather radar, named Weather Surveillance Radar-1988 Doppler (WSR-88D), provides observations of the atmosphere to estimate spectral moments such as reflectivity factor (Z_{h}), Doppler velocity (v_{r}), and spectrum width (σ_{v}) [4]. Since the upgrade of the WSR-88D network to dual-polarization capability, completed in 2012 [5], the radar also provides the estimates of polarimetric variables such as differential reflectivity (Z_{DR}), differential phase (Φ_{DP}), and co-polar correlation coefficient (ρ_{hv}). Z_{DR} is defined as the ratio of the H and V reflectivities, conventionally expressed in dB. Φ_{DP} is the difference between the phases of signals received at the H and V polarization channels. Lastly, ρ_{hv} is the normalized cross correlation coefficient between returned signals at the H and V polarization channels [6].

Severe weather, such as tornado-producing supercells, evolves fast and requires high-temporal resolution observations to improve our physical understanding of underlying processes driving tornadogenesis [7]. Yet, conventional radars with parabolic dish are constrained by the antenna's mechanical inertia due to rotation [4], which imposes an inherent limitation on temporal resolution. Therefore, Phased Array Radar (PAR) has been introduced as a promising technology for polarimetric weather observations with high-temporal resolution due to its unique capabilities, such as electronic steering and digital beamforming (DBF) [8].

The phase array antenna consists of small radiating elements, each of which has an amplitude and phase controller. PAR technology reduces the need for mechanical rotation by steering the beam electronically in the desired direction, which is know as *beam agility*. This is done by controlling the excitation phases applied to individual antenna elements [9]. For example, the beam multiplexing (BMX) technique exploits beam agility to reduce the variance of estimates without increasing scan time [10]. To optimize the flexibility in scanning strategies that support the use of the traditional pencil beams, spoiled beams, and/or multiple beams, all-digital PAR was introduced [11]. The "Horus" radar is an all-digital, S-band, polarimetric PAR developed by the Advanced Radar Research Center (ARRC) at the University of Oklahoma, funded by the National Oceanic and Atmospheric Administration (NOAA) [12]. The Horus antenna consists of 1024 (32×32) fully digital polarimetric antenna elements [13], that can be used to transmit (TX) and receive (RX) arbitrary waveforms. Horus is envisioned to operate as a rotating PAR (RPAR), steering the beam electronically (mostly in elevation) and rotating mechanically in azimuth. This is the most promising PAR architecture to achieve requirements needed for polarimetric meteorological observations [14, 15].

Radar imaging involves the transmission of a wide spoiled beam and the use of DBF to form multiple simultaneous narrow beams on reception, within the transmit beam [16, 17]. DBF is a unique capability of PARs, which can be used to improve temporal resolution. Hence, a wide sector can be observed quickly, reducing the overall scan time [18–20]. For example, the X-band phased array weather radar (PAWR) described by [21] performs DBF in a wide elevation sector (-5° to 87°) and rotates in azimuth to observe a three-dimensional space within 80 km in range and up to 17 km altitude every 30 s. DBF can be done by either predetermining the beamforming weights, considered a non-adaptive method, or calculating beamforming weights based on received signals, considered an adaptive method. For the non-adaptive methods, the beamforming weights (amplitude and phase) applied to the antenna elements determines the beam pattern, which affects performance [22]. The Fourier DBF method can be used to determine the beamforming weights by calculating the phase differences needed between antenna elements to electronically steer the beam in desired directions. That is, precomputed weights are multiplied by the received signals at every element to perform coherent summation and produce a synthesized beam in the angular direction of interest. Therefore, the beam pattern computed using the Fourier method depends only on the antenna geometry. In contrast, the Capon method (also known as the MVDR, minimum variance distortionless response method), adaptively calculates beamforming weights from the received signals to reduce signal interference contamination coming from the antenna sidelobes [23]. This method started from the idea of frequency-wavenumber power spectral density estimation [24], and it has better performance than the Fourier method. This is achieved by lowering antenna sidelobe levels in directions of interference and improving angular resolution of the scanning beam [25, 26]. However, it is challenging to implement the Capon method in real time since its computational complexity increases exponentially with the number of antenna elements. This is because it requires the inversion of the spatial covariance matrix, with a size of $N \times N$, where N is number of receivers.

1.2 Motivation

Artificial Intelligent (AI) has been widely applied in various fields because of its flexibility and adaptability. It established a new approach to tackle high-complexity, nonlinear problems. One of the AI subfields, known as Machine Learning (ML), is an efficient data analysis tool [27]. For example, in the radar meteorology domain, ML has been used for storm-mode classification, classification of precipitation type, tornado prediction, and hail prediction [28]. Deep learning (DL), a subset of ML, is a technique that makes use of a neural network with multiple layers to create complex mappings that connect inputs to outputs and learns complex relationships between data [29]. DL has been applied to weather radar observations to improve the temporal resolution while maintaining data quality using DBF [30]. In this research, a radial-basis-function neural network (RBFNN) that performs real-time PAR DBF is proposed. RBFNN has been trained with the covariance matrix of received signals as input and produces beamforming weights to implement adaptive DBF in real time. However, RBFNN does not support the use of complex numbers; instead, they ingest real and imaginary parts separately. Therefore, it results in a neural network model with doubled length and, more importantly, loss of critical signal phase information. ML with its flexibility and adaptability has shown the potential to overcome the computational burden of the adaptive DBF. Therefore, this thesis is focused on evaluating the potential of DL to improve weather observations using a DBF-capable PARs in real time.

1.3 Contributions and Outline

A deep neural network (DNN) is designed to perform adaptive DBF with improved training efficiency by preserving phase information, to overcome the computational burden of adaptive DBF. It is designed as a feed-forward network to take the time-series in-phase and quadrature (IQ) signals as inputs and produce beamforming weights as outputs. The network is trained using IQ signals and Capon-beamforming weights for two applications, point and distributed targets (i.e., weather). In general, complex-valued data are separated into real and imaginary parts prior to the learning process. However, loss of phase information from IQ signals and beamforming weights is critical and often leads to learning failure or time inefficiency [30]. The current version of Keras, one of the largest ML libraries implemented in Python, does not support complex values during the learning process and limits many applications of ML [31]. Therefore, transforming complex values into amplitude and phase, leads to the use of real-valued data, which makes full use of Keras' flexibility and improves the learning process.

The novelty of the network developed in this thesis is that it preserves the phase information in IQ signals and beamforming weights. This is done by expressing the complex values in phasor form, i.e., using amplitude and phase. Amplitude information contains only positive values while phase information contains values from $-\pi$ to π. Therefore, signal amplitude and phase are trained independently in parallel for efficient learning. One of the most important features of the neural network for efficient learning is the activation function. For amplitude information, a rectified linear unit (ReLU), well known for supporting positive values, is used as activation function [32]. For phase information, a hyperbolic tangent function, that can take negative values and support non-linear changes of data is used as activation function [33]. Moreover, the proposed neural network uses a combination of convolution layers and dense layers to optimize the quality of data that is extracted during training.

The remainder of the thesis is organized as follows. In Chapter 2, an overview of ML methods is presented, including its component functions, and example uses in different research fields. Also, a mathematical basis to describe the neural network, activation functions, and normalization is discussed. Chapter 3 starts with fundamental radar concepts including, the conventional parabolic-reflector antenna radar, the development of PAR for weather observations, and a discussion of different PAR architectures. A review of PAR signals is introduced in this section on simulations of IQ signals for a point target and distributed targets. The Fourier method and the Capon method are discussed as non-adaptive and adaptive methods with their advantages and limitations. Then, Chapter 3 concludes by summarizing the challenges of adaptive DBF, mainly, the exponentially increasing computational power needed as the antenna size increases. Chapter 4 proposes a potential solution to address the issue in the previous chapter. The proposed DNN is trained to take IQ signals as input and produce Capon-like beamforming weights as outputs. Details of the neural network, such as its architecture, training strategy, and data processing for training are discussed. Chapter 5 shows preliminary results of the proposed method. The method is tested for three different data sets: a simulated moving point target, real data of a moving point target, simulated weather cases (e.g., tornado, hail, thunderstorm). The performance of the proposed method is analyzed with qualitative and quantitative metrics. For the qualitative analysis, the following are evaluated: quality of the generated beam pattern and the performance of the adaptive DBF patterns. For the quantitative analysis, the following are evaluated: the computation time, the performance as a function of SNR, the mean square error (MSE) of beamforming weights values, the normalized cross-correlation of generated beam patterns. Finally, Chapter 6 provides a summary of this work, and directions for future research.

Chapter 2

Review of Artificial Intelligence Concepts

AI consists of a set of algorithms and numerical simulation tools to model the human intelligence using machines [34]. AI algorithms can quickly learn to perform desired tasks with proper training and network architecture. AI has been widely used to perform a variety of tasks in our daily lives, some simpler and others more complicated. For example, a traffic light controller that is optimized with AI can operate adaptively [35]. To handle various types of tasks, AI has been developed in many different forms. For example, machine vision [36], speech recognition [37], expert systems [38], AI robotics [39], and ML are subsets of AI. ML is defined as a computer algorithm that can learn or adapt to solve associated tasks on its own without being explicitly programmed or designed to do so [40]. Because of its flexibility and adaptability, it has been extremely useful in many different fields [41]. Especially ML is actively used in biomedicine and healthcare system as clinical decision support systems, where medical imaging is one of the most suitable applications for using AI [42].

2.1 Machine Learning

ML has the ability to learn from a specific collection of data to solve associated tasks without being programmed [40]. A simple linear ML model behaves like a function, such as,

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x} + b, \tag{2.1}$$

where $\boldsymbol{x}$ represents an input vector, $\boldsymbol{w}$ is the corresponding set of weights to each input, and b is the bias term. ML has been actively used in meteorology for weather prediction and classification [28]. The ridge classifier is a well known linear ML model, which is built based on the Ridge regression method and allows the model to achieve better distinction between classes of outputs [43]. For example, this linear ML was used to detect and classify the migration of birds and insects from radar echoes with data collected by the S-band WSR-88D in [44]. It takes images of radar variables, such as the Z_{DR}, Φ_{DP}, and ρ_{hv} as inputs. Additionally, [44] evaluated the use of extra features, such as sectors, textures, and/or range interval, and how these would impact the quality of the output. It extracts the bird migration echoes by blob coloring, then computes their texture. The function of the proposed ML model is described as follows,

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = |w_{Z_{DR}}\ w_{\Phi_{DP}}\ w_{\rho_{hv}}| \begin{vmatrix} Z_{DR} \\ \Phi_{DP} \\ \rho_{hv} \end{vmatrix} + b, \tag{2.2}$$

where $w_{Z_{DR}}, w_{\Phi_{DP}}$, and $w_{\rho_{hv}}$ are the weights for Z_{DR}, Φ_{DP}, and ρ_{hv}. There are total 8 ridge classifiers with different combinations of inputs and their results are shown in Fig. 2.1. The birds and insects are colored blue and red. These results indicate that the ridge classifiers have $94.4\% - 95.4\%$ detection of classified gates.

ML algorithms can also improve efficiency by ingesting image data. For instance, saliency is another type of ML that models the human attention system and extracts the eye-attracting spots in the meteorological images. Weather Radar Spatiotemporal Saliency (WR-STS) has been developed to quantify the amount of spatial and temporal information

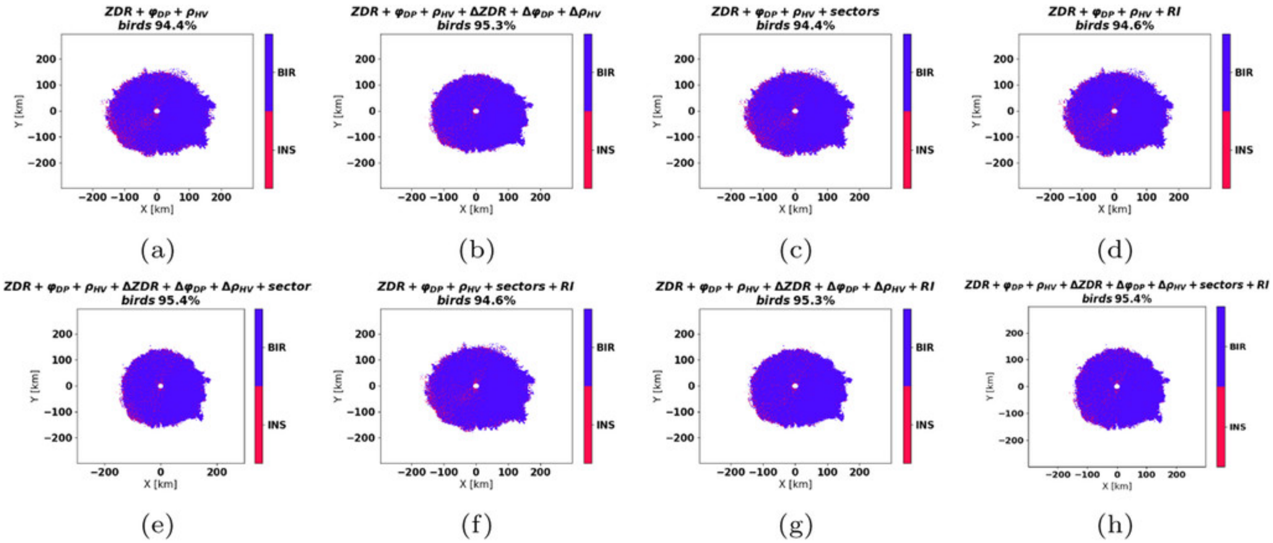


Figure 2.1: Ridge classification results for bird migration observed with KTLX radar at 0413 UTC 2 May 2015. BIR represents birds and INS represents insects [44]

in the weather radar reflectivity images in [45]. The proposed WR-STS is composed of 4 main steps: 1. Multi-scale decomposition and feature extraction 2. Spatial activation and fusion 3. Temporal activation 4. Spatio-temporal fusion. Illustrative results are shown in Fig. 2.2, where images show time series of reflectivity estimates going from 1932 UTC to 2256 UTC collected on 24 May 2011. The Fig. 2.2 shows a line of convective storms moving eastward, with active tornado warning polygons, severe thunderstorm polygons, and WR-STS contours that are indicated with white lines.

Despite its simplicity and efficiency, linear ML could lead to over-fitting of data. It predicts well for the examples used in the training, but frequently makes an error when it is applied to new data that have not been seen in the training phase. Also, there are limitations on the quality of learning for different levels of tasks. In contrast, DL, which is a ML-based artificial neural network, has become popular over linear ML by getting trained in more complex ways. More details of DL will be discussed in the following section.

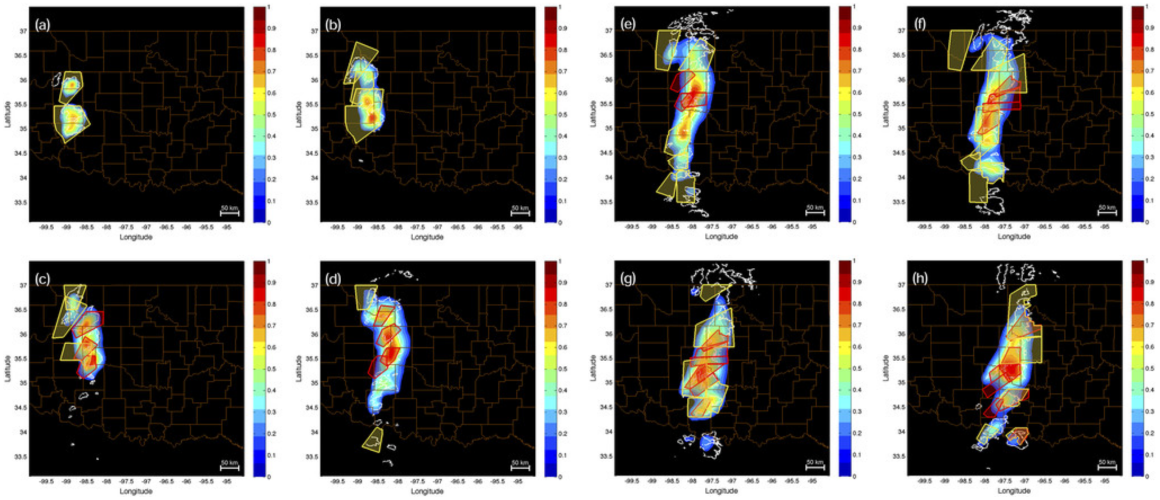


Figure 2.2: WR-STS maps for the reflectivity images of the tornadic storm on 24 May 2011 were obtained from the KTLX radar. Active tornado warning polygons (red), severe thunderstorm polygons (yellow), and 30-dBZ reflectivity contour lines (white) are overlaid on top of the WR-STS maps: (a)1932, (b)1957, (c)2027, (d)2057, (e)2127, (f)2157, (g)2226, and (h)2256 UTC.[45].

2.2 Deep Neural Networks

Deep neural networks, that consist of more than one hidden layer, create more complex networks with advanced operations such as convolution and multiple activation functions [46]. The relationship between ML and DL is shown in Fig. 2.3.

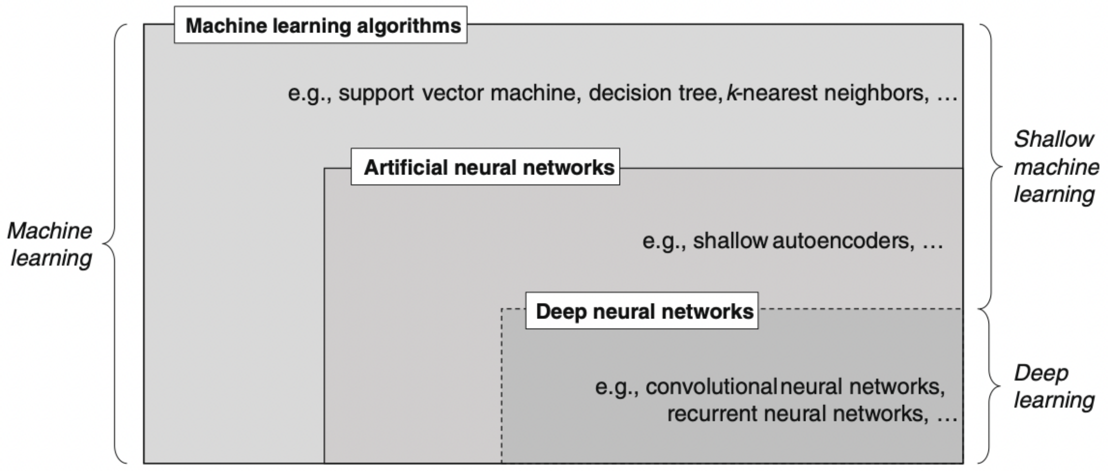

Machine learning algorithms
e.g., support vector machine, decision tree, k-nearest neighbors, ...
Artificial neural networks
e.g., shallow autoencoders, ...
Deep neural networks
e.g., convolutional neural networks,
recurrent neural networks, ...
Machine learning
Shallow machine learning
Deep learning

Figure 2.3: Process of analytical model building [40].

A neural network with i-layers has a nested function that is mathematically described as follows [34, 47],

$$y = F_{NN}(x) = \boldsymbol{f_i}(\boldsymbol{f_{i-1}}(...(\boldsymbol{f_2}(\boldsymbol{f_1})))), \tag{2.3}$$

where the inner layers, $\boldsymbol{f_1}, \boldsymbol{f_2}, ..., \boldsymbol{f_{i-2}}, \boldsymbol{f_{i-1}}$, are mathematically described as follows,

$$\boldsymbol{f_i}(\boldsymbol{z}) = \boldsymbol{g}_i(\boldsymbol{W}_i\boldsymbol{z} + \boldsymbol{b}_i), \tag{2.4}$$

where $\boldsymbol{g}_i$ is an activation function, $\boldsymbol{W}_i$ is the set of learned weights of the layer in matrix form, and $\boldsymbol{b}_l$ is the error between true and tested value. Neural networks with more than two non-output layers are called DNN as shown in Fig. 2.4. A Convolutional Neural Network (CNN) is an efficient neural network developed for data with spatial information (e.g., pictures). That is because it learns the input data by applying a convolution filter and extracting important features [48]. While CNN has been producing outstanding results for pattern recognition-related tasks, it might not be an efficient way to learn data that does not have any spatial information. For example, time-series signals, binary data, or audio data often do not have spatial information. For these datasets, dense (also called fully connected) layers have been used. A dense layer creates a full mapping connection among the data between layers [34].

DNN has been used for adaptive ultrasound beamforming in the biomedical domain to minimize the computational burden of data-driven reconstruction methods [50]. Their proposed DNN is built with four dense layers, three activation functions, and three dropout layers to optimize its learning process for non-spatial data without excessive over-fitting. Then the network is trained to take the real and imaginary parts of time-aligned array signals separately as inputs and produces the optimal set of beamforming weights to perform adaptive beamforming. The beam patterns produced by the proposed neural network (Adaptive

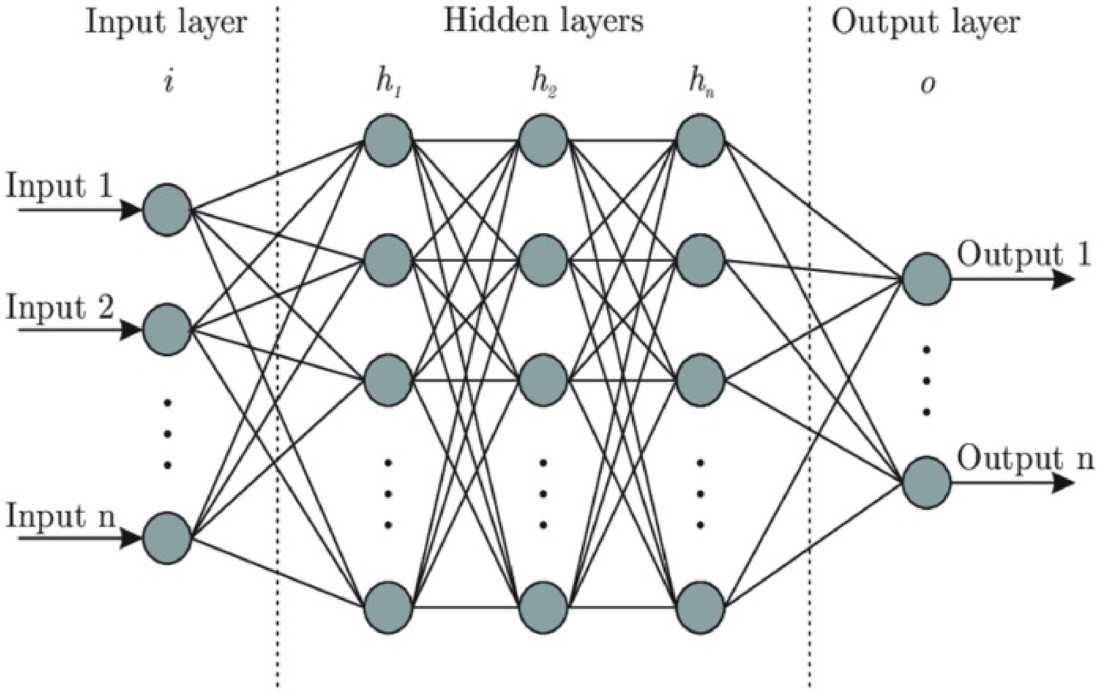


Figure 2.4: General deep neural network concept from [49].

Beamforming by deep-LEarning, ABLE) are compared to other conventional methods for beamforming in the biomedical domain, as shown in Fig. 2.5. Delay-And-Sum (DAS) and iterative maximum a posteriori (iMAP2) are the non-adaptive beamforming method. EBMV (i.e., the MVDR or Capon method) is the adaptive beamforming method used as ground truth for the training. Then, the reconstructed images obtained by each method are shown in Fig. 2.6

As shown in Eq. (2.4), a neural network is not built with only convolution or dense layers. To build an efficient neural network, activation functions ($\boldsymbol{g}$) and normalization are essential. These are introduced in the following subsection.

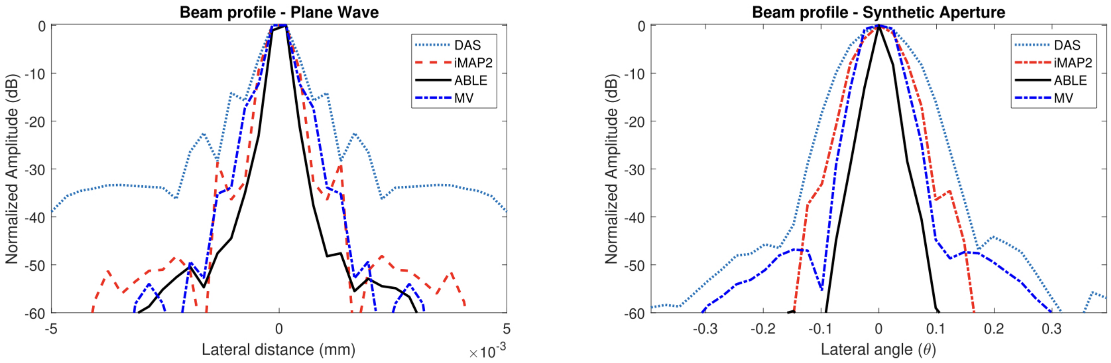


Figure 2.5: Normalized beamprofiles along a simulated pointscatterer for (left) Plane-wave imaging and (right) Synthetic Aperture (IVUS) imaging [50].

2.2.1 Activation functions

Activation functions ($g(x)$) transform the activation level of input neurons to output neurons [32, 51]. Activation functions play an important role in the learning process as they sort out complicated data and help the neural network connect neurons between layers to create meaningful mapping [49]. Careful selection of activation functions is required for different tasks so the neural network can learn properly and have good prediction accuracy. Also, the neural network without activation function is just a linear regression that will not be able to predict properly with new data and will likely fail to learn complex connections among the data. The activation function can be either linear or non-linear. One of the most well-known activation functions is the ReLU, which is a linear activation and often used for image-related tasks. ReLU can be mathematically described as follows [33],

$$g(x) = max(x, 0), \tag{2.5}$$

where x is the input data. Linear activations can be efficient when the model is learning using the back-propagation of errors. Because its behavior is similar to linear regression and it avoids easy saturation and provides sensitivity for non-zero data. However, the linear

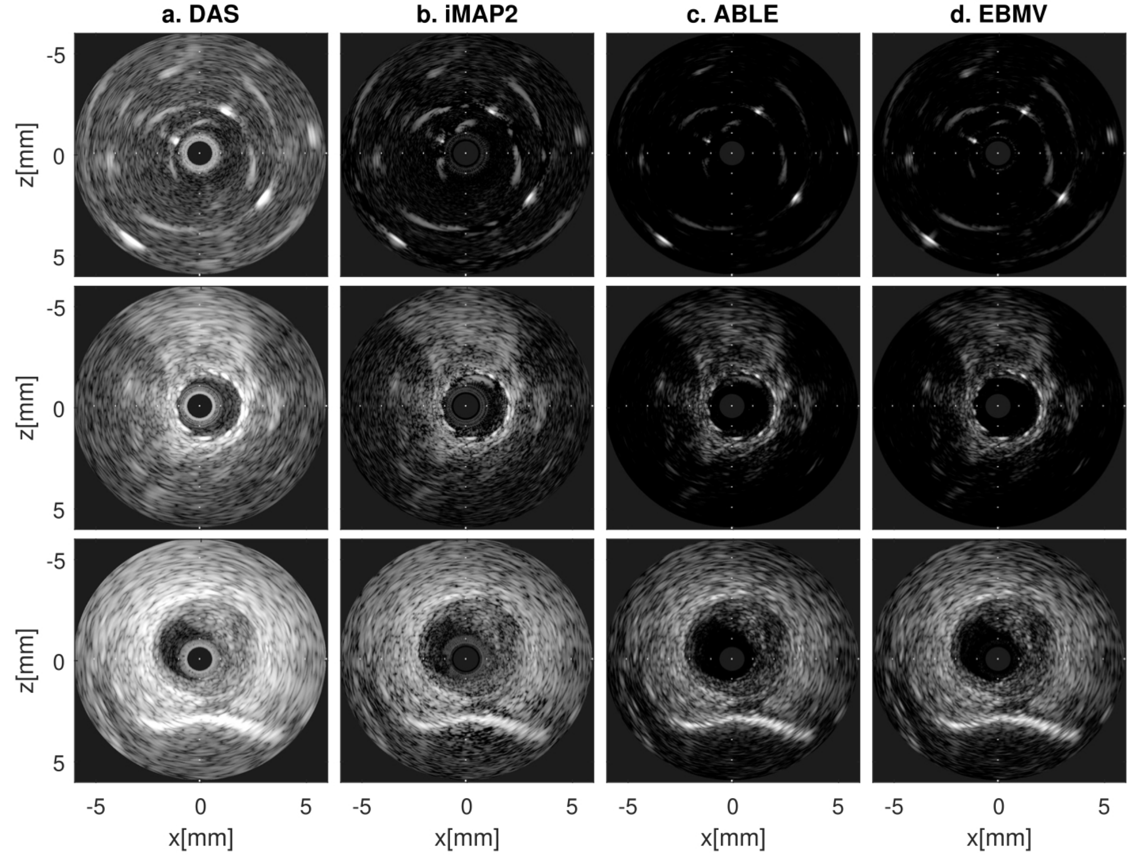


Figure 2.6: Reconstructed synthetic aperture *in-vivo* images using: a) DAS beamforming with Hanning apodization, b) iMAP2 beamforming c) ABLE, and d) Eigen-Based Minimum variance (EBMV) beamforming. Images are logarithmically compressed with a dynamic range of 60 dB. From top to bottom we have a wire phantom, an arterial stent, and a coronary artery with plaque [50].

activation function has a linear boundary and will fail with non-linear changes in data. As described in Eq. (2.5), the linear activation function will vanish nodes if the data contain negative values. This limitation of linear activation is often critical because many problems in the real world have non-linear behavior and non-positive values. In contrast, the non-linear activation function allows the neural network to extract complex information from data by letting the non-positive and non-linear changes pass through the function. One activation function may work better than an other one depending on the task of the neural network. Fig. 2.7 shows the behavior of six different activation functions. Sigmoid

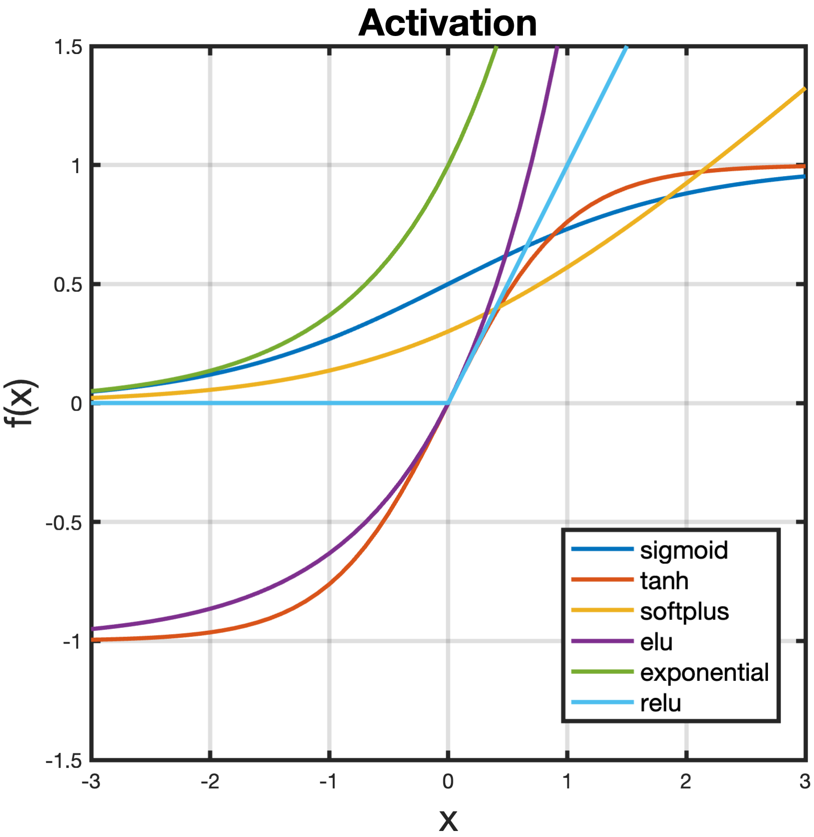


Figure 2.7: Behavior of several activation functions. Linear function, ReLU is in light blue. Non-linear functions are Sigmoid (dark blue), Tanh (red), Softplus (Yellow), Elu (purple), and exponential (green). The X-axis represents the input to the function and the Y-axis represents the output from the function.

