Verification and Calibration of the ICEBEAR Radar through GPU Acceleration, Noise Characterization and Calculation, and Radio Galaxy Phase Calibration

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Physics & Engineering Physics
University of Saskatchewan
Saskatoon

By
Draven Galeschuk

©Copyright Draven Galeschuk, March 2021. All rights reserved.
Unless otherwise noted, copyright of the material in this thesis belongs to the author

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Physics and Engineering Physics
163 Physics Building
116 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5E2 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

The research performed for this thesis focused on verifying, quantifying, calibrating, and improving the Ionospheric Continuous Wave (CW) E-region Bi-static Radar (ICEBEAR) data observations and quality. Graphical processing unit (GPU) acceleration was used to improve the computation speed of ICEBEAR data analysis. The ICEBEAR noise floor was studied to better understand the ICEBEAR noise environment and verify the signal to noise ratio (SNR), which affects all ICEBEAR data products. Finally, a calibration method using the radio galaxy Cygnus A was developed to enable improved phase calibration of the ICEBEAR receiver antennas.

GPUs enable high computational throughput through the use of parallel processing and specific hardware design. This part of my research used the properties of GPUs to accelerate the data analysis of ICEBEAR to be 48 times faster than the original processing capability, enabling real-time analysis of ICEBEAR data.

The current noise calculation technique of taking the median power calculation of the ICEBEAR field of view is reasonable, but it is recommended that ICEBEAR switch to using an average of the furthest ranges measured by the radar. The dominant noise sources in the radar changes based on ionospheric activity, where self-clutter dominates during active periods and cosmic noise dominates during quite periods. This impacts the computation of the SNR data product and is better quantified by a far range average for all 45 baselines in the ICEBEAR radar.

The detection of Cygnus A during quiet ionospheric periods was used to calculate phase self-calibrations for the radar by comparing the measured phase difference between antennas to the expected theoretical phase difference of Cygnus A. The technique is shown to generate similar and complementary results to the current spectrum analyzer calibration technique. Future improvements to ICEBEAR imaging analysis and future research into the improved observation of Cygnus A will allow this new phase self-calibration method to be actively used for ICEBEAR.

Acknowledgements

First and foremost I would like to thank my supervisor Dr. Glenn Hussey, for sparking my interest in space physics and giving me the opportunity to study in this field. His support and aid in this research cannot be overstated.

I would also like to thank the rest of the ICEBEAR team, Devin Huyghebaert and Adam Lozinsky. They have helped better my understanding of ICEBEAR, acted as a sounding board for my ideas, and have been instrumental in the completion of this thesis research.

The rest of the faulty and students in the ISAS department also deserves my thanks and gratitude, especially Kieth Kotyk, Kevin Krieger, Marci Detwiller, and Marina Schmidt who have all helped me better understand radars and advance my research.

My professors, especially Dr. Kathryn McWilliams, and Dr. Sasha Koustov. They have all helped me learn and grow in this field, granted me opportunities to participate in conferences and programs that were truly fascinating and impactful, and have motivated me throughout my research.

The organizations and programs that have provided me opportunities and funding throughout my research; NSERC Create, NSERC Discovery, CIF, the Province of Saskatchewan, and the University of Saskatchewan. Without them this research would not have been possible.

I must of course thank my parents Rhea and Carey, and my friends Bailey Culbertson, Shania Stewart, Kevin Krantz, Lavie Nguyen, and Ashanthi Maxworth. Without their support and advice, providing their insight to my research and writing, I could never have completed this work.

Finally, I would like to thank the rest of my friends and loving family. They have supported and encouraged me through every stage of this research, taking the time to listen to my trails and successes and showing genuine interest in my work. I am forever grateful for their support.

Contents

Permission to Use **i**

Abstract **ii**

Acknowledgements **iii**

Contents **iv**

List of Tables **vii**

List of Figures **viii**

List of Abbreviations **xiv**

1 Introduction **1**
- 1.1 The Ionosphere . . . 4
- 1.2 The E-region . . . 6
  - 1.2.1 Instability Formation . . . 6
  - 1.2.2 Coherent Backscatter . . . 7
- 1.3 The ICEBEAR instrument . . . 9
- 1.4 Thesis Outline . . . 12

2 ICEBEAR System and Data Processing **15**
- 2.1 Introduction to ICEBEAR . . . 15
- 2.2 ICEBEAR Hardware . . . 18
  - 2.2.1 Antenna Layout . . . 18
  - 2.2.2 Transmitter System . . . 19
  - 2.2.3 Receiver System . . . 24
- 2.3 Data Processing . . . 26
  - 2.3.1 Interferometry . . . 26
  - 2.3.2 ICEBEAR Analysis . . . 28
- 2.4 Summary . . . 33

3 GPU Optimization **35**
- 3.1 Graphical Processing Units (GPUs) . . . 36
  - 3.1.1 GPU Structure . . . 36
  - 3.1.2 Parallel Processing . . . 37
- 3.2 The ICEBEAR GPU . . . 39
  - 3.2.1 GPU Selection . . . 39
  - 3.2.2 NVIDIA GeForce GTX 1080 Ti GPU Structure . . . 39
- 3.3 ICEBEAR GPU Optimization . . . 44

3.3.1 GPU Language . . . 44
3.3.2 Integration into Analysis . . . 46
3.4 Validation . . . 51
3.4.1 Processing Times . . . 51
3.4.2 C++ and CUDA Output Comparison . . . 53
3.5 Summary . . . 55

4 Background Noise in ICEBEAR Measurements **57**
4.1 Introduction . . . 57
4.2 Noise in Radar Systems . . . 58
4.3 Determining Noise . . . 60
4.3.1 Analysis Methodology . . . 62
4.4 Processing Results . . . 65
4.4.1 Mean Noise Value Comparison and Clutter Correction . . . 65
4.4.2 Diurnal Trends . . . 75
4.4.3 Baseline Dependence . . . 78
4.5 Summary . . . 83

5 Cygnus A Phase Calibration **85**
5.1 Introduction . . . 85
5.2 Cygnus A Detection . . . 86
5.2.1 The Noise Signature . . . 86
5.2.2 Tracking Radio Galaxies . . . 91
5.3 Identifying the Radio Galaxy Source . . . 94
5.3.1 Cygnus A . . . 95
5.4 Phase Calibration of ICEBEAR . . . 98
5.4.1 3D Interferometry . . . 98
5.4.2 Generating Phase Corrections with Cygnus A . . . 101
5.5 Calibration Results . . . 105
5.5.1 Comparison to Spectrum Analyzer Phase Correction . . . 107
5.5.2 Cygnus A Detection Issues . . . 111
5.6 Future Considerations . . . 116
5.7 Summary . . . 118

6 ICEBEAR Observations **120**
6.1 Introduction . . . 120
6.2 Instability Echos . . . 120
6.2.1 ICEBEAR-Linear Receiver Observations . . . 121
6.2.2 ICEBEAR-3D Observations . . . 121
6.3 Meteor Trail Echos . . . 124
6.3.1 ICEBEAR Observation and SuperDARN Comparison . . . 125
6.4 Summary . . . 128

7 Conclusion **129**
7.1 Future Work . . . 132

References **133**

Appendix A GPU Accelerated CUDA Code **137**
A.1 CUDA code 137
A.2 Python Wrapper 141

Appendix B Local Horizon Position Code **143**

List of Tables

2.1 Specifications for ICEBEAR after (Huyghebaert et al., 2019) 22

3.1 After (Nickolls & Dally, 2010). Comparison of CPU and GPU performance on 2010 hardware. Assumes CPU cores are 5x faster and 50x the area of a GPU core. 37

3.2 Comparison of computation time for a number of baselines. Processing 5 s of data and 8 hrs of data for ICEBEAR analysis for C++ code and GPU code. Times rounded to 2 decimal places. 52

4.1 Data sets used in examination of the ICEBEAR radar noise 65

4.2 Percent difference of noise power calculation methods compared against power calculated using the far range method. Results are presented for quiet and active ionospheric conditions, and for noise calculated before and after the clutter correction is applied. Demonstrates the similarity of the FoV median method to the other averaging methods, and that the median becomes more similar during active ionospheric conditions. 68

4.3 Percent difference of noise phase calculation techniques compared against phase calculated using the far range technique. Results are presented for quiet and active ionospheric conditions, and for noise calculated before and after the clutter correction is applied. This demonstrates the similarity of the FoV median technique to the other averaging techniques, and that the phase pattern is very slightly disturbed by active ionospheric conditions 74

5.1 Average difference in SNR, azimuth AOA, and elevation AOA between ICEBEAR imaging outputs using the Cygnus A calibration technique and the spectrum analyzer calibration technique. The difference in the AOAs is small and shows agreement between the to calibration techniques. Note that the SNR difference between techniques is due to floating point errors in the data processing. 109

6.1 Comparison of 3 meteor trail events measured by ICEBEAR and SuperDARN. Velocity, range, and heading all agree between the two radars. As the power measurements are not absolute they are different, but the relative power between the events is the same. 126

List of Figures

1.1 Typical vertical electron density profile of mid-latitude ionosphere after (Hunsucker & Hargreaves, 2003). The peak density of each ionospheric layer is labeled with the layer letter (D, E, F). 5

1.2 The four classic types of instabilities, with the typical normalized power distribution (y-axis) plotted against the Doppler frequency shift (x-axis). Here C_s is the ion acoustic speed. Type I has a narrow spectrum centered around C_s, Type II has a broad spectrum centered around zero Doppler velocity, Type III is similar to Type I but centered at a Doppler velocity smaller than C_s, and Type IV is again similar to Type I, but with a center Doppler velocity much greater than C_s (Hussey, 1994). 8

1.3 ICEBEAR map for 10 March 2018. Displays the transmitter FoV (green), the receiver FoV (blue), magnetic aspect angle (gray scale), RF propagation distance/2.0 (pink contours), magnetic inclination angle (black contours), transmitter and receiver site locations, and Doppler velocity (color scale) of detected scatter. The scatter depicted in this figure is example data that outlines the spatial resolution capabilities of the radar and instability structures that can be observed. 12

2.1 Block diagrams of the receiver and transmitter hardware, and the associated communication connections for the ICEBEAR radar after (Huyghebaert, 2019). 17

2.2 Schematic of the new ICEBEAR-3D configuration. The position of each antenna is stated relative to a single antenna (antenna 0), and the orientation of the array is shown as 7° East of North (Lozinsky, personal communication, December 20, 2020). 20

2.3 Satellite image of the ICEBEAR receiver site (Google, n.d.). The location of the 10 receiving antenna positions have been denoted on the image and labelled as they are identified in processing (Lozinsky, personal communication, December 20, 2020). 21

2.4 Relationship between two antennas separated by a distance, d, measuring the same source or scattered signal. Showing the extra distance traveled by the wave, δx, between reception on antenna 1 (S_1) and reception on antenna 2 (S_2). The AOA, θ, can be in azimuth and/or elevation depending on the orientation of the antennas and the position of the source at a distance, r, from the antennas. 27

2.5 SNR plot of RF Propagation distance/2.0 vs Doppler Frequency, with all echo types detected by ICEBEAR labeled for March 10, 2018 at 3:14:19 UT . . . 33

3.1 After (NVIDIA, 2019a). Block diagram comparison between CPU and GPU hardware layouts with equal area for distribution of components. Yellow blocks are control units which manage communication with external devices, execution of code, and management of memory within the device. Green blocks are Arithmetic Logic Units (ALUs) which are the base processing block. Orange blocks (cache, DRAM) are memory locations. Dynamic Random Access Memory (DRAM) is external memory that holds program code and data for operation. The cache stores local data from the DRAM for quick access within the processing unit. GPU structures assume less program control and data management and focus on increasing computational throughput by using more ALUs. 38
3.2 After (NVIDIA, 2009). Hierarchy of threads, blocks, and grids with corresponding memory access. Represents the programming structure of the NVIDIA GPU 41
3.3 After (NVIDIA, 2016). The SM structure used by the NVIDIA GeForce GTX 1080 Ti GPU based on Fermi architecture. Instructions are stored in the instruction cache, while data is stored in the shared memory or the texture/L1 cache. The warp scheduler and dispatch unit receive commands from the instruction buffer and direct commands to cores or SFUs that are available. Load/Store (LD/ST) units manage moving data between the processing cores and SFUs and the device memory. Tex units are used by the GPU to manipulate images. Each SM contains 128 cores, 256 KB of register file capacity, a 96 KB shared memory unit, 48 KB of total L1 cache storage, eight Tex units, 32 SFUs, and 32 LD/ST units. 42
3.4 After (NVIDIA, 2019a). Image (A) shows the structure of blocks and threads. The grid holds a 2 dimensional array of blocks, and each block contains a 2 dimensional set of threads, all of which are identified by an ID number. Image (B) shows the execution of a set of blocks on GPUs with different numbers of SMs. Fewer SMs in the GPU results in the same number of blocks taking longer to process. 45
3.5 Analysis process flowchart. Analysis set-up and reading in data is performed by Python, which then enters into a loop to process analysis intervals. The time averaged cross-correlations are computed in parallel on the GPU using CUDA. When called, CUDA copies data from the CPU to the GPU, mixes, filters, decimates, and stores the resulting matrix, returning it to the CPU. . 47
3.6 Sequential addressing procedure to decimate code in parallel. The starting stride value is half the array size, and reduces by half for each subsequent step until all data has been summed into the first array element. 49
3.7 Output of the NVIDIA profiler program depicting total execution time for one 5 s time average. Shows Memory Copy (MemCpy) time for Host to Device (HtoD) and Device to Host (DtoH). In this scenario, the device is the GPU and the host is the CPU. Each call of the kernel and run time is also displayed. This snap shot depicts the program flow within CUDA on the GPU. This evaluation does not take into account the Python to CUDA interaction time. Only the CUDA execution is profiled. 53

3.8 5 s average SNR measured by ICEBEAR on 17 March 2018, 3:00:00 UTC. SNR calculated for receiver antennas 1 and 2 using the original C++ analysis code (top), and the new CUDA code (bottom). Only small variations in the SNR noise appear between plots, so both codes are shown to give the same result. 54

4.1 Raw noise power (top) and noise power after clutter correction (bottom) calculated using multiple averaging methods and the FoV median, on 7−8 March 2018. This is a quiet period with a 1λ antenna separation for the linear ICEBEAR antenna layout configuration. All averaging methods show essentially identical noise values. The FoV median noise power is weaker but follows the same trend. The clutter correction reduces biasing effects of strong clutter in the noise, causing the overall trend in the noise floor to become more pronounced. 66

4.2 Raw noise power (top) and noise power after clutter correction (bottom) calculated using multiple averaging methods and the FoV median, on 10 March 2018. This is a 30 minute period of high ionospheric activity measured using the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. Before the clutter correction, the median and average powers match during the active period, with the FoV average trending very slightly larger than the other noise calculation methods. The FoV average is slightly more distinct from the other methods after the clutter correction is applied. 69

4.3 Raw noise phase (top) and noise phase after clutter correction (bottom) calculated using multiple averaging methods and a FoV median on 7−8 March 2018. This is the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The clutter correction suppresses the background phase variation from the noise sources. All noise averaging methods as well as the noise median method follow the exact same trend with comparable magnitudes. The clutter correction removes most of the phase bias. 71

4.4 Raw noise phase (top) and noise phase after clutter correction (bottom) calculated using multiple averaging methods and a FoV median on 10 March 2018. This is a 30 minute period of high ionospheric activity measured using the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The clutter correction, unlike in the quiet data, does not suppress the phase activity, but changes it due to the presence of scattering signals shifting the clutter correction based on how they are interpreted in the noise calculation technique. 72

4.5 Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median on 12 April 2018. This is a quiet period for the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The FoV median noise follows the average noise powers at a slightly lower power, while the phase of all methods is nearly identical. 76

4.6 Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median starting on 29 May 2020 3:00:00 UTC for 24 hours. This is a quiet period for the ICEBEAR-3D configuration with a 4λ antenna separation using antennas 0 and 3. The FoV median power and phase values follow the trend of the average noise methods, but are shifted down. Baselines larger than 1λ can have a phase offset for the FoV median not seen in the 1λ noise data due to the grating lobes. 79
4.7 Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median on 7−8 March 2018. This is a quiet period using the ICEBEAR-linear configuration with a 4λ antenna separation. As expected, the FoV median noise power follows the same trend as the average noise methods at a slightly lower power. Due to grating lobes in larger baselines and the phase sensitivity of the ICEBEAR instrument, the FoV median noise phase is greatly influenced by phase variations in noise sources. Nevertheless, the general phase trend in the average noise data can still be seen in the FoV median data. 80
4.8 1λ (top) and 4λ (bottom) baseline azimuth beam patterns generated using a NEC model of the ICEBEAR antennas (Voors, 2020). This demonstrates that baselines with a larger antenna spacing develop more grating lobes within the same FoV. 82

5.1 Distinct peak in the noise power indicated by the red rectangle, occurring in 21 January 2018, (top) and 12 April 2018, (bottom) using the ICEBEAR linear antenna layout configuration. The noise power (magnitude) is in dB for the noise signature from the cross-correlation between antennas 1 and 2, which have a separation of 6 m (or 1λ). 87
5.2 Oscillation of the noise phase indicated by the red rectangle, occurring in 21 January 2018, (top) and 12 April 2018, (bottom) using the ICEBEAR linear antenna layout configuration. The phase is in degrees for the noise signature from the cross-correlation between antennas 1 and 2, which have a separation of 6 m (or 1λ). 88
5.3 Distinct peaks in the noise power indicated by the red rectangles, occurring in 12 April 2018, using the ICEBEAR linear antenna layout configuration. The power is in dB for the noise signature from cross-correlation between antennas 1 and 2 (top) with a separation of 6 m (1λ), and between antennas 4 and 8 (bottom) with a separation of 24 m (4λ). Note that the peak from the 1λ baseline appears as three peaks in the 4λ baseline. 89
5.4 (Top) Horizontal cross-section of the 10 antenna ICEBEAR-3D receiver array beam pattern generated using the 4nec2 modeling software (Voors, 2020), North is at 90°. (Bottom) Azimuth angles of 4 radio galaxies with respect to the ICEBEAR receiver location over a 12 hour period on 21 January 2018, North is at 0°. Red regions of the azimuth indicate that the radio galaxy is below the horizon, black indicates above the horizon, and green indicates that the radio galaxy azimuth and zenith angles are within the ICEBEAR receiver beam pattern FoV. 96