[52], softplus [53], and exponential [54] are the non-linear activation functions that only return positive values. The sigmoid activation function can be mathematically described as follows,

$$g(x) = \frac{1}{1 + e^{-x}} \ . \tag{2.6}$$

It returns a value close to zero when input x is smaller than -5 and a value close to one when input x is bigger than 5. Compared to the ReLU activation function, the sigmoid function does not vanish negative values as long as they are greater than -5. Therefore, it could be useful for data that contain negative values. However, it gets easily saturated and will lose many pieces of information if the data have a wide range of values. The softplus activation is another non-linear activation function that only produces positive values and it is mathematically described as follows,

$$g(x) = log(e^x + 1). \tag{2.7}$$

While the softplus function has a lower tolerance for negative values than the sigmoid function, it does not get saturated easily with large positive values. The exponential activation function is described as follows,

$$g(x) = e^x. \tag{2.8}$$

Even though it is a non-linear activation function, it acts similar to ReLU. It gets saturated with low values (< -5), but does not get saturated with positive x values. Exponential Linear Unit (ELU) and tanh are the non-linear activation functions that return both positive and negative values. The ELU function is described as follows,

$$g(x) = \begin{cases} x, & \textit{if}\, x > 0, \\ \alpha(e^x - 1), & \textit{if}\, x < 0. \end{cases} \tag{2.9}$$

The tanh function is described as follows:

$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}\ . \tag{2.10}$$

To understand the behavior of each activation function, their behaviors are tested on an example image, shown in Fig. 2.8. The Fig. 2.9 shows how each activation function sorts out the data. Note that image data contain positive values only. ReLU, softplus, and ELU do not easily get saturated with positive values and preserve the valuable information from the test image, while sigmoid and tanh get saturated as shown in Fig. 2.9. Therefore, ReLU, softplus, and ELU would work better for the neural network with image related tasks, while sigmoid and tanh would work better for non-spatial data such as time-series data.

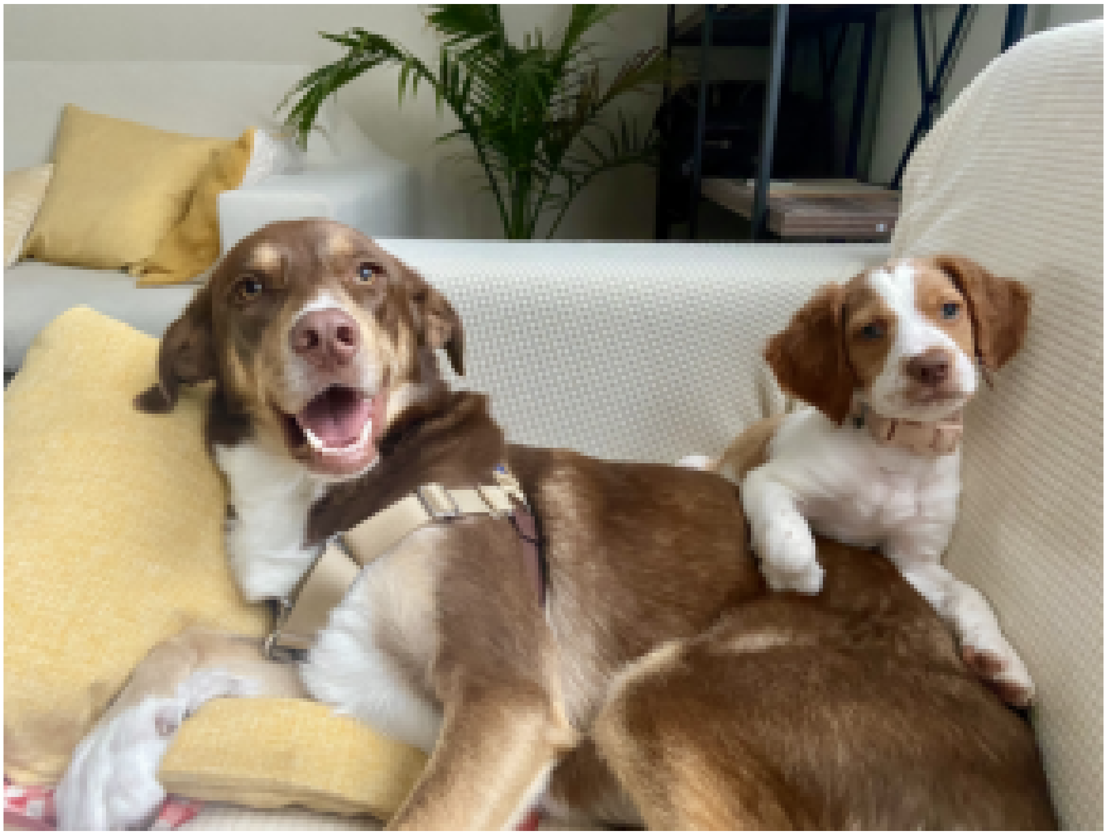


Figure 2.8: Test image for evaluating behavior of different activation functions.

2.2.2 Data Normalization

As the data used for neural networks are becoming more complicated, the training time increases. This is when normalization plays an important role that impacts the quality of training with reduced time [55]. Normalization is one of the necessary components to build an efficient neural network, especially with Stochastic Gradient Descent (SGD), because the normalization can improve the efficiency by standardizing inputs with the mean and standard deviation of the training data [56]. There are various ways to normalize the data

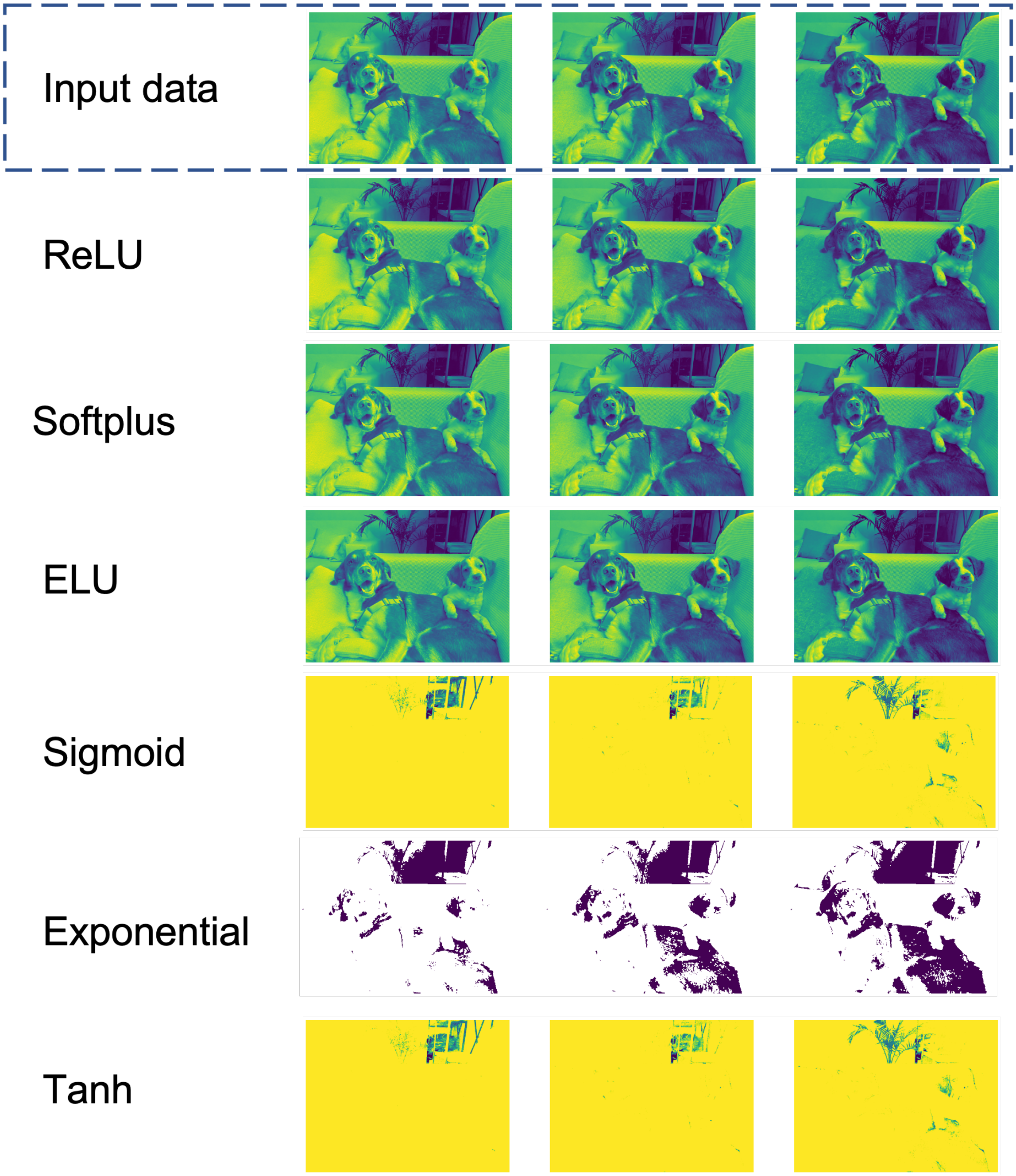


Figure 2.9: Output images from activation functions. The test image was separated into three different filters, RGB, shown in the first row. The outputs from each activation functions, ReLU, Softplus, ELU, Sigmoid, Exponential, and Tanh are shown in each row.

depending on the learning process and data structures. The general normalization process can be simply written as a transformation,

$$\hat{x} = \text{Norm}(x, X), \tag{2.11}$$

where x is the given training data and X is the set of training data.

Keras, Python's library for DL built on a TensorFlow platform [57], has its built-in normalization that can be used as a layer inside the neural network. Keras has many built-in functions that allows users to build their own network in an easier and faster way [31, 58]. Having a normalization layer inside the neural network keeps the consistency of the data throughout the learning process and reduces the training time. Batch normalization and layer normalization are the two normalization techniques implemented in the Keras library. Batch normalization processes a group of input data x, defined as a mini-batch $\boldsymbol{B} = x_1, x_2, ..., x_m$ [59]. Also, it has two different function modes for training and inference. During the training, it normalizes each dimension of a d-dimensional input with the mean (μ_{Bt}) and variance (σ_{Bt}) of the current batch of inputs as follows,

$$\mu_{Bt} = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{2.12}$$

$$\sigma_{Bt}^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{Bt})^2, \tag{2.13}$$

$$\hat{x}_i = \frac{x_i - \mu_{Bt}}{\sqrt{\sigma_{Bt}^2 + \epsilon}}, \tag{2.14}$$

where ϵ is a small constant parameter usually assigned as 0.001. However, using the mean and variance of the batch is not desirable when the network is in the inference mode. Unbiased (i.e., moving average) mean (μ_{Bi}) and variance (σ_{Bi}^2) for the inference mode are calculated based on the mean and variance estimated from the training. Therefore, the output depends only on the input deterministically.

$$\mu_{Bi} = \frac{1}{j} \sum^{j} \mu_{Bt}, \tag{2.15}$$

$$\sigma^2_{Bi} = \frac{m}{m-1} \left(\frac{1}{j} \sum^{j} \sigma^2_{Bt} \right), \tag{2.16}$$

where j represents the number of mini-batch used during the training [56]. However, batch normalization has a limitation when used for a Recurrent Neural Network (RNN). Averaging inputs of each batch becomes complicated when RNN does not have the same shape for all sequences of data. Also, batch normalization loses the advantage of grouping the inputs as a batch when the values of data are largely distributed. To overcome the weakness of batch normalization, layer normalization is introduced. Layer normalization is used to normalize the input data within each layer of the network, instead of normalizing over summed multiple input data [60]. Layer normalization calculates the mean and variance of i^{th} hidden units at the l^{th} layer as follows [61],

$$\mu_L = \frac{1}{h} \sum_{i=1}^{h} a_{li}, \tag{2.17}$$

$$\sigma^2_L = \frac{1}{h} \sum_{i=1}^{h} (a_{Li} - \mu_L)^2, \tag{2.18}$$

where h denotes the number of hidden units in a layer and a_{Li} is a vector of the summed inputs to the neurons in L^{th} layer. In summary, while batch normalization shares the same mean and standard deviation among the batch of multiple input data, layer normalization shares only among the neurons in the layer of single input dataset. The performance of batch and layer normalization tested on MNIST is shown in Fig. 2.10.

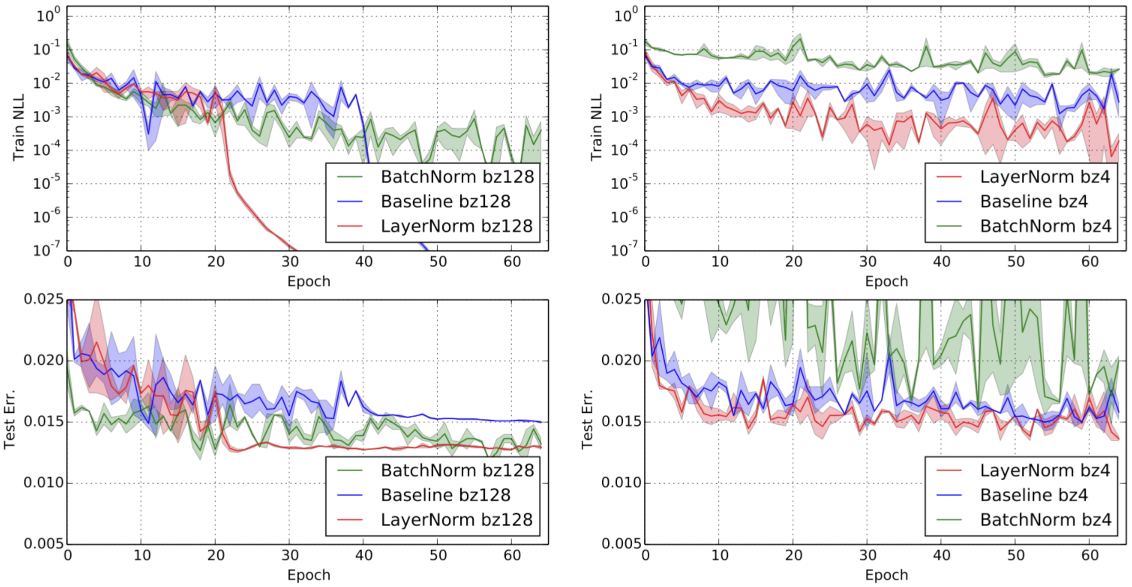


Figure 2.10: Evaluation of batch and layer normalization on MNIST model with negative log likelihood and test error. (Left) Batch size is 128. (Right) Batch size is 4.

Chapter 3

Simulation of Radar Signals

3.1 Radar Fundamentals

Radar technology has been actively used for weather observations for several decades. Weather radar monitors the occurrence and movement of precipitation systems, which can provide information about precipitation timing, probability, location, type, and intensity. It is one of the most important tools to detect hazardous weather conditions in real-time over a large coverage area [62]. Weather radars often use pulsed waveforms to estimate the location and Doppler velocity of hydrometeors. Pulsed Doppler radars consist of a waveform generator, transmitter, antenna, and receiver [3]. An example of a conventional pulsed-Doppler radar is the S-band WSR-88D, which remotely observes the atmosphere and collects high-quality data such as reflectivity, mean radial velocity, and spectrum width [4]. It uses a parabolic-reflector antenna to transmit pulses of electromagnetic energy forming a narrow pencil beam pattern, and receive energy back-scattered from hydrometeors or other scatterers in the atmosphere. The incident power density (S_i) of the transmitted pulse can be expressed as follows [63],

$$S_i(\theta,\phi) = \frac{P_t}{4\pi r^2} G, \tag{3.1}$$

$$G = g_t\, f^2(\theta,\phi), \tag{3.2}$$

where P_t is the peak transmit power, r is the range, g_t is antenna gain which defines how much the energy can be focused in the beam, and $f^2(\theta,\phi)$ is the normalized antenna radiation pattern [1]. As the transmitted wave impinges on hydrometeors, part of the energy is back-scattered to the radar. The back-scattering radar cross-section (RCS, σ_b) represents the scatterers' effective area that intercepts the propagating wave and reflects energy back. Then, the power density (S_r) received back at the antenna can be determined as follows [63],

$$S_r(\theta,\phi) = \frac{S_i(\theta,\phi)\sigma_b}{4\pi r^2} = \frac{P_t\sigma_b}{(4\pi r^2)^2}. \tag{3.3}$$

The power received by the antenna is [63],

$$P_r = S_r A_e(\theta,\phi) = \frac{P_t g_r \lambda \sigma_b f^4(\theta,\phi)}{(4\pi)^3 r^4} l^2, \tag{3.4}$$

where $A_e = \frac{g_r\lambda^2}{4\pi} f^2(\theta,\phi)$ denotes effective antenna area, g_r is received gain, l^2 is the two-way losses, and λ is the radar wavelength. Note that the received gain is equal to transmit gain ($g_t = g_r$) for pencil beam operation.

The echo voltage $V(t)$ from a point target at the receiver can be viewd as the time-delayed transmitted signal and is described as follows,

$$V(t,r) = A\left\{\exp[j2\pi f(t-2r/c) + j\psi]\right\} U(t-2r/c), \tag{3.5}$$

[1] Note that all the equations assume the use of variables in the international system of units.

where A is the complex amplitude, c is the speed of light, and $2r/c$ is the range time. The echo phase is defined as follows [63],

$$\psi_e = -\frac{4\pi r}{\lambda} + \psi_t + \psi_s, \tag{3.6}$$

where ψ_s is the phase shift upon scattering, and ψ_t is the transmitter phase. The received signal is split into I and Q channels. The I channel mixes the received signal with a local oscillator at the radar frequency and produces signal as follows [63]:

$$I(t) = (|A|/\sqrt{2})U(t - 2r/c)\cos(4\pi r/\lambda - \psi_t - \psi_s). \tag{3.7}$$

Similarly, the Q channel mixes the received signal with a local oscillator at the radar frequency but with a $90 - \deg$ phase shift and results in the following form [63],

$$Q(t) = (|A|/\sqrt{2})U(t - 2r/c)\sin(4\pi r/\lambda - \psi_t - \psi_s). \tag{3.8}$$

The weather signal received contains contributions from all the hydrometeors in the resolution volume. The locations of each hydrometeor are random, therefore, the amplitude and phase of the weather signal sample are considered random variables. Assuming the discrete range time delay received from echoes at r to be $\tau_s = r/(2c)$, the complex weather signal sample $V(\tau_s)$ is defined as [63],

$$V(\tau_s) = \frac{1}{\sqrt{2}} \sum_i A_i W_i e^{-j4\pi r_i/\lambda} = I + jQ, \tag{3.9}$$

where A_i and r_i are the amplitude and range of the i^{th} scatterer, and W_i is a range-dependent weight. Note that the effects from the antenna radiation pattern, $f^2(\theta, \phi)$, are considered to be part of A_i. The power at range time τ_s, $P(\tau_s)$, is derived as follows [63],

$$P(\tau_s) = VV^* = \frac{1}{2}\sum_{i,k}^{N_s} A_i A_k^* W_i W_k^* e^{j4\pi(r_k - r_i)/\lambda}. \tag{3.10}$$

Weather signals at the IQ channels can be defined as follows,

$$I(\tau_s, T_s) = \sum_i |A_i W_i| \cos\gamma_i = |V(\tau_s, T_s)| \cos(\theta_s, T_s), \tag{3.11}$$

$$Q(\tau_s, T_s) = -\sum_i |A_i W_i| \sin\gamma_i = |V(\tau_s, T_s)| \sin(\theta_s, T_s). \tag{3.12}$$

Assuming the elemental volume $dV = r^2 dr \sin\theta \; d\theta \; d\phi$, where θ and ϕ represent elevation and azimuth angle, in the polar coordinate system is filled with hydrometeors, then the expected power return from an elemental volume is [63],

$$E[dP] = \frac{P_t g^2 \lambda^2 f^4(\theta - \theta_0, \phi - \phi_0)}{(4\pi)^3 r^4 l^2} |W_s(r_0, r)|^2 dV \frac{E[\sigma_b]}{dV}, \tag{3.13}$$

where θ_0 and ϕ_0 indicate the beam direction in elevation and azimuth, and $\frac{E[\sigma_b]}{dV}$ is the expected backscatter cross-section per volume and can be defined as reflectivity, $\eta(\boldsymbol{r})$ with an equation,

$$\eta(\boldsymbol{r}) = \int_0^\infty \sigma_b(D) N(D, \boldsymbol{r}) dD. \tag{3.14}$$

Then, Eq. (3.13) can be rewritten as:

$$E[dP] = I(r_0, \boldsymbol{r}) dV \int_0^\infty \sigma_b(D) N(D, \boldsymbol{r}) dD, \tag{3.15}$$

with

$$I(r_0, r) = \frac{C f^4(\theta - \theta_0, \phi - \phi_0) |W_s(r_0, r)|^2}{l^4 r^4}, \tag{3.16}$$

$$C = \frac{P_t g^2 \lambda^2}{(4\pi)^3}, \tag{3.17}$$

where θ_0 and ϕ_0 are the beam direction, and r_0 is the range to the center of the resolution volume being considered. By integrating over space, Eq. (3.15) is converted to the following equation:

$$\overline{P}(\boldsymbol{r_0}) = \int_0^{r_2} \int_0^{\pi} \int_0^{2\pi} \eta(\boldsymbol{r}) \boldsymbol{I}(\boldsymbol{r_0}, \boldsymbol{r}) dV. \tag{3.18}$$

Assume that the volume is filled uniformlly, loss and reflectivity do not change inside the resolution volume, and the antenna pattern is circularly symmetric and has a Gaussian shape in Eq. (3.18). Then, the expected weather signal power at the receiver is,

$$\overline{P}(\boldsymbol{r_0}) = \frac{P_t g^2 \eta c \tau \pi \theta_1^2 \lambda^2}{(4\pi)^3 r_0^2 l^2 16 \ln 2}. \tag{3.19}$$

The angular resolution of the antenna can be measured by determining the half-power beamwidth beamwidth (i.e., -3 dB width), which can be approximated as [64],

$$\theta_1 = 2 \sin^{-1} \left(\frac{1.4\lambda}{\pi D} \right) \approx 0.89 \frac{\lambda}{D} \tag{3.20}$$

where D is the antenna diameter. Another important key feature that determines the quality of radar estimates is the Signal-to-Noise Ratio (SNR). In real measurements, there are two types of noise, one from external sources and the other one from the radar receiver itself. The SNR can be simply described as follows,

$$SNR = \frac{P_{r_0}}{N} \tag{3.21}$$

where P_{r_0} is the signal power at the receiver output, and N is the white noise power.

The information that can be obtained from single polarization weather radars is reflectivity factor (Z_{h}), mean Doppler velocity (v_{r}), and spectrum width (σ_{v}). Signal power from the dual-polarization of H and V channels can be estimated in terms of the received echo voltages V_h and V_v, as follows [63],

$$\hat{S}_h = \frac{1}{M}\sum_{k=0}^{M-1} |V_h(k)|^2 - N, \tag{3.22}$$

and

$$\hat{S}_v = \frac{1}{M}\sum_{k=0}^{M-1} |V_v(k)|^2 - N, \tag{3.23}$$

where M is the number of signal samples. Second, the mean Doppler velocity can be estimated as follows [63],

$$\hat{v} = -(\lambda/4\pi T_s)\arg\{\hat{R}(T_s)\}, \tag{3.24}$$

where $\hat{R}(T_s)$ is the autocorrelation estimated at lag T_s, and which is pulse repetition time. The autocorrelation estimated from the horizontal polarization is described as follows,

$$\hat{R}_h(T_s) = \frac{1}{M}\sum_{m=0}^{M-1} V_h^*(m)V_h(m+1). \tag{3.25}$$

Last, spectrum width can be calculated with the estimation of $|\hat{R}_{1h}| = |\hat{R}_h(T_s)|$, as follows [63],

$$\hat{\sigma}_{vh} = \frac{\lambda}{2\pi T_s\sqrt{2}}\left|\ln\left(\frac{\hat{S}_h}{|\hat{R}_{1h}|}\right)\right|^{1/2}. \tag{3.26}$$

Single-polarization weather radars are not able to obtain information about shape or aspect ratio of hydrometeors, or the phase of the hydrometeor (e.g., water/ice), due to transmitting only horizontally polarized waves [3]. Dual-polarization radars offer additional information by transmitting and receiving horizontally and vertically polarized electric fields . Therefore, weather radars have been upgraded to dual-polarization, enabling algorithms such hydrometeor classification, better rainfall rate estimation, hail detection, to cite a few [65].

Polarimetric variables derived include Z_{DR}, Φ_{DP}, and ρ_{hv} [66]. The Z_{DR} can be estimated as follows [63],

$$\hat{Z}_{DR} = 10 \log \frac{\hat{S}_h}{\hat{S}_v}, \tag{3.27}$$

where $\hat{S}_h$ and $\hat{S}_v$ are the mean signal powers estimated from H and V polarizations. The Φ_{DP} can be estimated as follows [67],

$$\hat{\Phi}_{DP} = \arg\{\hat{R}_{hv}(0)\} \tag{3.28}$$

where, $\hat{R}_{hv}(0)$ is the cross-correlation estimated at lag 0,

$$\hat{R}_{hv}(0) = \frac{1}{M} \sum_{i=0}^{M-1} {V_{hi}}^{*} V_{vi} \tag{3.29}$$

where V_h and V_v are the IQ signals received at horizontal and vertical polarizations. The correlation coefficient and spectrum width can be estimated as follows [63],

$$\hat{\rho}_{hv}(0) = \frac{\left|\hat{R}_{hv}(0)\right|}{(\hat{S}_h \hat{S}_v)^{1/2}}\ . \tag{3.30}$$

Conventional radars with a parabolic antenna can offer dual-polarization capabilities, increase receiver sensitivity, and provide real-time data that allow forecasters to identify severe weather events [4]. However, mechanical scanning with a rotating parabolic antenna limits the temporal resolution and the flexibility of the system. This is a critical limitation, especially for severe weather observation that requires fast scan update rates. Also, they cannot support adaptive scanning, which is one of the scanning strategies that can adaptively change the beam patterns to sample specific locations that have the potential for the severe weather [9].

3.2 Phased Array Radar

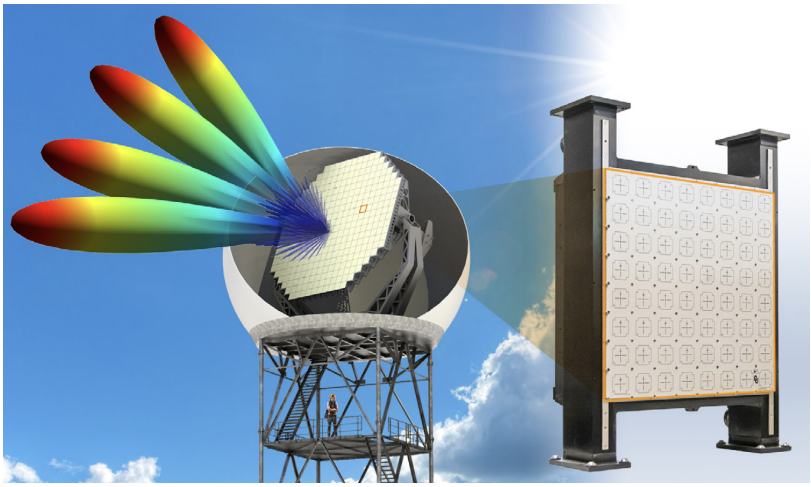


Figure 3.1: An example phased array radar with a potential beam pattern and a closeup look of its radiating elements [8].

The phased array antenna is composed of several antenna elements, through which electronic steering, beamforming, and other capabilities are possible [8]. It has the potential to improve temporal and spatial resolution through its unique capabilities, without impacting data quality. Instead of mechanically scanning the beam, the PAR steers the beam to the desired direction electronically by changing the relative phase of signals radiated by its elements, and creating wave fronts progressing in the desired direction [9]. Each element of the PAR, normally separated by $d = \lambda/2$ to mitigate the grating lobes, transmits and receives independent electromagnetic waves. Waves from each element are coherently added together to form a collimated beam in the desired direction. The main lobe can be electronically steered by varying the difference in phases between antenna elements. To steer the beam in a given direction, the phase difference between adjacent elements should be [8],

$$\varphi = kd \sin \theta, \tag{3.31}$$

where $k = 2\pi/\lambda$ is the wavenumber and θ is the steering angle. An example of this is shown in Fig. 3.2.

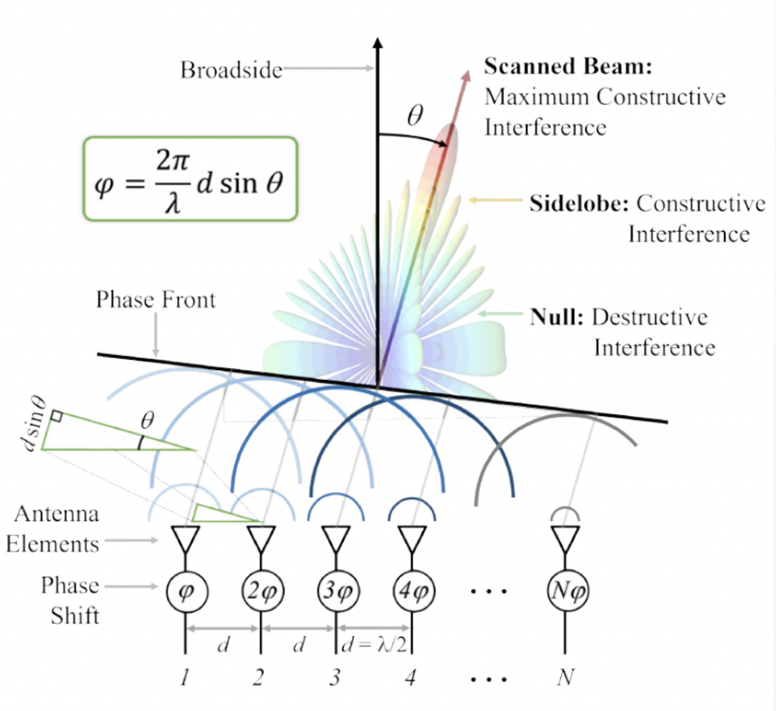


Figure 3.2: Fundamental concept of PAR with uniform linear array [8] is shown. φ is the phase shift needed to steer at θ for each elements that are spacing with d.

The National Weather Radar Testbed (NWRT) was a PAR built using an AN/SPY-1A antenna provided by the US Navy, for weather research installed in Norman, OK, and shown in Fig. 3.3. Compared to the WSR-88D radar, the NWRT was able to improve the temporal resolution, collecting volumetric scans 5 times faster (in ≤ 1 min). However, the NWRT was a single-polarization radar. More recently, the National Severe Storms Laboratory installed a dual-polarization phased array radar, the Advanced Technology Demonstrator (ATD), in the NWRT facility to evaluate the performance of polarimetric PAR for weather observations [68, 69]. ATD is a dual-polarization PAR with a scalable antenna panel architecture, deployed for testing weather and aircraft surveillance. It is shown in shown in Fig. 3.5 [68], along with other radar systems in the surrounding area. The Multi-function Phased Array Radar (MPAR) concept involves using the same radar system for multiple applications simultaneously. MPAR can reduce the system cost by combining several radars into one, and by reducing the need for maintenance due to lower wear and tear caused by mechanical rotation.

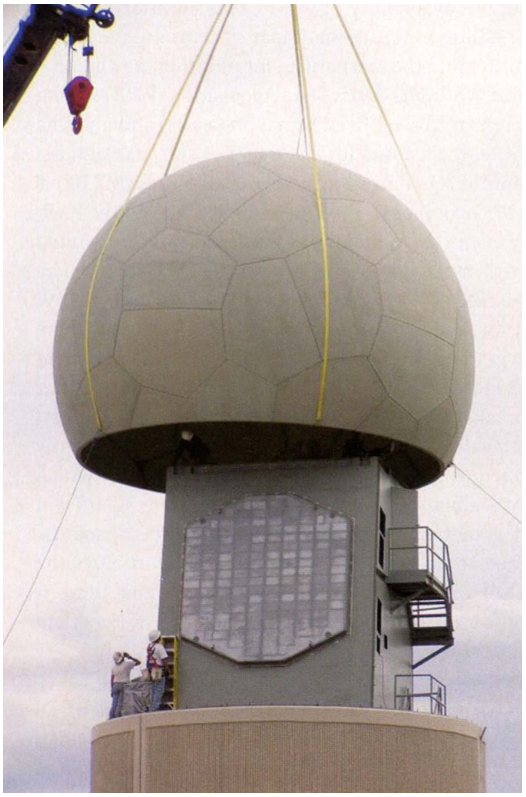


Figure 3.3: NWRT: Installation of the radome over the single aperture of the AN/SPY-IA radar antenna [9].

The Turbulent Eddy Profiler (TEP) was a vertically-pointing profiler with 2-D DBF capability to observe the intensity and motion of turbulence with high temporal and spatial resolution [70]. A picture of TEP is shown in Fig. 3.4. TEP transmits wide vertical beams to observe the motion of turbulence at altitudes from 200 m to 1.5 km. It provides three-dimensional estimates of the echo power, radial velocity, and spectrum width obtained by

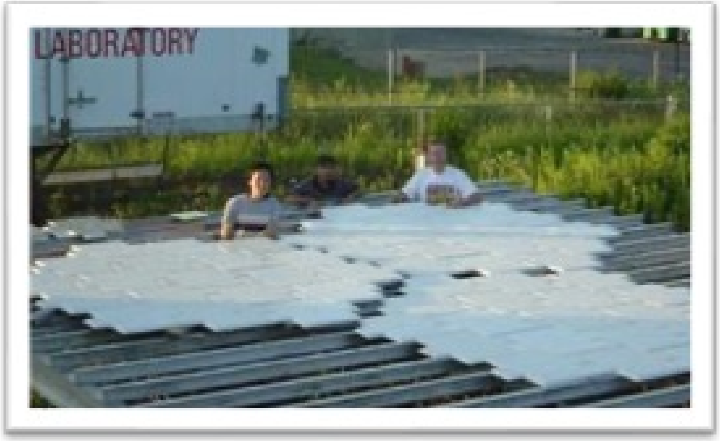


Figure 3.4: TEP array under construction in the field. Clutter fences, the transmit horn, and the trailer are visible in the background [70].

performing DBF.

The C-band mobile Parametric Atmospheric Imagining Radar (PAIR) offers high-temporal resolution, scanning flexibility, and fast update time with DBF in elevation [72, 73]. It is being developed by the ARRC at OU with funding from the National Science Foundation (NSF). Also, it has a novel polarimetric phased array antenna design that allows to achieve required dual-polarization data quality, due to its ultra-low cross-polarization levels (on the order of -50 dB) [74]. A picture of PAIR is shown in Fig. 3.6.

The Horus radar is an all-digital, S-band, polarimetric phased array radar for multi-mission surveillance, including weather and aircraft surveillance [71]. The antenna is composed of 1024 (32×32) dual-polarization antenna elements that offers great flexibility in scanning strategies, including traditional pencil beams and DBF with spoiled beams [75, 76]. A picture of Horus is shown in Fig. 3.7.

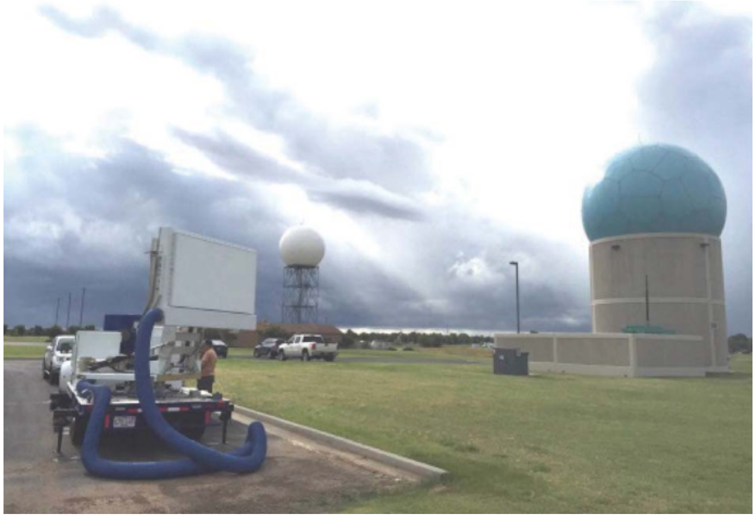


Figure 3.5: (left foreground) The MPAR 10-panel proof of concept demonstrator, based on the Lincoln Laboratory panel provided the first S-band dual-polarization weather data. (blue radome) The National Weather Radar Testbed is the background (right), and the (white radome) NSSL WSR-88D is in the background (left) [68].

3.3 Simulation of Digital Phased Array Radar

In this section, a simulation framework for point target and weather-like signals is presented. Point target signals are simulated based on the TEP radar specification [70], which is a vertical transmitting radar and performs DBF in two dimensions, north-south (NS) and east-west (EW). Therefore simulation of a point target is conducted for one range gate in two angular dimensions. Weather-like IQ voltage signals are simulated from existing Next-generation Weather Radar (NEXRAD) data following the methods described in [77] and [78]. Since NEXRAD is a reflector-antenna based radar with a single receiver, additional received signals from a PAR are simulated to evaluate DBF methods. In this thesis, the Horus radar specification is used to investigate the DBF in azimuth direction.

Figure 3.6: A picture of C-band mobile Polarimetric Atmospheric Imagining Radar (PAIR) is shown.

3.3.1 Simulation of Point Target Signals

The time-series IQ voltage signals for a point target are simulated based on TEP specifications [79], shown in Table 3.1. The geometry of the TEP receiving antenna elements is shown in Fig. 3.8. The simulation of IQ voltage signals is implemented using the following equation,

Table 3.1: Point target simulation parameters based on TEP specification

Parameter	Value
Frequency (f)	915 MHz
Number of subarrays (N)	56
Pulse repetition time (prt)	1/140 s
Spacing between adjacent antenna elements (d)	16.4 mm
Number of sample (M)	64
Number of scan (rec)	75
Height	2000 (m)
NS scan	-15° to 15°
EW scan	-15° to 15°

$$V_n(t) = A_n e^{jk(r_t(t)+r_{r_n}(t))} \tag{3.32}$$

where $r_t(t)$ is the range from transmitter to the target at time t, and $r_{r_n}(t)$ is the range from target to the n^{th} antenna element at time t. The simulated point target moves in the shape

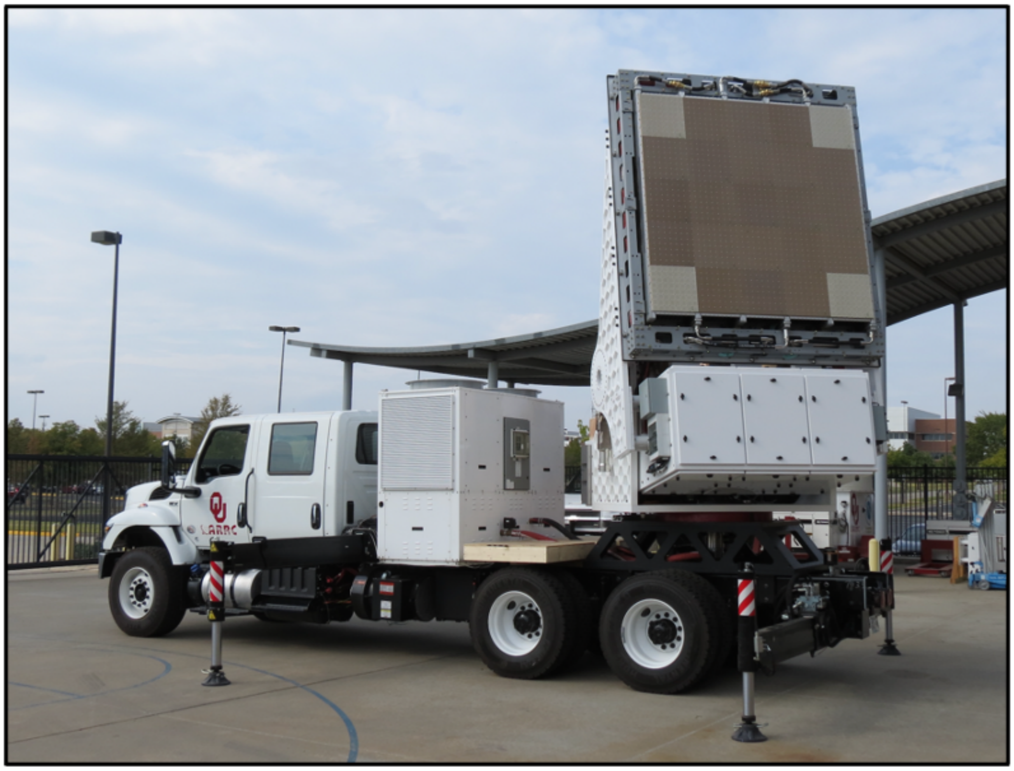


Figure 3.7: The mobile Horus is an S-Band, dual-polarization, fully digital radar [71].

of a heart; its location over time is shown in Fig. 3.9.

3.3.2 Simulation of Weather-like Signals

Three different weather cases are simulated based on Horus specification as shown in Table 3.2, each with different types of storms. These include a tornadic supercell storm, hail storm, and thunderstorm. First the NEXRAD data are used to produce realistic time-series IQ data. Radar variables such as ρ_{hv}, Φ_{DP}, σ_{v}, Z_{h}, and v_{r} for each case are shown in Fig. 3.10, Fig. 3.11, and Fig. 3.12.

In particular, NEXRAD data have two types of flags, one is for overlaid echoes labeled with -888 in Doppler variables and the other one is for censored data labeled with -999 in Doppler variables. Therefore, the collected radar variables need robust filling and

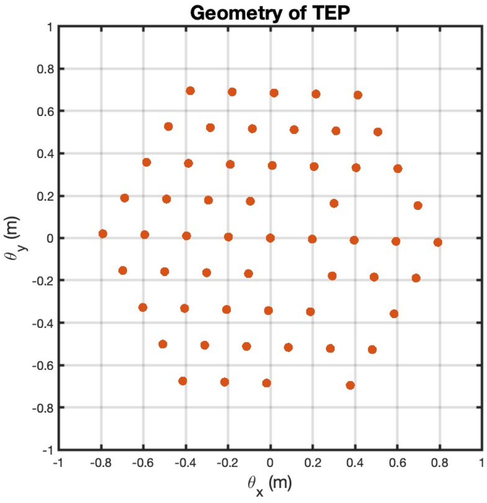


Figure 3.8: Geometry of TEP with 56 receiving channels is shown.

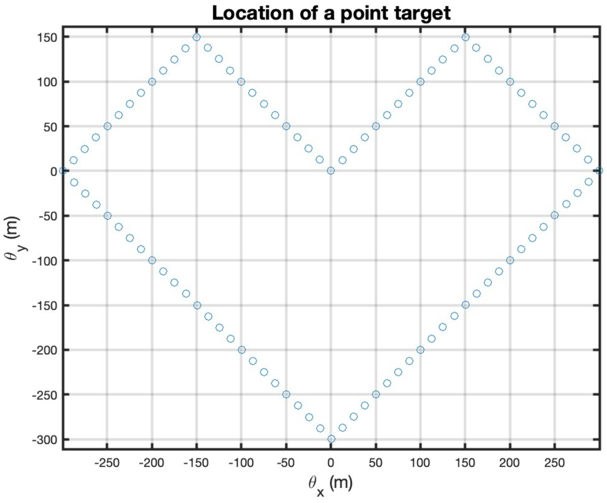
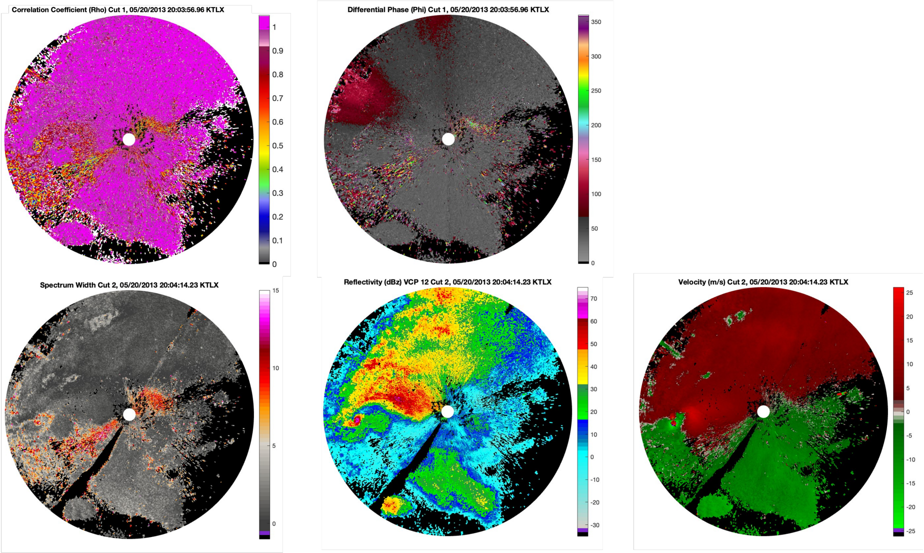


Figure 3.9: Location of a simulated point target. The point target is moving in a heart shape over 75 scans.



Correlation Coefficient (Rho) Cut 1, 05/20/2013 20:03:56.96 KTLX

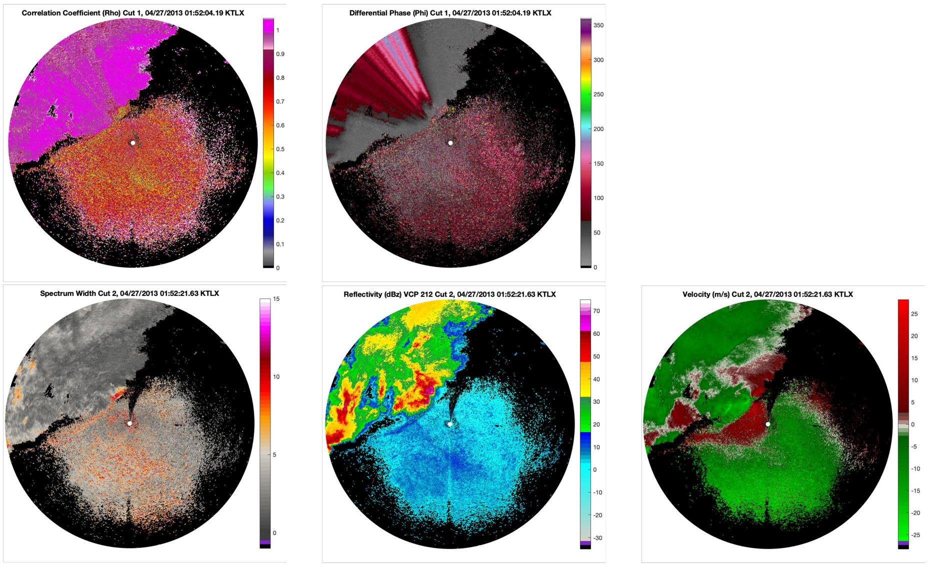
Differential Phase (Phi) Cut 1, 05/20/2013 20:03:56.96 KTLX

Spectrum Width Cut 2, 05/20/2013 20:04:14.23 KTLX

Reflectivity (dBz) VCP 12 Cut 2, 05/20/2013 20:04:14.23 KTLX

Velocity (m/s) Cut 2, 05/20/2013 20:04:14.23 KTLX

Figure 3.10: Tornadic supercell storm case is shown. Radar variables were collected by NEXRAD, 05/20/2013 20:04:14.23 KTLX. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).



Correlation Coefficient (Rho) Cut 1, 04/27/2013 01:52:04.19 KTLX

Differential Phase (Phi) Cut 1, 04/27/2013 01:52:04.19 KTLX

Spectrum Width Cut 2, 04/27/2013 01:52:21.63 KTLX

Reflectivity (dBz) VCP 212 Cut 2, 04/27/2013 01:52:21.63 KTLX

Velocity (m/s) Cut 2, 04/27/2013 01:52:21.63 KTLX

Figure 3.11: Hail storm case is shown. Radar variables were collected by NEXRAD, 04/27/2013 01:52:04.19 KTLX. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).

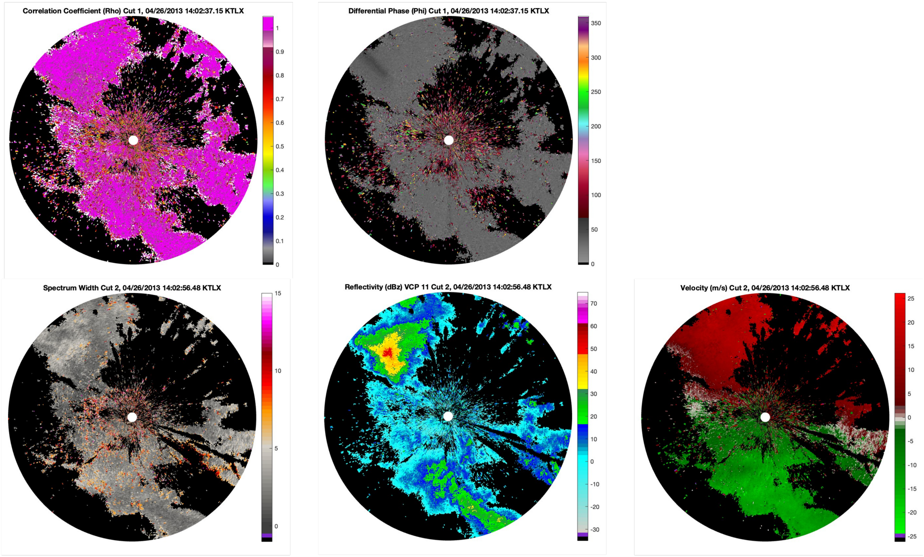


Figure 3.12: Thunderstorm case is shown. Radar variables were collected by NEXRAD, 04/26/2013 14:02:37.15 KTLX. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).

de-aliasing before they can be used to simulate IQ signals. The overlaid echoes and the censored data are filled up with interpolation and Gaussian noise, respectively. Processed data are shown in Fig. 3.13, Fig. 3.14, and Fig. 3.15.

The three important spectral moments, Z_{h}, v_{r}, and σ_{v}, are used to simulate weather-like IQ signals following the method discussed in [77]. First, power (S) can be estimated from Z_{h} as follows [63],

$$\hat{Z}_{dBZ} = 10\log_{10}\hat{S}_h + 20\log_{10}R + R\cdot\mu + C_0, \tag{3.33}$$

where R is the range of the radar in km, μ is atmospheric attenuation in dB/km, and C_0 is the radar calibration constant. Second, Gaussian-shaped Doppler spectrum S_k is generated with power S, σ_{v}, and v_{r} as follows,

$$S_k = \frac{S}{\sqrt{2\pi}\sigma_v}\exp\left[-\frac{(v_k - v)^2}{2\sigma_v^2}\right]. \tag{3.34}$$

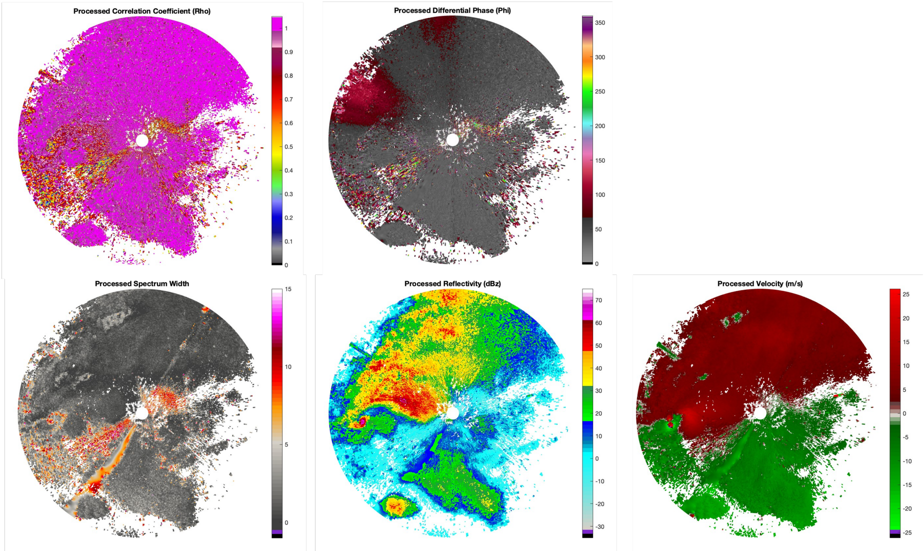


Figure 3.13: This is the output from robust filling and de-aliasing for the tornadic supercell storm case. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).

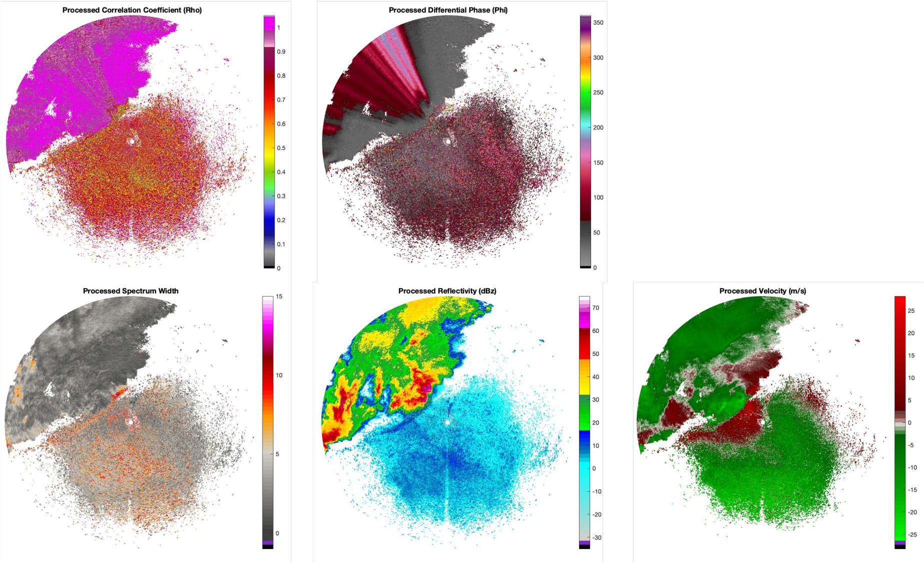


Figure 3.14: This is the output from robust filling and de-aliasing for the hail storm case. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).

Third, the time-series IQ voltage signals ($V(i)$) for distributed targets are computed

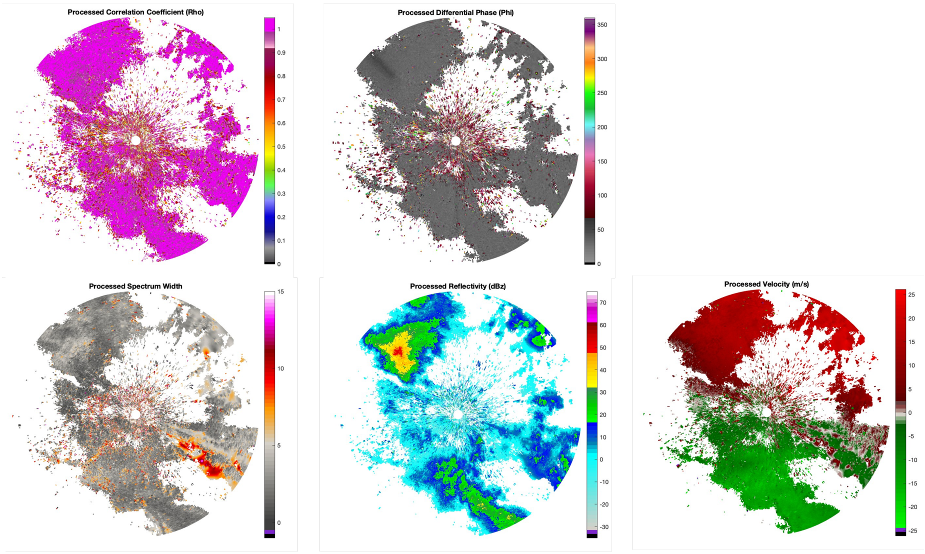


Figure 3.15: This is the output from robust filling and de-aliasing for the thunderstorm case. Top left: correlation coefficient (ρ_{hv}), top right: differential phase (Φ_{DP}), bottom left: spectrum width (σ_{v}), bottom middle: reflectivity factor (Z_{h}), bottom right: velocity (v_{r}).

through inverse Fourier transform of the simulated power spectrum.

$$V(i) = I(i) + jQ(i) = \frac{1}{n}\sum_{k=1}^{n} P_k^{1/2} \exp\left[j\theta_k\right] \exp\left[-j\frac{2\pi}{n}ki\right]. \tag{3.35}$$

Finally, a stationary point target was added at the location of 99.7476° (azimuth) and 21.125 km (range), where there are no significant weather echoes, to evaluate the ability of adaptive nulling using the Capon method.

The simulated IQ signals from NEXRAD variables are for one receiver, yet, additional received signals are required to evaluate DBF. Therefore, the simulation is extended to include digital PARs. The synthesized transmit pattern for Horus radar is used for for simulation to generate realistic data and is in Fig. 3.16. This pattern is 60° wide.

A schematic representing the digital PAR simulation is shown in Fig. 3.17 with an

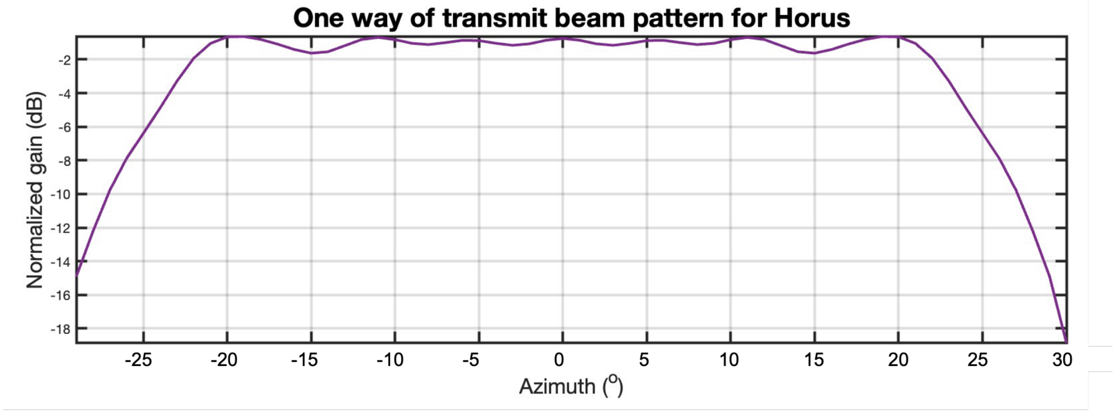


Figure 3.16: One way of transmit pattern for Horus is shown. The *x*-axis shows the angle (azimuth) and the *y*-axis shows the power in dB.

equation as follows,

$$V_n(t) = \sum_{i=1}^{nt} f_\theta^2 V_i \exp\left[-jk(r_t(t) + r_{r_n}(t))\right] + n_t(t) \tag{3.36}$$

where V_i is the simulated IQ signals from NEXRAD variables, nt is the number of simulated IQ signals used to form a spoiled beam, and $N_t(t)$ is Gaussian-distributed white noise with mean value extracted from the NEXRAD KTLX, equal to 1.0357×10^{-8} for the tornadic storm case, 1.0096×10^{-8} for the hail storm case, and 1.0008×10^{-8} for the thunderstorm case. This process is repeated for every receiving antenna element (32 in this case). The simulated spoiled beams are 60° wide ($nt = 120$) and overlap $20\deg$ to mitigate discontinuities at each edge. A total of 36 spoiled beams were simulated to cover 360° (azimuth). The SNR from one antenna element is shown in Fig. 3.18. SNR is calculated with Eq. (3.21).

3.3.3 Digital Beamforming

Digital beamforming is a unique capability of PAR, which can be used to improve temporal resolution. The improved temporal resolution of PAR has the potential to meet the

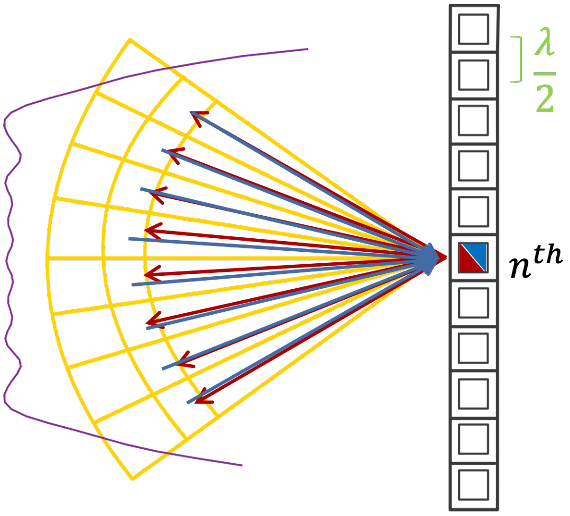
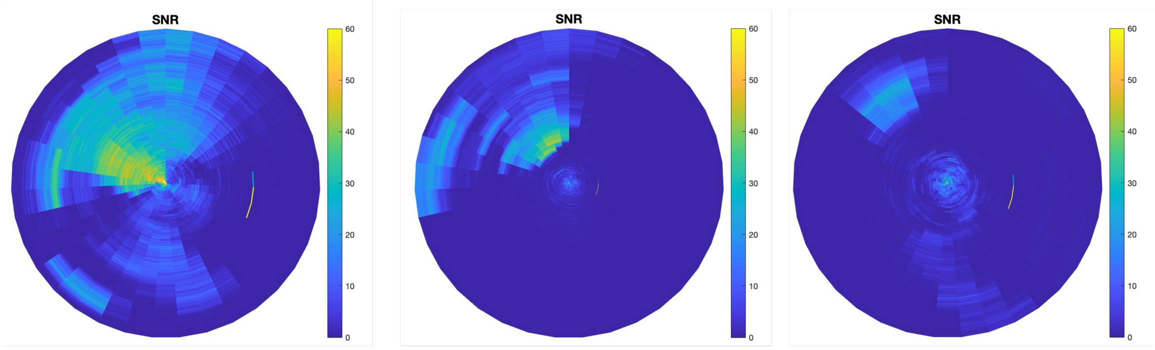


Figure 3.17: Illustration of PAR simulation based on Horus specifications with a spacing of $\lambda/2$ between antenna elements. Yellow indicates the simulated IQ signals from NEXRAD, red represents the transmitting range from n^{th} antenna element to the target, blue represents the receiving range from the target to n^{th} antenna element, and purple indicates Gaussian two-way electric field pattern.



a. Tornado case

b. Hail case

c. Thunderstorm case

Figure 3.18: SNR of simulated spoiled beams for (left) tornadic supercell storm case covering 360° in azimuth and 37.375 km in range, (middle) hail storm case covering 360° in azimuth and 99.875 km in range, and (right) Thunderstorm case covering 360° in azimuth and 62.375 km in range.