5.5 (Top) Vertical cross-section of the 10 antenna ICEBEAR-3D receiver array beam pattern generated using the 4nec2 modeling software (Voors, 2020), horizon is at 90°. (Bottom) Zenith angles of 4 radio galaxies with respect to the ICEBEAR receiver location over a 12 hour period on 21 January 2018, horizon is at 0°. Red regions of the zenith indicate the radio galaxy is below the horizon, black indicated above the horizon, and green indicates that the radio galaxy azimuth and zenith angles are within the ICEBEAR receiver beam pattern FoV. 97
5.6 After Palmer et al., (1996). General coordinate system relating the positions of two antennas (i and j) and a stellar source (x, y, z) with respect to a frame of reference. The antenna vectors $\boldsymbol{D}$ represent the distance of the antennas from the origin, d is the distance between the two antennas, $\boldsymbol{n}$ is the unit vector pointing along vector $\boldsymbol{P}$ towards the stellar source, α is the elevation angle measured with respect to the zenith, β is the azimuth angle measured from North, and γ is the angle between North and the baseline. 99
5.7 Comparison of measured noise power on antenna baseline 2-7 and the corresponding theoretical beam pattern gain of the baseline along the Cygnus A path using data measured on 20 December 2019. The beam pattern has been normalized by 30 dBi and the noise power has been shifted down by the average power (∼42.5 dB) and then normalized by the max power after shifting (∼5 dB). Note that the power peaks and nulls in the measurements align with the bean pattern of the baseline along the Cygnus A path. 103
5.8 Noise power and phase measurements (blue) when Cygnus A was within the ICEBEAR FoV from 20 December 2019. The orange line is the data set smoothed by a running average. The running average was applied to the real and imaginary components, and then converted to power and phase. The overall behaviour of the data is preserved through the average while noise is reduced. 104
5.9 Measured phase (blue), corrected phase (orange), and theoretical phase (green) for baselines 4-9 (top) and 0-3 (bottom) using data from 20 November 2019. After the correction is applied, the data more closely aligns with the theoretical phase values. 106
5.10 FoV Doppler velocity distribution (aziumithal top-left and altitude top-right) and FoV SNR distribution (azimuthal bottom-left and altitude bottom-right) of measured E-region scatter by the ICEBEAR receiver between 5:00 and 6:00 UTC, 19 December 2019. Generated using Cygnus A phase corrections on each baseline. Baselines 1-5, 1-6, 2-5, 2-8, 4-5, 4-6, 5-6, 5-8, and 6-8 were not used in processing this data. 108
5.11 FoV Doppler velocity distribution (aziumithal top-left and altitude top-right) and FoV SNR distribution (azimuthal bottom-left and altitude bottom-right) of measured E-region scatter by the ICEBEAR receiver between 5:00 and 6:00 UTC, 2019, December 19. Generated using antenna phase corrections measured by a spectrum analyzer. Baselines 1-5, 1-6, 2-5, 2-8, 4-5, 4-6, 5-6, 5-8, and 6-8 were not used in processing this data to compare with Cygnus A phase correction results. 110

5.12 Comparison of measured phase difference (blue) and the theoretical phase difference (green) over an 8 hour period on 20 December 2019. The top plot is a responsive baseline (2-3) and the bottom plot is an unresponsive baseline (6-8) from the ICEBEAR-3D antenna layout configuration. Baseline 6-8 shows no notable pattern and therefore cannot be matched to the theoretical values, and corrections generated for baseline 6-8 will be meaningless. When there is a clear pattern in the phase response as in baseline 2-3, comparison of the measured and theoretical values will generate valid phase calibrations. . . . 113
5.13 Noise power measured over the same 8 hour period as in Figure 5.12 on 20 December 2019. The top plot is a responsive baseline (2-3), and the bottom plot is an unresponsive baseline (2-6) in the ICEBEAR-3D antenna layout configuration. The distance between the antennas used in the baseline d, is shown in meters. Baselines that have a higher average noise power display less response to the presence of Cygnus A. In this example, the more responsive baseline has a power level $\sim$10 dB lower than the unresponsive baseline. . . . 114
5.14 Each data point represents a baseline in the ICEBEAR radar assessed when Cygnus A was visible on either 20 November 2019 or 20 December 2019. There is a clear exponential decrease in the phase spread as the noise power increases, indicating a poorer response to the presence of Cygnus A 115
5.15 Cross-correlation in time of 6 s of raw data between antennas 4 and 8 taken around 2 UTC 21 January 2021. Cygnus A was not present at the time of this correlation analysis and no ionospheric scatter was occurring. This indicates that some noise source, such as the background cosmic noise or a terrestrial signal, was correlating across this baseline in the ICEBEAR data. 117

6.1 ICEBEAR data averaged over 5 seconds on 10 March 2018, at 3:14:19 UTC, mapped over the FoV using all 10 antennas of the ICEBEAR-linear receiver array. This plot depicts the Tx and Rx locations and FoVs, along with the magnetic inclination angle and Radio Frequency (RF) propagation distance. Doppler velocity of the measured signals is depicted using a color scale. The gray scale contour represents the magnetic aspect angle. 122
6.2 ICEBEAR data averaged over 5 seconds on 10 March 2018, at 3:14:19 UTC, depicted in a range-Doppler plot made using all 10 antennas (9 identical baselines) of the ICEBEAR-linear receiver array. The color scale represents the SNR measurements for each range-Doppler value. 123
6.3 Comparison ICEBEAR and SuperDARN measurements on 7 March 2018, at 16:09 UTC. The meteor trail detected by the two radar is identified by a red circle on the FoV plot. While the two maps do not use the same projection, examining the location of the trail with respect to near by lake features shows the signal is located in the same general location by both radars. For ICEBEAR a 0.5 s measurement of the FoV is shown, and for the Saskatoon SupderDARN a 1 minute scan of the FoV is shown when beam 3 was scanning at the same time as the ICEBEAR measurement. 127

List of Abbreviations

ALU	Algorithmic Logic Unit
AOA	Angle of Arrival
BPF	Band Pass Filter
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CW	Continuous Wave
EM	Electromagnetic
FFT	Fast Fourier Transform
FoV	Field of View
GPS	Global Positioning System
GPU	Graphical Processing Unit
GST	Greenwhich Sidereal Time
HDD	Hard Disk Drive
HDF5	Hierarchical Data Format 5
ICEBEAR	Ionospheric Continuous wave Bi-static E-region Auroral Radar
IQ	In-phase and Quadrature
LD/ST	Load/Store
LNA	Low Noise Amplifier
LST	Local Sidereal Time
MU	Middle and Upper atmosphere
NEC	Numerical Electromagnetics Code
PDU	Power Distribution Unit
PRN	Pseudo-Random Noise
PSF	Point Spread Function
PSK	Phase Shift Keying
RADAR	Radio Detection And Ranging
RF	Radio Frequency
Rx	Receiver
SDR	Software Defined Radio
SM	Streaming Multiprocessor
SNR	Signal to Noise Ratio
SuperDARN	Super Dual Auroral Radar Network
Tx	Transmitter
UT	Universal Time
UTC	Coordinate Universal Time
VHF	Very High Frequency

1 Introduction

The Ionospheric Continuous Wave (CW) E-region Bi-static Experimental Auroral Radar (ICEBEAR) is a new coherent scatter radar designed to study the high-latitude E-region of the ionosphere (Huyghebaert et al., 2019). The ionosphere is region of the atmosphere where photons and ionizing particles from the sun and magnetosphere of the Earth ionize the atoms and molecules of the atmosphere, causing a plasma to from. The high-latitude E-region is a dynamic section of this plasma layer that develops plasma structures driven by activity in the magnetosphere of the Earth and solar activity. The term space weather is used to define sun-magnetosphere-ionosphere-Earth interactions and conditions in the region of space close to Earth that can affect technology and human interactions. Space weather impacts many technological systems on Earth. Some examples are communications systems like cell-phones, radio stations, satellites, as well as defense and weather radar, all of which are systems that impact our daily lives (Schrijver et al., 2015). Studying plasma activity in the E-region provides insights into how solar activity affects this layer of the ionosphere. This develops further understanding of how the ionosphere interacts with the magnetosphere of the Earth, and how the magnetosphere interacts with the Sun. Improving our understanding of space weather enables better modeling of the space-Earth environment and prediction of how space weather activity will affect technology.

The ICEBEAR radar will significantly contribute to E-region research due to the high temporal and spatial resolution of this radar (Huyghebaert, 2019). High temporal resolution allows for detailed observation of how plasma structures develop, evolve and move over time, and high spatial resolution provides accurate mapping and determination of the size and location of plasma structures. The ability to have both high temporal and spatial resolution simultaneously is a new development in radar systems made possible through the use of modern

hardware and software defined radio (SDR) techniques (Huyghebaert, 2019). ICEBEAR is the first radar of its kind and promises to greatly advance the science and understanding of E-region physics and its roll in space weather. The research presented in this thesis is focused on verifying, quantifying, calibrating, and improving ICEBEAR capabilities to ensure the best performance of the instrument. Three main topics were the focus of this research:

1. Improving the ICEBEAR analysis computation time
2. Quantifying the noise measured by the radar and ensuring the validity of the technique used to determine the noise
3. Creating a phase calibration technique for the ICEBEAR receiver using the detection of a radio galaxy in the ICEBEAR noise data

The high resolution capability of ICEBEAR results in large amounts of data generated by the radar. One antenna can collect 2.8 GB of data per hour, and since ICEBEAR has 10 receiving antennas it generates 28 GB/hour. Current operation of the radar collects data for 14 hours a day and fills a 12 TB hard drive in a little under a month. The analysis required to process ICEBEAR data is computationally intensive, and the original processing code was unable to process the data in real-time. Acceleration of the ICEBEAR analysis computation speed was possible through the implementation of parallel processing on a graphical processing unit (GPU). Improvement of this computation time enabled real-time analysis of ICEBEAR data.

As ICEBEAR is a new system, it is important to ensure that the noise in the measured data (effects from radio signals that are not from the desired target) is quantified and well understood. Initially selected mostly for computational efficiency, the technique ICEBEAR currently uses to calculate the noise in the radar is to take the median value of an intermediate analysis step where the power is calculated across the entire ICEBEAR field of view (FoV) (the median value of what will be defined as range-Doppler matrices in Chapter 2). It was of interest to verify that this method of noise calculation was reasonable for ICEBEAR and that it generates noise values that are representative of noise in the radar. Alternate techniques of noise calculation, such as averages in regions of the FoV where only noise is expected to be measured, provide another means of noise determination to compare against the median

calculation. These other methods have been shown by this research to be better suited for calculating the noise in ICEBEAR measurements. Also of interest to this research is characterizing the typical noise measured by the radar and identifying the dominant sources of this noise. Depending on the ionospheric activity and time of year different sources of noise may be dominant in radar measurements, which could impact the calculation of the signal to noise ratio (SNR) data product; therefore, it is important to quantify trends and typical values in the ICEBEAR noise.

Finally, it is observed that the radio galaxy Cygnus A is detectable in the noise data of ICEBEAR. The presence of Cygnus A provides the opportunity to create a phase self-calibration technique that can be generated any time the radio galaxy is observed by the radar. It is important to ensure that the radar is properly calibrated if the measurements made by the radar are to be meaningful. Phase measurements are used to determine the angle of arrival (AOA) of signals, which is critical for identifying the spatial extent and location of plasma structures. The ten antenna ICEBEAR receiving antenna array is made up of antenna pairs (45 unique pairs). Each antenna pair is like a Young's two-slit experiment for measuring AOA. All these pairs must be calibrated to the proper phase within $\sim \pm 1^\circ$ for ICEBEAR to properly determine AOA. If not properly calibrated, the AOAs determined from the phases of the antenna pair will be random, resulting in structures being fragmented and scatted throughout the FoV. These measurements would loose all physical meaning and could not be used for scientific study. Phase calibrations can be generated for ICEBEAR using a spectrum analyzer instrument which determines the phase error by sending signals through the radar components and measuring the response. However, this method requires the spectrum analyzer to be physically taken to the radar site, so calibrations are made infrequently. As well, small segments of the radar system cannot be measured using the spectrum analyzer, therefore there is a small chance of introducing phase error which has not been accounted for. Using Cygnus A to perform a self-calibration of the radar allows for more frequent calibrations to ensure proper ICEBEAR performance, and can also be used to verify and complement the spectrum analyzer measurements.

1.1 The Ionosphere

The ionosphere of the Earth is formed by photons and high energy particles from the Sun ionizing the neutral atmosphere, creating a conductive quasi-neutral plasma containing free electrons and positive ions within the atmospheric layer. Ionizing photons and particles overcome the ionizing potential of outer electrons in atomic or molecular constituents in the atmosphere, creating an ion-electron pair. In general ion and electron generation is dependent on the altitude, the intensity of radiation, the density of the neutral atmosphere, and the atmospheric constituents (Kivelson & Russell, 1995). Loss of ions and electrons from the plasma is primarily through recombination back into neutral particles. Motion of the plasma is influenced by electromagnetic (EM) forces as well as collisions with the neutral atmospheric constituents. An important property of a plasma is the plasma frequency, ω_p. This value represents the natural oscillation frequency of the electrons and ions in the plasma as a result of charge density perturbations. It can be expressed as (Kivelson & Russell, 1995),

$$\omega_{ps} = \left(n_s e^2/\epsilon_0 m_s\right)^{\frac{1}{2}} \tag{1.1}$$

where the subscript s is used to denote either electrons or ions, n_s is the electron or ion density, m_s is the electron or ion mass, ϵ_0 is the permittivity of free space, and e is the charge of an electron. It is important to note that not all of the particles in a gas need to be ionized for the formation of a plasma, as long as the density of neutral particles within the gas is sufficiently small such that collisions with the charged particles occurs infrequently a plasma can form.

The ionosphere is separated into 4 distinct regions, these are listed here with their daytime properties (Hunsucker & Hargreaves, 2003),

- D-region, 60-90 km: electron density $10^2 - 10^4$ cm^{-3}
- E-region, 90-160 km: electron density 10^5 cm^{-3}
- F1-region, 160-180 km: electron density $10^5 - 10^6$ cm^{-3}

- F2-region, maximum height varies around 300 km: electron density 10^6 cm^{-3}

These regions are denoted by the electron densities in each, and are separated by distinct peaks in the electron densities. The density of these regions decreases at nighttime as the sun is no longer present for solar ionization. The typical electron density profile of the ionosphere is shown in Figure 1.1. The primary ions in each region differs as the atmospheric composition changes with altitude. The atmospheric density, the primary ions formed, and the recombination rates help define each of the ionospheric regions (Hunsucker & Hargreaves, 2003).

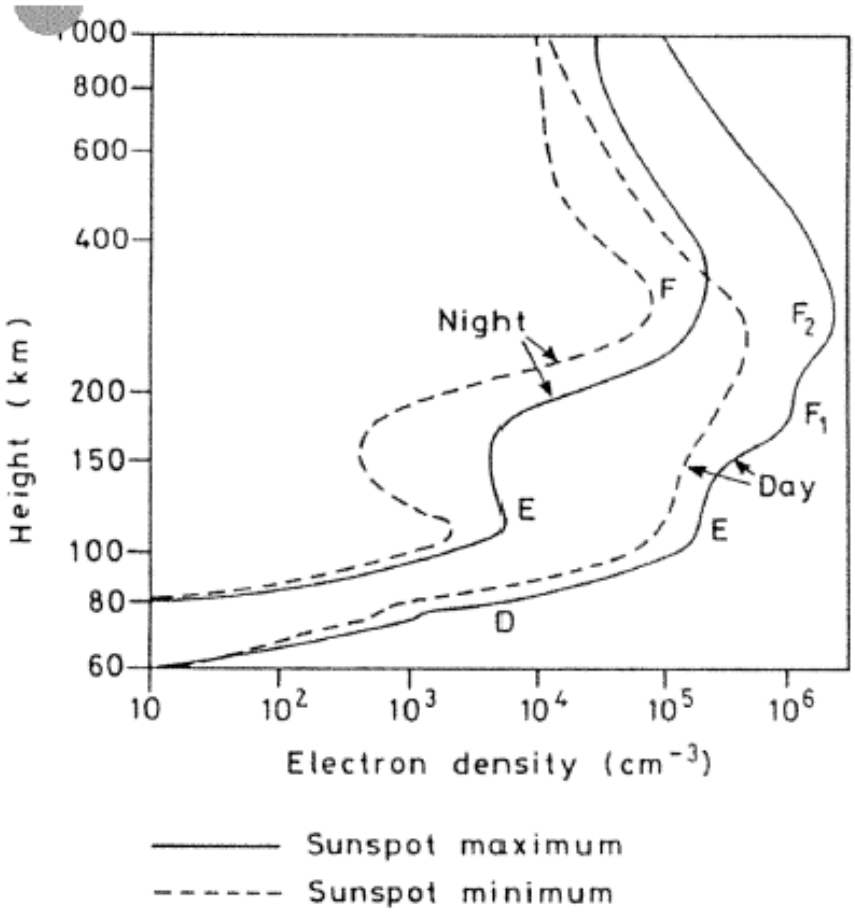


Figure 1.1: Typical vertical electron density profile of mid-latitude ionosphere after (Hunsucker & Hargreaves, 2003). The peak density of each ionospheric layer is labeled with the layer letter (D, E, F).

1.2 The E-region

Since the ICEBEAR instrument is used to observe the high-latitude E-region (>60°-latitude), this aspect of the ionosphere will be the main focus in this description. The E-region reaches a peak electron density around 105−110 km and is driven by charged particles and parts of the solar EM spectrum that are capable of penetrating through the higher layers of the ionosphere/atmosphere. The most numerous ions in this region are NO^+ and O_2^+, and at night the E-region decreases in election density down to 10^3 cm^{-3}. The conditions in the nighttime E-region at high-latitude, along with geomagnetic activity driven by the Sun produce the conditions wherein E-region coherent radar backscattering from plasma instabilities can be observed. During periods of high geomagnetic activity, the dominating form of ionization is precipitating particles (charged particles from the Sun and magnetosphere penetrating into the ionosphere), developing strong currents which flow through the E-region, driving instabilities in the plasma. Increased ionization expands the active regions of the E-region and generates a much more dynamic environment connected to the magnetosphere above and its plasma activity (Hunsucker & Hargreaves, 2003; Schlegel, 1996).