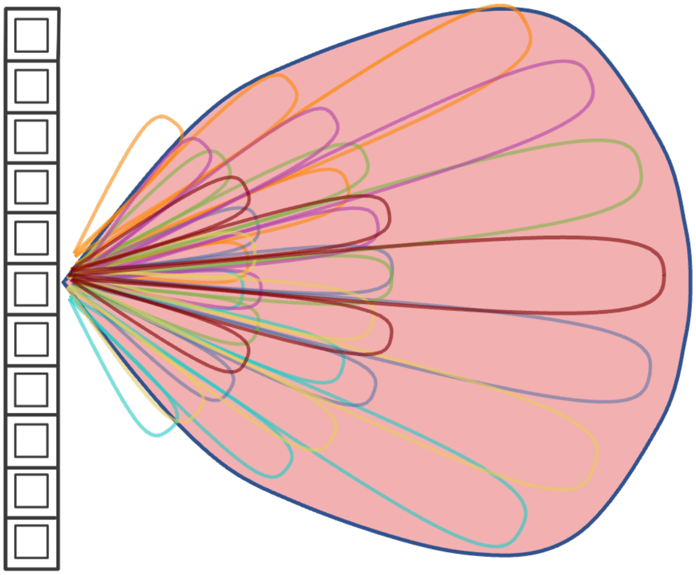


Figure 3.19: Illustration of DBF. The transmitting antenna element transmits a spoiled beam (pink) and multiple narrow receiving beams at each receiving antenna element (multiple colors) are formed simultaneously.

requirement needed for high-impact weather observations [21]. DBF is accomplished by transmitting the spoiled beams and forming narrow receive beams within the transmit beam simultaneously to sample a wide sector of the atmosphere, as illustrated in Fig. 3.19. DBF can be done in either a non-adaptive (i.e., deterministic) or adaptive way. The Fourier method is a non-adaptive DBF method that pre-computes the beamforming weights based

Table 3.2: Weather Simulation Parameters: 1D Horus

Parameter	Tornado	Hail	Thunderstorm
Date	05/20/2013	04/27/2013	04/26/2013
Time	20:04:14.23	01:52:04.19	14:02:37.15
Range Gate coverage (rg)	37.375 km	99.875 km	62.375 km
Azimuth coverage (az)		360°	
Frequency (f)		3 GHz	
Range resolution		250 m	
Number of antenna element (N)		32	
Spacing (d)		16.4 mm	
Number of sample (M)		64	

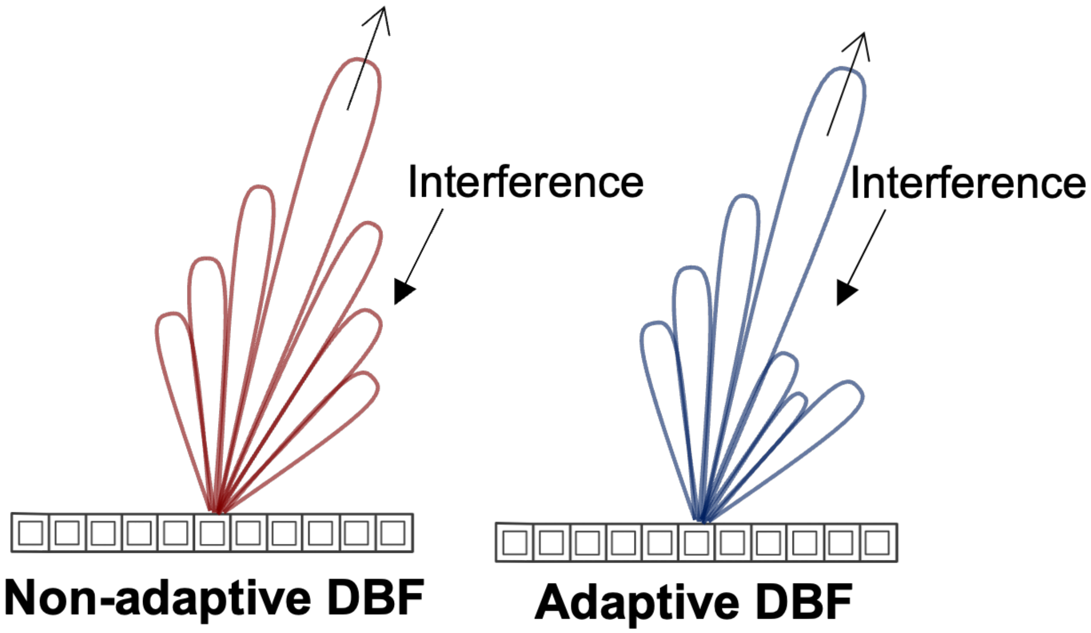


Figure 3.20: Illustration of non-adaptive and adaptive DBF methods. The non-adaptive method has fixed sidelobes while the adaptive method lowers the sidelobe when there's an interference source.

on the geometry of the antenna elements. Therefore it results in fixed sidelobe levels and fixed angular resolution, which could lead to high sidelobe levels in undesirable directions that have high return power, often caused by a ground clutter or interference. In contrast, the Capon method (also known as the MVDR) is an adaptive DBF method that computes the beamforming weights using received signals. Therefore the Capon method is able to suppress the high power return from an undesired direction by placing a null adaptively as illustrated in Fig. 3.20.

Fig. 3.21 shows simulated digitally formed beams based on the TEP specifications. The beams generated using the Fourier method are shown on the left two plots, and the beams generated using the Capon method are shown on the right two plots. The location of an interfering source is marked in black and the steering direction is marked in blue. The beam generated by the Capon method puts a null at the location of the interfering source.

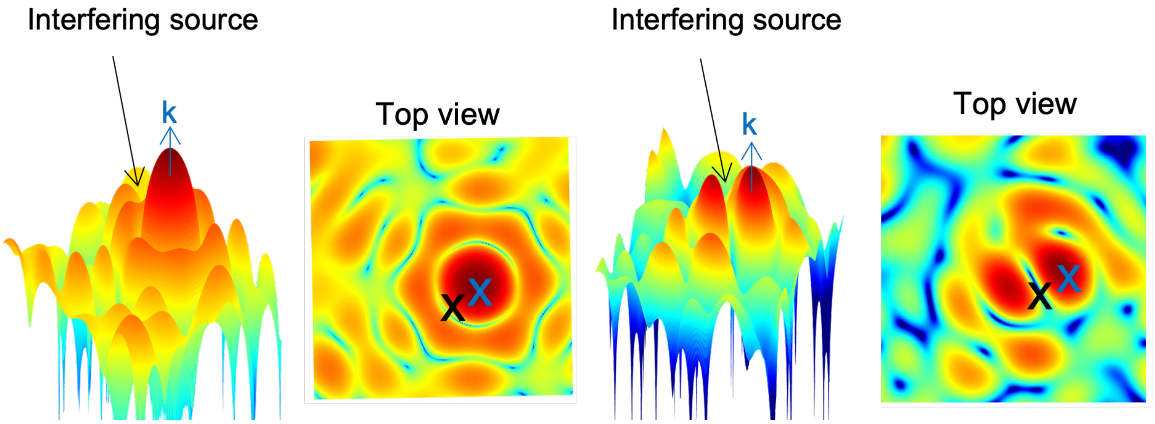


Figure 3.21: The generated beams by the (left two plots) Fourier and the (right two plots) Capon methods based on TEP specification. The location of the interfering source is marked in black and the steering direction is marked in blue.

Non-adaptive digital beamforming

The Fourier method computes beamforming weights by calculating the needed phase shift between antenna elements to electronically steer the beam in desired directions, as follows [16],

$$\boldsymbol{\omega}_f = [e^{j\boldsymbol{k}\cdot\boldsymbol{D_1}}\ e^{j\boldsymbol{k}\cdot\boldsymbol{D_2}}\ \ldots\ e^{j\boldsymbol{k}\cdot\boldsymbol{D_N}}]^{\boldsymbol{T}}, \tag{3.37}$$

where $\boldsymbol{D}$ represents the position vector of the N^{th} antenna element, and $\boldsymbol{k}$ is the wavenumber vector determined as $\boldsymbol{k} = 2\pi/\lambda\ [\sin\theta\sin\phi\ \ \sin\theta\cos\phi\ \ \cos\theta]$. Note that the Fourier beamforming weights are the conventional steering vector. Then, the power of the generated beam can be obtained as follows,

$$\hat{P}_f(\boldsymbol{k}, t) = \boldsymbol{\omega}_f^H \boldsymbol{R} \boldsymbol{\omega}_f, \tag{3.38}$$

where H indicates the Hermitian operator, and $\boldsymbol{R}$ is the autocovariance matrix of the received signals from N elements at time t. Since the Fourier method is independent of the received data and the beamforming weights are pre-computed based on the antenna geometry and steering angle, it does not optimize sidelobe levels based on contaminating target locations [26].

Adaptive digital beamforming

The Capon method is a data-dependent technique that adaptively calculates the beamforming weights from the returned signals to improve the quality of data. It attempts to lower the sidelobe levels in directions with high return power and minimize the power in all directions except the desired direction $\boldsymbol{k}$ [25, 26]. This minimization problem is described mathematically as follows,

$$\min_{\boldsymbol{w}} P(\boldsymbol{k}, f) \text{ subject to } \boldsymbol{e}^H \boldsymbol{w} = 1. \tag{3.39}$$

To solve this problem, standard Lagrange methods [80] was applied to the general form of power given Eq. (3.38), as follows,

$$L(\boldsymbol{w}, \gamma) = \boldsymbol{w}^H \boldsymbol{V} \boldsymbol{w} + \gamma(\boldsymbol{e}^H \boldsymbol{w} - 1), \tag{3.40}$$

where γ is the Lagrange multiplier shown below [25],

$$\gamma = \frac{-2}{\boldsymbol{e}^H \boldsymbol{V}^{-\mathbf{1}} \boldsymbol{e}}. \tag{3.41}$$

Finally, the Capon beamforming weights are computed as follows [16],

$$\boldsymbol{\omega}_C = \frac{\boldsymbol{R}^{-1} \boldsymbol{e}}{\boldsymbol{e}^H \boldsymbol{R}^{-1} \boldsymbol{e}}, \tag{3.42}$$

where $\boldsymbol{e} = \boldsymbol{\omega}_f$ is the steering vector, $\boldsymbol{R}^{-1}$ is the inverse of the autocovariance matrix with a size of $N \times N$. The power of the formed beam can be computed as follows,

$$\hat{P}_c(\boldsymbol{k}, t) = \boldsymbol{\omega}_c^H \boldsymbol{R} \boldsymbol{\omega}_c. \tag{3.43}$$

The computation time of the autocovariance matrix inversion will exponentially grow with N which limits the real-time implementation of the Capon method for large digital PARs. Details for calculating algorithm complexity will be discussed later in this chapter.

DBF Results

The results of the non-adaptive and adaptive DBF methods for the simulated point target, an actual point target observed by TEP, and the three different weather cases are presented next. First, DBF has been done in two dimensions (NS and EW) for the simulated point target and actual point target cases, and 121 narrow receiving beams generated. For the simulated point target case, the DBF electronic scan covers -15° to 15° in NS (θ_y) and EW (θ_x) with 3° sampling. The power of the generated beam by each method is calculated with Eq. (3.38) and Eq. (3.43) and is shown in Fig. 3.22 and Fig. 3.23 with every 5^{th} scan out of total 75 scans plotted. The second point target case is real data of a moving bird collected by TEP with 56 sub-arrays on June 14, 2003. The location of the bird is shown in Fig. 3.24. The DBF electronic scan covers -10° to 10° in NS (θ_y) and EW (θ_x) with 2° sampling. The power of the Fourier and the Capon methods is shown in Fig. 3.25 and Fig. 3.26 with the first 15 scans out of total 21 scans plotted.

DBF has been done in one dimension for 360° azimuth and oversampled at 0.5° azimuthal sampling for the weather cases based on the Horus specifications. The normalized power and reflectivity fields of the Fourier and Capon methods for each weather case are shown in Fig. 3.27, Fig. 3.30, and Fig. 3.33. The stationary point target, which was arti-

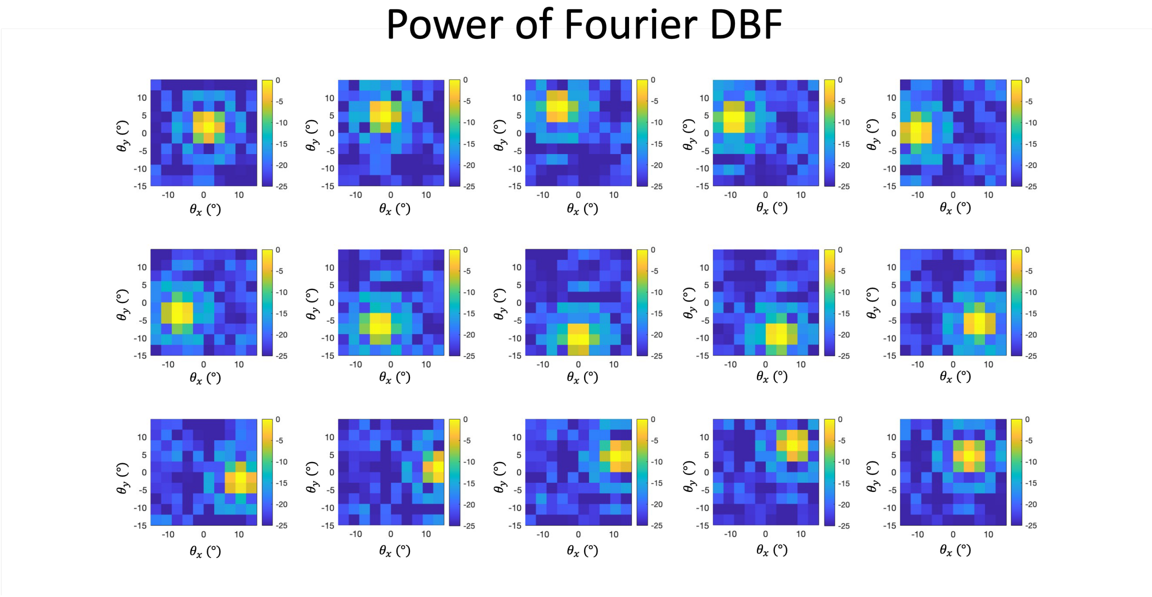


Figure 3.22: Normalized power of the Fourier method with simulated point target data. The point target is moving in a heart-shaped path over 75 scans. Every 5^{th} scan is plotted. X-axis is EW (θ_x) and y-axis is NS (θ_y).

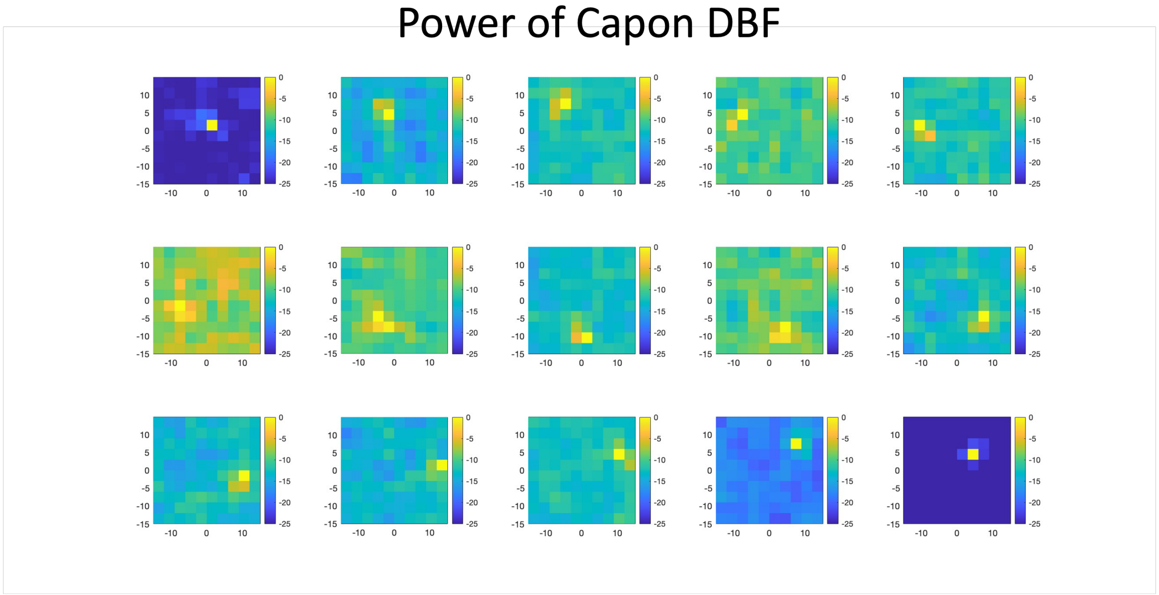


Figure 3.23: Normalized power of the Capon method with simulated point target data. The point target is moving in a heart-shaped path over 75 scans. Every 5^{th} scan is plotted. X-axis is EW (θ_x) and y-axis is NS (θ_y).

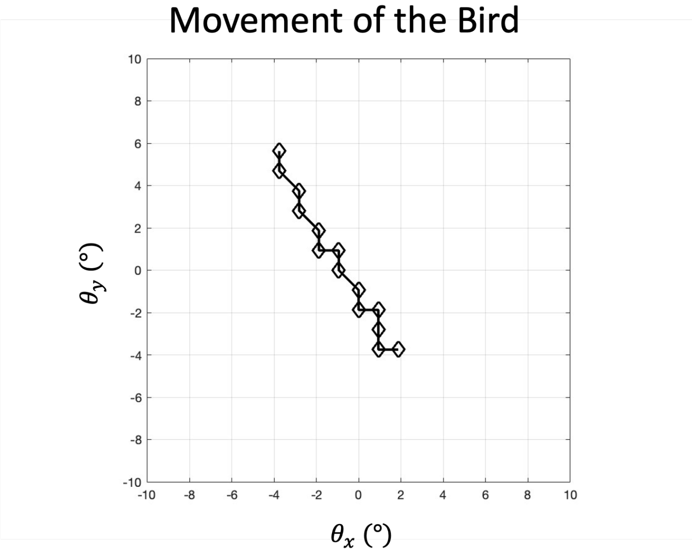


Figure 3.24: The movement of a bird that was observed by TEP with 21 scans. It starts at (-4° , 6°) and ends at (2° , -4°). The X-axis is EW (θ_x) and the y-axis is NS (θ_y).

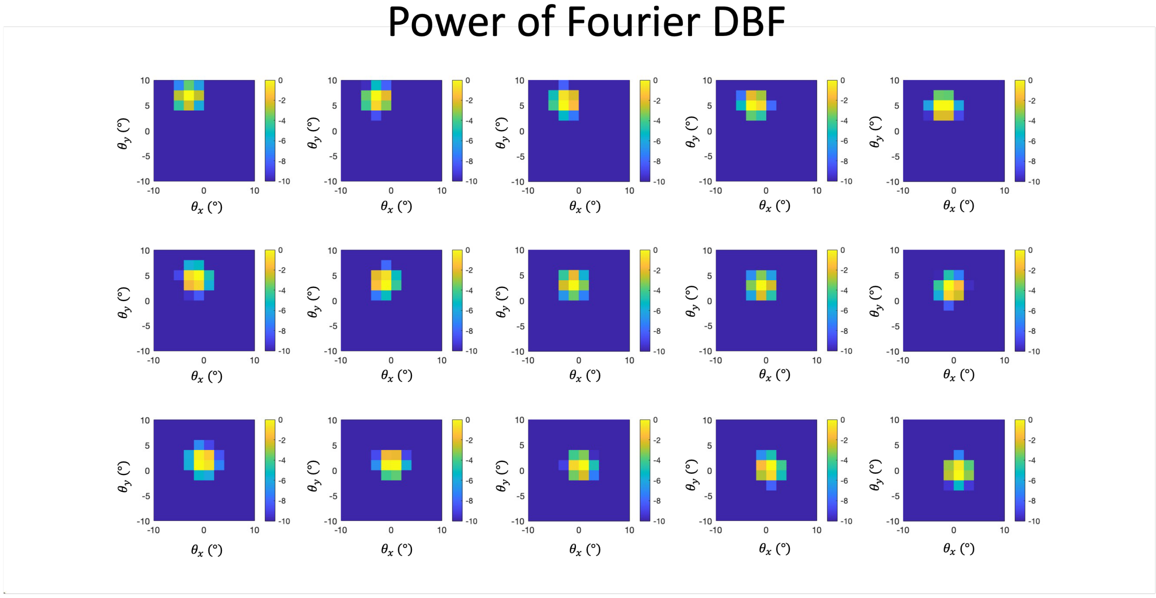


Figure 3.25: Normalized power of the Fourier method with real TEP data of a moving bird. A bird is moving over 21 scans. Every 5^{th} scan is plotted. X-axis is EW (θ_x) and y-axis is NS (θ_y).

ficially added, is circled in red. This figures show that the Fourier method was not able to suppress the high power coming back from the interference caused by the stationary point

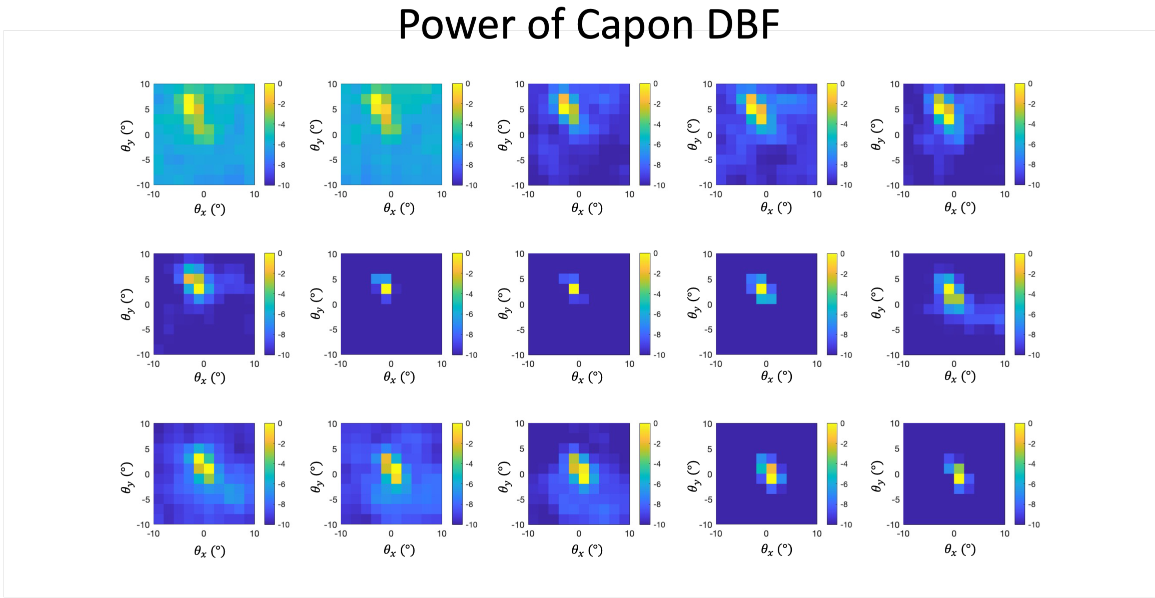


Figure 3.26: Normalized power of the Capon method with real TEP data of a moving bird. The point target is moving in a heart-shaped path over 75 scans. Every 5^{th} scan is plotted. X-axis is EW (θ_x) and y-axis is NS (θ_y).

target while the Capon method was able to suppress the high power coming back from the undesired directions.

Normalized antenna patterns of the Fourier and the Capon methods around the stationary point target for each weather case are shown in Fig. 3.28, Fig. 3.31, and Fig. 3.34. The Fourier beam pattern is in red and the Capon one is in blue. The location of the stationary point target is marked with a green dotted line and the boresight angle is marked with a black line. These figures are showing that the beam pattern generated by the Capon method places a null adaptively at the location of the stationary point target while the Fourier method generates beam patterns with a fixed sidelobe level. Normalized antenna patterns of the Fourier and the Capon methods in the weather area for each weather case are shown in Fig. 3.29, Fig. 3.32, and Fig. 3.35.

The Capon method shows better performance achieved by adaptive nulling. However, it becomes challenging to implement in real-time when the number of antenna elements

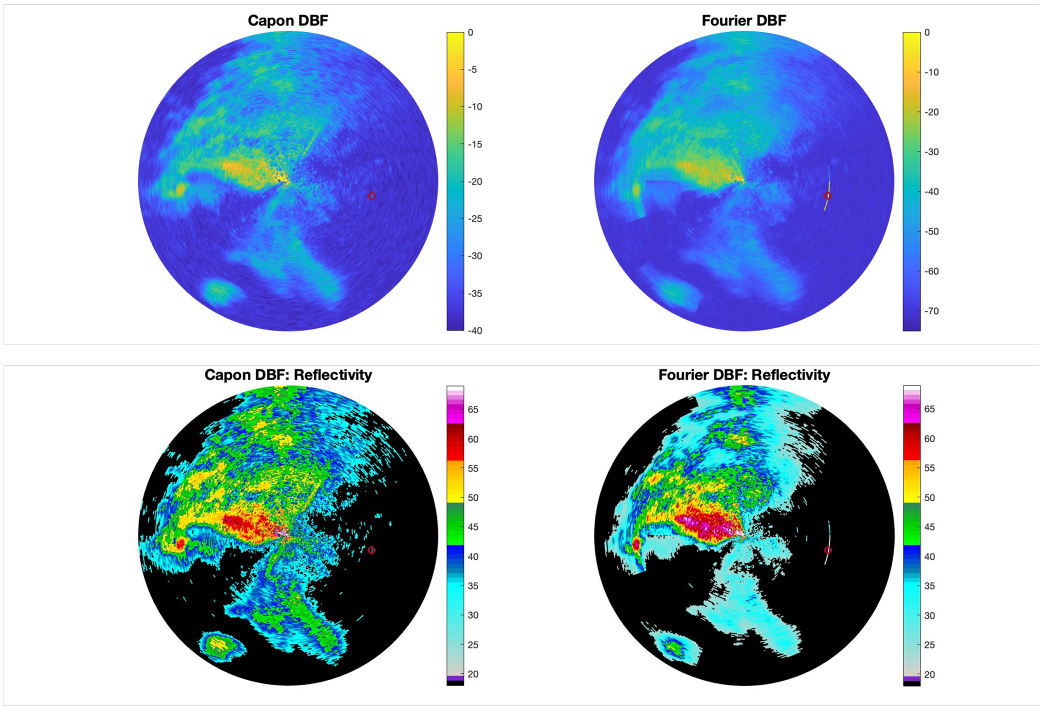


Figure 3.27: Normalized power in dB (top row) and reflectivity fields in dBZ (bottom row) by the Capon method (left column) and the Fourier method (right column) for the tornadic supercell storm case. The location of a stationary point target that was artificially added is circled in red.

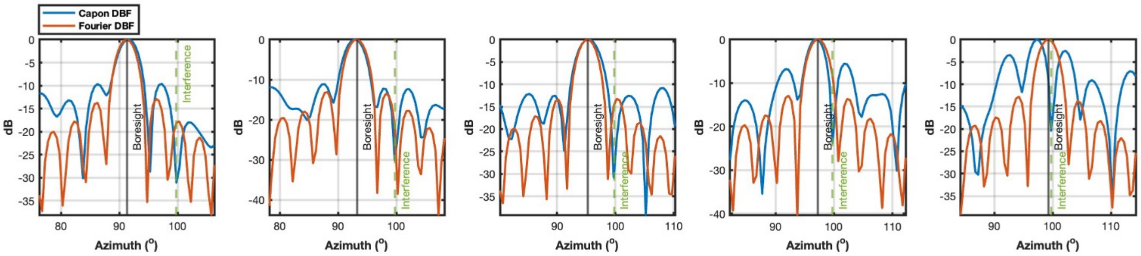


Figure 3.28: Normalized antenna pattern generated by the Fourier and the Capon methods around a stationary point target for the tornadic supercell storm case. A stationary point target is located at 99.7476° in azimuth, marked with a dotted green line. From the left to the right, the beam steers at 91.257° , 93.2767° , 95.2295° , 97.2372° , and 99.2615° , marked with a solid black line.

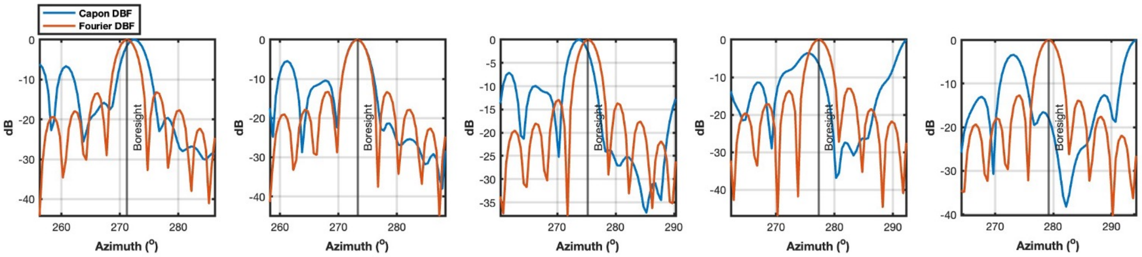


Figure 3.29: Normalized antenna pattern generated by the Fourier and the Capon methods for a tornadic supercell storm case. From left to right, the beam steers at 271.2387° , 273.2602° , 275.2570° , 277.2729° , and 279.2587° , marked with a solid black line.

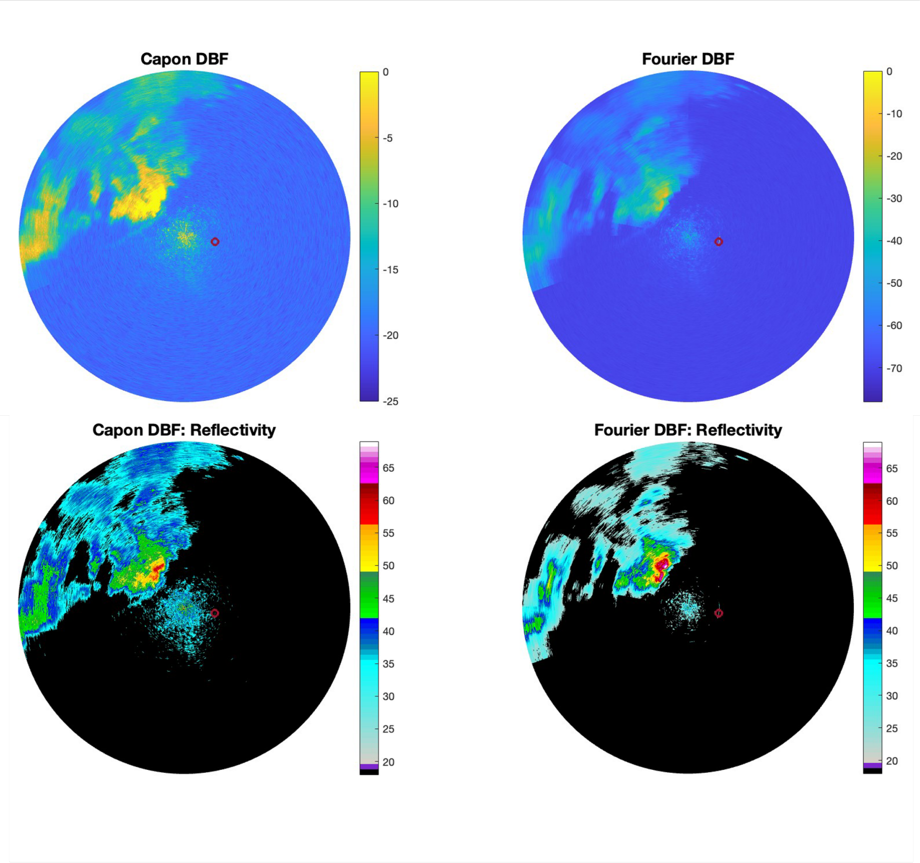


Figure 3.30: Normalized power in dB (top row) and reflectivity fields in dBZ (bottom row) by the Capon method (left column) and the Fourier method (right column) for a hail storm case. The location of a stationary point target that was artificially added is circled in red.

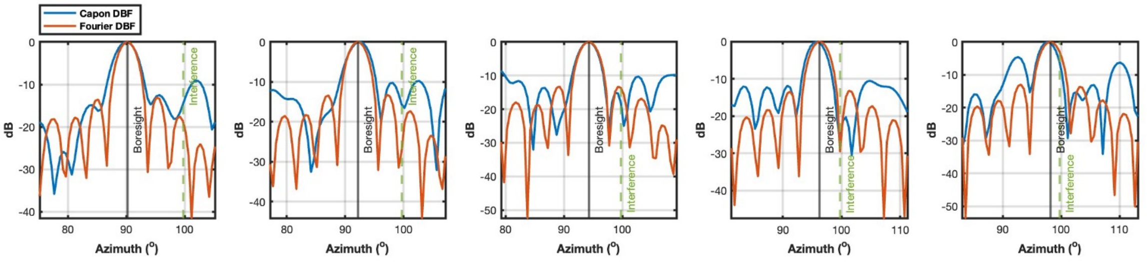


Figure 3.31: Normalized antenna pattern generated by the Fourier and the Capon methods around a stationary point target for the hail storm case. A stationary point target is located at 99.7476° in azimuth, marked with a dotted green line. From the left to the right, the beam steers at 90.2472° , 92.2330° , 94.2517° , 96.2512° , and 98.2343° , marked with a solid black line.

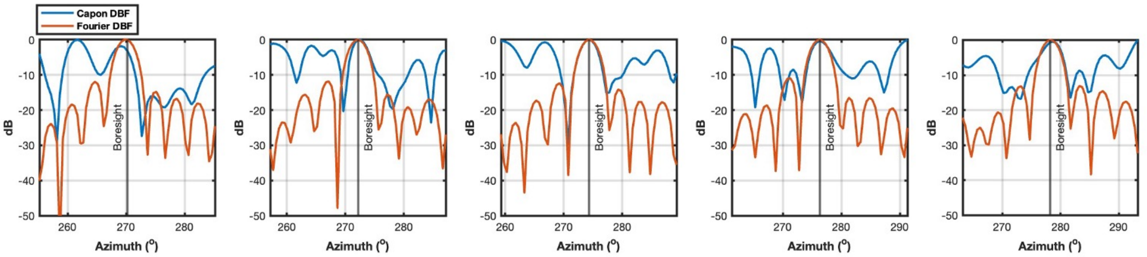


Figure 3.32: Normalized antenna pattern generated by the Fourier and the Capon methods in weather area for the hail storm case. From the left to the right, the beam steers at 270.2280° , 272.2495° , 274.2627° , 276.2512° , and 278.2617° , marked with a black line.

grows. That is because its computational complexity increases exponentially due to the needed covariance matrix inversion. Computational complexity of each DBF method is discussed in the following section.

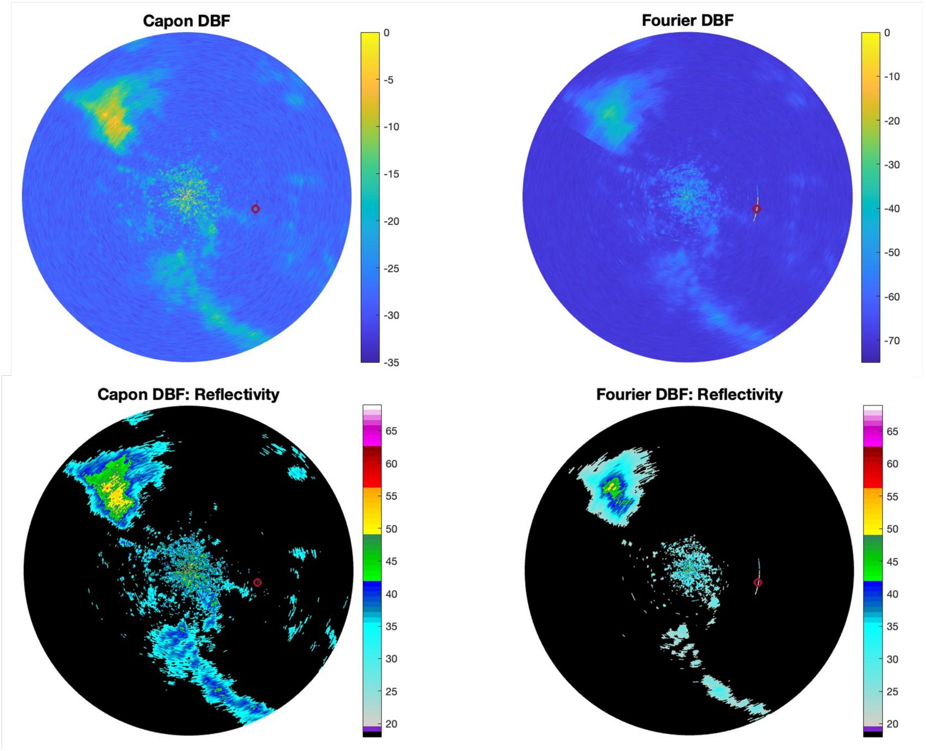


Figure 3.33: Normalized power in dB (top row) and reflectivity fields in dBZ (bottom row) by the Capon method (left column) and the Fourier method (right column) for the thunderstorm case. The location of a stationary point target that was artificially added is circled in red.

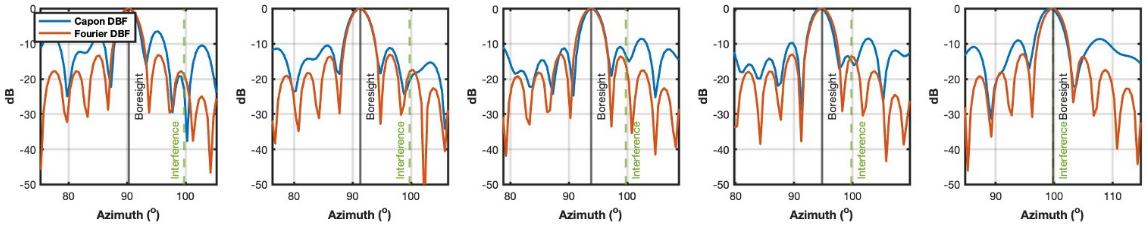


Figure 3.34: Normalized antenna pattern generated by the Fourier and the Capon methods around a stationary point target for the thunderstorm case. A stationary point target is located at 99.7476° in azimuth, marked with a dotted green line. From the left to the right, the beam steers at 90.2472° , 91.2579° , 93.7766° , 94.7296° , and 99.7394° , marked as a solid black line.

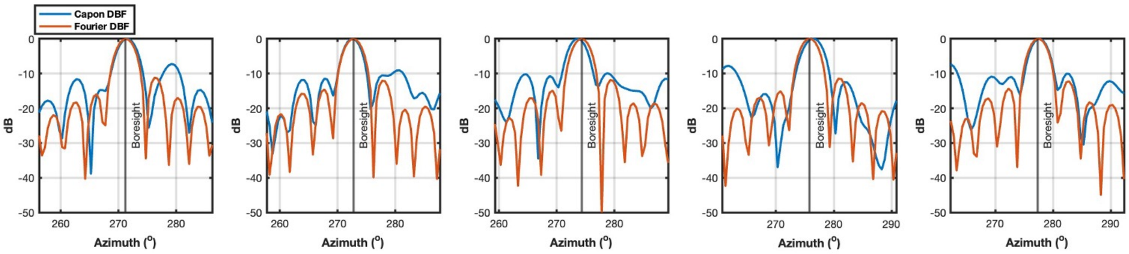


Figure 3.35: Normalized antenna pattern generated by the Fourier and the Capon methods in weather area for the thunderstorm case. From the left to the right, the beam steers at 271.2387° , 272.7301° , 274.2627° , 275.7980° , and 277.2729° , which is marked with a black line.

3.4 Computational Complexity of Digital Beamforming Methods

The Fourier method involves only matrix multiplication, namely $\boldsymbol{k} \cdot \boldsymbol{D}$, where $\boldsymbol{k}$ has a size of 1×3 and $\boldsymbol{D}$ has a size of 3×1 as described in Eq. (3.37). The complexity order of matrix multiplication of a $n \times m$ matrix and a $m \times p$ matrix that produces a $n \times p$ matrix will be $O(nmp)$ [81]. Therefore the complexity order of the Fourier method is $O(3)$ which is independent of the size of the receivers and remains constant.

The Capon method involves the matrix inversion to calculate R^{-1}, and the auto-covariance matrix R has a size of the number of receivers by the number of receivers (N×N), as described in Eq. (3.42). The complexity order of matrix inversion done by Gauss-Jordan elimination with a $n \times n$ matrix that produces a $n \times n$ will be $O(n^3)$ [81]. Therefore, the complexity order of the Capon method will be $O(N^3)$, which increases exponentially with N. Fig. 3.36 shows the order of complexity as a function of the number of receivers for each DBF method.

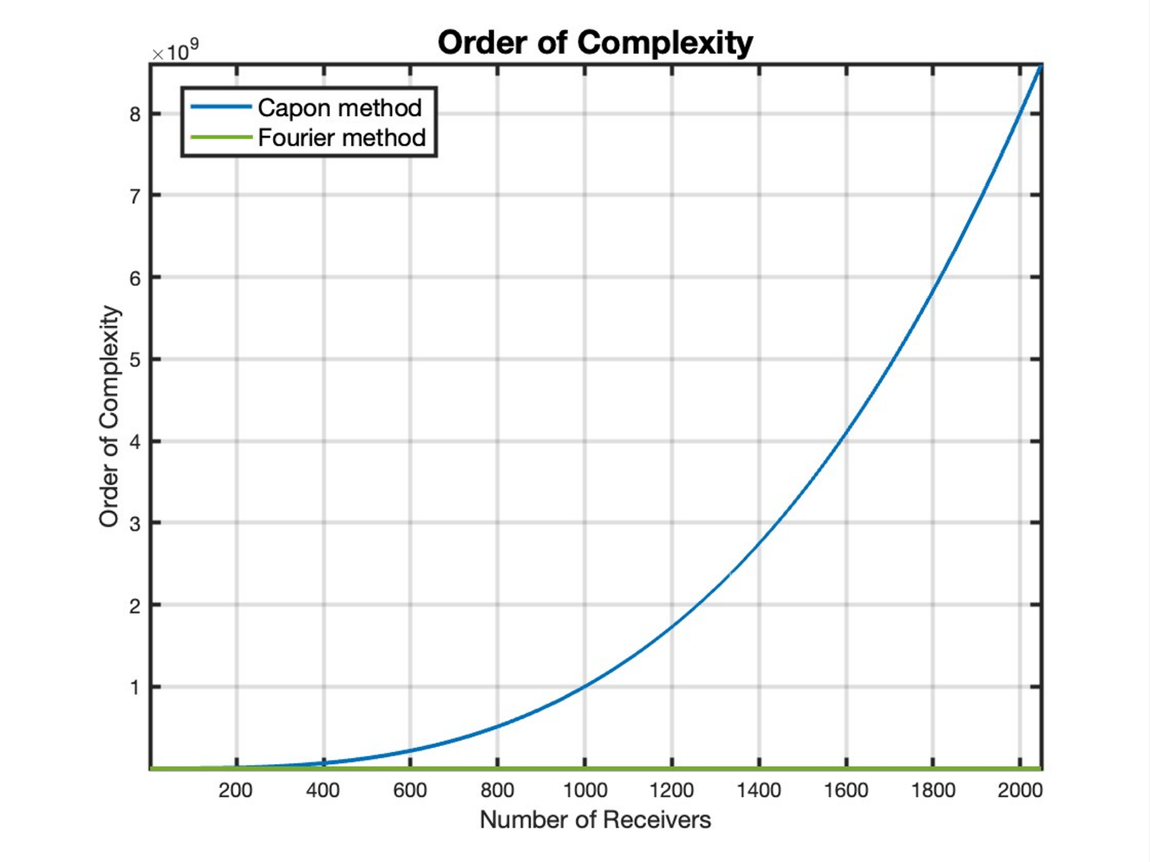


Figure 3.36: Complexity order of the Fourier and the Capon methods as a function of the number of the receiver. The blue line represents the complexity order of the Capon method and the green line represents the complexity order of the Fourier method.

Chapter 4

Deep Learning for Adaptive Digital Beamforming

To overcome the computational burden of the Capon method, the Deep Learning for Adaptive digital beamForming (DLAF) algorithm is developed in this thesis. DLAF is a feed-forward deep neural network and is trained through deep learning. The overview of DLAF's network is described in Fig. 4.1. The model is designed to take the time-series IQ signals

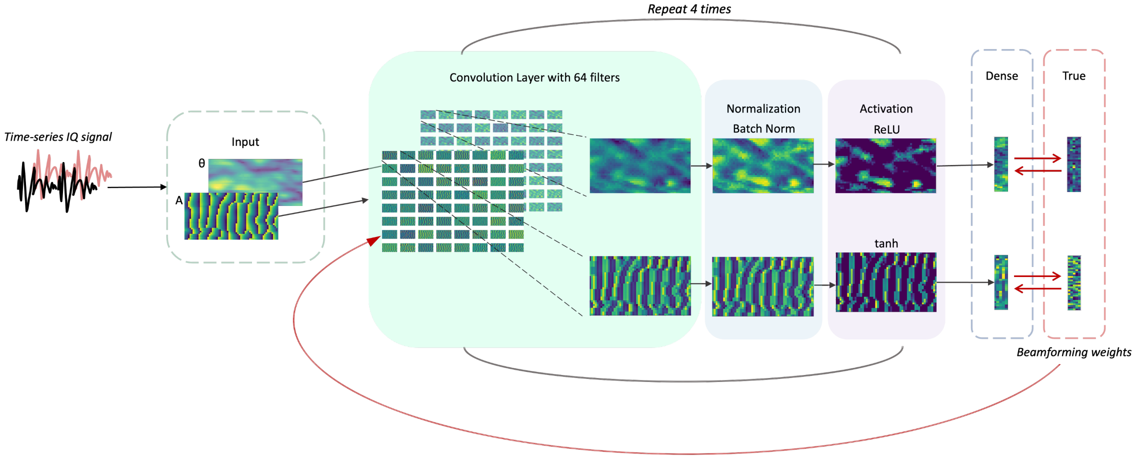


Figure 4.1: The proposed convolutional neural network that takes the time-series IQ signals and produces the beamforming weights to perform adaptive DBF. The network is built with convolution layer, batch normalization layer, activation layer, and dense layer. It extracts important features of inputs, compares with ground truth data, and learns from the estimated bias.

as inputs and produce beamforming weights for each antenna element at specific angular directions as output, based on the Keras library in Python. Keras is one of the most widely used libraries for ML users, since it offers many different layers including activation, convolution, and dense to help users build flexible and efficient networks [82]. Yet, Keras does not support complex values in the network. Separating complex values into real and imaginary parts is not desirable for this case, because it loses the phase information. Therefore, complex values of signals are transformed into phasor form for the full use of Keras' flexibility and preserving the phase information of the signals. The transformation between rectangular and polar forms of complex values are shown below,

$$x = a + jb, \tag{4.1}$$

$$\theta = \tan^{-1}\left(\frac{b}{a}\right), \tag{4.2}$$

$$A = \sqrt{a^2 + b^2}, \tag{4.3}$$

$$a = A\cos(\theta), \tag{4.4}$$

$$b = A\sin(\theta). \tag{4.5}$$

For efficient learning, signal phase and amplitude are trained independently in parallel with different activation functions.

The remainder of this section is organized as follows. The structure of the training datasets is discussed in Section 4.1. The components of the network and their functions are discussed in Section 4.2. Lastly, the training strategy is discussed in Section 4.3.

4.1 Training Data for DLAF

This section will discuss how to prepare datasets for DLAF training. The size and order of data dimension are important for the proper training of the network, because the convolution filter is applied on the two dimensions of the data (e.g., image). Therefore, the first dimension of the input data indicates the different sets of training data, and the second and

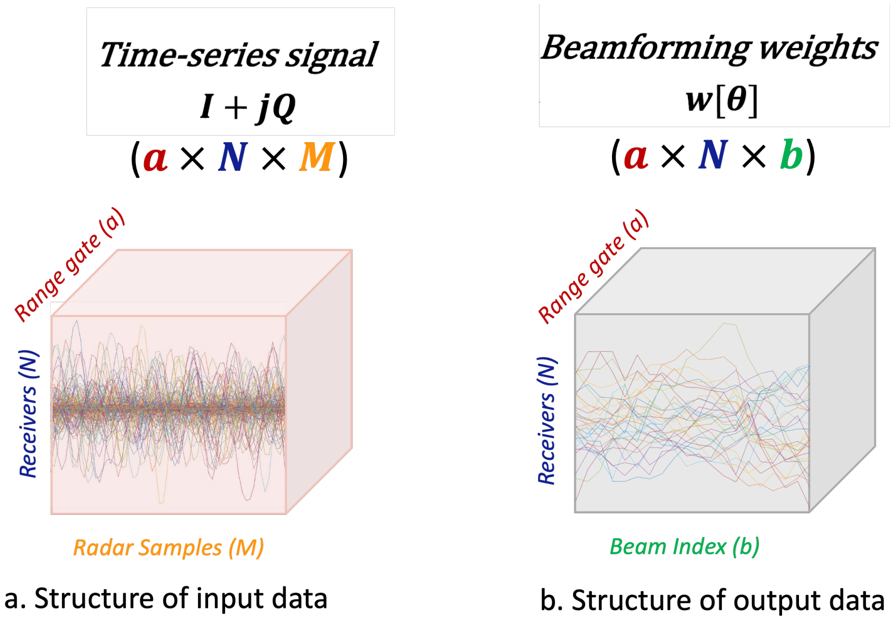


Figure 4.2: (a) Structure of input data for DLAF training. Input data have a 3D matrix format with a size of $a \times N \times M$, where a is the number of range gate, N is number of receivers, and M is number of radar samples. (b) Structure of output data for DLAF training. Output data have a 3D matrix format with a size of $a \times N \times b$, where b is the number of beam directions.

third dimension indicate the two dimensional input data. For DLAF, the time-series IQ signals and the Capon beamforming weights are used as input and output. The time-series IQ signals have four dimensions, each indicating the number of transmit beams, range gates, antenna elements, and radar samples. The first two dimensions are combined together to form a 3D matrix with a size of $a \times N \times M$, where a is the range number of IQ signals, and M is number of radar samples, as shown in Fig. 4.2. The Capon beamforming weights have four dimensions, each indicating the number of transmit beams, range gates, antenna elements, and beam index. The first two dimensions are combined together to form a 3D matrix with a size of $a \times N \times b$, where b is the number of beam directions. Then, the training data are normalized to minimize the loss and time of the learning before being fed to the network. Phase is normalized by π and amplitude is normalized by their maximum value. For point target cases, 200 IQ realizations of a moving point target were simulated for training, the number of receivers is 56 (TEP), with 64 samples. For weather cases, total 1800 IQ realizations (36 transmit beams and 50 range gates) of the tornadic supercell storm case were used for the training. Since the IQ signals of distributed targets (i.e., weather) are more complicated than the point target IQ signals, more datasets were used for DLAF training. The N is 32 (Horus), and M is 64 for the weather IQ signals.

4.2 Architecture of the Neural Network

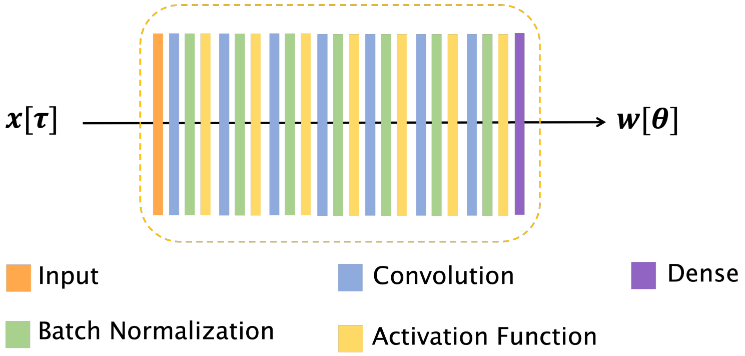

$\boldsymbol{x}[\boldsymbol{\tau}]$
$\boldsymbol{w}[\boldsymbol{\theta}]$
Input
Convolution
Dense
Batch Normalization
Activation Function

Figure 4.3: Architecture of the neural network. It contains 7 convolution layers followed by normalization layers and activation layers, and a dense layer as an output layer.

Two different networks were built to train phase and amplitude information independently. Both networks are composed of 7 convolution layers, 6 normalization layers, 6 activation layers, a dense layer, as shown in Fig. 4.3, to extract important features from input data. Yet, different activation functions are used for phase and amplitude information to improve the efficiency of the training. Tanh function, which can support negative values and non-linear changes of the data, is used as activation function for phase inputs, and ReLU, which can only support the positive values with the linear change, is used as activation function for amplitude inputs. The convolution layer uses the 64 convolution filters. A dense layer fully connects the neurons of current layer to the next layer, which could help to improve the accuracy. Then the network produces phase and amplitude of beamforming weights for each receiving elements and each beam index.

A set of phase and amplitude training data with a size of $N \times M$ are shown in Fig. 4.4. As shown, phase contains values from $-\pi$ to π and amplitude contains only positive values (≤ 0.07). Phase and amplitude data are going through the network separately. Then, they are combined to form complex valued beamforming weights at the end of the network. The convolution layers learns each data with 64 convolution filters. The results from the first convolution layer for the phase and amplitude are shown in Fig. 4.5 and Fig. 4.6. The spatial information of the time-series IQ signals may seem insignificant compared to image data. Therefore, the 64 convolution filters are used to improve the efficiency of learning spatial information in the input data.

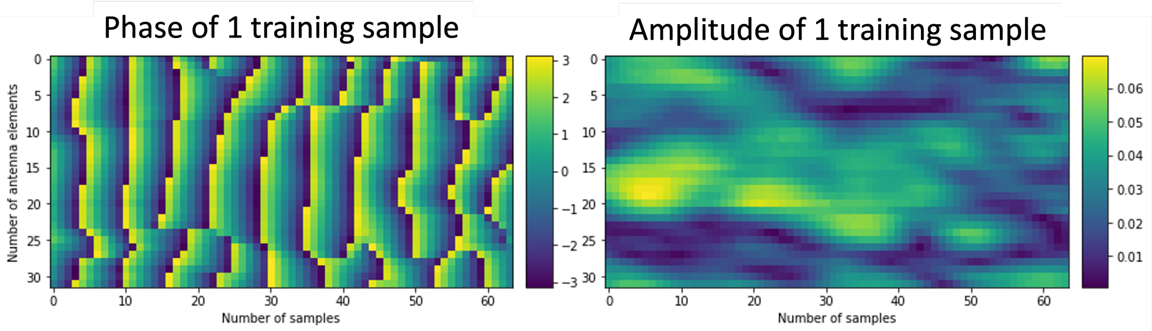


Figure 4.4: Illustration of phase (left) and amplitude (right) of training data for DLAF. The *x*-axis-axis shows number of samples and the *y*-axis-axis shows number of antenna elements. The value at each pixel is indicated with color. Phase has a range of $-\pi$ to π (radians) and amplitude has a range of 0 to 0.07

Data are normalized after every convolution layers to avoid excessive data loss during the activation layers. Batch normalization is used inside the neural network as it gets trained with a batch size of 36. Batch normalization takes 36 inputs of training data and uses the averaged statistical parameters to normalize data. Also, normalizing data throughout the training preserves the consistency of data. As data gets more complex, the role of normalization becomes critical for proper learning. The number of batch size, 36, indicates

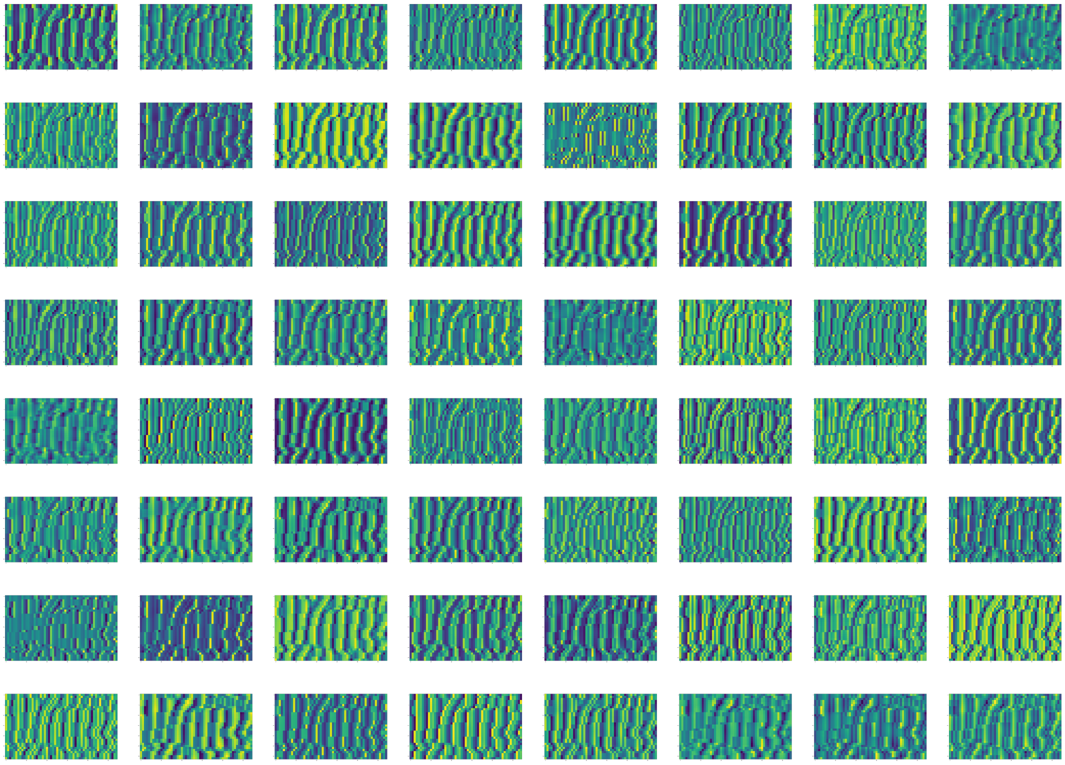


Figure 4.5: Illustration of outputs from 64 convolution filters applied to the phase. The color-scale is constant. Note that they have the same color scales from Fig. 4.4.

the number of transmit beams that is used to cover 360° in azimuth for one range gate. Activation functions allow the neural network to learn complex features of the data that cannot be analyzed by simple linear regression, as it was discussed in Chapter 2 with Eq. (2.4). Choosing the right type of activation function will impact the training significantly. Since phase and amplitude data types are different, two different activation functions are used. Linear activation function ReLU, which does not get saturated easily Eq. (2.5), is used for amplitude data and tanh activation layer, which support negative values and non-linear changes Eq. (2.10), is used for phase data. The function of the first convolution, normalization, and activation layers for each phase and amplitude data are illustrated in Fig. 4.7 and Fig. 4.8.

This process of convolution, normalization, and activation is repeated for 7 times. The size of data reduces due to the kernel size of convolution filters. The convolution layer has a kernel size of (1×10) for 6 times and (1×6) for the 7^{th} convolution layer. The size of the convolution kernel was chosen considering to have the right dimension for outputs. The second dimension of the data is constant between input and output, but the third dimension of the data changes from size of radar samples to size of DBF steering angles. It would be easier to start with a larger size of inputs and produce a smaller size of outputs. However, it is also possible to upsize the dimension of data from inputs to outputs with some extra layers such as maximum pooling or deconvolution layers. The last set of this process for phase and amplitude are shown in Fig. 4.9 and Fig. 4.10.

At the end of the convolution process, two dimension of neurons are transformed into one dimension with a flatten layer and a dense layer fully connects all the neurons from 64

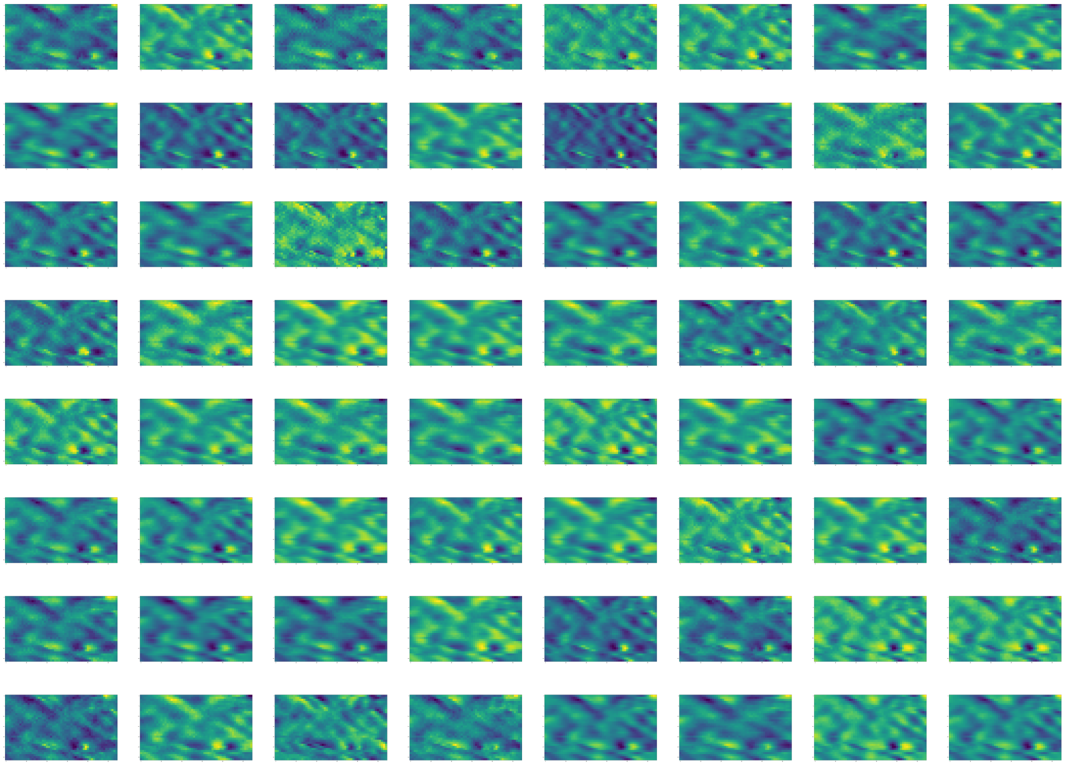


Figure 4.6: Illustration of outputs from 64 convolution filters applied to the amplitude. The color-scale is constant. Note that they have the same color scales from Fig. 4.4.

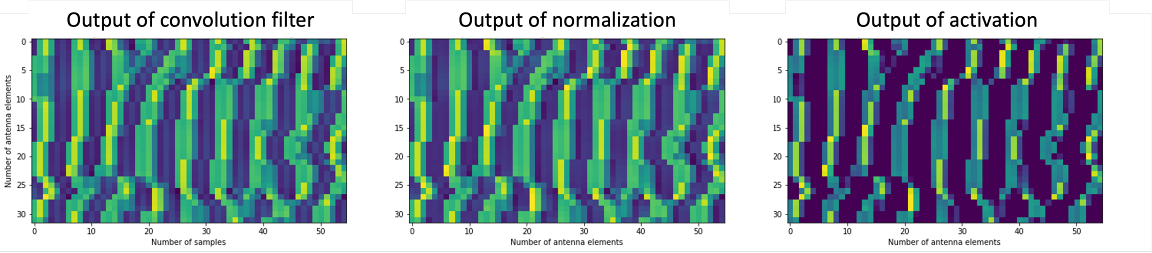


Figure 4.7: First set of convolution (left), normalization (middle), and activation (right) applied to the phase training data.

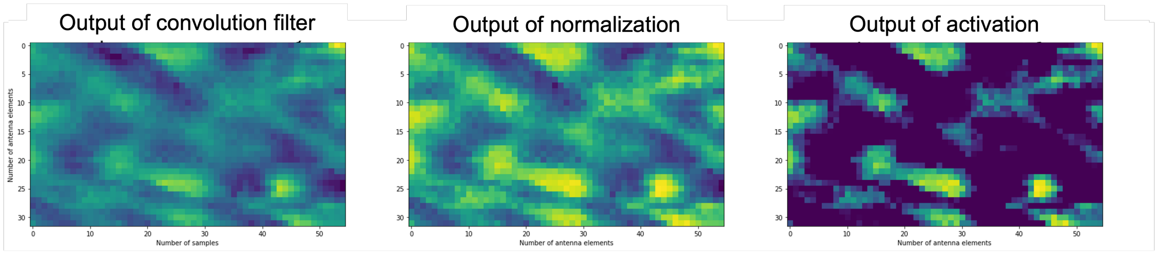


Figure 4.8: First set of convolution (left), normalization (middle), and activation (right) applied to the amplitude training data.

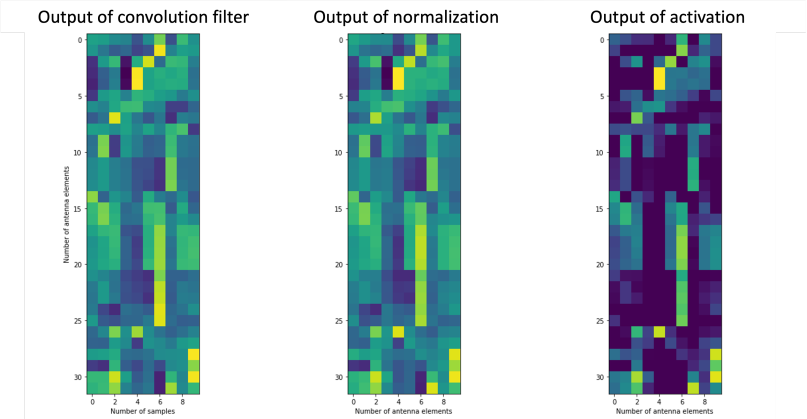


Figure 4.9: Last set of convolution (left), normalization (middle), and activation (right) applied to the phase training data.

filters of convolution layers as shown in Fig. 4.11. The number of neurons gets matched with the number of groundtruth data neurons through the dense layer. Then the phase and amplitude outputs from the network are shown in Fig. 4.12 and Fig. 4.13. From this state, the neural network calculates the bias between the output and truth, and update the

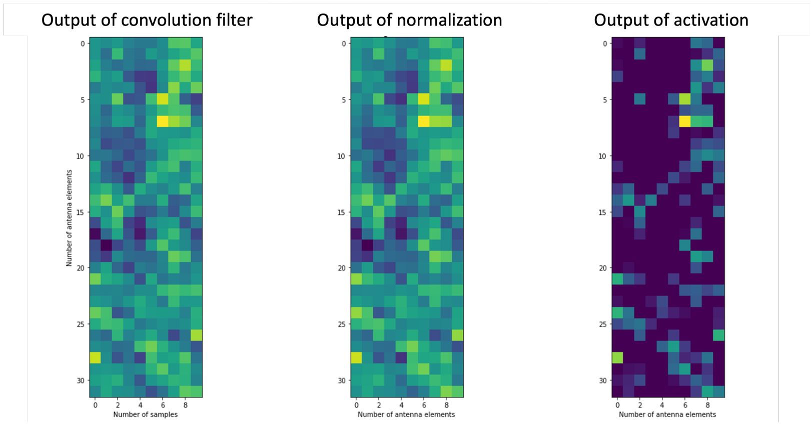


Figure 4.10: Last set of convolution (left), normalization (middle), and activation (right) applied to the amplitude training data.

learning weights over many epochs. After couple hundreds of epochs, the network is able to minimize the errors with MSE loss function and produce similar data to the truth as shown in Fig. 4.14.