1.2.1 Instability Formation

In the E-region, electron motion is mainly governed by the $E \times B$ drift (also known as Hall drift), while the movement of the larger and heavier ions is mainly controlled by collisions with neutral atoms and molecules in the atmosphere and the local electric fields (Baumjohann & Treumann, 1997; Farley, 2009). As such, especially during periods of enhanced precipitation, currents in the E-region can generate instabilities. Instabilities are oscillations in the plasma density which are driven by the separation of ions and electrons and their resulting electrical attraction. Small density perturbations form in E-region plasma due to $E \times B$ drift, gradients, and collisions. These density perturbations can be enhanced by currents in the E-region from charged particles sourced from the magnetosphere above. Magnetospheric physics is driven by solar activity (such as solar flares or coronal mass ejections) from the Sun (Kivelson & Russell, 1995). Instabilities are formed perpendicular to the geomagnetic field lines, while the free movement of the electrons and ions along the magnetic field strongly damps plasma

oscillations. This creates standing waves in the plasma density oriented perpendicular to the geomagnetic field lines (Hunsucker & Hargreaves, 2003). The motion of charged particles moving perpendicular to the magnetic field lines is confined by EM forces (the Lorentz force). Electrons oscillate in bulk about the ions creating wave fronts.

One very effective way to study these plasma instabilities is through coherent radar (Fejer & Kelly, 1980; Schlegel, 1996), with some example radars being SuperDARN (Greenwald et al., 1995), SAPPHIRE (Koehler et al., 1995), and the University of Saskatchewan FMCW E-region radar (Cooper, 2006). Many studies have used radar instruments to probe the ionosphere and the instabilities observed have been classically categorized as 4 distinct types (Sahr & Fejer, 1996). These are presented in Figure 1.2 demonstrating their typical Doppler frequency spectrum. Type I instabilities are classified as having a narrow distribution centered around the ion-acoustic speed C_s, and is classified as a Farley-Buneman or two stream instability (Kivelson & Russell, 1995). Type II is defined by a broad spectrum centered around 0 Hz Doppler shift and is classified as a gradient-drift instability (Hunsucker & Hargreaves, 2003). Type III and Type IV are centered below and above C_s respectively, and the driving mechanism for these types are not fully understood yet (Hussey, 1994). While these classifications provide clear-cut definitions of the instabilities that can form, they do not fully encompass the true dynamics of these instabilities and the possible relationships between them. For example. recent studies have suggested that Type III and IV are phenomenon related to Type I (Chau & St.-Maurice, 2016). The understanding of instabilities and their formation is constantly evolving and ICEBEAR will provide further insights into this field.

1.2.2 Coherent Backscatter

The general equation for power measured by a radar from a scattered signal, known as the radar equation, is defined as (Richards et al., 2010),

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{1.2}$$

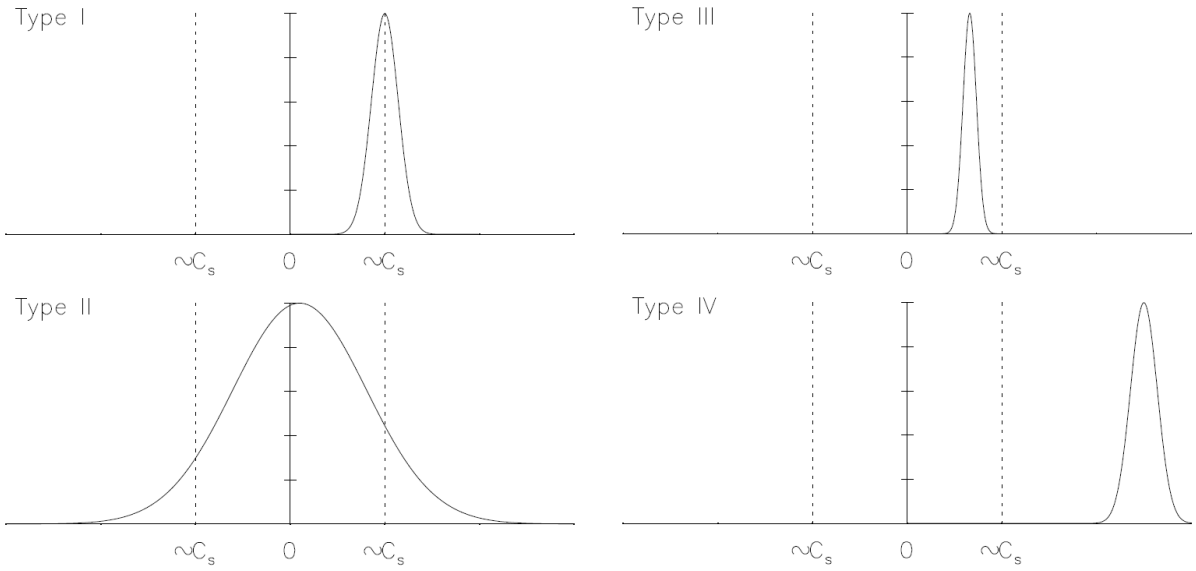


Figure 1.2: The four classic types of instabilities, with the typical normalized power distribution (y-axis) plotted against the Doppler frequency shift (x-axis). Here C_s is the ion acoustic speed. Type I has a narrow spectrum centered around C_s, Type II has a broad spectrum centered around zero Doppler velocity, Type III is similar to Type I but centered at a Doppler velocity smaller than C_s, and Type IV is again similar to Type I, but with a center Doppler velocity much greater than C_s (Hussey, 1994).

Where P_r is the received radar power, P_t is the transmitted radar power, G_t is the transmitter antenna gain, G_r is the receiver antenna gain, λ is the wavelength of the radar signal, σ is the radar cross section of the scattering target, and R is the range of the target (distance from the radar). This demonstrates the dependence of radar power measurements on the radar properties, scattering target properties, and target range. Coherent backscatter cannot be directly defined by the radar equation, but follows from the same basic principles.

Radio signals, such as for ICEBEAR, can scatter coherently from the plane waves of instabilities formed in the ionospheric plasma, sometimes called Bragg planes (Schlegel, 1996). Due to this radars that observe Bragg plane scattering are known as coherent backscatter radars. These radars use low power transmissions, and rely on the scattered signals to interfere coherently from the Bragg planes formed by the plasma waves and therefore increase the scatter power observed at the radar. For coherent backscatter from E-region instabilities, two conditions must be met:

1. The scattering structures must have a width that is half the radar wavelength (following the Bragg scattering condition) (Schlegel, 1996).

2. The transmitted radio wave must be propagating near perpendicular to the orientation of the magnetic field lines (Schlegel, 1996).

The Bragg scattering condition for coherent scatter can be expressed as (Hunsucker & Hargreaves, 2003),

$$\lambda_i = \frac{\lambda}{2\cos(\theta/2)} \tag{1.3}$$

where λ_i is the wavelength of the instability, λ is the wavelength of the radio signal, and θ is the angle between the transmitted and reflected radio signal. For mono-static radar configurations (e.g. SuperDARN), the transmission and reflection path of the radio signal is the same, so $\theta = 0^\circ$ and $\lambda_i = \lambda/2$. For ICEBEAR, which is a bi-static radar configuration (Tx and Rx are in separate locations), θ will be non-zero. The θ of a scattered signal will change for ICEBEAR depending on where in the FoV the scatter occurs. Typical scattering angles range from 0° to 15°, so the 49.5 MHz (6.06 m wavelength) radar will coherently scatter from instabilities with a wavelengths in the range $\lambda_i = 3.03$ m to $\lambda_i = 3.06$ m.

For the perpendicularity condition, the angle between the RF propagation path and the geomagnetic field is defined as the magnetic aspect angle α, where $\alpha = 0^\circ$ for perpendicular propagation (Hunsucker & Hargreaves, 2003). As the signal propagation deviates from the perpendicularity condition, the backscatter power decreases as the plasma waves are strongly damped along the geomagnetic field direction. Once $\alpha >\sim 5^\circ$, the plasma waves are strongly damped and no coherent backscatter is typically observable by coherent backscatter radars (Hall & Moorcroft, 1992; Kustov et al., 1994).

1.3 The ICEBEAR instrument

The ICEBEAR instrument is a very high frequency (VHF) coherent scatter radar developed at the University of Saskatchewan to study the E-region of the ionosphere (Huyghebaert et al.,

2019). It is a fully digital, CW radar operating at a center frequency of 49.5 MHz. Utilizing SDR, modern hardware, and advanced signal processing techniques, the radar generates high temporal and spatial resolution observations of the E-region. There are two types of radar scatter observed by the ICEBEAR system. The first is coherent scatter as discussed above, the second is direct reflection off metallic ions in meteor trails (Bronshten, 1983).

Nominally ICEBEAR has a 1.5 km range resolution (where range is the distance from a target to the radar) and a 0.1 s temporal resolution, however, as a fully digital radar these parameters can be readily adjusted. The transmitter is located near Prelate, SK, 300 km south west of the University of Saskatchewan in Saskatoon, SK, and the receiver site is located just outside of Saskatoon, SK. The FoV of the radar is 600 km x 600 km, centered around $\sim$58°N, 106°W with an antenna bore-sight of 7° East of North (Huyghebaert et al., 2019). Bore-sight is the forward looking direction for an antenna or array FoV. The ICEBEAR FoV falls within the auroral zone where the geomagnetic field is near perpendicular with respect to the surface of the Earth. ICEBEAR is designed such that the main antenna beam pattern is centered at 6° above the horizon with an azimuthal FoV (East-West) spanning $\pm$45° about the bore-sight. For a radio wave of 49.5 MHz propagating through the atmosphere at this angle, it will experience minimal refraction and meet the perpendicularity condition for coherent scatter around 700 km from the radar (Huyghebaert, 2019). Two different receiver antenna array layouts have been used for ICEBEAR. The initial array was only linear, with each of the 10 antennas separated by 6.0 m (which is 1λ at 50 MHz) in a roughly East-West orientation. The distance between a pair of antennas is called the baseline. Interferometry between antenna pairs provided the azimuth position of detected signals (see Chapter 2 Section 2.3.1). This layout provided redundancy in measurements, but came at the price of phase wrapping for signals received beyond $\pm$30° from the bore-sight. Phase wrapping is a form of aliasing in signal measurements that affects determination of the AOA. Phase measured on the antennas can range from $-\pi$ to π, and the AOAs that these phase values correspond to is determined by the combined antenna beam patterns (gain patterns) used in the interferometry process. For the linear receiver antenna array, the range of AOAs was limited to $\pm$30° from the bore-sight due to the antenna spacing (this is described in more detail in Chapter 2 Section 2.3.1). In summer 2019, the array layout was altered,

implementing non-lambda spacing between antennas, as well as implementing a 2-dimensional spread of the antenna locations. Azimuth and elevation information of scattered signals can now be determined by ICEBEAR using interferometry by combining all the receiver antenna pairs of the new antenna layout (Lozinsky, personal communication, December 20, 2020). Due to non-lambda spacing of the antennas, aliasing of the AOA is non-existent through the data processing method implemented.

The ICEBEAR FoV is displayed in Figure 1.3, as well as the transmitter and receiver locations. Some example scatter from the linear receiver antenna array configuration, pre-summer 2019, is plotted over geographical coordinates within the FoV. Data is plotted over the mapped geographical location of the scattering instabilities, and is given a color to represent the Doppler velocity. Positive velocity indicates movement towards the radar, and negative velocity indicates movement away from the radar. Magnetic aspect angle of the signal propagation is displayed using a grey-scale contour beneath the plotted data. The pink contours represent the radio frequency (RF) propagation distance or range of the radar. Since ICEBEAR is a bistatic radar, it is important to remember that the range is not exactly half the RF propagation distance, but the close proximity of the transmitter and receiver sites makes the halfway point a good range approximation for the scale of measurements being performed, although $\textit{RF propagation distance}/2 - 200$ km is a more realistic estimate.

ICEBEAR was developed to study the E-region of the ionosphere, with a focus on field-aligned plasma instabilities as the main radar backscatter. With the high temporal and spatial resolution provided by ICEBEAR, the dynamics and detailed structure of the E-region can be resolved in much greater detail than previously possible. This greatly enhanced observation resolution will develop a better understanding of E-region physics, how the E-region interacts with the rest of the ionosphere and magnetosphere, and improve our understanding of space weather. One aspect that can now be investigated using ICEBEAR is the elevation that different types of instabilities originate from, and if the altitude has a role in determining the type of instability that forms (St.-Maurice & Chau, 2016).

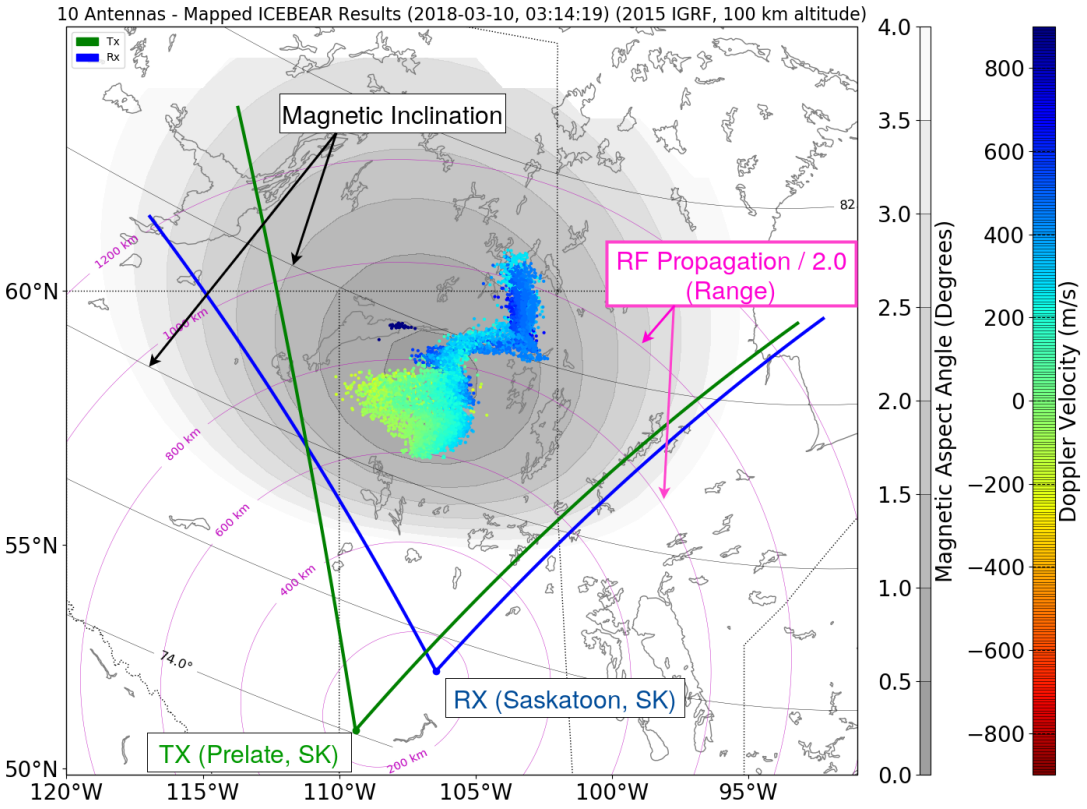


Figure 1.3: ICEBEAR map for 10 March 2018. Displays the transmitter FoV (green), the receiver FoV (blue), magnetic aspect angle (gray scale), RF propagation distance/2.0 (pink contours), magnetic inclination angle (black contours), transmitter and receiver site locations, and Doppler velocity (color scale) of detected scatter. The scatter depicted in this figure is example data that outlines the spatial resolution capabilities of the radar and instability structures that can be observed.

1.4 Thesis Outline

The research performed for this thesis focused on verifying, quantifying, and improving ICEBEAR data observations and quality. GPU acceleration was used to improve the computation speed of ICEBEAR data analysis. The noise in ICEBEAR measurements was studied to better understand the ICEBEAR noise environment and verify the SNR data product. Finally, a calibration method using the radio galaxy Cygnus A was developed to enable improved phase calibration of the ICEBEAR receiver antennas.

The ICEBEAR radar was designed to to provide detailed observations of E-region plasma density instabilities. However, to fully achieve this, improvements in data analysis speeds and better understanding of measured radar parameters is needed. For example, data processing of the high resolution measurements took longer than real-time to compute. This made the data set problematic to work with for timely analysis until I implemented GPUs for ICEBEAR analysis. In addition, no study of the radar noise parameters had been performed in detail to identify dominant noise sources or verify the noise estimates used in generation of the ICEBEAR SNR. Additionally, phase calibration of the ICEBEAR receiver antennas could be improved by using known measured signals, such as radio galaxies, to complement the current spectrum analyzer phase calibration technique.

Chapter 2 covers a detailed description of the ICEBEAR instrument. This begins with describing the antennas used for signal transmission and reception, as wells as the configuration of the transmitting, and receiving antennas arrays. Next the key radar components of the transmitter and receiver system are presented along with their role in the radar operation. The data processing technique is covered by first outlining the principals of interferometry, followed by how data is extracted and interpreted from the received signals.

Chapter 3 presents the optimization of ICEBEAR data processing using parallel programming and GPUs. First, the general structure and function of GPUs in computers and processing systems is reviewed, followed by a detailed description of the GPU used in the ICEBEAR processing computer. Next the data analysis technique presented in the previous chapter is programmed from the original C++ code into a parallel algorithm using the programming language CUDA. Finally, a performance comparison presents the new processing speeds obtained using the parallel algorithm.

Chapter 4 presents an analysis performed to verify the current technique ICEBEAR uses to determine the noise. Namely, using the median value of all measured signals over the ICEBEAR measurement time period (currently 0.1 s). This result is checked using data from the ICEBEAR FoV where only noise is detected. Comparisons to the median value technique were made to determine if it creates a reasonable noise estimate. The method used to remove clutter from ICEBEAR data is also evaluated to examine its effects. Lastly,

the noise data is examined to identify the dominant noise sources and any trends that are observed in the noise data.

Chapter 5 presents a method of phase calibration for the radar using the stellar radio noise source Cygnus A. First the presence of Cygnus A within the radar measurements is identified, along with the response of signal changes which depend on the receiving interferometry antenna pairs and their orientations. It is then shown how to use a stellar source with a known position to generate phase corrections. Using data collected during periods when there was low ionospheric activity, phase corrections were generated for the radar and compared to phase corrections determined using the spectrum analyzer phase calibration technique.

Chapter 6 demonstrates the observational capabilities of ICEBEAR presenting typical examples. This includes detection of E-region plasma activity and the identification of meteor trails in the ICEBEAR FoV. Comparison of meteor trail observations from both ICEBEAR and SuperDARN were made to demonstrate the accuracy of ICEBEAR measurements, and that ICEBEAR data can be used to perform collaborative experiments with other instruments.