4.3 Methodology

The performance of the training phase is evaluated with the loss over the number of epochs for two different normalization methods. One is called layer normalization, Eq. (2.18), and the other one is called batch normalization, Eq. (2.14). Details of these two normalization were discussed in Chapter 2. The results are shown in Fig. 4.15 and Fig. 4.16. Batch normalization starts with larger loss, yet it can be reduced over the epoch. In contrast, layer normalization could start with smaller loss, yet the training quality is not stable throughout the epoch. That is because strong IQ signals for the tornadic supercell storm can dominate the learning process over the weak weather IQ signals with layer normalization, which normalizes one layer at a time. However, batch normalization can normalize data for one range gate over the 360° azimuth to keep the consistency of the data.

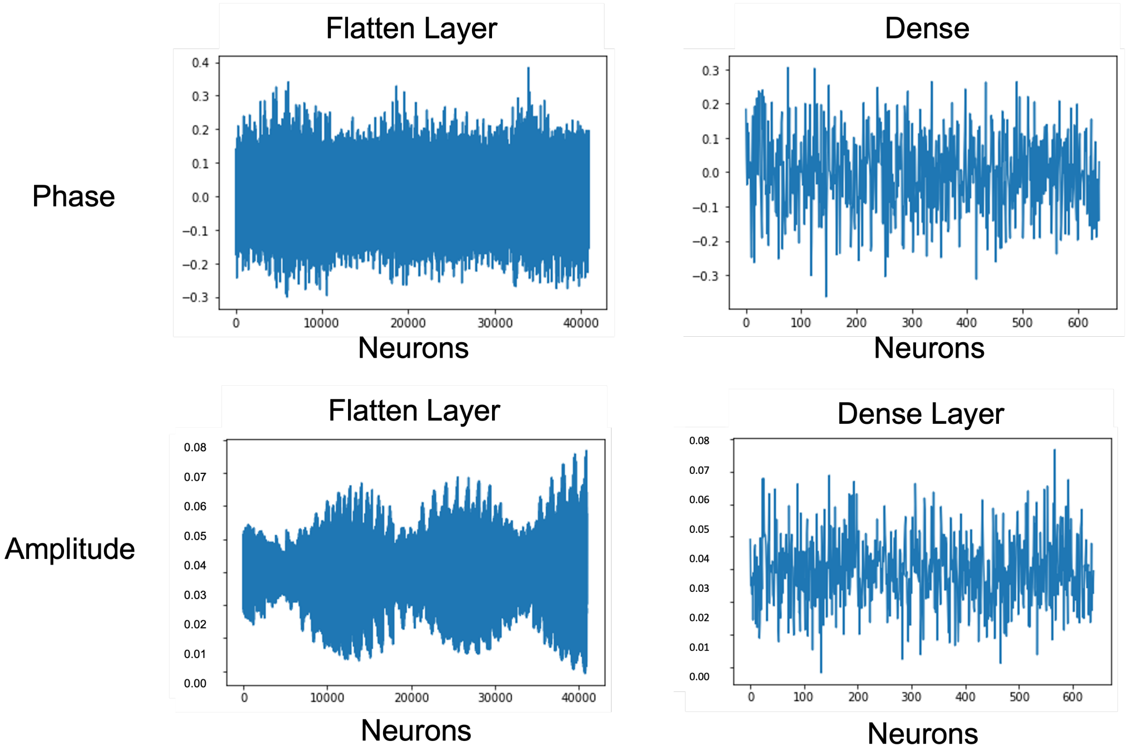


Figure 4.11: Flatten layer (left column) and Dense layer (right column). The first row shows the phase and second row shows the amplitude. The *x*-axis shows the number of neurons from the convolution layer and the *y*-axis shows the value of each neuron.

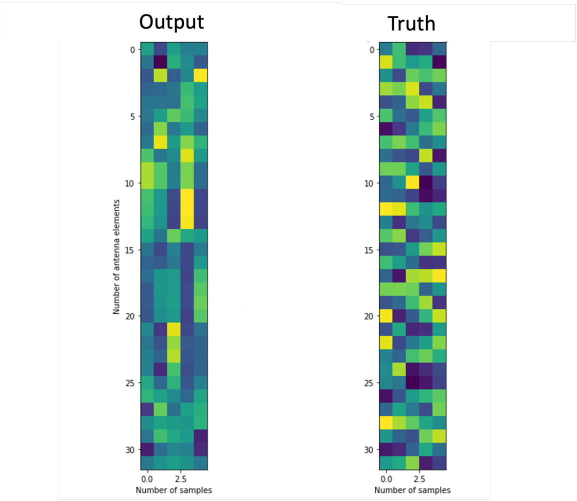


Figure 4.12: Training output from DLAF for phase values (left) and the true values, which are the Capon beamforming weights (right).

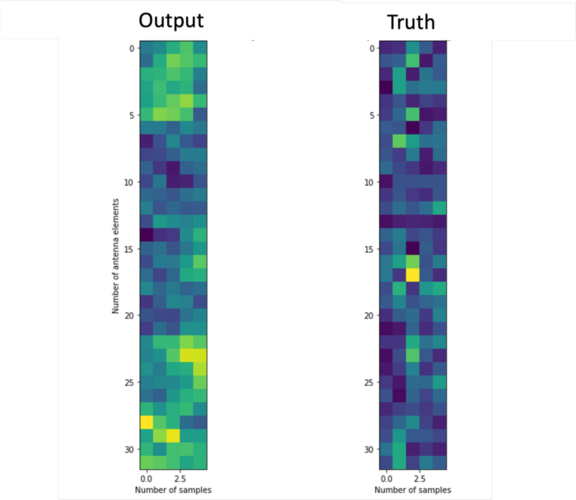


Figure 4.13: Training output from DLAF for amplitude values (left) and the true values, which is the Capon beamforming weights (right).

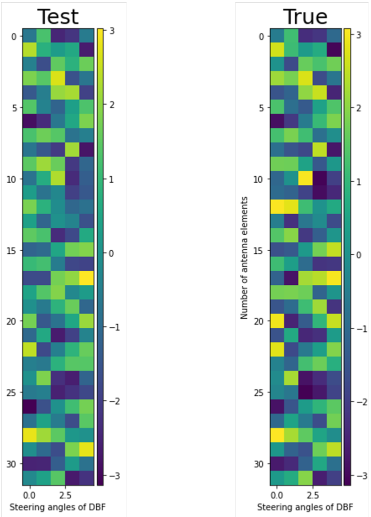


Figure 4.14: Result of the DLAF after the training for a couple hundreds epochs (left) and the true values (right).

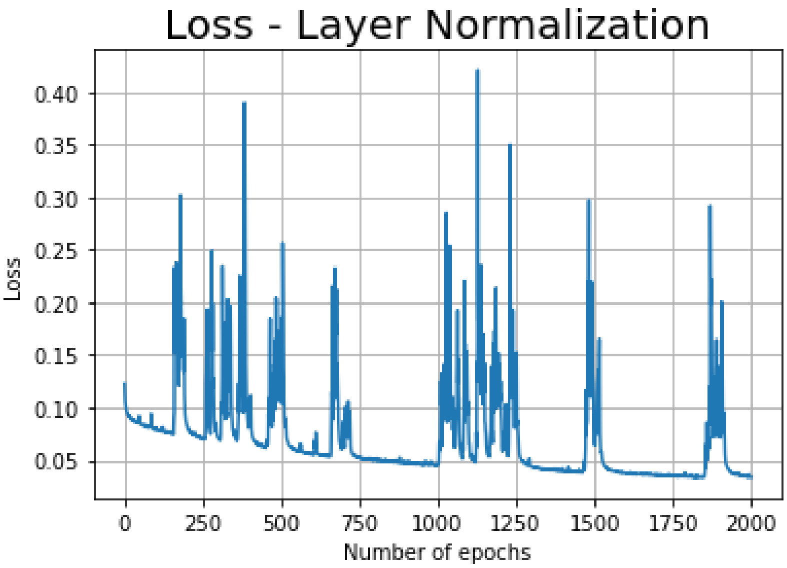


Figure 4.15: Loss during the training with the layer normalization.

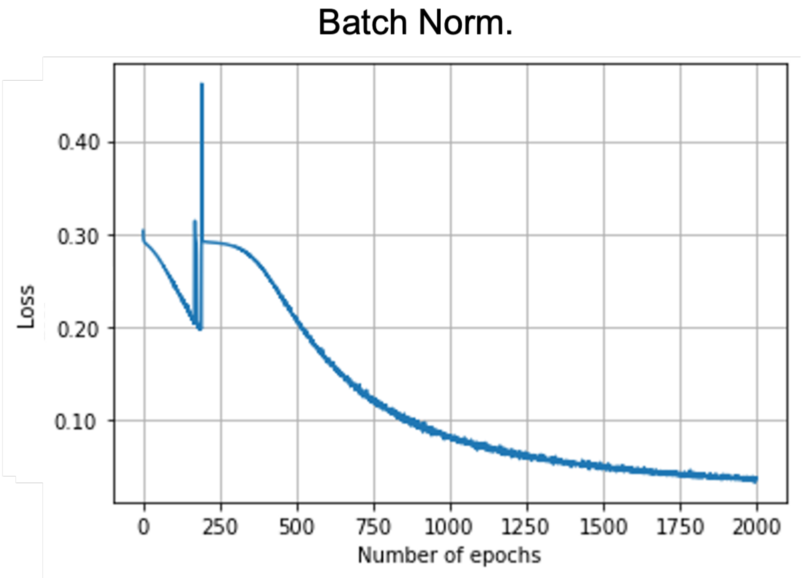


Figure 4.16: Loss during the training with batch normalization.

4.3.1 Complexity of the Neural Network

The complexity of this neural network is defined as $O(n^s L^d)$, where n is the number of neuron in the hidden layer, L is number of layer, s is the first dimension of the layer, and

d is the second dimension of the layer [83]. The complexity of the neural network will increase linearly depending on the number of layers and neurons. In this case, number of neurons will increase depending on the size of the time-series IQ signals. Therefore, the complexity order of training for DLAF will increase linearly with increasing antenna size or the number of radar samples. However, it would not impact the computation time for the inference mode once it is trained.

Chapter 5

Application and Evaluation of DLAF

DLAF has been tested on three different sets: 1. simulated moving point target 2. real data of a bird collected with the TEP 3. simulated weather cases 3a. supercell 3b. hail storm 3c. thunderstorm. The point target and weather simulations are described in Chapter 3. As DLAF has been trained with time-series IQ signals and Capon beamforming weights, the expected performance of DLAF is to produce similar beam patterns as the Capon patterns. In this chapter, the performance of DLAF is evaluated and compared with the Capon and Fourier methods. Two qualitative metrics are used for the evaluation of DLAF performance: 1. Quality of the generated beam pattern, such as the -3 dB beamwidth and artifacts from sidelobes 2. The capability of adaptive DBF in placing a null at the location of the interference. Four quantitative metrics are used for the evaluation: 1. Computation time 2. Performance of DLAF as a function of SNR 3. MSE of beamforming weight values and 4. Normalized cross-correlation of generated beam patterns.

5.1 Qualitative Analysis

The quality of the generated DLAF beams is analyzed and compared to the beam patterns generated by the Capon and Fourier methods. The generated patterns by DLAF DBF are evaluated based on how well they agree with the Capon patterns. DLAF has been tested for a moving point target that was simulated based on TEP configuration as described in Chapter 3, with simulation specifications described in Tab. 3.1. Then it was tested with real point target data that was collected by TEP, described in Chapter 3. Finally, DLAF was tested with weather data, which are distributed targets, of three different weather cases, supercell, hail storm, and thunderstorm. The results of DLAF for the simulated and real point target data are analyzed with power plots, 3D antenna pattern plots, and 2D antenna pattern plots. The application of DLAF to weather data is analyzed using PPI maps of signal power, line plots of power at 21.125km (range) for 360° (azimuth), reflectivity fields, and 2D antenna pattern plots.

A total of 5 different scans of the normalized power by each DBF method for the simulated point target are shown in Fig. 5.1. The power is normalized to have a peak at 0 dB and plotted with the same color scale from -25 dB to 0 dB. Based on the TEP configuration, DBF has been done in 2D scanning from -15° to 15° for both θ_x and θ_y. The top view of the 3D normalized antenna pattern plots by three different DBF methods of the 5^{th} scan are shown in Fig. 5.2. The interference is located at $\theta_x = -3.5^\circ$ and $\theta_y = 3^\circ$, marked with an X, and the beam is steered at $\theta_x = 3^\circ$ and $\theta_y = 3^\circ$. The 2D normalized antenna pattern plots from EW cut and NS cut of the Capon, the Fourier, and the DLAF methods are shown in Fig. 5.3. In the EW cut plot (left), the beam is steered to 3° and the point target is located at 3.5° . As it is shown in Fig. 5.3, the Fourier pattern (red) has high return power from the location of the point target due to its fixed sidelobe level. On the other hand, the Capon pattern (blue) and the DLAF (yellow) are placing a null at the location of the point target and lowering the sidelobe levels.

The real data of bird case collected by TEP as described in Fig. 3.24 is used to test the DLAF performance. DBF has been done in 2D scanning from -10° to 10° for both θ_x and θ_y. The power of each DBF method is shown in Fig. 5.4. The power is not normalized for this case due to different power levels and dynamic ranges. The Capon DBF power is

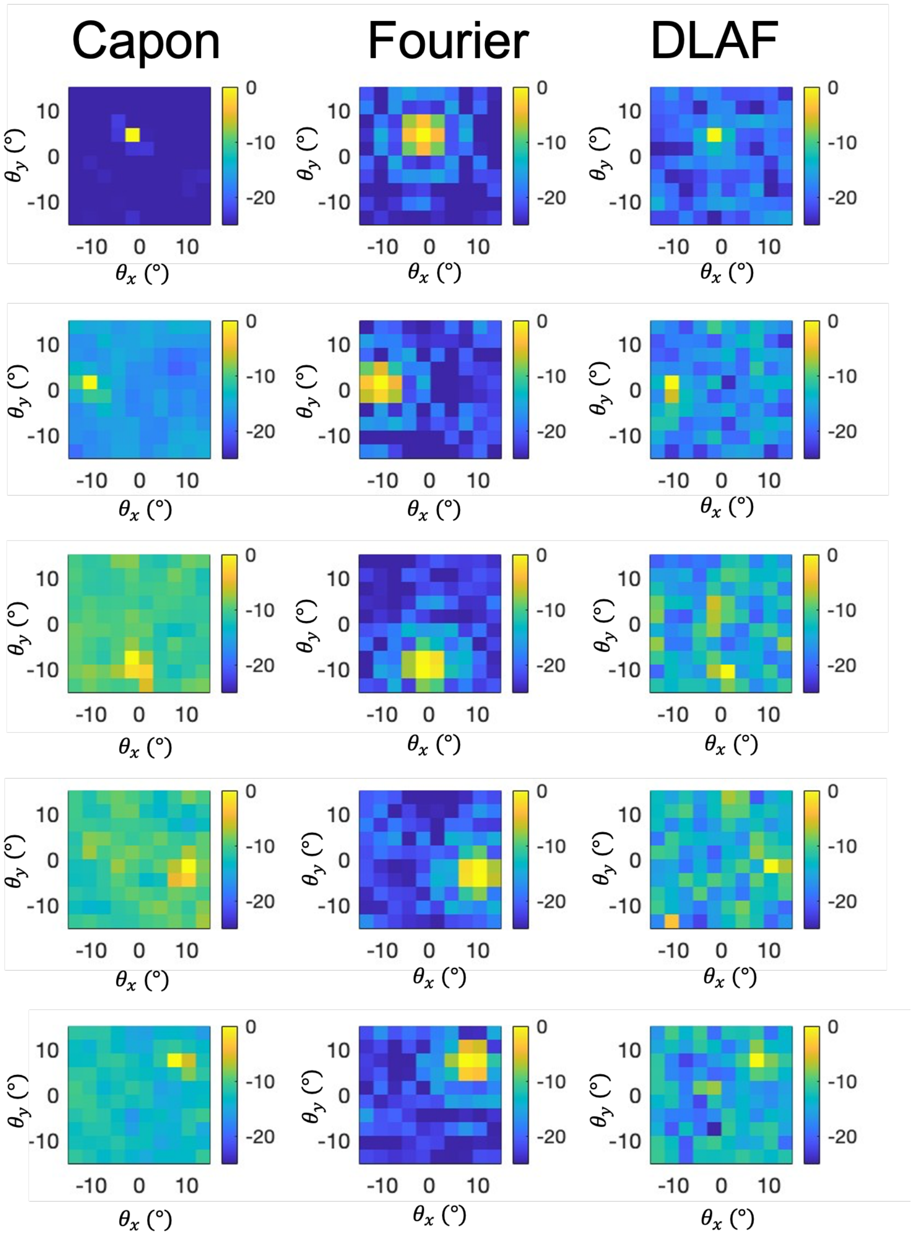


Figure 5.1: Normalized power (dB) plot of simulated moving point target by the Capon method (left), the Fourier method (middle), and DLAF (right). Note that all the power is normalized and has the same color scale for easy comparison. DBF has been done in 2D scanning, NS, and EW direction. The *x*-axis represents θ_x and the *y*-axis represents θ_y.

plotted in a range of 45 dB to 55 dB, the Fourier DBF power is plotted in a range of 50 dB to 85 dB, and the DLAF DBF power is plotted in a range of 40 dB to 50 dB. The top

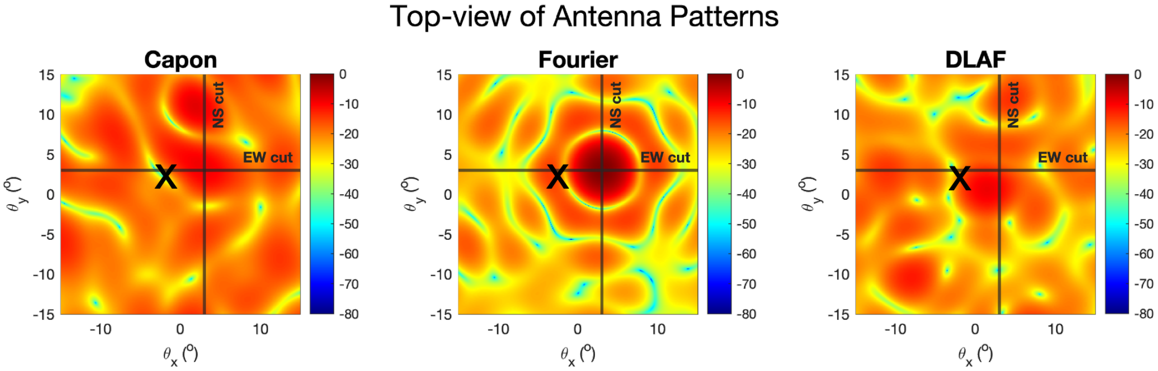


Figure 5.2: Top view of the 3D normalized antenna pattern of simulated moving point target by the Capon method (left), the Fourier method (middle), and DLAF (right). Note that all patterns are normalized and have the same color scale for easy comparison. DBF has been done in 2D. The *x*-axis represents θ_x (EW) and the *y*-axis represents θ_y (NS). The steering direction of the beam is marked with EW cut and NS cut. The location of the point target is marked as X.

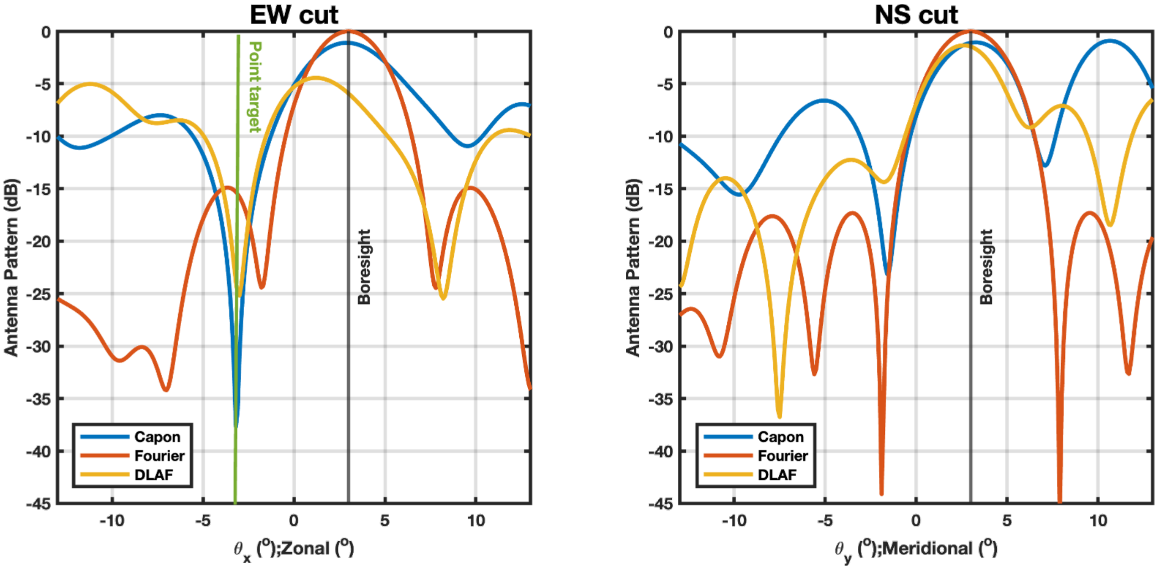


Figure 5.3: 2D normalized antenna pattern plot of simulated moving point target by the Capon method (blue), the Fourier method (red), and DLAF (yellow). The left plot is a 2D antenna pattern at EW cut and the right plot is a 2D antenna pattern at the NS cut. The *x*-axis represents θ_x (left) and θ_y (right). The *y*-axis represents the power of the antenna pattern in dB. The boresight of the beam is marked as a black solid line and the location of the point target is marked as a green solid line.

view of 3D antenna patterns plots by three different DBF methods of 60^{th} scan are shown in Fig. 5.5. The beam is steered to $\theta_x = 4°$ and $\theta_y = 2°$ and the interference is located at

$\theta_x = 3°$ and $\theta_y = 3°$. The 2D normalized antenna patterns from EW cut and NS cut are shown in Fig. 5.6. The Fourier beam patterns (red) have fixed sidelobe levels and result in high return power in the direction of the interference. Yet, the Capon and DLAF are minimizing power coming from the direction of the interference by placing a null. As it is shown in the Fig. 5.6, the point target is located in the mainlobe. However, the Capon and DLAF methods are still able to suppress the power in that direction. Furthermore, the overall shape of the beam patterns of DLAF are matched well with the Capon beam patterns.

DLAF is also tested for weather data, which is composed of distributed targets. Three different weather cases, supercell, hail storm, and thunderstorms are used to evaluate the performance of DLAF. As described in Chapter 3, a point target was artificially added at a location of 21.125km (range) and 99.7475° (azimuth) to test the capability of adaptive nulling. First, DLAF is tested for the supercell case. The power PPI of the three methods under consideration, Capon, Fourier, and DLAF are shown in Fig. 5.7. The point target that was artificially added is circled with red in each plot. There are some areas, especially around the tornado hook echo, showing discontinuity in the power PPI by the Fourier DBF method, while DLAF is showing better angular resolution. Also, the Fourier DBF method was not able to suppress the artificially added interference and result in high power from the undesired direction as shown in the power PPI. On the other hand, the Capon and DLAF DBF methods are able to suppress the power coming from the undesired direction as shown in the PPI plots. The power (in dB) of the three DBF methods at the ranges of 21.125km for 360° in azimuth are plotted in Fig. 5.8. To compare with the original IQ signals, which was simulated with NEXRAD radar variables, the power of IQ signals from one of the receiver is calculated and plotted in Fig. 5.8. The overall power level by DLAF method (purple) matches well with the power level by the Capon method (orange) and the power of the simulated IQ signal (blue). Furthermore, it shows that DLAF method was able to sup-

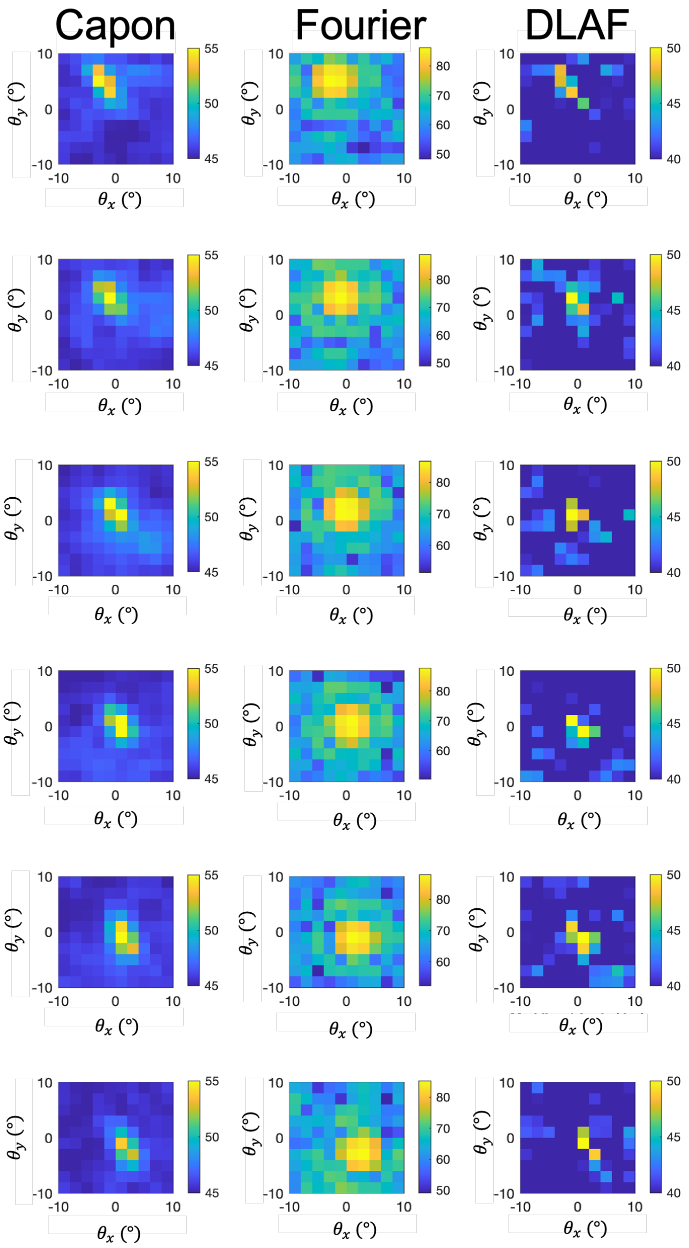


Figure 5.4: Normalized power plot (dB) of real data of a bird observed by TEP, with the Capon method (left), the Fourier method (middle), and DLAF (right). Note that all the power is normalized and has the same color scale for ease of comparison. DBF has been done in 2D scanning, NS, and EW direction. The *x*-axis represents θ_x and the *y*-axis represents θ_y

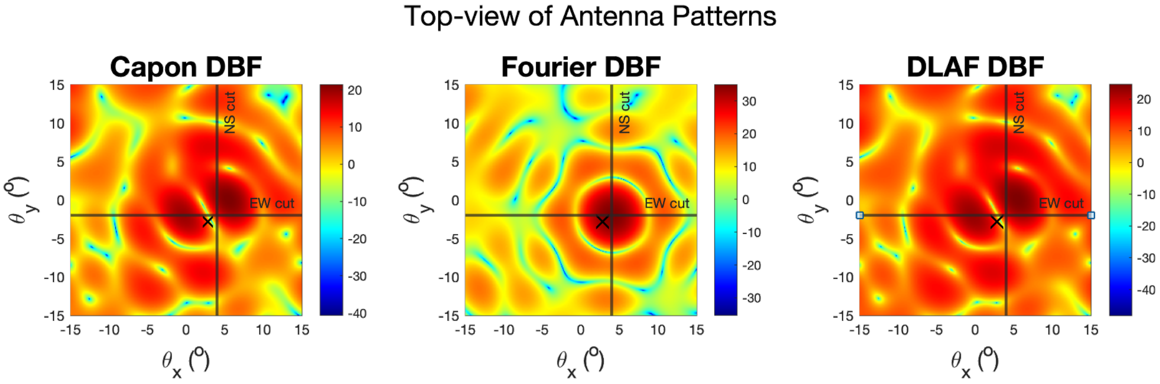


Figure 5.5: Top view of the 3D normalized antenna pattern of real data of a bird observed with TEP, by the Capon method (left), the Fourier method (middle), and DLAF (right). Note that all patterns are normalized and have the same color scale for easy comparison. DBF has been done in 2D. The *x*-axis represents θ_x (EW) and the *y*-axis represents θ_y (NS). The steering direction of the beam is marked with EW cut and NS cut. The location of the point target is marked as X.

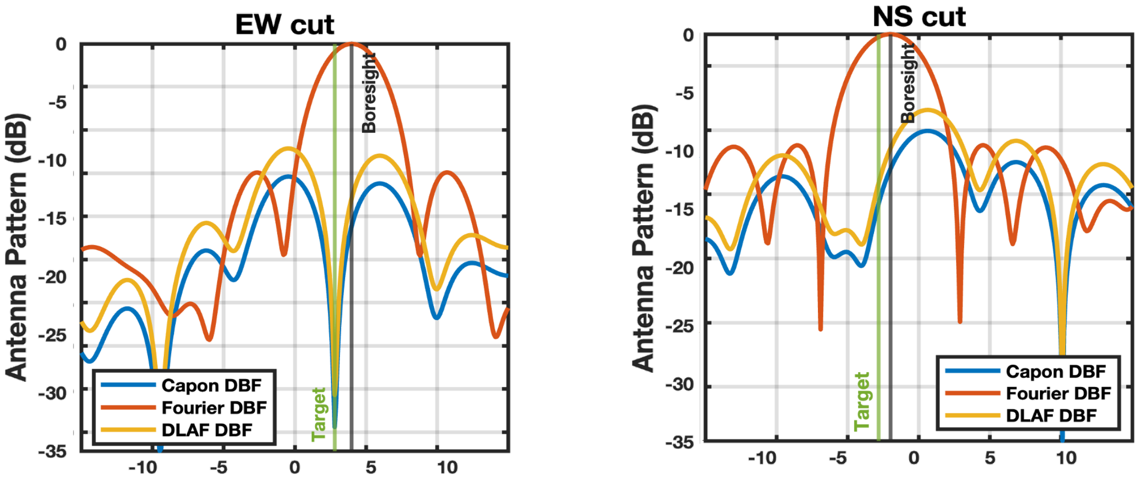


Figure 5.6: 2D normalized antenna pattern plot of real data of a bird observed with TEP, by the Capon method (blue), the Fourier method (red), and DLAF (yellow). The left plot is a 2D antenna pattern at EW cut and the right plot is a 2D antenna pattern at the NS cut. The *x*-axis represents θ_x (left) and θ_y (right). The *y*-axis represents the power of the antenna pattern in dB. The boresight of the beam is marked as a black solid line and the location of the point target is marked as a green solid line.

press the high power coming from undesired direction just like the Capon method, while it shows high power coming back from the undesired direction for the Fourier method. The reflectivity is calculated with Eq. (3.33). The PPI of the reflectivity field is plotted in Fig. 5.9 with the location of the point target circled in red. In the reflectivity field, the adaptive nulling of DLAF can be clearly seen, in contrast to the Fourier pattern. Normalized antenna patterns by three different methods for the point target area and the strong weather area at 21.125km (range) are plotted in Fig. 5.10 and Fig. 5.11. The location of the interference is marked with a dotted green line and the steering angle for each case is marked with a solid black line. The antenna patterns of DLAF (yellow) are well matched with the Capon antenna patterns (blue). Both method places a null at the location of the interference while the Fourier method (red) has the fixed beam patterns.