Chapter 7 summarizes the information presented throughout the thesis and presents examples of future research that expands on the work performed in this thesis.

2 ICEBEAR System and Data Processing

2.1 Introduction to ICEBEAR

The Ionospheric Continuous wave (CW) E-region Bi-static Experimental Auroral Radar (ICEBEAR) is a radar developed at the University of Saskatchewan to study the E-region of the ionosphere in the auroral zone (Huyghebaert et al., 2019). CW transmission is used by ICEBEAR to maintain constant illumination of the radar field of view (FoV), providing high time resolution observations of E-region dynamics. Constant illumination over the 600 km x 600 km FoV results in scatter from multiple locations reaching the receiver simultaneously. To differentiate the multiple scattered signals, phase encoding is used on the transmitted signal to make each period, or range, of the transmission unique.

Due to the CW transmission, the receiving antennas cannot be located near the transmitting antennas because they would be saturated by the transmitted signal. As a bi-static system the ICEBEAR receiver is spatially separated from the transmitter and uses directional Yagi antennas at both the transmitter and receiver sites. Distance between antennas is defined by wavelength of the radar, λ, and for ICEBEAR 1λ=6.06 m. The transmitter site uses a linear layout of 10 antennas uniformly spaced at 1λ intervals, with the array oriented 16° East of North. For the receiver there are 2 antenna layouts that have been used. ICEBEAR was originally built using a linear antenna array layout of 10 antennas uniformly spaced at 1λ intervals in an East-West orientation just like the transmitter. This array allowed for interferometry to determine the angle of arrival (AOA) of scatter in azimuth, however, it was subject to phase aliasing from signals outside ±30° from the array bore-sight. This array layout is referred to as the linear receiver array, or ICEBEAR-linear. The current array layout, known as the 2D receiver array configuration or the ICEBEAR 3D configuration, is

designed to make optimal use of the land available for antenna placement. Each baseline created between antennas is unique, making use of non-integer λ spacing between antenna pairs, and antenna positions are distributed in 2-dimensions. Implementation of this array has removed phase aliasing in ICEBEAR observations and has added the capability to measure the elevation AOA, in addition to the azimuth AOA, of signals. The new configuration allows ICEBEAR to locate scatter in 3-dimensions using, azimuth AOA, elevation AOA, and range.

Figure 2.1 presents the ICEBEAR hardware layout and depicts how each component is connected for the transmitter (Tx) and receiver (Rx) sites. At the transmitter site, to generate and transmit the phase modulated CW signal, the ICEBEAR transmitter site utilizes a number of hardware components: computers are used to control and manage the transmitter system; GPS synchronized clocks provide accurate timing; while software defined radio transceivers (Ettus X300s (Ettus, 2018b)) generate the signal for transmission; finally the Tx signal is passed through amplifiers and is transmitted on the antennas. The ICEBEAR receiver site has a very similar set of hardware: antennas detect the Rx signal; band pass filters to isolate the ICEBEAR signal from interfering signals, both man-made and natural; low noise amplifiers (LNAs) increase the Rx signal; software defined radio (SDR) transceivers (again X300s) digitize the received signal; GPS synchronized clocks maintain accurate timing; and computers control and manage the system and store the digitized measurements. The X300 transceivers digitize the Rx signal in-phase and quadrature (IQ) which creates voltage measurements with real and imaginary components that represent the strength and phase of the measured signal. The complex voltage sample data stream is then saved as HDF5 files on an external hard drive. This data is then manually retrieved from the receiver site and taken back to the University of Saskatchewan for processing. Processing this data involves retrieving the scattered signal from the measurements through match filtering with the phase modulation code that was transmitted, decimating the data, and performing a cross-correlation between each antenna pair in Fourier space. This generates a set of 2D arrays representing the range and Doppler velocity of the measured signals for each sample time.

In this chapter, first the antenna layout of the transmitter and receiver sites will be outlined,

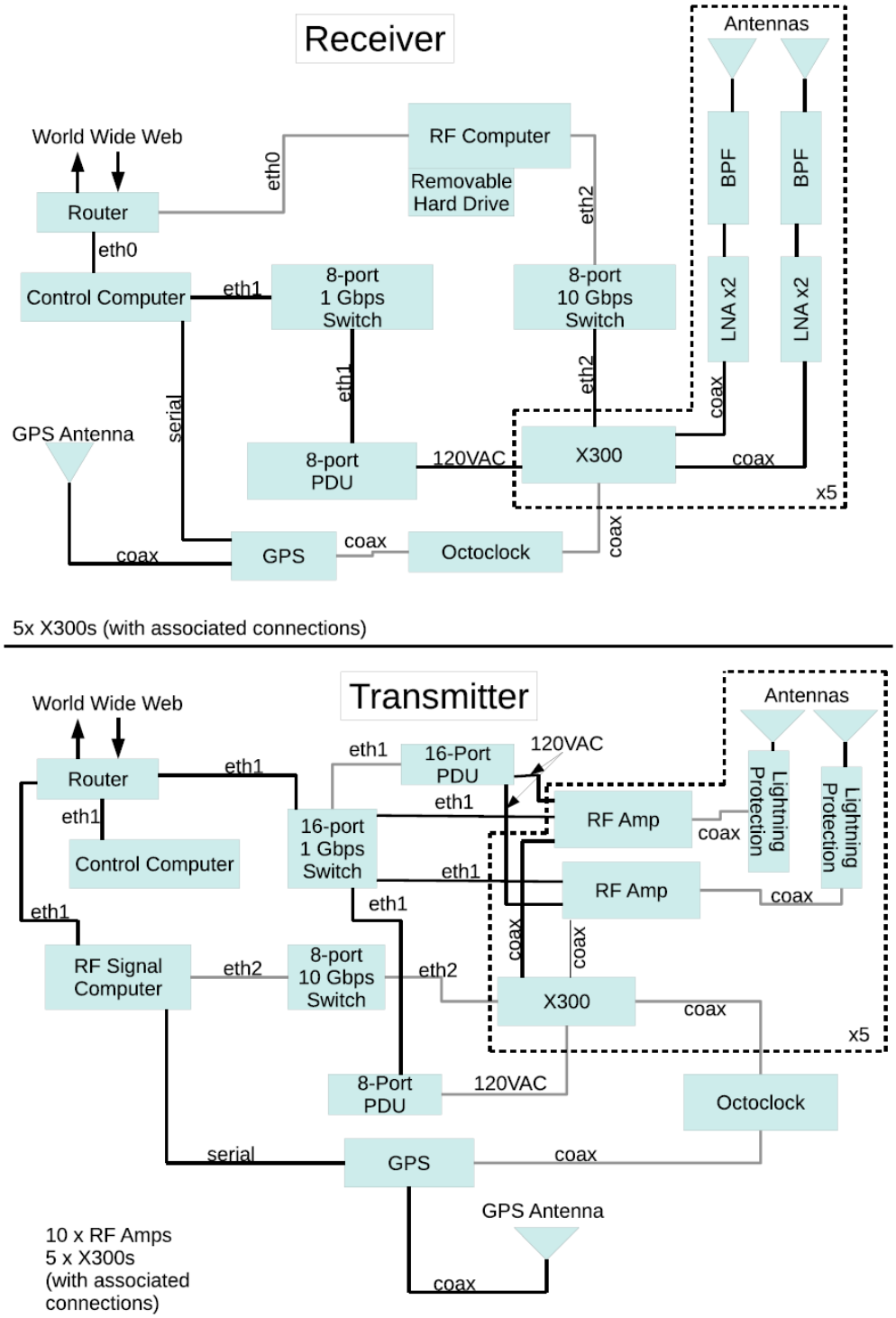


Figure 2.1: Block diagrams of the receiver and transmitter hardware, and the associated communication connections for the ICEBEAR radar after (Huyghebaert, 2019).

then the ICEBEAR hardware for the transmitter and receiver sites will be discussed, followed by an outline of the low level data analysis. For the transmitter site, the signal chain from control, to generation, to transmission will be discussed. For the receiver site, the signal chain from reception, to digitization, to storage will be covered. Finally, a discussion of the necessary processing steps will be presented.

2.2 ICEBEAR Hardware

2.2.1 Antenna Layout

The ICEBEAR transmitter has 10 Cushcraft 50-5 antennas designed for 50 MHz transmissions (Cushcraft, 2018b). These antennas were chosen for their ease of assembly and broad beam-width; they provide a signal gain of 10.5 dBi and have a maximum power output of 1000 W. Each antenna is mounted 15 m above the ground and they are configured in a uniformly spaced linear array, with 1λ spacing between each antenna. The array is oriented with a bore-sight of 16° East of North. Modeling of the transmitter beam pattern determined that the antennas produce a multi-lobed pattern in elevation, with the peak power centered at 6° in elevation and a vertical beam-width of 3°.

The ICEBEAR receiver has 10 Cushcraft 617-6B 6-m Boomer antennas that were reused from a previous radar system (Cushcraft, 2018a). Similar to the transmitter, the receiver antennas are mounted 15 m above the ground and provide a forward gain of 16.15 dBi (Cushcraft, 2018a). Modeling the 617-6B antenna beam pattern determined the receiver antennas produce a multi-lobed pattern in elevation, with a peak power centered at 6° in elevation with a 3 dB vertical beam-width of 3°. The azimuthal beam-width spans 30° around the antenna bore-sight, which is oriented 7° East of North.

Two antenna configurations have been used for the ICEBEAR receiver antenna layout, a linear antenna array similar to the transmitter set-up pre-summer 2019, and a newly implemented 2D antenna array configuration post-summer 2019. These are referred to as the ICEBEAR-linear and ICEBEAR-3D respectively. The ICEBEAR-linear receiver array antennas had a uniform spacing of 1λ between each antenna. These antennas were from the

existing hardware of a previous radar that was re-purposed for ICEBEAR (Huyghebaert, 2019). The array was oriented to have all antennas facing 7° East of North. Interferometry between antennas provided azimuthal AOA determination, however due to the uniform spacing of the antennas, phase aliasing occurred for scatter with an AOA greater than 30° with respect to the bore-sight. During summer in 2019, the receiver antenna array was re-configured to improve ICEBEAR observation capability. The new antenna locations were constrained by the available land, terrain, and environment, and were optimized to better constrain the point-spread function (PSF) of the receiver, and therefore improve the resolving capability of the interferometry layout. Due to non-integer wavelength spacing used between antennas (Thompson et al., 2017) and the 2D interferometry configuration, ICEBEAR can now better isolate the origin of scatter. Figure 2.2 defines the layout of the new antenna configuration as designed by Adam Lozinsky (Lozinsky, personal communication, December 20, 2020). In the new configuration, 5 antennas were arranged along the same East-West direction as ICEBEAR-linear, and 5 antennas situated in a North-South direction to provide elevation interferometry. The geographical layout of these antennas is shown in Figure 2.3 which displays the antenna positions of the new array on a satellite image of the receiver site. The antenna placement used non-uniform spacing so that each baseline of the new configuration is unique. The bore-sight of the receiver array was kept as 7° East of North. For more information on the design and implementation of the 2D receiver array, refer to the M.Sc research by Adam Lozinsky (Lozinsky, personal communication, December 20, 2020).

2.2.2 Transmitter System

This Section and Section 2.2.3 describe the system layout and operation of the ICEBEAR transmitter and receiver sites. Both sections are based on the Ph.D. work of Devin Huyghbaert who designed and built ICEBEAR. More detailed information regarding the ICEBEAR system can be found from the ICEBEAR instrument paper (Huyghebaert et al., 2019), and the Ph.D. thesis (Huyghebaert, 2019). A block diagram of the hardware is depicted in Figure 2.1, and a list of specifications for the ICEBEAR radar are in Table 2.1.

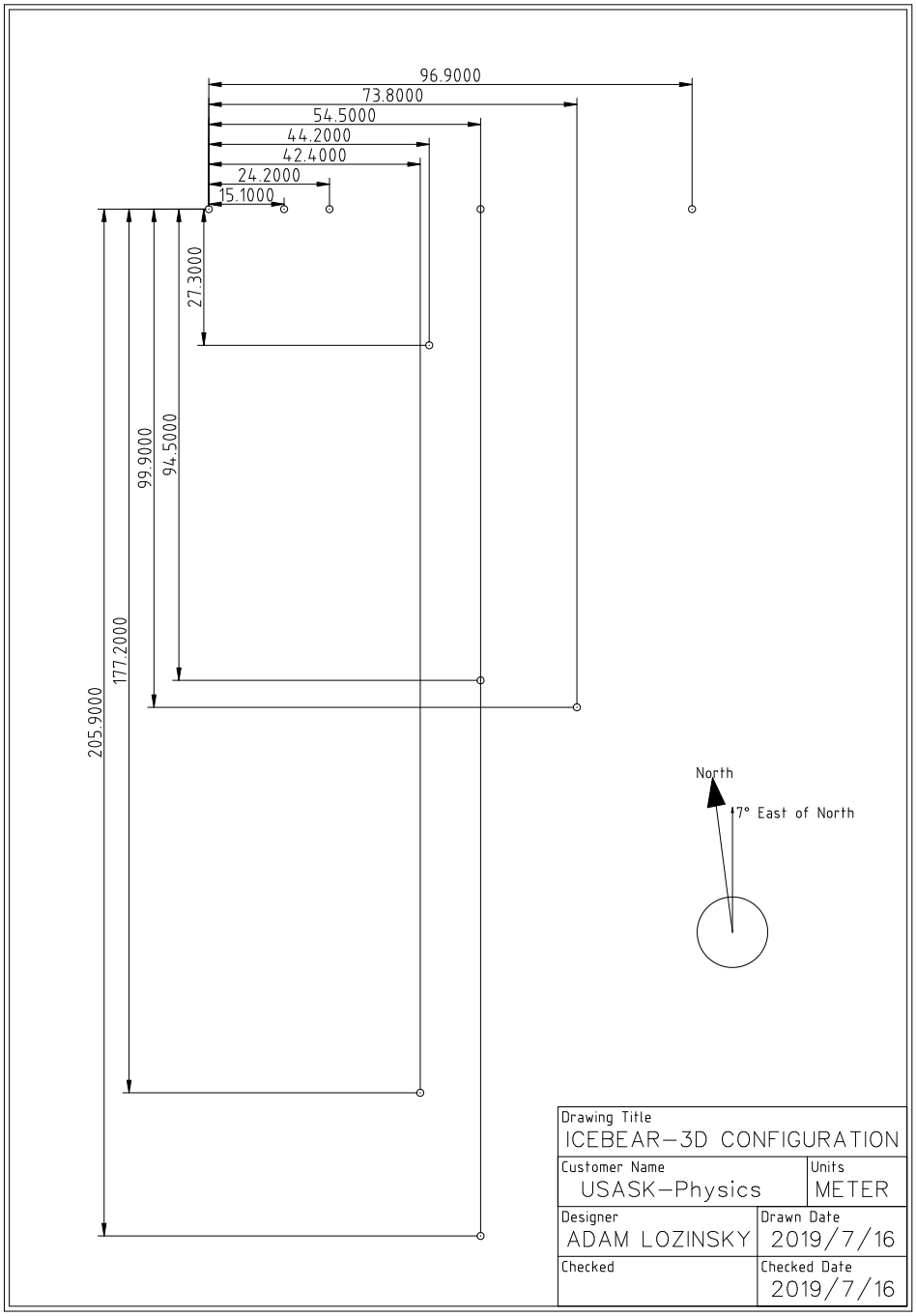


Figure 2.2: Schematic of the new ICEBEAR-3D configuration. The position of each antenna is stated relative to a single antenna (antenna 0), and the orientation of the array is shown as 7° East of North (Lozinsky, personal communication, December 20, 2020).

Figure 2.3: Satellite image of the ICEBEAR receiver site (Google, n.d.). The location of the 10 receiving antenna positions have been denoted on the image and labelled as they are identified in processing (Lozinsky, personal communication, December 20, 2020).

Specification @ 49.5 MHz	Value
Rx location	52.24319°, −106.450191°
Rx pointing direction	7° East of North
Tx location	50.893467°, −109.403151°
Tx pointing direction	16° East of North
Tx Peak Power	300 W
Modulation Type	Binary PSK
CW modulation code length	N=10,000 elements
Symbol length	10 μs
Rx sampling rate	200 kHz
Range resolution	1.5 km
Temporal resolution	0.1 s
Frequency resolution	10 Hz
Range aliasing	30,000 km
Frequency aliasing	± 100 kHz sampling
Sample size	32-bit IQ
Data rate (10 Rx)	8 MB/s

Table 2.1: Specifications for ICEBEAR after (Huyghebaert et al., 2019)

System Control and Timing

The bi-static nature of ICEBEAR is required since it uses a CW signal, which would saturate the receiver system if it was at or close to the transmitter. As a bi-static CW radar, for ease of operation both sites are remotely controlled and monitored. They must remain time synchronized to maintain coherence between the transmitted and received RF waveform signals. At the transmitter site, a Control Computer and an RF Computer are used to manage the site remotely. The control computer manages the operations of the amplifiers and the power distribution unit (PDU), as well as monitors system diagnostics. Overall control and RF signal generation is managed by the RF computer. Both computers are connected to a local internet network via a 1 Gbps Ethernet connection.

To maintain accurate timing, the radar uses the Fury GPSDO GPS-disciplined clock (Jackson, 2018) to accurately determine the time using GPS locked clocks, and maintain low clock drift in the event of lost GPS signal. Each device is accurate within 20 ns when using a GPS lock and has a drift of 7 μs/24 hours when a lock is lost. The clock is distributed to each transmitter chain through the Ettus Research Octoclock (Ettus, 2018a) to synchronize all ten transmitter antennas. Phase matching of the equipment and cables, as well as maintaining a uniform transmission path for the RF signal from generation to the antenna, ensure the synchronization of the outgoing signal is maintained.

Waveform Generation and Modulation

The radar waveform is generated using the Ettus Research X300 Software Defined Radio Transceiver (Ettus, 2018b). ICEBEAR uses pulse-compression techniques to modulate the waveform and signal filtering to ensure the transmitted signal falls within the radio license. Pulse-compression is implemented by using Pseudo-Random Noise (PRN) codes to modulate the phase of the outgoing signal, encoding the transmission to enable ICEBEAR signals to be separated from radar noise (Sulzer, 1989). This modulation results in radar signal appearing noise-like as the phase of the signal has a random pattern. For ICEBEAR signal modulation, a single PRN code was generated to use to modulate ICEBEAR transmissions. This PRN code is a sequence that randomly causes a 180° phase shift of the transmitted signal. The code used has a 10 μs symbol length and is 10,000 elements long, giving a 0.1 s code length. This results in a bandwidth of 100 kHz. Measurements do no alias until $\pm$ 50 kHz in frequency and 30,000 km in range which is well beyond the expected observation ranges from the E-region (Schlegel, 1996). To meet E-region observation needs and reduce processing only a subset of these abilities are processed, namely $\pm$500 Hz and a range of 3000 km, but if needed these parameters can be changed.

The transmitter signal is over sampled at a rate of 800 kHz to ensure sufficient sampling of the outgoing signal with the X300s while avoiding an excessive streaming bandwidth. Each X300 mixes the waveform to the 49.5 MHz center frequency and converts the waveform into an analog output that can then be amplified and transmitted. Timing of each device is maintained through the GPS and clock distribution system for a synchronized output.

Signal Transmission Line

First the RF computer generates the modulated waveform in software. This waveform is then passed to the X300s via a 10 Gbps switch where the analog signal is then generated. Each X300 has 2 output channels, so the 10 transmitting antennas are fed by 5 X300s. An RF signal for each antenna passes from the X300 through coaxial cable to the custom made 3 stage RF amplifiers built by Devin Huyghebaert at the University of Saskatchewan (Huyghebaert et al., 2019). Through a combination of the amplifier, the PRN code, and amplitude modulation in software to remove sidebands in the transmission waveform, the transmission is kept within the 160 kHz bandwidth allowed by the radio license. Each antenna is connected to amplifiers by 107 m of cable. The amplified signal travels through the coaxial cables to the antennas where the signal is then transmitted.

2.2.3 Receiver System

To be able to use the signal propagation time to determine the range of scatter, the two sites need to have temporal coherence. This is achieved by using the same GPS clock and clock distribution system that is at the transmitter site at the receiver site. Each signal path uses phase matched coaxial cable so all signals have the same phase offset. Combined with the low drift in the GPS clock and the accurate clock during GPS lock results in accurately matched timing between both transmitter and receiver sites. If the GPS lock was lost for 24 hours, ICEBEAR would experience a possible max of 7 μs clock drift, introducing a range error of less than 1.5 km (an ICEBEAR range gate). The receiver system also has the same computer set-up scheme as the transmitter, a Control Computer for monitoring status and power control, and an RF Computer which receives measurements from the radio transceivers and stores data. The Ettus Research X300 Software Defined Radio Transceiver is also used at the receiver site to sample the incoming signals and generate digital representation (Ettus, 2018b). Voltage samples are measured in-phase and quadrature (IQ) to generate 32-bit complex voltage samples.

Signal Reception Line

Incoming radio waves are collected by the Cushcraft 617-6B antennas (Cushcraft, 2018a) and the RF signal is transmitted through 183 m of coaxial cable to 3333-SMA Band Pass Filters (BPFs) (KR, 2018). The BPF attenuates signals from outside the desired bandwidth about 49.5 MHz (eg: TV and radio station transmitters) to only receive scatter from the ICEBEAR radar transmitted signal. Next the signal is amplified using two Low Noise Amplifiers (LNAs) in series, which increases the signal power to a level which can be digitized by the X300 (Mini, 2018); each LNA has a 28 dB gain. Due to signal loss in the coaxial cables and the BPFs, approximately 12 dB of signal is lost before the signal reaches the LNAs (Huyghebaert, 2019). Once the signal has been amplified, the X300s digitize the RF signal with a sampling rate of 200 kHz. This sampling is required by Nyquist sampling theorem to sample the 100 kHz signal bandwidth signal. Each of the 10 receiving antennas has the 183 m of coaxial cable, the BPF, and the LNA in their signal chain. The X300s have two receiver channels, so each of the 5 X300s collect samples from 2 antennas.

Data Storage

The samples digitized by the X300s are streamed to the RF computer and stored on an internal HDD using the open source MIT Haystack DigitalRF software suite (R. et al., 2021). When the radar is not actively recording, measured data is transferred to a larger 12 TB external HDD where the data is stored. Once full, the external HDD is retrieved from the site and taken to the University of Saskatchewan for processing and a new HDD is required to store data for the next experiments. Samples are stored as HDF5 files, which is a non-platform-based file format. The DigitalRF software also tracks meta data, corrects for lost samples, and provides functions to read the HDF5 files. As ICEBEAR is a CW radar system there is a continuous stream of data from all 10 Rx antennas. At the 200 kHz sampling rate for 10 antennas, ICEBEAR has a data rate of 8 MBps or 675 GB per day. Due to the large data rates and the lack of sufficient internet upload speed, the radar was run in campaign mode during the initial years of operation. During expected active times, the receiver and transmitter were turned on, typically for a few days at a time, to collect data. In October

2020, ICEBEAR began operating on a daily basis from 0−14 UTC.

2.3 Data Processing

2.3.1 Interferometry

To retrieve information about scattering targets from the measured data, interferometry is used. Interferometry is the process of correlating measurements made from two differently located antennas and using the correlation to determine the location of the scattering source. The basic concept of interferometry is shown in Figure 2.4 and is an application of Young's experiment from optics (Hecht, 2016; Hysell et al., 2016). If the scattering target is sufficiently far from the radar, a few tens of km, it can be treated as returning to the receiving antennas as a plane wave. Scattered signals will arrive at one antenna first, and then have an extra distance, Δx, to travel before reaching the second antenna as is shown in Figure 2.4. As the wave travels that extra distance, the phase will change, resulting in a phase difference measured between the two antennas that is dependent on the extra distance the wave had to travel. Since this extra distance, Δx, is dependent on what angle the signal arrives from, the phase difference in the measurements can be related to the AOA, θ. In the ICEBEAR radar, baselines in an East-West orientation will determine azimuth AOAs, and North-South oriented baselines will determine elevation AOAs. For two antennas with some baseline, or separation distance d, the phase difference, $\Delta\Phi$, is given by (Hariharan, 2007),

$$\Delta\Phi = \frac{2\pi d \sin\theta}{\lambda} \tag{2.1}$$

where λ is the radar wavelength. This technique is used by the ICEBEAR receiver system and configuration to retrieve azimuthal and elevation AOAs from the complex voltage measurements, which contain the information needed to locate where in the FoV the scattering signal was sourced.

It is important to note that phase measurements always fall within the range -π to π. Phase values that pass outside these ranges are wrapped back into the range. So if the extra distance

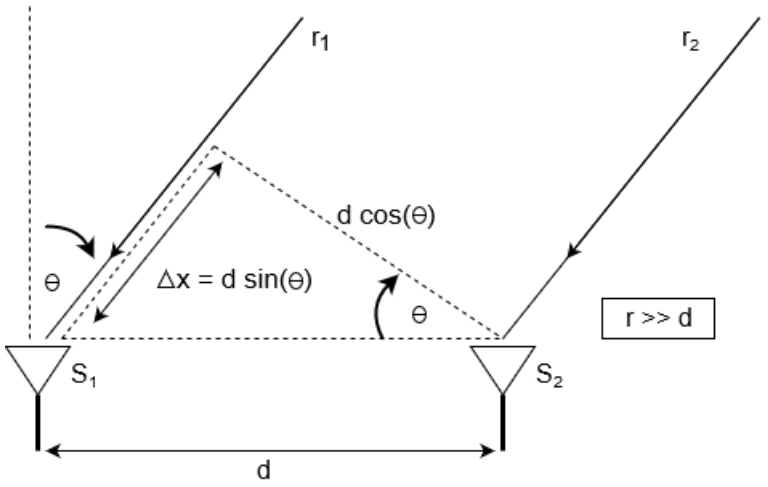


Figure 2.4: Relationship between two antennas separated by a distance, d, measuring the same source or scattered signal. Showing the extra distance traveled by the wave, δx, between reception on antenna 1 (S_1) and reception on antenna 2 (S_2). The AOA, θ, can be in azimuth and/or elevation depending on the orientation of the antennas and the position of the source at a distance, r, from the antennas.

traveled by the signal is $d \sin\theta = \lambda$, the phase difference measured on the antennas will be 2π, which is the same as 0π due to phase wrapping, thus the AOA will be calculated as if there is no phase difference. This property of interferometry is called AOA aliasing, as signals beyond a certain AOA are aliased back into the bore-sight of the radar FoV when the path difference is a multiple of λ. The aliasing point depends on the baseline distance between antennas and the wavelength of the RF signal. The beam pattern of the antennas used in interferometry act like a Young's double slit experiment, where each lobe in the beam pattern is created by constructive interference of the signal, and nulls are created by destructive interference. The edges of a lobe in the beam pattern correspond to the maximum and minimum phase measurements ($-\pi$, π), so the range of AOAs that can be determined through interferometry are constrained by the angular width of the beam pattern lobes. The angular position of the lobe peaks and nulls are given by (Hecht, 2016),

$$d \sin\theta_{peak} = m\lambda \tag{2.2}$$

$$d \sin\theta_{null} = (m + \frac{1}{2})\lambda \tag{2.3}$$

where $m = 0, 1, 2...$, θ_{peak} is the angle of maximum intensity, and θ_{null} is the angle of minimum

intensity. While the Young's double slit formulas do not take into account other factors that influence the beam pattern, such as ground reflection, it is a very good analogy when considering determining AOAs. For example, consider the linear receiver array described in Section 2.2.1 which had a 1λ spacing between antennas. From Equation 2.1, a 30° AOA gives a phase difference of π, and conversely an AOA of -30° give a phase difference of $-\pi$. So between 30° and -30° there is a full 2π phase difference. Furthermore, Equations 2.2 and 2.3 can be used to show that the first peak $m = 0$ is located at an AOA of 0°, and the lobe edge or null for this lobe is at 30°, with the other edge at -30° by symmetry. This matches the beam width from Equation 2.1 which confines ICEBEAR observations within this FoV and introduces aliasing of measurements from outside the main lobe of the beam pattern for the ICEBEAR-linear Rx antenna array configuration. ICEBEAR has now removed this aliasing ambiguity from measurements with the new ICEBEAR-3D receiver antenna array layout that was described in Section 2.2.1. Non-integer wavelength spacing between antennas and many different baseline sizes are used to isolate the scattering location by combining each baseline in the analysis (Thompson et al., 2017).

2.3.2 ICEBEAR Analysis

ICEBEAR data analysis extracts physical properties from the complex voltage values measured on the receiver. As this is very computationally intensive, even by current standards, processing is performed on a Graphical Processing Unit (GPU) to optimize the analysis processing speed. The implementation of the GPU in ICEBEAR data analysis is discussed in Chapter 3. This section will review the individual analysis steps which include: retrieval of scattering information from the complex voltage measurements; decimation of the data; determination of the range-Doppler spectrum for one antenna; and finally the range-Doppler spectrum generated by a cross-correlation between two antennas. These analysis steps assume the nominal operating parameters of ICEBEAR, and generates the low level processed data for a single baseline. The analysis calculates 0.1 s of data, the temporal resolution of the radar. For ten receiving antennas, there are a total of 45 possible unique baselines for the new 2D receiver antenna array configuration (9 unique baselines for the linear (1D) receiver array configuration). Thus typical processing of 0.1 s of ICEBEAR data requires this analysis

to be repeated 45 times.

The scattered received signals can be retrieved from the measured voltage samples by using the CW transmitted PRN modulated code as a matched filter (Sulzer, 1989). Consider a number of voltage samples, V, of length $N + r$ and a transmitted complex code, C, of length N:

$$V[t] = [V[0], V[1], V[2], \ldots, V[N + r - 1]] \tag{2.4}$$

$$C[t] = [C[0], C[1], C[2], \ldots, C[N - 1]] \tag{2.5}$$

where N is the length of the complex code, r is the maximum range index of the RF propagation distance, and t is the time of the sample. In this analysis, the complex transmitted code, C, is the PRN code. For ICEBEAR, the typical PRN code has 10,000 elements with a 10 μ s symbol length (See Table 2.1 in Section 2.2), for a code 0.1 s in length. This defines the time resolution Δt, giving Δt=0.1 s. The range resolution, Δr, is determined by the sampling rate and is given as,

$$\Delta r = \frac{c}{f_s} \tag{2.6}$$

where c is the speed of light, and f_s is the sampling frequency. At a sampling rate of f_s = 200 kHz, the range resolution is Δr = 1.5 km. Given a maximum range nominally used in processing of 3000 km, the maximum range index is r = 2000. While the range resolution is defined by the sampling frequency, the spatial resolution is provided by the CW signal and is dependant on the bandwidth of the radio signal (Richards et al., 2010). The total propagation path length resolution is then given by,

$$p_t = \frac{c}{BW} \tag{2.7}$$

where p_t is the total path length, and BW is the bandwidth. Since the bandwidth of the modulated CW transmission is 100 kHz, the total path resolution is 3 km.

Scattered information is decoded by matching the PRN code to the measured voltage samples. There is 1 sample generated every 5 μs, resulting in 20,000 voltage samples collected over the 0.1 s PRN code. Since the symbol length of the transmitted PRN code is 10 μs, to compare the code to the measured voltage samples each element of PRN code must be repeated to create a complex code, C, with the same number of samples as the measured data. Thus for this analysis, the complex code length is N=20,000. In the decoding process, the voltage samples are time shifted by the number elements equal to the range index being evaluated, and then multiplied by the complex conjugate of the decoding PRN code to generate a matrix of voltage samples that is N columns by $r+1$ rows:

$$V_{spec}[r,t] = \begin{bmatrix} V[0]C^*[0] & V[1]C^*[1] & \dots & V[N-1]C^*[N-1] \\ V[1]C^*[0] & V[2]C^*[1] & \dots & V[N]C^*[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ V[r]C^*[0] & V[r+1]C^*[1] & \dots & V[N+r-1]C^*[N-1] \end{bmatrix} \tag{2.8}$$

where V_{spec} is the decoded range-time spectrum of voltage samples. In this matrix, each row represents one range bin, from an index of 0 to 2000 or from 0 km to 3000 km. The columns represent the time samples from each range, where each column represents a measurement 5 μs after the previous column. As can be seen in Equation 2.8, the range index desired is obtained by shifting the voltage values from the starting point by the index to be calculated. For example, range index 1 is obtained by multiplying the complex conjugate of the PRN code with the voltage samples from element 1 to element N. Therefore, range index 1024 is obtained using voltage samples from element 1024 to N+1023. In this way the PRN code has been matched to measurements and used to filter out ICEBEAR data by demodulating the measured data, hence the term match filter to describe the process. To reduce the amount of data, and average out noise fluctuations in the measurements, the original sampling rate of 200 kHz is be down sampled by coherently averaging multiple measurements together. ICEBEAR nominally uses a decimation rate of 200 to convert the 200 kHz sampling rate to 1 kHz, as this covers the Doppler frequencies expected in the measurements (Huyghebaert, 2019). At 1 kHz, the Doppler range is $\pm$ 500 Hz centered around 0 Hz. The decimation step

is performed simultaneously with the decoding step to reduce computation time.

After the decimation, only 100 columns remain in the range-time spectrum, V_{spec}, each holding coherently averaged time measurements averaged over 1 ms. Next a Fast Fourier Transform (FFT) is applied to every row of the matrix, converting time domain measurements to frequency domain measurements. The new range-Doppler matrix now represents the range and Doppler shift (frequency shift of the received signal from the RF wave frequency) of the scattering signal. The Doppler resolution, Δf is defined by,

$$\Delta f = f_s/N \tag{2.9}$$

so the Doppler resolution for this analysis where $f_s = 200$ kHz and $N = 20{,}000$ is 10 Hz. The Doppler shift represents the velocity of the scattering target, with a positive Doppler shift indicating movement towards the radar with respect to the Bragg scattering condition, and a negative value indicating movement away. The Doppler velocity of a target in the ICEBEAR radar is described by (Alabaster, 2012) and follows the bi-static form of Equation 1.3 from Chapter 1,

$$v = \frac{\lambda f_d}{2\cos(\theta/2)} \tag{2.10}$$

where v is the velocity of the scattering target with respect to the radar, f_d is the Doppler shift of the signal, λ is the wavelength of the radar, and θ is the angle between the Tx and Rx locations from the scattering target. From the 10 Hz frequency resolution the Doppler velocity resolution for ICEBEAR, using the signal wavelength $\lambda = 6.06$ m and assuming a $\theta \sim 0°$, is 30.3 m/s and the total Doppler velocity range is therefore $\pm$ 1.5 km/s.

At this point, the range and Doppler shift of a signal measured by a single antenna has been retrieved. Finally, a cross-correlation must be performed to determine the azimuth/elevation AOAs of the received signals. In frequency space, one range-Doppler matrix is multiplied by the conjugate of the range-Doppler matrix forming the antenna pair as given by,

$$V_{xspec}[r, f] = V_{spec1} V^*_{spec2} \tag{2.11}$$

where V_{xspec} is the cross-correlation range-Doppler matrix for an antenna pair, and V_{spec1} and V_{spec2} are the range-Doppler spectra calculated for antennas 1 and 2 respectively. The units of data in V_{xspec} will be [V^2] (volts squared). The magnitude of the new cross-correlated range-Doppler matrix represents the "power" of the cross-correlated signals measured in each range-Doppler bin, and the angle of the matrix values represents the phase difference measured between the antenna pair. As was discussed in Section 2.3.1, the phase difference can now be used to determine the AOA using Equation 2.1. The power is interpreted using a signal to noise ratio (SNR). This compares the strength of each signal measured to the background noise in the radar measurements and is defined as (Richards et al., 2010),

$$SNR = \frac{S - N_0}{N_0} \tag{2.12}$$

where S is the total power measured on the radar, and N_0 is the noise power measured on the radar. In the radar system the desired signal and noise add together to create the total power measured. However, the noise will not be correlated with the scattered signal power so they signal power can be calculated by taking the difference of the total power and the noise power. Noise in ICEBEAR is mainly a result of cosmic noise and self-clutter which is typical for a coherent radar, although other sources of noise (e.g. FM/AM radio stations and thermal noise) are also present. Chapter 4 will discuss the determination of the noise power in the ICEBEAR radar (Note when trying to consider the "absolute" noise of the system in Chapter 4, the noise is the [V^2] measurements and not a ratio like the SNR). Figure 2.5 shows the result of the cross-correlations between two antennas, 1 and 2, in the linear array, which are separated by one wavelength. The SNR has been computed by taking the median of the cross-correlation matrix as the noise value, and any SNR value below 3 dB has been masked. Each type of E-region instability that can be measured has been indicated in the figure, showing the Doppler Shift, range from the radar.