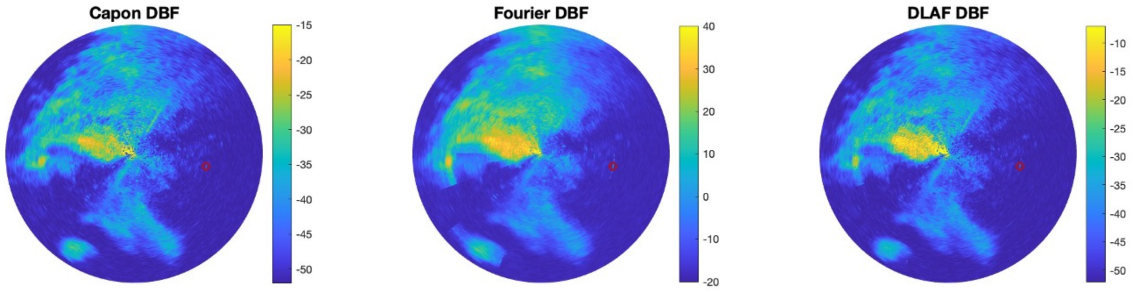


Figure 5.7: Power PPI (dB) by the Capon (left), the Fourier (middle), and DLAF (right) methods of the supercell case. The location of the point target, 99.7475° is circled in red.

Second, DLAF is tested for the hail storm case. The power PPI by the Capon (left), the Fourier (middle), and DLAF (right) methods are shown in Fig. 5.12. They are normalized to have a max value at 0dB. The power by the Capon and DLAF methods has the similar dynamic ranges and the same color scale, from -25dB to 0dB. On the other hand, the power by the Fourier has wider dynamic ranges and plotted from -75dB to 0dB. The point target that was artificially added is circled in red. The power by this three DBF methods at the range 21.125km for 360° in azimuth is plotted in Fig. 5.13 to compare the return power by each method around the point target. Note that there is no weather data around this area.

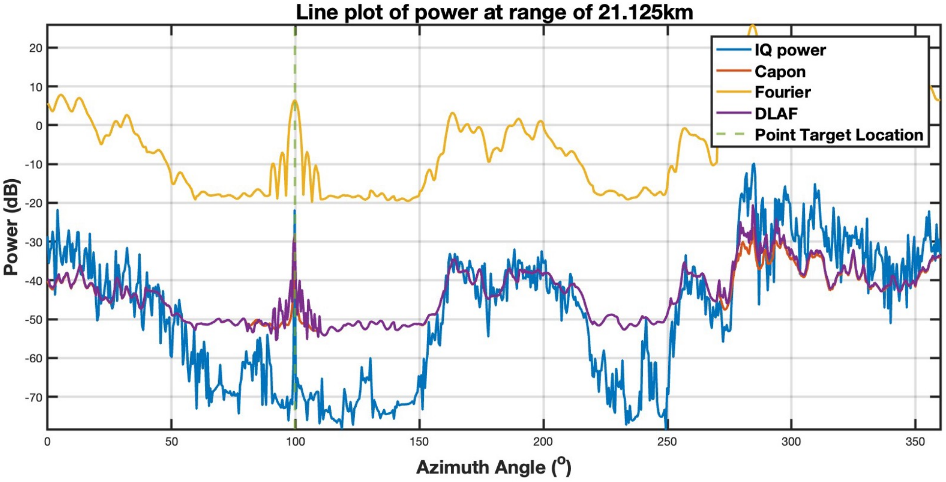


Figure 5.8: Power (dB) by the Capon, the Fourier, and DLAF methods at the range 21.125km for 360° in azimuth of the supercell case.

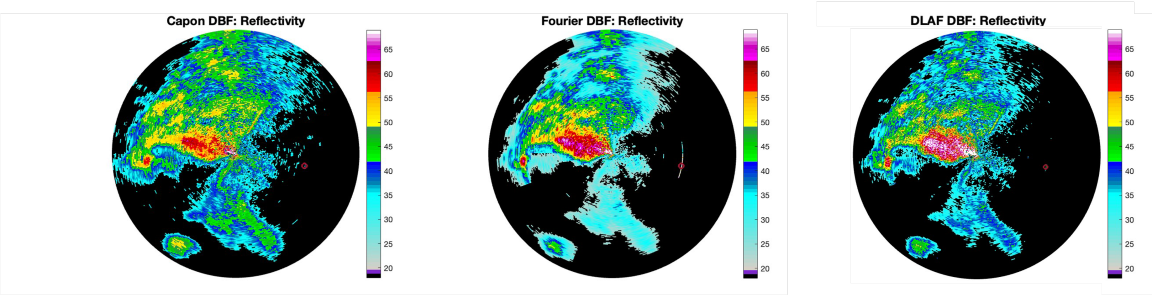


Figure 5.9: Reflectivity field of the supercell case. The Capon (left), the Fourier (middle), and DLAF (right) methods are plotted in the same color scale, from 17 dBZ to 68 dBZ. The point target is circled in red.

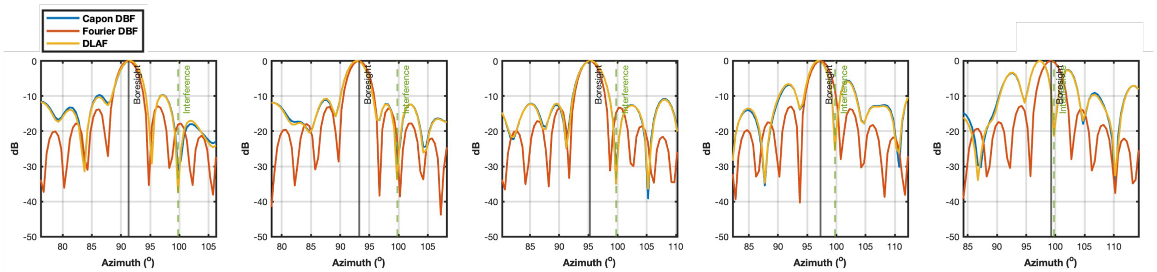


Figure 5.10: Antenna pattern of the Capon (blue), the Fourier (red), and DLAF (yellow) methods of supercell case in the point target area. The steering angle is marked with a solid black and the location of the point target with marked as a dotted green line.

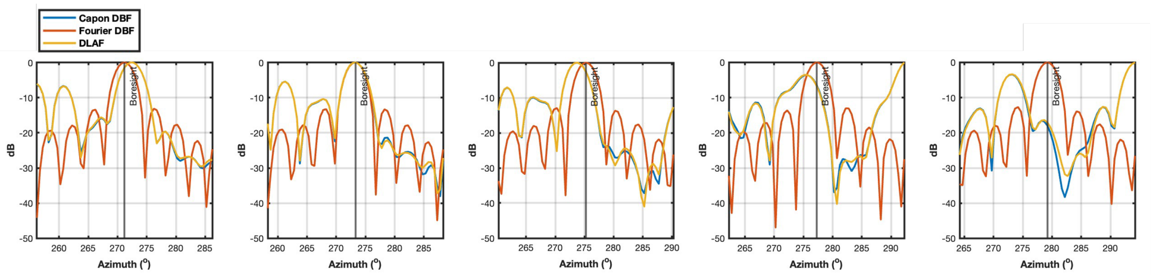


Figure 5.11: Antenna pattern by the Capon (blue), the Fourier (red), and DLAF (yellow) methods of the supercell case in the weather area. The steering angle is marked with a solid black.

The return power by the Fourier method (yellow) is high from 80° to 120° while the Capon (orange) and DLAF (purple) methods are able to suppress the power coming from undesired direction. The reflectivity field by the Capon (left), the Fourier (middle), and DLAF (right) methods is plotted in Fig. 5.14. The location of the point target is circled in red. Antenna patterns by this three different DBF methods for the point target area and for the weather area are plotted in Fig. 5.15 and Fig. 5.16. Fig. 5.15 shows the adaptive nulling capability of DLAF (yellow) even though overall shape of the pattern is not matched to the patterns by the Capon method (blue) as much as it did for the previous supercell case. This is because DLAF has been trained only with a part of the supercell case and the hail storm case has different SNR as shown in Fig. 3.18. This shows that the function of DLAF is sensitive with SNR, which will be discussed more in the following section. Yet, DLAF is still able to perform the adaptive nulling skill which means that DLAF has successfully learned from the training data, the Capon beamforming weights. For the weather area which contains distributed targets, DLAF has produced similar patterns to the Capon patterns.

Lastly, DLAF is tested for the thunderstorm case which has the low SNR. The power PPI by the Capon (left), the Fourier (middle), and DLAF (right) methods are shown in Fig. 5.17. The location of the artifically added point target is circled with red. For this case, DLAF shows the wider dynamic range which is from -48dB to 40dB. The power by

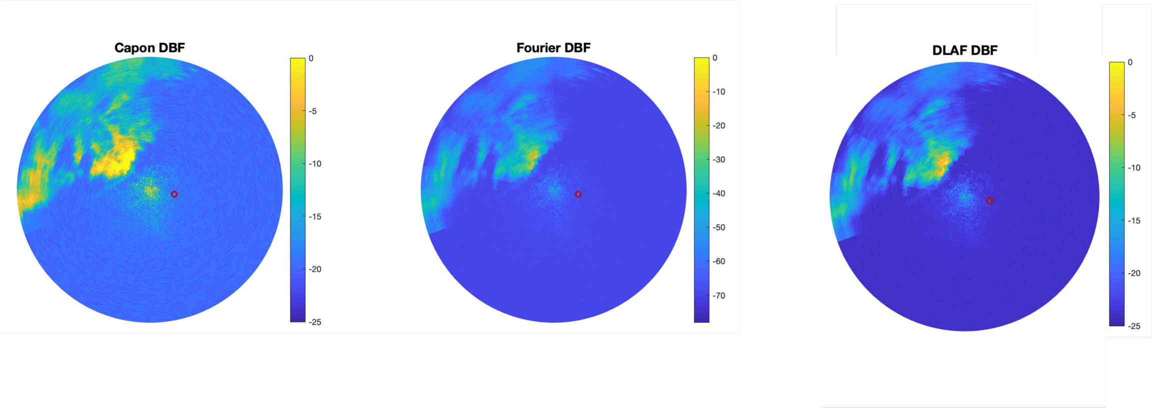


Figure 5.12: Power PPI (dB) by the Capon (left), the Fourier (middle), and DLAF (right) methods of the hail storm case. The location of the point target, 21.125km (range) and 99.7475° (azimuth), is circled in red.

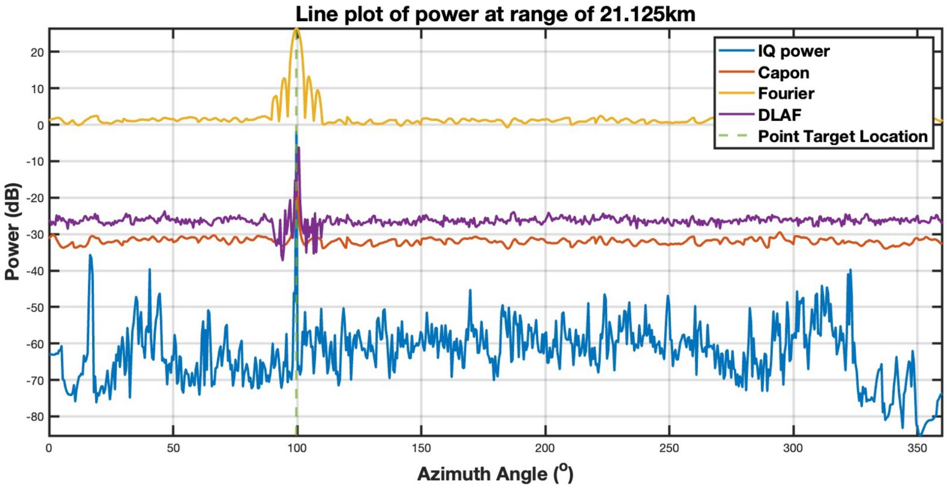


Figure 5.13: Power (dB) by the Capon, the Fourier, and DLAF methods at the range 21.125km for 360° in azimuth of the hail storm case.

this three DBF methods at the range 21.125km for 360° in azimuth are plotted in Fig. 5.18 to compare the return power by each method around the point target. The Fourier method (yellow) has high return power from undesired directions, 80° to 120° , while the Capon (orange) and DLAF (purple) methods suppress the power except the desired direction which is 99.7474° . The reflectivity field by each method is plotted in Fig. 5.19 and the location of the point target is circled in red. There is strong reflectivity appearing at the undesired

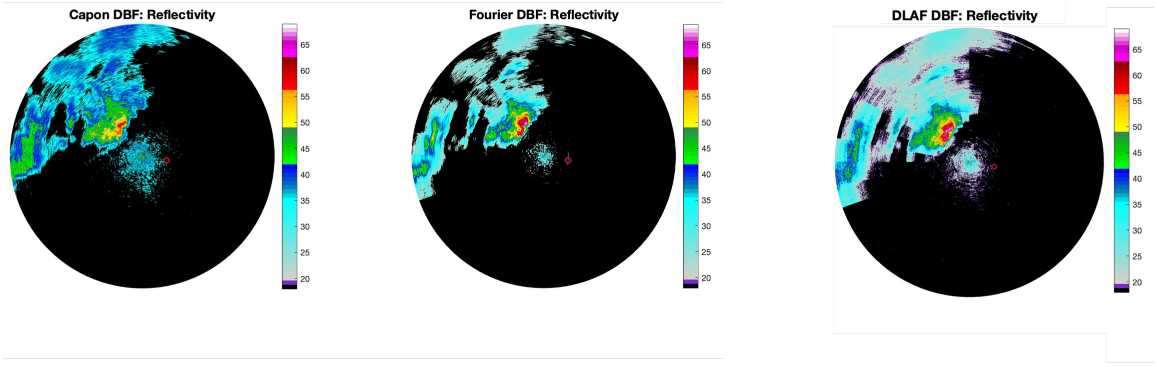


Figure 5.14: Reflectivity field (dBZ) of the hail case. The Capon (left), the Fourier (middle), and DLAF (right) methods for the hail storm case are plotted in the same color scale, from 17 dBZ to 68 dBZ. The point target is circled in red.

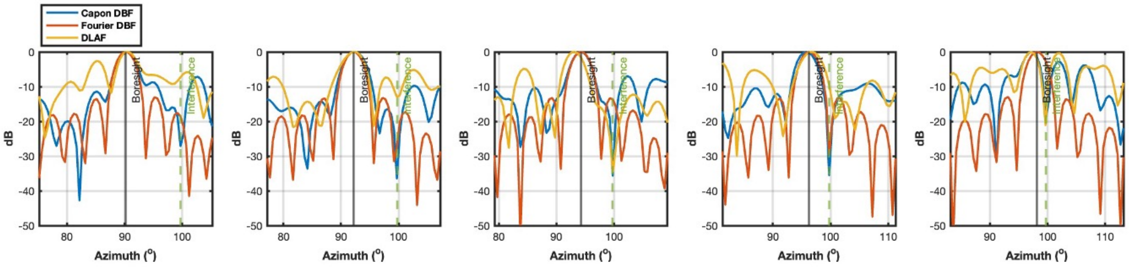


Figure 5.15: Antenna pattern by the Capon (blue), the Fourier (red), and DLAF (yellow) methods for the hail storm case. The steering angle is marked with a solid black and the location of the point target is marked with a dotted green line.

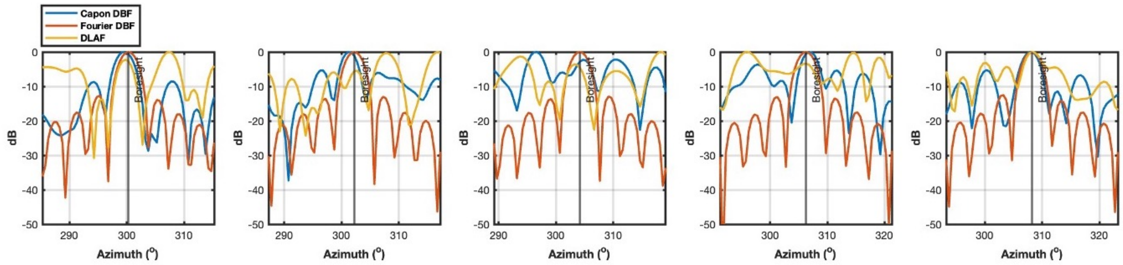


Figure 5.16: Antenna pattern by the Capon (blue), the Fourier (red), and DLAF (yellow) methods for the hail storm case. The steering angle is marked in a solid black.

location around the point target in the Fourier method (middle) plot while the Capon (left) and DLAF (right) method have strong reflectivity at the desired location. Antenna patterns by the Capon (blue), the Fourier (orange), and DLAF (yellow) methods for the point target area and for the weather area are plotted in Fig. 5.20 and Fig. 5.21. Fig. 5.20 shows that DLAF has successfully placed a null at the location of interference, yet the overall shape of the patterns may not be matched well as in the previous supercell case. Note again that DLAF has been only trained with supercell case which has high SNR. Since the thunderstorm case has the lowest SNR, patterns generated by DLAF may not be matched well with the Capon patterns even for the weather area as shown in Fig. 5.21.

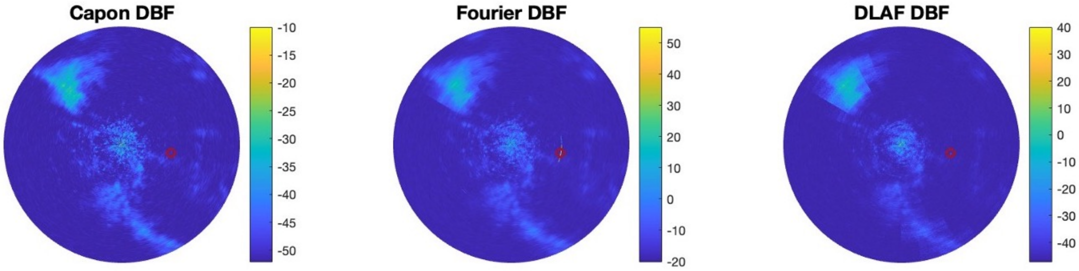


Figure 5.17: Power PPI (dB) by the Capon (left), the Fourier (middle), and DLAF (right) methods of the thunderstorm case. The location of the point target, 99.7475° is circled in red.

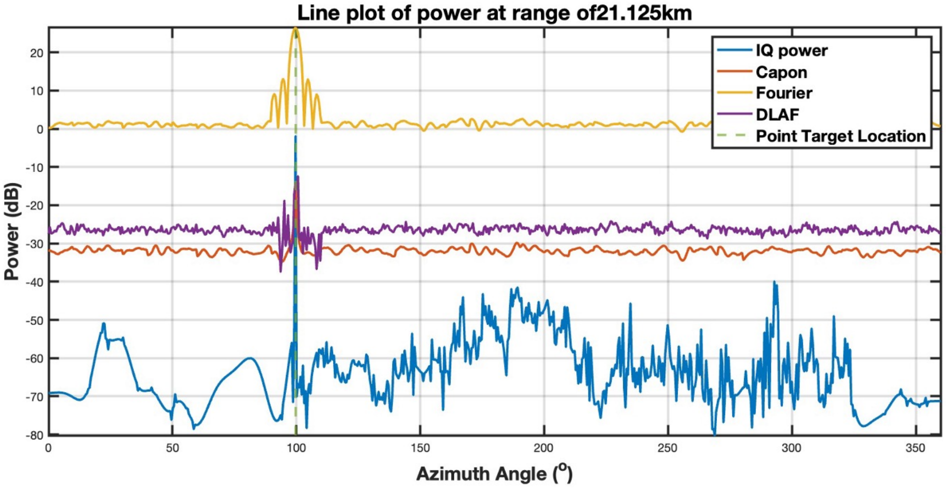


Figure 5.18: Power (dB) by the Capon, the Fourier, and DLAF methods of the thunderstorm case at the range 21.125km for 360° in azimuth.

DLAF has a better quality of performance for cases with higher SNR, such as a tornadic

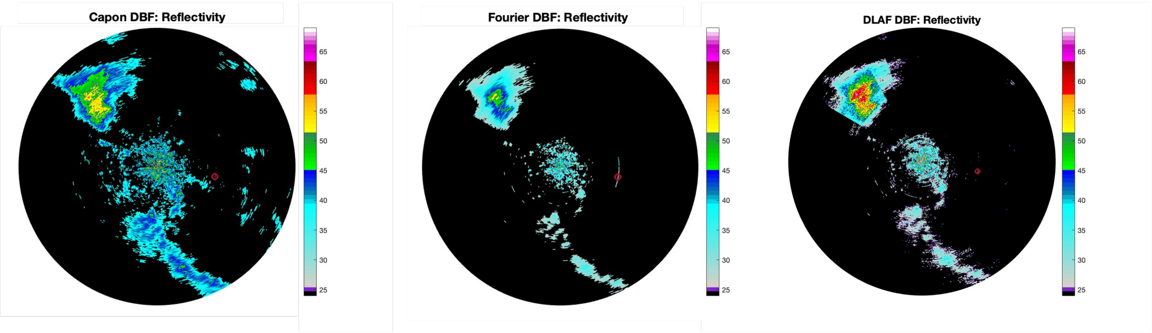


Figure 5.19: Reflectivity field (dBZ) of the thunderstorm case. The Capon (left), the Fourier (middle), and DLAF (right) methods are plotted in the same color scale, from 17 dBZ to 68 dBZ. The point target is circled in red.

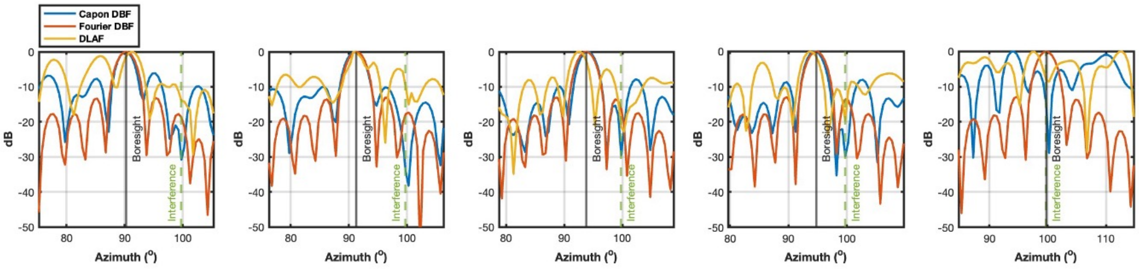


Figure 5.20: Antenna pattern by the Capon (blue), the Fourier (red), and DLAF (yellow) methods for the thunderstorm case. The steering angle is marked with a solid black and the location of the point target is marked with a dotted green line.

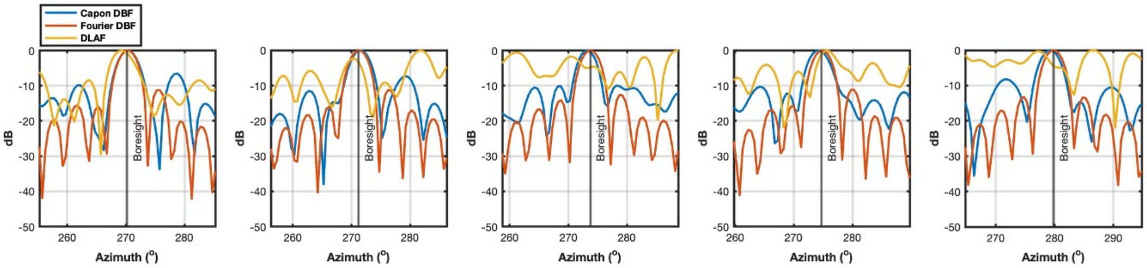


Figure 5.21: Antenna pattern of the Capon (blue), the Fourier (red), and DLAF (yellow) methods for the thunderstorm case. The steering angle is marked with a solid black.

supercell storm, compared to less impactful weather like a thunderstorm. Yet, DLAF has shown good quality performance of adaptive DBF for the point target. In the following section, DLAF will be analyzed quantitatively and discussed how the SNR would impact the performance of DLAF.

5.2 Quantitative Analysis

The DLAF performance is analyzed for the supercell case with four quantitative metrics. These include, the computation time, the MSE of the Capon and DLAF beamforming weights, the SNR impact on DLAF performance, and the correlation coefficient between the Capon and DLAF beam patterns.

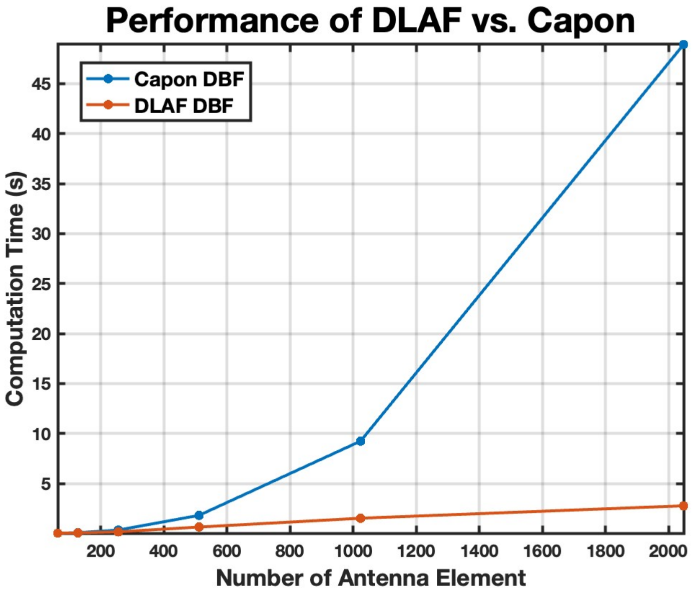


Figure 5.22: Measured computation time of the Capon DBF (blue) and DLAF DBF (red) for one spoiled beam with 20 receiving beams generated and 150 range gates. Computation time is measured for different sizes of the antenna element. The X-axis is the number of the antenna element and Y-axis is the computation time in seconds.

DBF has been done with 20 receiving beams generated for one transmitted spoiled beam to cover 10° in azimuth as discussed in Chapter 3. The computation time of performing

DBF with the Capon and DLAF methods to cover 360° azimuth and 150 range gates are measured with different number of receivers and it is shown in Fig. 5.22. The computation time of the Capon method is in blue and DLAF method is in red. As expected from Fig. 3.36 and Chapter 4, the computational complexity of the Capon method exponentially grows with the number of receivers while the complexity of the DLAF method linearly grows with the number of receivers. Especially at the size of Horus antenna, 1024 (32×32), the Capon method takes about 9 seconds and the DLAF method takes about 2 seconds. Therefore, DLAF has shown the potential to overcome the computational burden of the Capon method for radars with the bigger antenna size while it still performs adaptive DBF.

The MSE of the beamforming weights can show how much the values are different, which could be a good evaluation of DLAF training. MSE is calculated between beamforming weights of the Capon and DLAF methods. Note again that DBF generates complex valued beamforming weights of 20 received beams for one transmit spoiled beam and every receivers. The MSE of complex valued beamforming weights was calculated in two different ways, one with imaginary and real parts and the other one with phase (radians) and amplitude, as follows,

$$MSE_{X,Y} = \sqrt{\sum_{i=1}^{N} \Delta X^2 + \sum_{i=1}^{N} \Delta Y^2}, \tag{5.1}$$

where ΔX is the difference between the Capon and DLAF beamforming weights in the real part, ΔY is the difference between the Capon and DLAF beamforming weights in the imaginary part, and N is the number of receivers. The other way uses amplitude and phase of beamforming weights, as follows,

$$MSE_{\theta,A} = \sqrt{\sum_{i=1}^{N} \Delta \theta^2 + \sum_{i=1}^{N} \Delta A^2}, \tag{5.2}$$

where $\Delta\theta$ is the difference between the Capon and DLAF beamforming weights in phase and ΔA is the difference between the Capon and DLAF beamforming weights in amplitude. Each MSE is plotted in Fig. 5.23 and Fig. 5.24. The MSE calculated for the point target area is in blue and MSE calculated for the weather is in red. The *x*-axis represents number of receive beams, which is 20, and the *y*-axis represents MSE values. Both figures show that MSE is smaller around the weather area than the point target area, which means the beamforming weights for the weather area are matched better than the point target area. This result makes sense since the shape of the beam patterns depends on the location of the targets. For the point target area, as long as it can minimize the power coming from the undesired direction, the sidelobes do not have to be matched. They shows high MSE values for 17^{th} to 20^{th} receiver. This could be related to the kernel size of the convolution

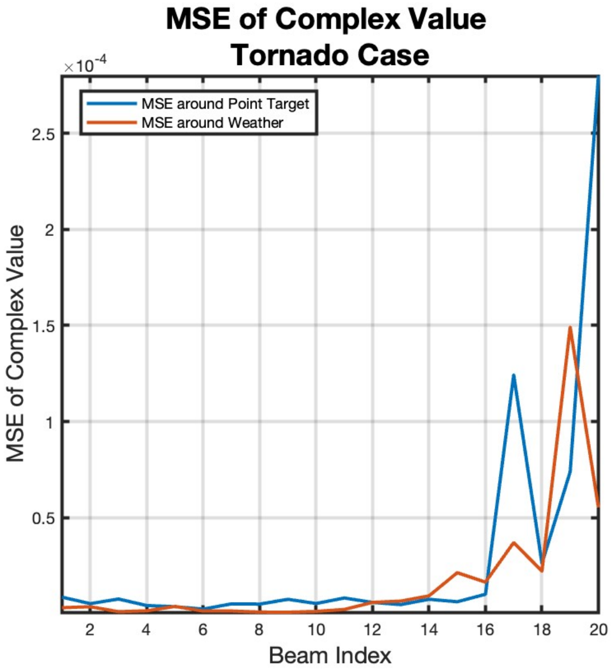


Figure 5.23: MSE between the Capon and DLAF beamforming weights in phase and amplitude for 32 antenna elements. The *x*-axis represents the number of receive beams and the *y*-axis represents the MSE value. The blue line represents the MSE for the point target area and the red line represents the MSE for the weather area.

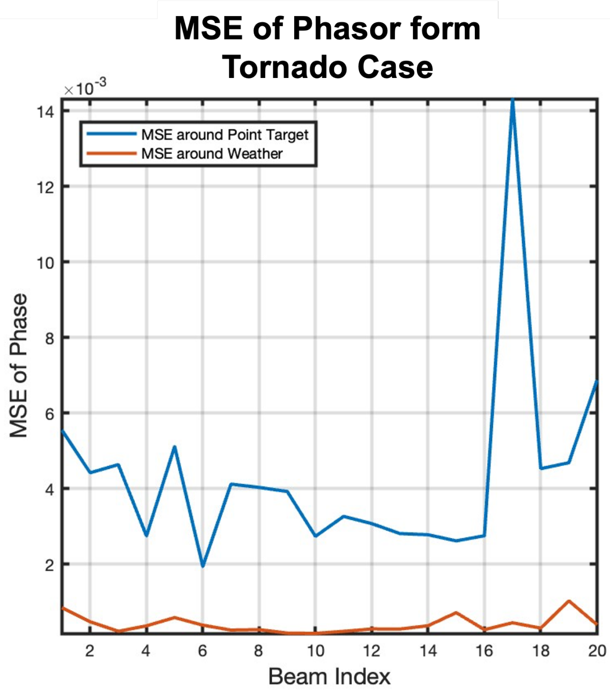


Figure 5.24: MSE between the Capon and DLAF beamforming weights in phase and amplitude for 32 antenna elements. The *x*-axis represents the number of receive beams and the *y*-axis represents the MSE value. The blue line represents the MSE for the point target area and the red line represents the MSE for the weather area.