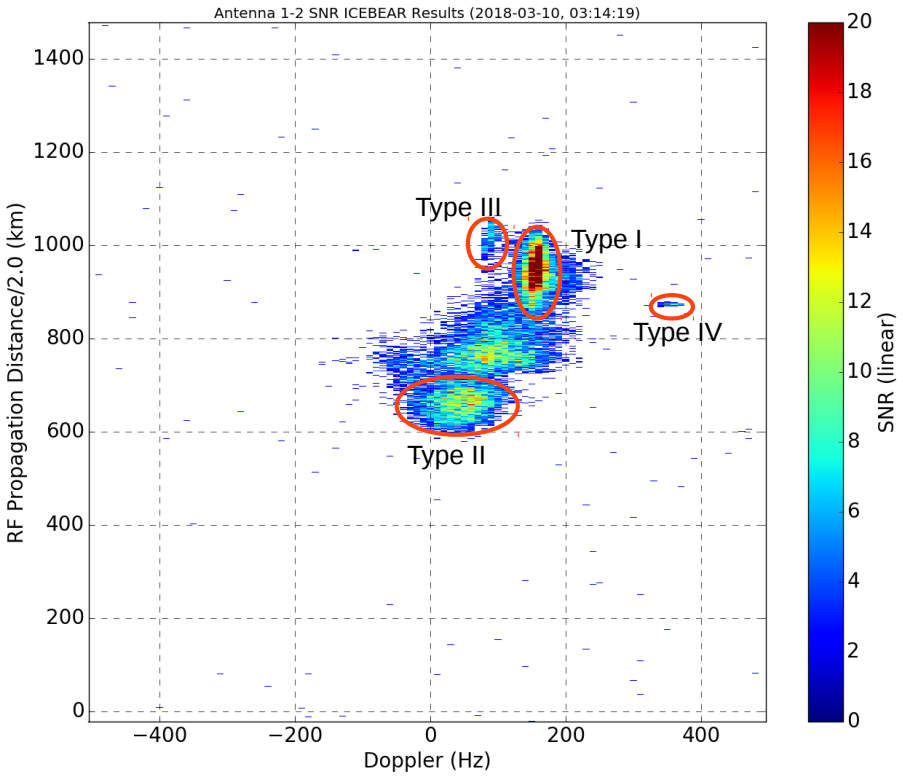


Figure 2.5: SNR plot of RF Propagation distance/2.0 vs Doppler Frequency, with all echo types detected by ICEBEAR labeled for March 10, 2018 at 3:14:19 UT

2.4 Summary

In this chapter, the hardware and operation of the ICEBEAR radar has been described. Starting from the transmitter site, the generation and transmission of the RF waveform has been outlined, and the receiver site reception and storage has also been discussed. The RF signal is generated by a computer and modulated in software. Using SDR X300 transceivers, the signal is digitally mixed at 49.5 MHz and then converted into an analog signal. The signal is amplified through custom amplifiers, and is finally transmitted from the radio antennas. At the receiver, antennas measure the scatter. The received signals are passed through a BPF and are amplified through two LNAs before being digitized by the X300s. This data is then stored on local HDDs at the receiver site in HDF5 format.

The process for retrieving physical information from measured signals was also discussed.

ICEBEAR data analysis takes the measured antenna voltages and matches them with the known PRN modulation code (known as match filtering) to demodulate the signal and identify range and Doppler frequency of the scatter. The data from a single antenna is first matched with the modulation scheme to retrieve the scattered signal from the noise-like measurements. The data is then decimated into range and time bins defined by the radar timing and FoV. ICEBEAR is currently operated with a 1.5 km range resolution and a 0.1 s temporal resolution, or 10 Hz in frequency; however, either of these can be changed within the radar software. This process produces a matrix of complex voltages, where the rows and columns represent the range and time bins respectively. Taking a Fast Fourier Transform (FFT) along each range in this matrix generates the Doppler frequencies of the scatter. Finally, a cross-correlation can be performed between two of these data sets to perform interferometry between an antenna baseline pair. The final complex valued matrix that results holds range and Doppler shift information in the number of rows and columns of the matrix. The magnitude of the matrix values represents the power of the signal for each range and Doppler shift bin, and the angle of each matrix value represents the phase difference measured between the antenna pair. For interferometry including and combining all the multiple antenna pairs, see Chapter 6.

3 GPU Optimization

Graphical Processing Units (GPUs) are a type of computer processor designed for rapidly processing and rendering of graphics and images. The GPU industry has grown rapidly over the past 20 years due to the ever increasing demands for fast real-time processing of complex 3D graphical environments for video games. This advancement in rapid computing capability has also been expanded into the scientific community for processing of numerical computations (Nickolls & Dally, 2010). GPUs offer the ability to perform scientific analysis efficiently by utilizing specialized hardware and a parallel processing model of computing. Parallel computing spreads computationally similar tasks among many different processing cores within the GPU architecture, enabling simultaneous processing of large amounts of data compared to conventional sequential computing methods.

The ICEBEAR analysis process is well suited to a parallel implementation, which is required for real-time calculations. Although the radar is capable of generating very high resolution observations of the E-region of the ionosphere, it comes at the price of generating a large quantity of data; 28 GB/hour of raw complex voltage data. Conventional sequential processing on modern computer hardware and Central Processing Units (CPUs) is unable to process this quantity of data in a reasonable time frame. To calculate the 45 unique receiver baselines, the optimized ICEBEAR C++ code takes 33.39 hours to calculate 1 hour of data. Implementing parallel processing of key parts of the ICEBEAR data analysis on a GPU enables processing of ICEBEAR data in effectively real-time.

First, this chapter will introduce the basic hardware structure of a GPU. Next, the basic concepts of parallel processing and how ICEBEAR analysis can be made parallel will be presented. The current ICEBEAR GPU will be discussed, and an example of how this GPU is used will be provided. Finally, the optimization of ICEBEAR analysis will be presented,

covering the GPU coding language selection, implementation of the ICEBEAR analysis onto a GPU, and validation of the processing results.

3.1 Graphical Processing Units (GPUs)

The graphics card, or more commonly the GPU, has been around for many years. Initially these devices were used to generate the graphic environment of computers and process vectors and geometries for computer and console games. Over the years, these devices have rapidly advanced to keep up with the high resolution and 3D graphic requirements of games, and are now capable of far more processing power than a regular Central Processing Unit (CPU) (Nickolls & Dally, 2010). In recent years their uses have been expanded and are now used in many scientific fields, utilizing the parallel processing capabilities of GPUs to process data and images (Memeti et al., 2017).

3.1.1 GPU Structure

GPU hardware is built with hundreds of processors and designed for efficient execution of many computations simultaneously. In simplest terms, more transistors are dedicated to data processing rather than caching or data flow. GPUs are most efficient at tasks with a high amount of arithmetic operations compared to the amount of memory management required (NVIDIA, 2019a). Table 3.1 shows a comparison between CPU and GPU performance when executing mostly sequential code versus mostly parallel code. CPUs are most efficient at running sequential code, while GPUs drastically improve parallel code. Repetitive calculations or intensive tasks are broken up and divided among the multiprocessors on the GPU, which can then execute tasks on each multiprocessor concurrently, increasing computation speeds through parallel execution on top of the efficient data processing hardware design.

Figure 3.1 shows a block diagram comparison between the general layout of a CPU and a GPU. CPUs require large caches and a significant control unit for general operation. While fast processing is important on a CPU, they need to be able to handle the wide variety of tasks needed for the operation of a computer. Data needed for the processor tasks are held

		Processing time (s)			
Program Type	Configuration	1 CPU core	500 GPU cores	10 CPU cores	1 CPU core + 450 GPU cores
Parallel-intensive program	0.5 % serial	1.0	5.0	1.0	1.00
	99.5 % parallel code	199.0	0.4	19.9	0.44
	Total	**200.0**	**5.4**	**20.9**	**1.44**
Mostly sequential program	75 % serial code	150.0	750.0	150.0	150.0
	25 % parallel code	50.0	0.1	5.0	0.11
	Total	**200.0**	**750.1**	**155.0**	**150.11**

Table 3.1: After (Nickolls & Dally, 2010). Comparison of CPU and GPU performance on 2010 hardware. Assumes CPU cores are 5x faster and 50x the area of a GPU core.

in the cache, while the control unit manages the execution of code and the management of memory within the processor. GPUs are built with a focus on computational efficiency and data throughput (Trobec et al., 2018). As such, the cache and control units of a GPU are much smaller to allow room for more Arithmetic Logic Units (ALUs). This design choice is made under the assumption that information passed to the GPU is designed for parallel execution and requires little program control beyond copying data to and from the processor from the computer memory. Multiple ALUs are sectioned into blocks within the GPU that each have their own control unit and memory cache. Section 3.2.2 describes the NVIDIA GeForce 1080 Ti GPU used by the ICEBEAR processing computer and describes a more detailed GPU structure layout relevant to ICEBEAR operations.

3.1.2 Parallel Processing

Parallel processing refers to computer algorithms that implement multiple operations in a single step (Trobec et al., 2018). Where a sequential algorithm executes the commands of a program one step at a time, parallel algorithms spread tasks between multiple processors or systems. All computer systems have become parallel to some extent, as there are always some

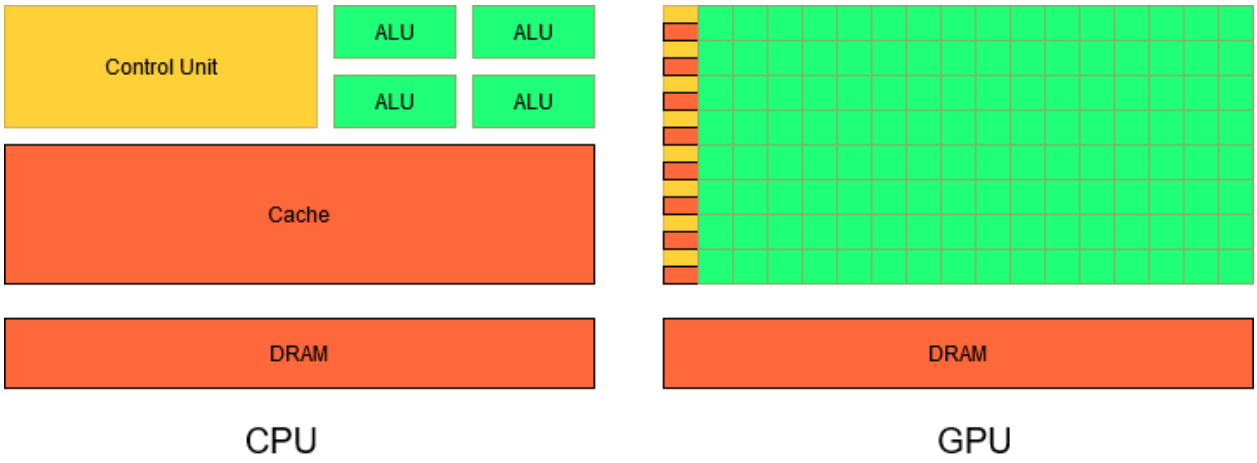


Figure 3.1: After (NVIDIA, 2019a). Block diagram comparison between CPU and GPU hardware layouts with equal area for distribution of components. Yellow blocks are control units which manage communication with external devices, execution of code, and management of memory within the device. Green blocks are Arithmetic Logic Units (ALUs) which are the base processing block. Orange blocks (cache, DRAM) are memory locations. Dynamic Random Access Memory (DRAM) is external memory that holds program code and data for operation. The cache stores local data from the DRAM for quick access within the processing unit. GPU structures assume less program control and data management and focus on increasing computational throughput by using more ALUs.

activities operating at the same time. Multiple processors can be used to run programs in parallel, processors can have multiple cores capable of executing their own instructions independently within the processor, and graphic processors within a GPU are capable of running hundreds of instructions in parallel (Trobec et al., 2018). The concept of breaking a sequential set of instructions across multiple processors to execute the instructions simultaneously is logical, and the resulting increase in computational efficiency is clear; more instructions run in less time. However, the complexity of designing parallel algorithms is much greater than a sequential algorithm. Designing parallel algorithms requires identifying what instructions can be performed in parallel, and what instructions are required to be sequential.

Matrix operations are the simplest examples of parallelism in an algorithm. Each element of a matrix undergoes some mathematical operation, but the operations on any one element are not dependent on the operations performed on other elements. Therefore, each element of the matrix may be computed simultaneously. Individual steps of **for**-loops may also be executed in parallel as long as none of the steps are dependent on the result of a previous

step of the loop. The design of a parallel algorithm requires a clear understanding of the order of instruction execution, as well as instruction dependence on other operations. It is also important to understand the parallel environment, a GPU in the case of ICEBEAR, that the algorithm will be executed on. Since the goal of parallel processing is to improve computational efficiency, the parallel algorithm must be designed to optimize the use of resources available in the parallel environment. Section 3.2.2 describes the GPU environment used in ICEBEAR for parallel processing, and Section 3.3.2 outlines how ICEBEAR implemented parallel processing into the analysis.

3.2 The ICEBEAR GPU

3.2.1 GPU Selection

While parallel processing can be performed on a CPU or spread across multiple processing computers, a GPU is the best solution to implement parallel processing on a local computer. The ICEBEAR analysis is too intensive for a CPU to process in a reasonable amount of time, but not so intensive as to require access to a computation network or supercomputer system. Therefore a GPU is required to accelerate the analysis of ICEBEAR data. Top GPU manufacturing companies AMD and NVIDIA currently are leading GPU development, offering cutting edge performance in their GPUs, as well as marketing their GPUs to the scientific community. The GPU in the ICEBEAR processing computer was chosen in mid 2017. At the time it was one of the best GPUs produced by NVIDIA to preform parallel data processing. ICEBEAR uses the NVIDIA GeForce GTX 1080 Ti GPU, which is capable of running user made parallel algorithms (NVIDIA, 2017).

3.2.2 NVIDIA GeForce GTX 1080 Ti GPU Structure

The GeForce GTX 1080 Ti GPU (NVIDIA, 2016) is an advanced GPU based on the NVIDIA Fermi architecture (NVIDIA, 2009). This section will focus on a description of the GPU hardware as it pertains to the implementation of parallel processing on the GPU. For more information about the GPU design and the Fermi architecture, refer to the whitepaper doc-

uments published by NVIDIA.

Processing Structure

First, the computing structure of NVIDIA GPUs needs to be outlined. A parallel process or program to be run on the GPU is known as a kernel (NVIDIA, 2009). While the actual code may include multiple kernels and some intermediary commands, a kernel is the block of user written code that executes in parallel within the GPU.

The basic building blocks of programs in the GPU are threads (NVIDIA, 2009). Threads are like a thought process, they execute a set of instructions or a segment of code. Multiple threads are run simultaneously within the GPU to run a parallel program. Instructions are divided among a number of threads determined by the programmer, and the GPU then manages directing what processors within the GPU run the threads based on allocated memory and processing space available. The management of allocating threads to be run on different processors within the device is referred to as scheduling, and the GPU is limited to scheduling 32 threads at a time; this set of threads is called a warp (NVIDIA, 2016). Due to the structure of this implementation each thread within a warp must execute the same set of commands, however, the individual threads are capable of accessing separate memory locations.

While each thread has private memory available only to it, warps of threads are grouped into blocks in the program (NVIDIA, 2009). Each block has a shared memory that all the threads within the block are capable of accessing. Blocks that are executing the same kernel are then grouped into arrays called grids which have access to the device global memory (NVIDIA, 2009). A figure depicting the thread, block, grid hierarchy of a NVIDIA GPU is shown in Figure 3.2. Each grid, block, and thread has an assigned ID number so it can be referenced by the program. There is no limit to the number of blocks within a grid, but there is a limit on the number of threads per block. Each block is expected to be run on a processor and must share the memory available to that processor, therefore each block can have up to 1024 threads assigned to it (NVIDIA, 2019a). The specific limitations and capabilities of the GPU are defined by its compute capability as defined by the GPU manufacturer, the parameters

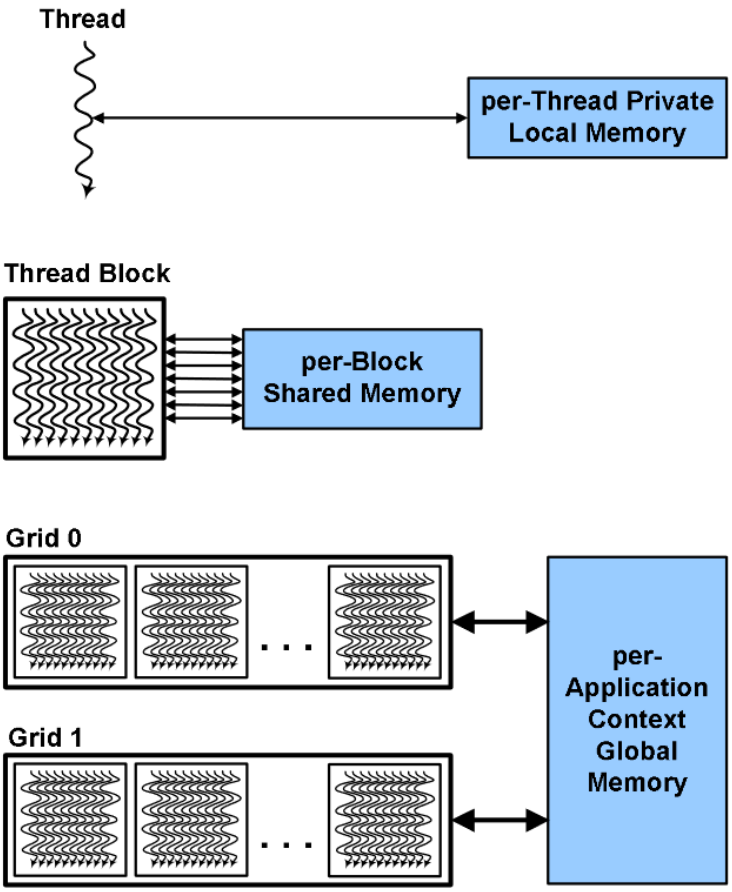


Figure 3.2: After (NVIDIA, 2009). Hierarchy of threads, blocks, and grids with corresponding memory access. Represents the programming structure of the NVIDIA GPU

of which are outlined in the NVIDIA toolkit documentation (NVIDIA, 2019a). The NVIDIA GeForce GTX 1080 Ti GPU has a compute capability of 6.1 (NVIDIA, 2019a).

Hardware Structure

The most important hardware unit of the 1080 Ti GPU is the streaming multiprocessor (SM), which can execute threads in the fashion described in Section 3.2.2. Figure 3.3 shows the block diagram layout of a single SM in the 1080 Ti GPU. The processing cores of the SM are called CUDA cores (NVIDIA, 2016) which comprise the main processing capability of the GPU. One SM is capable of running 2048 threads simultaneously, spread among the available cores on the device; the number of cores increases the computational throughput, but does not increase the number of concurrent threads the SM can handle (NVIDIA, 2009).

Figure 3.3: After (NVIDIA, 2016). The SM structure used by the NVIDIA GeForce GTX 1080 Ti GPU based on Fermi architecture. Instructions are stored in the instruction cache, while data is stored in the shared memory or the texture/L1 cache. The warp scheduler and dispatch unit receive commands from the instruction buffer and direct commands to cores or SFUs that are available. Load/Store (LD/ST) units manage moving data between the processing cores and SFUs and the device memory. Tex units are used by the GPU to manipulate images. Each SM contains 128 cores, 256 KB of register file capacity, a 96 KB shared memory unit, 48 KB of total L1 cache storage, eight Tex units, 32 SFUs, and 32 LD/ST units.

This means that in terms of threads and blocks, if each block contained the maximum 1024 threads allowable, the SM could only run two blocks. However, a maximum of 32 blocks can be run on a single SM (NVIDIA, 2019a), therefore the maximum thread count of 2048 could also be achieved by 32 blocks each containing 64 threads. Blocks and threads per block can be chosen in any fashion as needed by the program as long as they fall within these constraints.

Program commands are passed through the GPU and distributed among the SMs within the device through the instruction cache and buffer. Data being processed by the SM is transferred to local shared memory and processing begins as the warp scheduler directs the the processing cores and the Special Function Units (SFUs) to run scheduled threads. Load/Store (LD/ST) units control the transfer of data between the shared memory, L1 caches, and the private memory of the individual cores. L1 memory is a type of fast access memory that allows quicker read times than the main shared memory. Parallel programming of the ICEBEAR data analysis does not actively make use of any of the Texture (Tex) units within the device; however, built in commands within the GPU may make use of these units.

The 1080 Ti GPU has 128 CUDA cores per SM, and 28 SMs, for a total of 3584 CUDA cores (NVIDIA, 2017). It is capable of running 282,048 threads simultaneously. Beyond this maximum thread limit, since the minimum number of threads that can be scheduled is 32, blocks, grids, and the total number of threads must all be a factor of 2 in size. If not all of the threads in a warp are necessary for the kernel being executed, there will be idle threads in the warp that are scheduled but will not perform any computations.

Programming Example

The following example is used to describe how a sequential program can be implemented in parallel on the NVIDIA GeForce GTX 1080 Ti GPU:

Consider a matrix that is 100 x 100 elements and the value in each element must be doubled. One solution to this problem is to use a **for**-loop to step through each element of the matrix, multiplying the value in each element by 2. This would result in a **for**-loop that has 10,000 steps, with each step acting on a different element of the array. In parallel processing, these

10,000 individual steps can be instead run on 10,000 threads that can all be computed in parallel. Since this operation only needs to be performed once on a single matrix, only one parallel kernel is needed to solve this problem. Now the programmer must decide how to structure the 10,000 threads to be executed on the GPU.

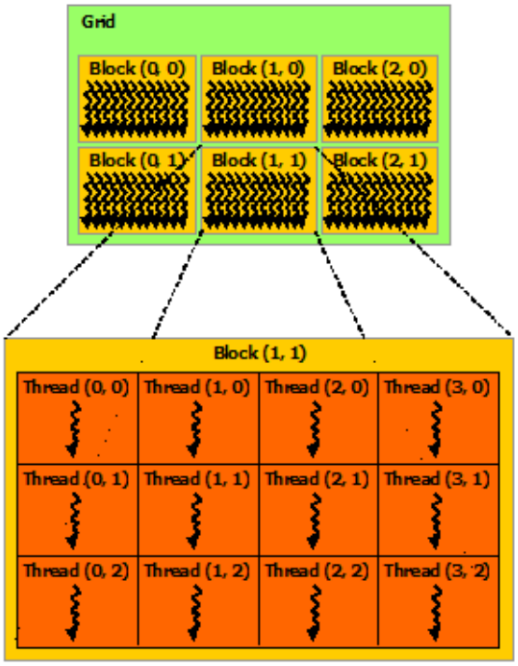
Since threads must be scheduled as warps, to obtain at least 10,000 threads 313 warps are needed for a total thread count of 10,016. A single block can have a maximum of 32 warps, so a single block with 313 warps is not possible. Since each SM can only run at most 32 blocks, the maximum number of concurrent blocks on the GPU is 896. The goal of this parallel design is to maximize occupancy of the GPU while at the same time minimizing the number of idle threads. For this example, it makes the most sense to use 313 blocks, each containing 1 warp. This allows all the threads to be processed at once, performing 10,000 arithmetic operations simultaneously while only having 16 idle threads.

Considering this is a simple problem, all the blocks can be stored in 1 grid for execution by the kernel. This example did not require more blocks than can be run on the GPU, but in the event that more blocks are required, the GPU scheduler holds the extra blocks in a buffer until there is processing space available to run the blocks. Figure 3.4 shows how a set of blocks and threads are identified, and how blocks are run based on the number of SMs in the GPU. Note that in Figure 3.4(B), each SM is only computing a single block for illustration purposes.

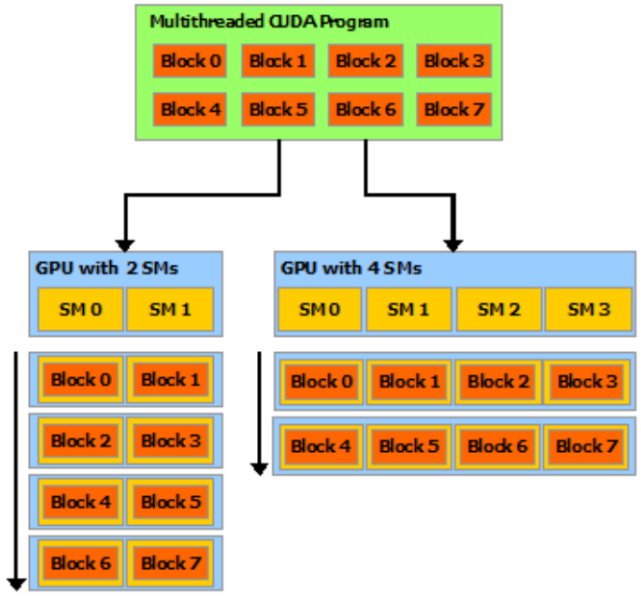
3.3 ICEBEAR GPU Optimization

3.3.1 GPU Language

There are multiple coding languages for GPUs, however the most prominent languages are CUDA, OpenACC, OpenMP, and OpenCL (Memeti et al., 2017). CUDA is a C++ based language developed by NVIDIA specifically for running parallel code on their GPUs, while OpenACC, OpenMP, and OpenCL are all open source languages for general GPU usage, also C++ based. These languages mostly have similar performance capability, with CUDA and OpenCL as the top performers. Language choice depends on the processing GPU and



(A)



(B)

Figure 3.4: After (NVIDIA, 2019a). Image (A) shows the structure of blocks and threads. The grid holds a 2 dimensional array of blocks, and each block contains a 2 dimensional set of threads, all of which are identified by an ID number. Image (B) shows the execution of a set of blocks on GPUs with different numbers of SMs. Fewer SMs in the GPU results in the same number of blocks taking longer to process.

how it is being used (Memeti et al., 2017). Since the ICEBEAR processing computer uses an NVIDIA GPU, CUDA is the preferred parallel processing language for this project and ICEBEAR analysis. CUDA was designed by NVIDIA, so there is less overhead programming required and the program commands are designed to utilize the NVIDIA GPU hardware and CUDA cores. Therefore implementing a parallel version of the analysis code is best done using CUDA. The CUDA language is freely provided by NVIDIA from their website (NVIDIA, 2019b). It comes with profilers to assess the GPU usage and performance, as well as debugging tools for programming. The most recent version of CUDA and the CUDA Toolkit, as well as all legacy releases, can be downloaded from the NVIDIA website.

3.3.2 Integration into Analysis

Section 2.3 in Chapter 2 presented the analysis procedure used to process ICEBEAR data. Here the steps taken to implement the calculation of a cross-correlation between two antennas on the GPU will be presented. The decimation, match filtering with the Pseudo-Random Noise (PRN) code, the Fast Fourier Transform (FFT), the cross-correlation, and the time averaging have all been implemented onto the GPU. While the cross-correlation analysis is implemented in CUDA, the programming language Python (Python-Software-Foundation, 2020) is used for loading the data to memory, combining the cross-correlation results, and plotting of the data. The optimized CUDA code and the Python wrapper that calls the CUDA code are in Appendix A. Figure 3.5 is a flowchart for the Python and CUDA portions of the analysis.

The match filtering process consists of a set of nested **for**-loops that involve multiplying the transmitter PRN code with the received measured voltages, and then performing repeated additions within the loops to perform the decimation. From the example in Section 3.2.2, this type of problem can be easily converted to parallel code with some additions to take into account the extra complexity of this situation. Since each thread and block is assigned an index, these indices can be used to represent the individual steps of a **for**-loop. As an example, for the stepping variables i and j in a nested loop, the thread with the 2 dimensional ID (9, 14) will execute the instance of the **for**-loop when $i = 9$ and $j = 14$. The cross-correlation

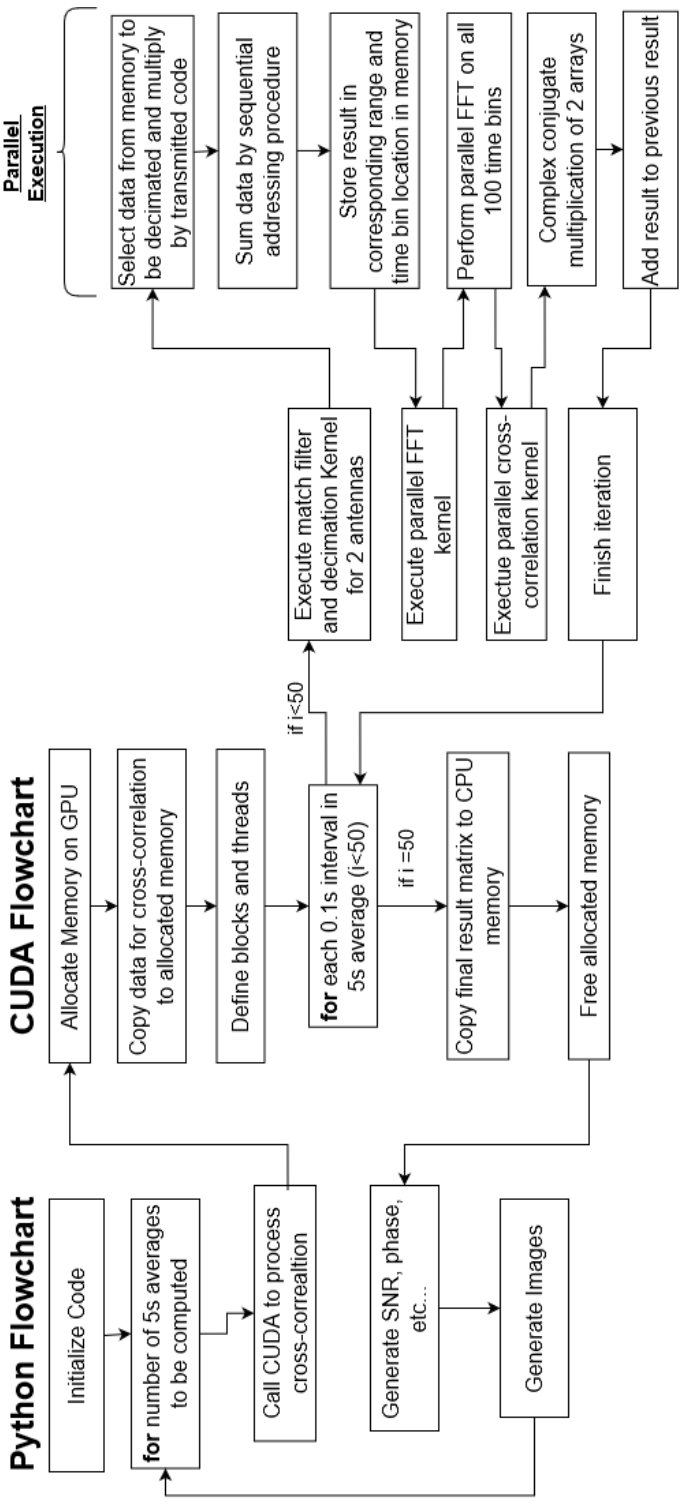


Figure 3.5: Analysis process flowchart. Analysis set-up and reading in data is performed by Python, which then enters into a loop to process analysis intervals. The time averaged cross-correlations are computed in parallel on the GPU using CUDA. When called, CUDA copies data from the CPU to the GPU, mixes, filters, decimates, and stores the resulting matrix, returning it to the CPU.

is also an element-wise combination of two matrices that can be easily made parallel. In addition, CUDA has built in functions that execute the FFT in parallel, so the entire range-Doppler cross-correlation analysis can be implemented in parallel processing on the GPU, except for the time averaging. The time averaging is implemented by adding each 0.1 s data matrix to the result variable to create a 5 s average. Since this method relies on the result variable to be updated after each matrix is processed this cannot be implemented in parallel.

First, the Python script reads in 5 s of data plus an extra 2000 data points (for a total of 1,002,000 complex voltage samples) for two antennas. Next, the result variable and the input data are converted to C type pointers that can be interpreted by the CUDA code. Finally, the Python script calls the optimized CUDA code to process the data and then stores the returned 5 s average range-Doppler matrix.

When the CUDA code is executed, it starts by allocating memory space on the GPU to hold the antenna data and the output result. It then transfers the data from the CPU to the GPU for processing. The 5 s time average is performed by a **for**-loop that calls the match-filter and decimation kernel, followed by the FFT and the cross-correlation, and then adds the result to the result variable for each 0.1 s matrix generated by the analysis. For the match filtering and decimation steps of the analysis, each element of the range-time matrix is generated by decimating 200 voltage samples that have been multiplied by the conjugate of the PRN code. The match filter and decimation kernel uses a grid that contains an array of 100 x 2000 blocks, so that each block represents one element of the output matrix. Each block then has 128 threads assigned to it. The first step of decimation reduces the 200 elements to 100 elements. This can be done on the same step that loads the data to the shared memory of the threads, so only 100 threads are needed per block and 28 of the threads will be idle. Once the kernel is invoked, shared memory is allocated to each block for faster read/write times for the threads.

The decimation is performed by implementing a technique known as sequential addressing reduction (Harris & NVIDIA, 2019). The principal concept is to perform a summation of all the elements in an array. First a stride value (also known as step size) is defined as half the length of the array. Elements of the array that are one stride apart are summed together and

stored in the first element. For example, an array of length N would have a stride value, s, of $N/2$. Then elements 0 through $N/2-1$ would be summed with elements $N/2$ through $N-1$ respectively and stored in elements 0 through $N/2-1$. Next the stride value is halved and the process is repeated until all data in the array is summed into the first element. Figure 3.6 provides a visual example of the sequential addressing reduction procedure for an example array length of 16. This method works only for arrays with a base 2 size, but the idea can be expanded to incorporate various array sizes. The initial method is used until the stride value can no longer be divided into a integer number, then the remaining elements are summed together.

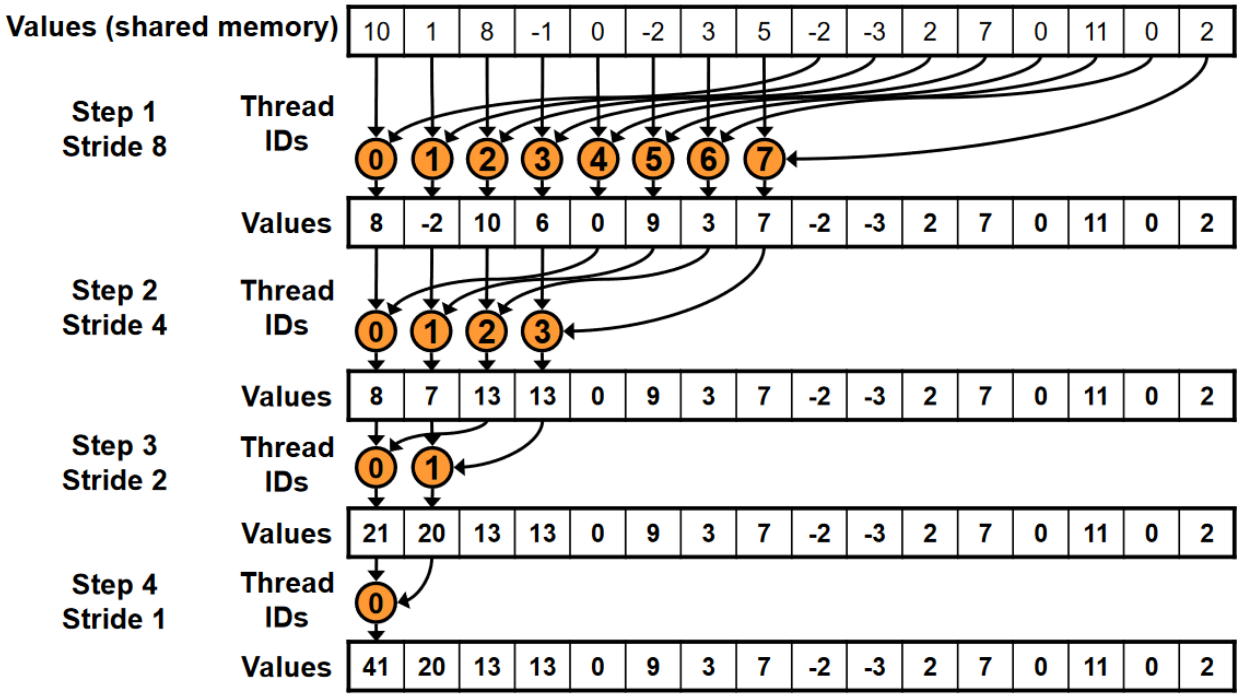


Figure 3.6: Sequential addressing procedure to decimate code in parallel. The starting stride value is half the array size, and reduces by half for each subsequent step until all data has been summed into the first array element.