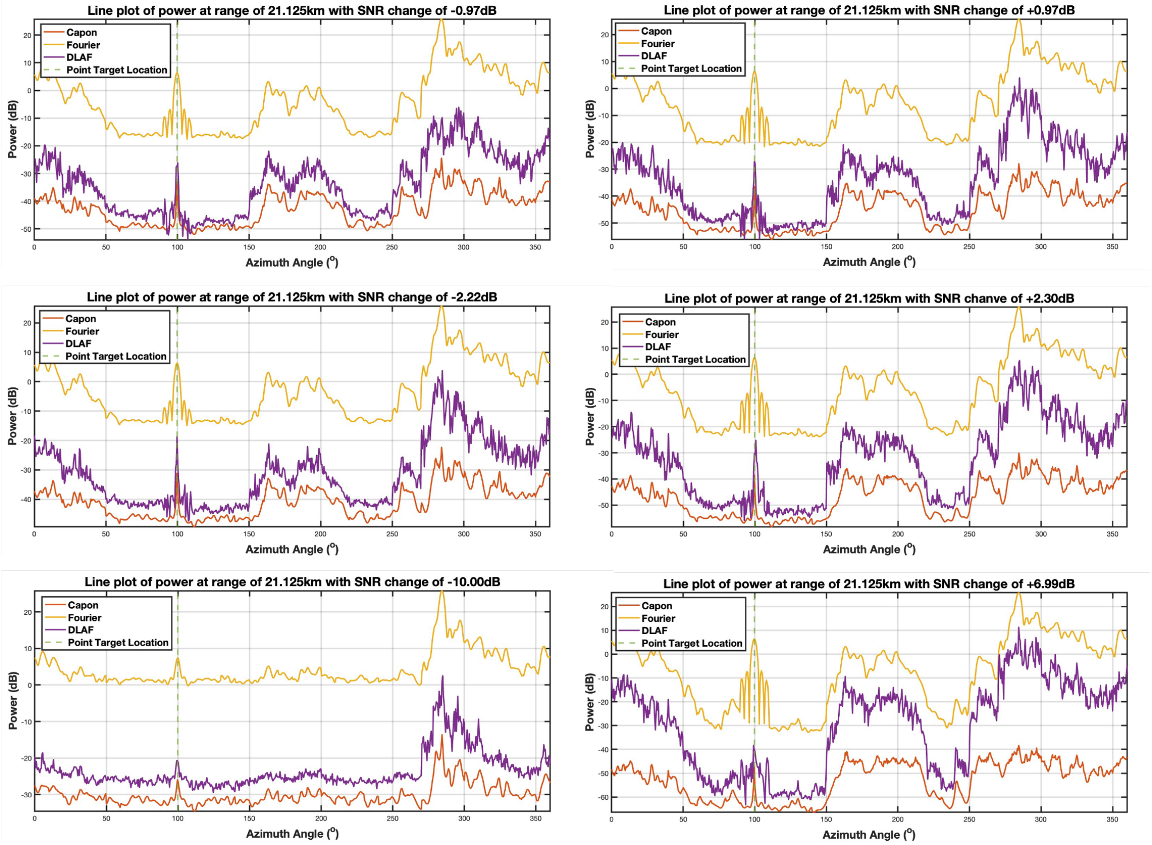
layers during the training. However, the overall MSE value is very small, the max value is approximately 3×10^{-4} from the complex values, and 14×10^{-3} from the phase values. Therefore MSE figures show that the beamforming values of the Capon and DLAF methods are very similar.

The impact of SNR on the DLAF performance is analyzed by testing different noise levels applied to the supercell case. The original SNR of the supercell case is 30.29dB at the point target area and 30.22dB around the hook echo area. Note that the noise power measured in the WSR-88D system is used in the simulation as described in Chapter. 3. Then SNR was changed to test the performance of DLAF with the changes of -10, -2.22, -0.97, +0.97, +2.30, and +6.99 (dB). The power by the Capon (orange), the Fourier (yellow), and DLAF (purple) methods at 21.125km (range) for 360° (azimuth) for different SNR is shown in Fig. 5.25. The power calculated by the three DBF methods with the original SNR is shown in Fig. 5.8. The location of the point target is 99.7475° and is indicated with a dotted green line in each plot. The *x*-axis represents the azimuth angle from 0° to 360° . The *y*-axis represents power in dB. DLAF is still able to suppress the high return power from the undesired direction around the point target, yet the overall power level is not match well as the original one. The antenna patterns by the Capon (blue), the Fourier (red), and DLAF (yellow) methods for different SNRs around the point target area (top row) and the weather area (bottom row) are shown in Fig. 5.26. The location of the point target is indicated with a dotted green line. The beam is steering at 99.2615° for the point target area and the beam is steering at 273.2602° for the weather area. The *x*-axis represents the azimuth angle from 1° to 360° . The *y*-axis represents the power of the pattern in dB. These plots shows DLAF can tolerate the change of SNR for the point target area. DLAF still places a null at the location of the interference and the sidelobe levels are well matched with the Capon patterns for all of the different SNR. However, the sidelobe levels of the pattern generated by DLAF are not matched with the Capon patterns when SNR is reduced by 10dB or increased by 6.99dB for the weather area. This can be dependent on how the targets are distributed within the transmit beam. As a result, it has been verified that SNR impacts the DLAF performance, yet DLAF has the tolerance for SNR changes of $\pm$10dB for the point target area and $\pm$2dB for the weather area.

The Pearson correlation coefficient is calculated with generated beam patterns by the Capon and DLAF methods as follows:

$$C_{norm} = \frac{(\omega_{DLAF_i} - \bar{\omega}_{DLAF})(\omega_{C_i} - \bar{\omega}_C)}{\sqrt{\sum(\omega_{DLAF_i} - \bar{\omega}_{DLAF})^2 \sum(\omega_{C_i} - \bar{\omega}_C)^2}}, \tag{5.3}$$

where $\bar{\omega}_{DLAF}$ and $\bar{\omega}_C$ are the mean values of generated beam pattern power by DLAF and the Capon methods. The Pearson correlation coefficient measures the linear correlation



Line plot of power at range of 21.125km with SNR change of -0.97dB

Line plot of power at range of 21.125km with SNR change of +0.97dB

Line plot of power at range of 21.125km with SNR change of -2.22dB

Line plot of power at range of 21.125km with SNR chanve of +2.30dB

Line plot of power at range of 21.125km with SNR change of -10.00dB

Line plot of power at range of 21.125km with SNR change of +6.99dB

Figure 5.25: SNR impact on the performance of DLAF is shown in this figure. Power by the Capon (red), the Fourier (yellow), and DLAF (purple) methods at different SNR are plotted in dB. The location of the point target is marked as a dotted green line. The *x*-axis represents the azimuth angle from 0° to 360° . The *y*-axis represents power in dB.

between two sets of data. In this case, it tells how much the DLAF patterns are similar to the Capon patterns. The correlation coefficient of 10 different beam patterns each for the point target area and the strong weather area (hook echo) are plotted in in Fig. 5.27 and Fig. 5.28. The steering angle of each case is listed in the top corner of the figure. Both figures show the maximum correlation coefficient value at angle shift 0. This means that the DLAF patterns are the most similar to the Capon patterns when there is no shift in angle. Furthermore, in Fig. 5.28, the maximum value of the normalized correlation coefficient at angle shift 0 is close to 1, which means that the DLAF pattern and the Capon pattern are almost identical when there is no angle shift. This result makes sense since the sidelobe

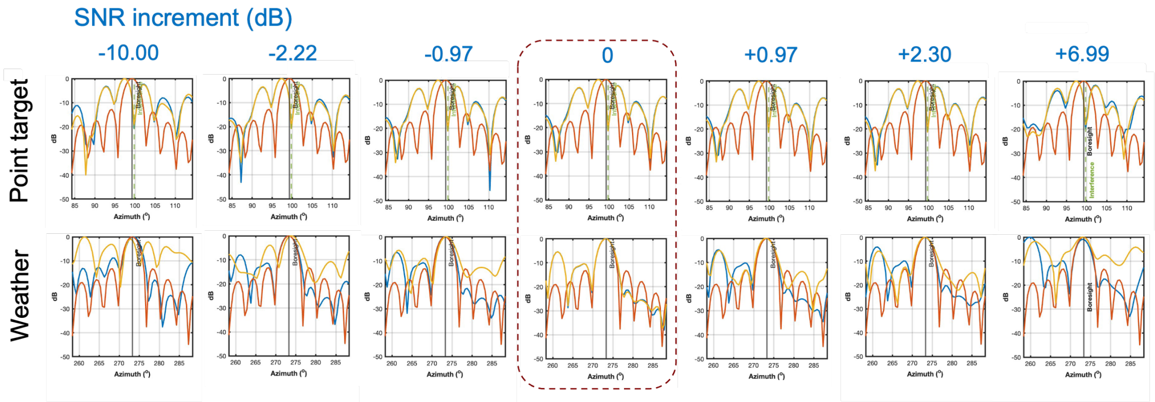


Figure 5.26: Antenna pattern of the Capon (blue), the Fourier (red), and DLAF (yellow) methods for the point target area (top row) and the weather area (bottom row) with different SNR. Each plot has a different SNR increment ratio, as indicated on the left side of the figure. The beam is steering at 99.2615° for the point target area and the beam is steering at 273.2602° for the weather area. The location of the point target is marked as a dotted green line. The *x*-axis represents the azimuth angle from 1° to 360° . The *y*-axis represents the power of the pattern in dB.

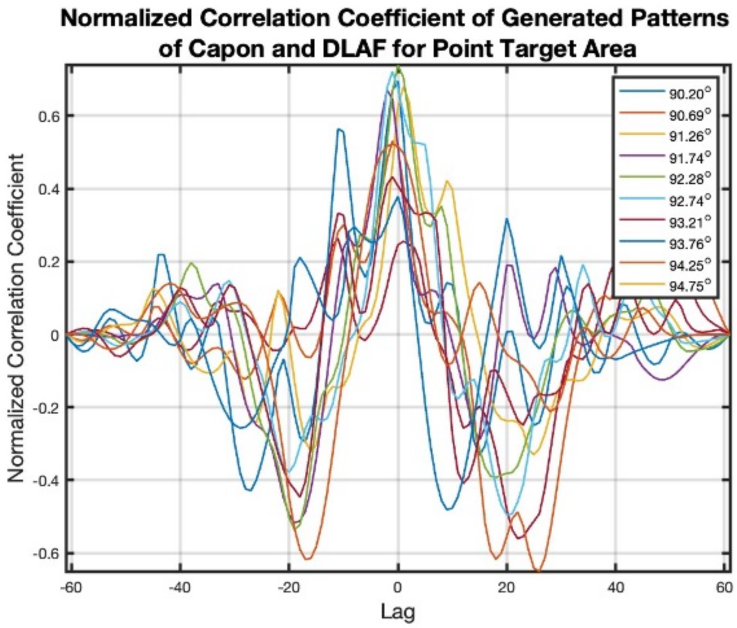


Figure 5.27: Pearson correlation coefficient of generated patterns by the Capon and DLAF methods for the point target area. Ten different patterns are used to calculate the correlation coefficient. The steering angle of each beam pattern is indicated in the top right corner. The *x*-axis shows angle shift and the *y*-axis shows the normalized correlation coefficient.

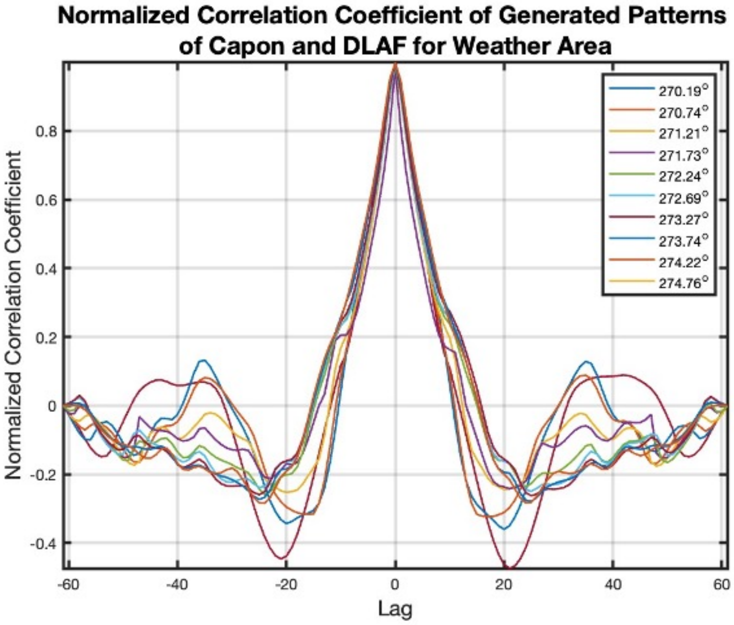


Figure 5.28: Pearson correlation coefficient of generated patterns by the Capon and DLAF methods for the weather area. Ten different patterns are used to calculate the Pearson correlation coefficient. The steering angle of each beam pattern is indicated in the top right corner. The *x*-axis is angle shift and the *y*-axis is the normalized correlation coefficient.

levels for the point target area can have more freedom as long as it can place a null at the interference direction, while sidelobe levels for the weather area, which has distributed targets, have to be matched well with the Capon.

Chapter 6

Conclusion and Future Work

6.1 Summary and Conclusion

This thesis started with a brief introduction on radar history and recent development in phased array radar technology. Parabolic-reflector radar was briefly discussed with its capabilities and limitations. Then, phased array radar and its unique capability, digital beamforming with two different methods was introduced. DBF can be done by either pre-computing the beamforming weights based on the configuration of the antenna or adaptively calculating the beamforming weights from received signals. The motivation of this thesis is to mitigate the computational burden of the adaptive DBF method. A potential solution based on DL was proposed.

AI has been widely adopted in various fields and developed into many different forms to resolve a variety of issues. ML, which is a subsection of AI, is a flexible and useful data analysis tool that can learn to perform tasks without human supervision or explicitly programmed. DL is a subsection of ML that is composed of multiple hidden layers and can learn complicated algorithms which can not be easily resolved by humans. Examples of ML use in different research fields and components of ML, such as neural networks, activation functions, and normalization were discussed with mathematical description in Chapter 2.

Fundamental radar equations for reflector antenna and phased array antenna were introduced in Chapter 3. The development of different PARs and their own characteristics were discussed. DBF, which is a unique capability of PAR that has the potential to improve the temporal resolution, relies on a spoiled transmit beam to digitally form multiple narrow beams simultaneously. The Fourier method is the conventional non-adaptive DBF method that predetermines the beamforming weights based on the antenna configuration. Since the Fourier method does not reflect the received signals, it results in fixed sidelobe levels and fixed angular resolution. The Capon method is one of the conventional adaptive DBF methods to calculate the beamforming weights with the received signals. It calculates the inverse of auto-covariance matrix, R^{-1}, with the size of $N \times N$. Therefore, the Capon method is able to place nulls adaptively to lower the sidelobes and minimize the power coming back from undesired directions caused by interference or ground clutter. However, it gets computationally expensive which could limit the real-time implementation. Real data of a moving bird collected by TEP was used to evaluate each DBF method. Furthermore, a moving point target based on TEP specifications and distributed targets (i.e., weather) based on Horus specification were simulated. The simulation of distributed targets has two parts. First the simulation of time-series IQ signals from the NEXRAD variables, then the simulation of multiple received IQ signals created by adding apropriate phase shifts using the array geometry.

To overcome the computational burden of the Capon method, a DNN called DLAF was proposed in Chapter 4. DLAF is a DNN that was trained with time-series IQ signals and the Capon beamforming weights. DLAF contains a combination of convolution layers and dense layers followed by activation and normalization layers. For training data, the complex values of time-series IQ signals and beamforming weights are transformed into phase and amplitude, instead of separating them into real and imaginary parts to preserve the phase information. Then, phase and amplitude are trained independently. Phase values are trained with tanh activation function while amplitude values are trained with ReLU activation function since phase contains negative values while amplitude contains only positive values. Then the network is trained with 200 realizations of time-series IQ signals for the point target case and 1800 realizations of time-series IQ signals for the weather case. The training performance was also evaluated for layer and batch normalization.

DLAF was tested with simulated point-target data, real data of a moving bird collected by TEP, and simulated weather data. The performance of DLAF was compared with the Capon and the Fourier methods in Chapter 5, using qualitative and quantitative metrics. The two qualitative metrics are the quality of the generated beam patterns and the capability of adaptive DBF. The four quantitative metrics are the computation time, the performance of DLAF as a function of SNR, mean square error of beamforming weights, and cross-correlation of generated beam patterns by the Capon method and DLAF.

To conclude the work, DLAF was trained successfully to perform adaptive DBF and mimic the behavior of the Capon method for point and distributed targets. Yet, DLAF was trained faster and produced better quality beam patterns for point target cases than distributed target cases. Also, DLAF was more robust at lower SNRs than higher SNR. The Pearson correlation coefficient of beam patterns generated by DLAF and the Capon method was used to analyze pattern similarity. DLAF produced well-matched main lobes and general sidelobe structure compared to the beam patterns generated by the Capon method. Then the computation time of the Capon method and DLAF are measured for different number of receivers, showing that DLAF was about 8 times faster than the Capon method at the size of Horus antenna ($N = 1024$).

6.2 Future Work

DL is a flexible and efficient data analysis tool that has great potential to learn complex algorithms. Using this advantage of DL, future efforts could involve expanding DLAF in many different directions. First, to enhance and improve the adaptive DBF quality, the network could be trained with multiple inputs such as the location of targets and interference, and the steering angle of the antenna. In this case, the network is trained to perform the adaptive DBF instead of just mimicking the Capon DBF method. Second, future efforts could involve expanding DLAF to include information related to the physical properties of the observed hydrometeors. Instead of only calculating beamforming weights, DLAF could be trained with more radar variables, such as correlation coefficient, differential reflectivity, or velocity. Lastly, DLAF could be also trained to predict the movement of the storm. ML is a flexible tool that could be used for many different tasks. Furthermore, ML is capable to learn complex algorithms with proper training, which could be a solution to many difficult problems in the real world.

References

[1] L. Brown, “A radar history of World War II,” *J. Am. Hist. Res*, 1999.

[2] D. S. Zrnic and A. V. Ryzhkov, “Polarimetry for weather surveillance radars,” *Bulletin of the American Meteorological Society*, vol. 80, no. 3, pp. 389–406, 1999.

[3] A. V. Ryzhkov and D. S. Zrnic, *Radar polarimetry for weather observations*. Springer, 2019, vol. 486.

[4] T. D. Crum and R. L. Alberty, “The WSR-88D and the WSR-88D operational support facility,” *Bulletin of the American Meteorological Society*, vol. 74, no. 9, pp. 1669–1688, 1993.

[5] M. J. Istok *et al.*, “WSR-88D dual polarization initial operational capabilities,” in *25th Conf. on Int. Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, Amer. Meteor. Soc. Phoenix, AZ, 2009.

[6] M. R. Kumjian, “Principles and Applications of Dual-Polarization Weather Radar. Part I: Description of the Polarimetric Radar Variables.,” *Journal of Operational Meteorology*, vol. 1, 2013.

[7] H. B. Bluestein *et al.*, “Radar in atmospheric sciences and related research: Current systems, emerging technology, and future needs,” *Bulletin of the American Meteorological Society*, vol. 95, no. 12, pp. 1850–1861, 2014.

[8] R. D. Palmer *et al.*, “A Primer on Phased Array Radar Technology for the Atmospheric Sciences,” *Bulletin of the American Meteorological Society*, 2022, Under review.

[9] D. Zrnic *et al.*, “Agile-beam phased array radar for weather observations,” *Bulletin of the American Meteorological Society*, vol. 88, no. 11, pp. 1753–1766, 2007.

[10] T.-Y. Yu, M. B. Orescanin, C. D. Curtis, D. S. Zrnić, and D. E. Forsyth, “Beam multiplexing using the phased-array weather radar,” *Journal of Atmospheric and Oceanic Technology*, vol. 24, no. 4, pp. 616–626, 2007.

[11] R. D. Palmer, C. J. Fulton, J. Salazar, H. Sigmarsson, and M. Yeary, “The “Horus” radar—an all-digital polarimetric phased array radar for multi-mission surveillance,” in *99th American Meteorological Society Annual Meeting*, AMS, 2019.

[12] R. Palmer *et al.*, “Horus – A Polarimetric Digital Phased Array Weather Radar Developed at the University of Oklahoma,” in *2022 IEEE International Symposium on Phased Array Systems and Technology*, 2022.

[13] C. Fulton, N. Goodman, M. Yeary, R. Palmer, H. Sigmarsson, and J. McDaniel, “Preliminary System Integration and Performance Features for an S-Band, Dual-Polarized, All-Digital Phased Array Radar,” in *2022 IEEE/MTT-S International Microwave Symposium-IMS 2022*, IEEE, 2022, pp. 862–864.

[14] K. Hondl and M. Weber, “NOAA’s Meteorological Phased Array Radar Research Program,” in *2019 IEEE International Symposium on Phased Array System & Technology (PAST)*, IEEE, 2019, pp. 1–6.

[15] D. Schvartzman, “Signal Processing Techniques and Concept of Operations for Polarimetric Rotating Phased Array Radar,” Ph.D. dissertation, The University of Oklahoma, Norman, OK, USA, 2020.

[16] B. Isom *et al.*, “The atmospheric imaging radar: Simultaneous volumetric observations using a phased array weather radar,” *Journal of Atmospheric and Oceanic Technology*, vol. 30, no. 4, pp. 655–675, 2013.

[17] D. Schvartzman, S. M. Torres, and T.-Y. Yu, “Distributed beams: Concept of operations for polarimetric rotating phased array radar,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 11, pp. 9173–9191, 2021. DOI: 10.1109/TGRS.2020.3047090.

[18] W. K. Hocking, J. Röttger, R. D. Palmer, T. Sato, and P. B. Chilson, *Atmospheric radar: Application and science of MST radars in the Earth’s mesosphere, stratosphere, troposphere, and weakly ionized regions*. Cambridge University Press, 2016.

[19] S. Drabowitch, A. Papiernik, H. Griffiths, J. Encinas, and B. L. Smith, *Modern antennas*. Springer Science & Business Media, 2010.

[20] R. J. Mailloux, *Phased array antenna handbook*. Artech house, 2017.

[21] T. Adachi and W. Mashiko, “High temporal-spatial resolution observation of tornadogenesis in a shallow supercell associated with Typhoon Hagibis (2019) us-

ing phased array weather radar," *Geophysical Research Letters*, vol. 47, no. 19, e2020GL089635, 2020.

[22] M. I. Skolnik, *Radar handbook*. McGraw-Hill Education, 2008.

[23] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[24] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[25] R. D. Palmer, S. Gopalam, T.-Y. Yu, and S. Fukao, "Coherent radar imaging using Capon's method," *Radio Science*, vol. 33, no. 6, pp. 1585–1598, 1998.

[26] T.-Y. Yu, R. D. Palmer, and D. L. Hysell, "A simulation study of coherent radar imaging," *Radio Science*, vol. 35, no. 5, pp. 1129–1141, 2000.

[27] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "An overview of machine learning," *Machine learning*, pp. 3–23, 1983.

[28] A. McGovern *et al.*, "Making the black box more transparent: Understanding the physical implications of machine learning," *Bulletin of the American Meteorological Society*, vol. 100, no. 11, pp. 2175–2199, 2019.

[29] Y. LeCun, Y. Bengio, G. Hinton, *et al.*, "Deep learning. nature, 521 (7553), 436-444," *Google Scholar Google Scholar Cross Ref Cross Ref*, 2015.

[30] T. Sallam, A. B. Abdel-Rahman, M. Alghoniemy, Z. Kawasaki, and T. Ushio, "A neural-network-based beamformer for phased array weather radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5095–5104, 2016.

[31] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[32] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," in *Bio-inspired neurocomputing*, Springer, 2021, pp. 203–224.

[33] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, 2020, pp. 281–286.

[34] A. Burkov, *The hundred-page machine learning book*. Andriy Burkov Quebec City, QC, Canada, 2019, vol. 1.

[35] M. Wiering, J. v. Veenen, J. Vreeken, A. Koopman, *et al.*, "Intelligent traffic light control," 2004.

[36] R. Jain, R. Kasturi, B. G. Schunck, *et al.*, *Machine vision*. McGraw-hill New York, 1995, vol. 5.

[37] D. R. Reddy, "Speech recognition by machine: A review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 501–531, 1976.

[38] D. S. Prerau, "Selection of an appropriate domain for an expert system," *AI magazine*, vol. 6, no. 2, pp. 26–26, 1985.

[39] R. R. Murphy, *Introduction to AI robotics*. MIT press, 2019.

[40] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[41] S. Ray, “A quick review of machine learning algorithms,” in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, IEEE, 2019, pp. 35–39.

[42] I. Castiglioni *et al.*, “AI applications to medical images: From machine learning to deep learning,” *Physica Medica*, vol. 83, pp. 9–24, 2021.

[43] D. W. Marquardt and R. D. Snee, “Ridge regression in practice,” *The American Statistician*, vol. 29, no. 1, pp. 3–20, 1975.

[44] P. Jatau, V. Melnikov, and T.-Y. Yu, “A Machine Learning Approach for Classifying Bird and Insect Radar Echoes with S-Band Polarimetric Weather Radar,” *Journal of Atmospheric and Oceanic Technology*, vol. 38, no. 10, pp. 1797–1812, 2021.

[45] D. Schvartzman, S. Torres, and T.-Y. Yu, “Weather radar spatiotemporal saliency: A first look at an information theory–based human attention model adapted to reflectivity images,” *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 1, pp. 137–152, 2017.

[46] A. Thakur and A. Konde, “Fundamentals of neural networks,” *Volume*, vol. 9, pp. 407–426, 2021.

[47] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Fundamentals of Artificial Neural Networks and Deep Learning," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer, 2022, pp. 379–425.

[48] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[49] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[50] B. Luijten *et al.*, "Adaptive ultrasound beamforming using deep learning," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 3967–3978, 2020.

[51] D. L. Elliott, "A better activation function for artificial neural networks," Tech. Rep., 1993.

[52] H. Pratiwi *et al.*, "Sigmoid activation function in selecting the best model of artificial neural networks," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1471, 2020, p. 012 010.

[53] K. Sun, J. Yu, L. Zhang, and Z. Dong, "A convolutional neural network model based on improved Softplus activation function," in *International Conference on Applications and Techniques in Cyber Security and Intelligence*, Springer, 2019, pp. 1326–1335.

[54] A. V. Savchenko, “Probabilistic neural network with complex exponential activation functions in image recognition,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 651–660, 2019.

[55] J. Sola and J. Sevilla, “Importance of input data normalization for the application of neural networks to complex industrial problems,” *IEEE Transactions on nuclear science*, vol. 44, no. 3, pp. 1464–1468, 1997.

[56] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[57] N. K. Manaswi, “Understanding and working with Keras,” in *Deep Learning with Applications Using Python*, Springer, 2018, pp. 31–43.

[58] N. Ketkar, “Introduction to keras,” in *Deep learning with Python*, Springer, 2017, pp. 97–111.

[59] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?” *Advances in neural information processing systems*, vol. 31, 2018.

[60] J. Lei Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *ArXiv e-prints*, arXiv–1607, 2016.

[61] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[62] F. Fabry, *Radar meteorology: principles and practice*. Cambridge University Press, 2018.

[63] R. J. Doviak *et al.*, *Doppler radar and weather observations*. Courier Corporation, 2006.

[64] M. A. Richards, *Fundamentals of radar signal processing*. McGraw-Hill Education, 2014.

[65] V. N. Bringi and V. Chandrasekar, *Polarimetric Doppler weather radar: principles and applications*. Cambridge university press, 2001.

[66] P. Meischner, *Weather radar: principles and advanced applications*. Springer Science & Business Media, 2005.

[67] V. M. Melnikov and D. S. Zrnić, “”Autocorrelation and Cross-Correlation Estimators of Polarimetric Variables”,” *Journal of Atmospheric and Oceanic Technology*, vol. 24, no. 8, pp. 1337–1350, 2007. DOI: 10.1175/JTECH2054.1.

[68] J. E. Stailey and K. D. Hondl, “Multifunction phased array radar for aircraft and weather surveillance,” *Proceedings of the IEEE*, vol. 104, no. 3, pp. 649–659, 2016.

[69] I. Ivić *et al.*, “An Overview of Weather Calibration for the Advanced Technology Demonstrator,” in *2019 IEEE International Symposium on Phased Array Systems and Technology (PAST)*, 2019, pp. 1–7. DOI: 10.1109/PAST43306.2019.9021053.

[70] J. B. Mead *et al.*, “A volume-imaging radar wind profiler for atmospheric boundary layer turbulence studies,” *Journal of Atmospheric and Oceanic Technology*, vol. 15, no. 4, pp. 849–859, 1998.

[71] R. D. Palmer, C. J. Fulton, J. Salazar, H. Sigmarsson, and M. Yeary, “The “Horus” radar—an all-digital polarimetric phased array radar for multi-mission surveillance,” in *99th American Meteorological Society Annual Meeting*, AMS, 2019.

[72] J. L. Salazar *et al.*, “An ultra-fast scan c-band polarimetric atmospheric imaging radar (pair),” in *2019 IEEE International Symposium on Phased Array System & Technology (PAST)*, IEEE, 2019, pp. 1–5.

[73] D. Schvartzman, J. D. D. Díaz, J. L. Salazar-Cerreño, T.-Y. Yu, R. D. Palmer, and M. S. McCord, “A Hybrid Antenna Pattern Synthesis Method for the Polarimetric Atmospheric Imaging Radar (PAIR),” in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 01–06. DOI: 10.1109/RadarConf2248738.2022.9764359.

[74] J. D. Díaz Díaz, D. Schvartzman, J. L. Salazar-Cerreño, T.-Y. Yu, R. D. Palmer, and M. S. McCord, “Polarimetric Atmospheric Imaging Radar (PAIR): Antenna Design, Testing, and Validation,” in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 1–5. DOI: 10.1109/RadarConf2248738.2022.9763907.

[75] R. Palmer *et al.*, “Horus – A polarimetric digital phased array weather radar developed at the University of Oklahoma,” in *2022 IEEE International Symposium on Phased Array Systems and Technology*, 2022.

[76] D. Schvartzman, R. Palmer, T.-Y. Yu, R. Reinke, and F. Nai, “A Pattern Synthesis Method for Polarimetric Weather Observations with the All-Digital Horus Phased Array Radar,” in *102nd American Meteorological Society Annual Meeting*, AMS, 2022.

[77] D. Zrnic, “Simulation of weatherlike Doppler spectra and signals.,” 1975.

[78] D. Schvartzman and C. D. Curtis, “Signal processing and radar characteristics (SPARC) simulator: A flexible dual-polarization weather-radar signal simulation framework based on preexisting radar-variable data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 1, pp. 135–150, 2018.

[79] B. L. Cheong, M. Hoffman, R. Palmer, S. J. Frasier, and F. López-Dekker, “Phased-array design for biological clutter rejection: Simulation and experimental validation,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 4, pp. 585–598, 2006.

[80] D. G. Luenberger, Y. Ye, *et al.*, *Linear and nonlinear programming*. Springer, 1984, vol. 2.

[81] D.-Z. Du and K.-I. Ko, *Theory of computational complexity*. John Wiley & Sons, 2011, vol. 58.

[82] F. Chollet *et al.*, *Keras*, https://github.com/fchollet/keras, 2015.

[83] M. A. Kon and L. Plaskota, “Information complexity of neural networks,” *Neural Networks*, vol. 13, no. 3, pp. 365–375, 2000.

Appendix A

Acronyms

ABLE Adaptive beamforming by deep learning

AI Artificial Intelligent

DAS Delay-and-sum

DBF Digital Beamforming

DL Deep Learning

DLAF Deep learning for adaptive digital beamforming

DNN Deep neural network

EBMV Eigen-based minimum variance

ELU Exponential linear unit

EW East-west

H Horizontal polarization

ML Machine Learning

MPAR Multi-function phased array radar

MSE Mean square error

NEXRAD Next generation weather radar

NS North-south

NWRT National weather radar testbed

PAIR C-band mobile parametric atmospheric imaging radar

PAR Phased Array Radar

PAWR X-band phased array weather radar

PPI Plane-position indicator

RBFNN Radial-basis-function neural network

RI range interval

RNN Recurrent neural network

RX Receiver

SGD Stochastic gradient descent

SNR Signal-to-Noise ratio

tahn Hyperbolic tangent

TEP Turbulent eddy profiler

TX Transmitter

V Vertical polarization

WR-STS Weather radar spatiotemporal

WSR-88D Weather Surveillance Radar - 1988 Doppler