ICEBEAR uses a decimation rate of 200, so each block of threads performs a sequential addressing reduction on a data array with 200 elements. The first step of the parallel reduction and the combination of the data with the PRN code takes place when the threads load the data from the GPU memory to the local block shared memory. This uses a stride value of 100, and then the decimation follows the steps of the sequential addressing reduction with

each thread running one element addition step. It is necessary to synchronize the threads between each step, waiting for all currently running threads in the block to finish their current commands, to prevent possible memory read/write errors as the next step of the decimation depends on the results of the previous. After the step with a stride value of 25, it can no longer be halved to an integer value. At this point the sequential addressing method is changed slightly, where every five elements of the remaining 25 elements are summed. The final step sums the last 5 elements to finish the decimation step.

Next, the parallel CUDA FFT is applied to each row of the range-time matrix, generating a range-Doppler shift matrix. Now the cross-correlation can be calculated by multiplying the range-Doppler matrix of one antenna with the complex conjugate of another antenna. Another kernel is called to execute the multiplication and complex conjugate in parallel. This kernel uses 391 blocks with 512 threads per block for a total of 200,192 threads. Since there are only 200,000 elements in the range-Doppler matrices, 192 threads will be idle in one of the blocks. In the kernel, the cross-correlation is calculated for each element of the range-Doppler matrix, and the result is added to the result matrix that will be the final return value of the CUDA code. This step adds the new cross-correlation result to the previous iterations to create the time average.

Once the time average loop completes, the result matrix is transferred from the GPU back to the Python script. This implementation of the ICEBEAR analysis makes use of the parallel capabilities of the GPU to greatly increase the data analysis speed.

The current decimation rate of 200 introduces many idle threads into the analysis since it is not a base 2 value, and changing the decimation rate to a base 2 value has been investigated. This would involve either padding the samples with zeros or dropping samples to adjust the array size to base 2 to accommodate the fact that a base 2 decimation rate would no longer evenly divide with the total number of samples which is a base 10 value. Computation time improvements of these implementations were minor, and the benefits were lost due to the extra time needed to accommodate decimating a base 10 sample set by a base 2 decimation rate. The decimation rate has been therefore left as 200 for ease of use and to avoid sample loss.

Future Improvements

There are two notable improvements that may be implemented in the future to expand the functionality of the code:

1. Implement a user defined decimation rate to account for possible changes to the radar.
2. Expand the design of the CUDA code to be usable by any NVIDIA GPU.

Changing the decimation rate may be necessary if the sampling rate of the ICEBEAR receiver is changed or the Doppler resolution needs to be changed for an experiment. This would require the addition of a method of determining the initial stride value and making the match filter and decimation kernel more modular. This is possible, but would be a complicated and involved process to design the code to still be efficiently using the GPU. The current sampling parameters and Doppler resolution are not expected to change for general ICEBEAR experiments, and small changes can be easily implemented through direct changes of the code.

The current code assumes a compute capability of 6.1 and to be used on the NVIDIA GeForce GTX 1080 Ti. CUDA has functions capable of identifying GPU parameters, so the code could be expanded to adjust processing parameters (such as thread and block limits) to work with the local GPU used by the CUDA code. This would allow for new GPUs to be installed in the ICEBEAR computer and allow the analysis code to still function as expected. The current GPU is sufficient for ICEBEAR analysis with the new CUDA code, but new GPUs will provide further improved computation times.

3.4 Validation

3.4.1 Processing Times

The original Python and C++ analysis code was typically able to compute a single cross-correlation averaged over 5 s of data in 3.71 s. Using the new CUDA analysis, the same 5 s averaged cross correlation can typically be computed in 0.0774 s, which is 48 times faster than the highly optimized C++ code. The scaling of computation time is linear with the

Language, Amount of Data Processed	Number of Baselines					
	1	**2**	**5**	**10**	**15**	**45**
C++, 5 s	3.71 s	7.42 s	18.55 s	37.10 s	55.65 s	166.95 s
CUDA, 5 s	0.08 s	0.15 s	0.39 s	0.77 s	1.16 s	3.48 s
C++, 8 hrs	5.94 hrs	11.87 hrs	29.68 hrs	59.36 hrs	89.04 hrs	267.12 hrs
CUDA, 8 hrs	0.12 hrs	0.25 hrs	0.62 hrs	1.24 hrs	1.86 hrs	5.57 hrs

Table 3.2: Comparison of computation time for a number of baselines. Processing 5 s of data and 8 hrs of data for ICEBEAR analysis for C++ code and GPU code. Times rounded to 2 decimal places.

increase in the number of cross-correlations computed. So the time of computation for any number of baselines can be calculated from these base times. Table 3.2 shows that the current parallel implementation has enabled many more baseline computations for analysis in a reasonable time frame compared to the original code. Multiple instances of the code can also be run simultaneously on the GPU. The scheduler on the GPU will allocate resources to all programs being executed on the GPU at any one time; however, processing speeds will be reduced for both programs. The current parallel implementation has enabled real-time computation of up to 64 baselines when one program is running. There are 45 unique baselines in the ICEBEAR 3D receiver antenna array, and only 9 unique baselines in the old linear receiver array, so all of the required ICEBEAR baselines can be calculated in real-time as shown in Table 3.2.

The NVIDIA GPU profiler (a program in the CUDA Toolkit) can be run with the CUDA program to determine how efficiently the GPU hardware is being used. Testing the analysis code with 10 iterations of the analysis process showed that the compute capability is near 49.6%. Figure 3.7 shows the output of the NVIDIA profiler. The compute capability is determined by comparing the total run time to the parallel kernel run time, so it should be kept in mind that this value also takes into account the unavoidable start up and allocation times. Of the parallel kernels executed, the match filter and decimation kernel is the most important, taking up 96% of the all kernel run time. The main aspects that lower the compute capability according to the profiler is idle threads during processing, and lack of simultaneous

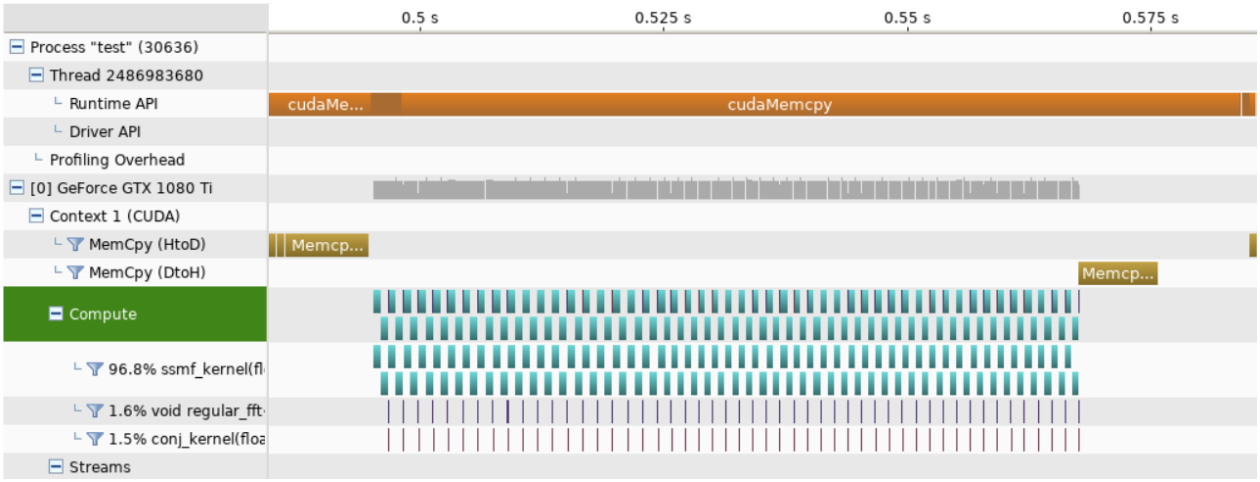


Figure 3.7: Output of the NVIDIA profiler program depicting total execution time for one 5 s time average. Shows Memory Copy (MemCpy) time for Host to Device (HtoD) and Device to Host (DtoH). In this scenario, the device is the GPU and the host is the CPU. Each call of the kernel and run time is also displayed. This snap shot depicts the program flow within CUDA on the GPU. This evaluation does not take into account the Python to CUDA interaction time. Only the CUDA execution is profiled.

memory transfers. The idle threads are mostly a result of the base 10 decimation rate, which was discussed in Section 3.3.2. Currently, all data for the processing is loaded to the GPU first, then parallel kernels are invoked. Once the data is processed, the results are transferred back. Further improvements could be potentially made to this implementation by staggering memory transfer, so that some of the data could be loaded simultaneously with some of the parallel processing. However, since all 45 unique baselines of the ICEBEAR receiver can be calculated faster than real time, the parallel GPU code has achieved the desired computation speed improvements.

3.4.2 C++ and CUDA Output Comparison

Since the data has been efficiently implemented in parallel, it is important to ensure that the numerical performance of the code has not been significantly altered by the change in analysis device and implementation. While some floating point differences are expected due to the different computational hardware, the main system output should be effectively identical. Figure 3.8 shows the computed SNR based on the analysis output of the original C++ analysis and the new CUDA analysis. Visual inspection between the plots shows that

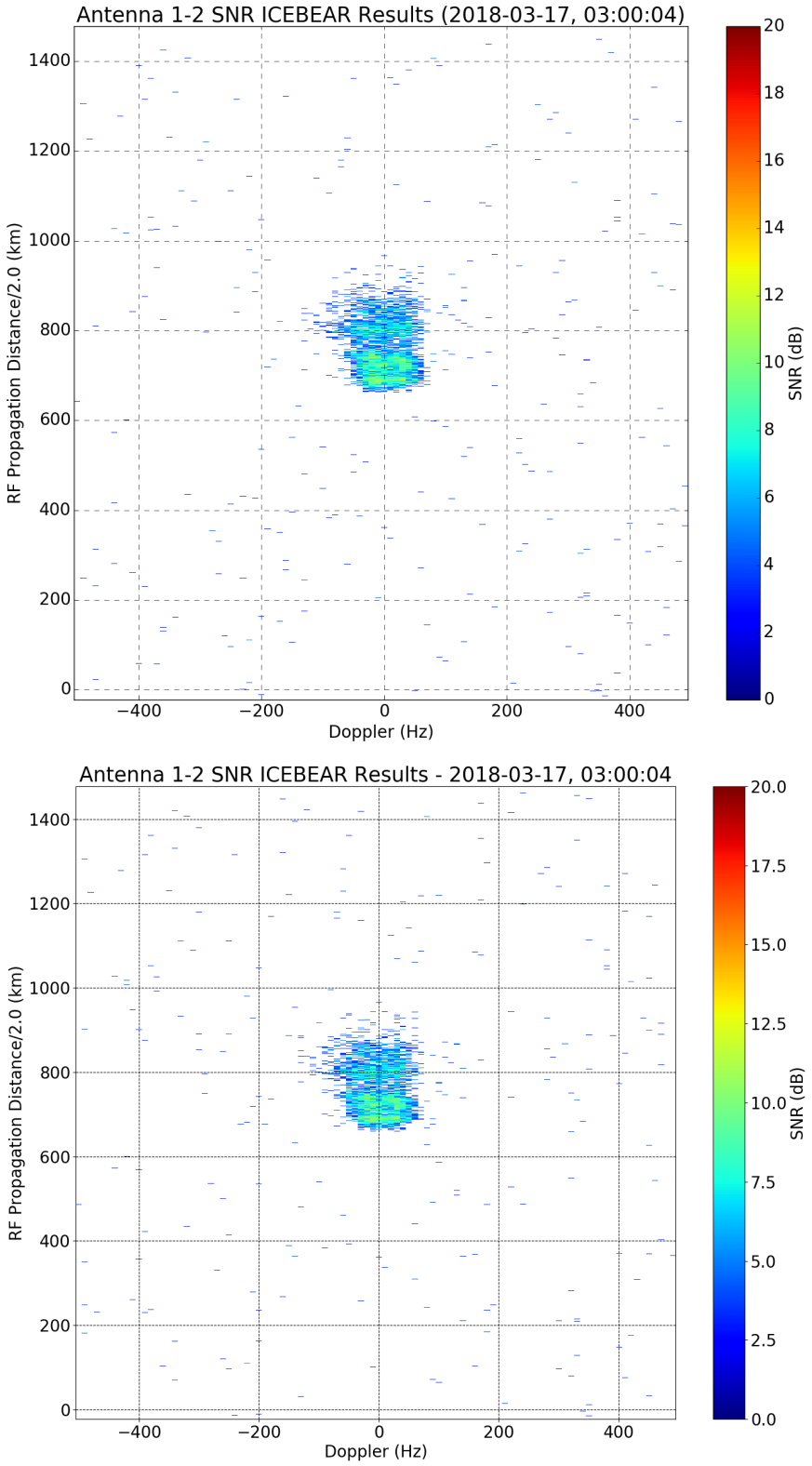


Figure 3.8: 5 s average SNR measured by ICEBEAR on 17 March 2018, 3:00:00 UTC. SNR calculated for receiver antennas 1 and 2 using the original C++ analysis code (top), and the new CUDA code (bottom). Only small variations in the SNR noise appear between plots, so both codes are shown to give the same result.

the two outputs are essentially identical. To compare the two data sets numerically the average difference is defined as,

$$\mu = \frac{\sum_{i=0}^{N} X_{c++}(i) - X_{CUDA}(i)}{N} \tag{3.1}$$

where μ is the average difference, N is the total number of data points, and X is the data calculated by the C++ or CUDA codes. The variance is then defined as,

$$\sigma^2 = \frac{\sum_{i=0}^{N} \left([X_{c++}(i) - X_{CUDA}(i)] - \mu\right)^2}{N} \tag{3.2}$$

where *sigma* is the variance of the difference. This comparison found that for the C++ and CUDA processed data $\mu = -0.016952 + j0.004263$ and $\sigma = 0.056683 - j0.001150$. For comparison, the average value of the range-Doppler values is on the order of 10^5, so on average the difference between the two code outputs is 7 orders of magnitude smaller and these are negligible differences that are a result of floating point errors in the calculation.

3.5 Summary

In this chapter the use of GPUs and parallel computing to improve the ICEBEAR analysis computation time was introduced. GPUs are specialized computer processors designed to quickly perform numerically heavy computations. While these devices were originally developed to provide real-time graphic simulation for video games, over the past 20 years GPU computing in scientific fields has become more prevalent. Implementation of numerical analysis into a parallel processing algorithm greatly improves efficiency of the calculations, improving the processing times of complex problems. The ICEBEAR analysis process is well suited to be implemented in parallel, which allows for real-time analysis capabilities.

The NVIDIA GeForce GTX 1080 Ti GPU was selected to be used as the processing GPU in the ICEBEAR data analysis computer. Programming of the GPU was done using the NVIDIA developed CUDA programming language as it was designed to best utilize the

NVIDIA GPU hardware and required less intricate understanding of the GPU by the programmer. Using CUDA, the cross-correlation analysis of the ICEBEAR antenna data was implemented on the NVIDIA GPU to improve the computation times. The original C++ analysis code was able to compute a 5 s averaged cross-correlation in 3.71 s, and the new CUDA analysis code computes the same data in 0.0774 s. This code is 48 times faster than the original C++ code and is now capable of computing all 45 unique baselines of the ICEBEAR receiver array in real-time. Comparison of the range-Doppler outputs of the CUDA and C++ code shows that the new analysis code does not change the numerical result of the analysis and accurately calculates the ICEBEAR data product. The GPU code can also be further generalized to be adaptable to changes in radar processing parameters such as the decimation rate, PRN code length, and range-Doppler matrix size.

4 Background Noise in ICEBEAR Measurements

4.1 Introduction

Understanding the noise in a radar system is crucial to understanding measured signals. Noise represents the sum of all external and internal sources that contribute to the measured data, but are not from the desired target. It is used in the computation of the signal to noise ratio (SNR) which was described by Equation 2.12 in Chapter 2.4 and is one of the primary methods used to assess and interpret ICEBEAR data. If noise is not properly quantified, the desired signal will be obscured by, and possibly lost behind, the background noise sources. Therefore, the noise of the system must be well understood so the SNR can be properly computed to account for noise and best represent the collected data. Through a combination of different processing methods, and the study of data collected over multiple times at various levels of ionospheric activity, a thorough study of ICEBEAR noise will be presented in this chapter with two main goals. First, to verify that the current method of determining noise is reliable and accurate. Second, to observe any possible patterns within the radar noise values to determine what the dominating source of noise is in the radar.

In this chapter, the possible noise sources that can influence a radar system, and how these sources influence ICEBEAR are outlined. Next, the current processing method of determining ICEBEAR noise and alternate noise calculation methods will be discussed. Since it is of interest to examine the noise during periods of high and low ionospheric activity, the global geomagnetic activity index, Kp, will be used as a measure of ionospheric activity. The data sets used in the analysis of ICEBEAR noise will then be presented. Using the different processing methods, a comparison of the different calculated noise data will be made to

verify the current ICEBEAR noise calculation. Finally, the daily and monthly trends of the noise value will be evaluated to identify the dominant noise sources.

4.2 Noise in Radar Systems

Noise is a term used to refer to data collected by the radar that is not a result of the desired target. These sources add to the desired signal power and need to be correctly quantified to accurately compute the SNR (Richards et al., 2010). There are various sources that contribute to the noise of a radar system:

1. **Thermal Noise:** Electrons are thermally agitated in electrical and RF components of the radar by the ambient and operating temperatures to generate a source of noise known as white noise. The magnitude of the electron oscillation is directly related to the temperature of the system and the bandwidth of the receiver (Hysell, 2018). Since the electron oscillation direction is random, thermal noise is a random noise source. Individual noise measurements will oscillate around zero power and phase, but will always average to zero phase and a constant power with a large enough sample size. No matter what voltage may be applied to the system, the thermal energy generated by the components and radiated from the surrounding environment will always generate thermal noise that must be considered in the data analysis.

2. **Terrestrial Radio Sources:** FM/AM radio, and HAM (amateur) radio operators are examples of terrestrial radio sources that may influence ICEBEAR noise. Any man-made device that outputs radio signals will have the potential to be detected by the ICEBEAR receiver depending on the frequency of the source. If the frequency falls within the bandwidth of the ICEBEAR radar, than the signal will be detected and add to the ICEBEAR noise.

3. **Cosmic Noise:** Cosmic noise, also known as sky noise, is radio signals that are constantly bombarding the Earth, coming from deep space. The strength and intensity varies as the Earth rotates and different regions of space become visible from the Earth (Hey, 1983). Cosmic noise, while not attributed to particular sources, is strongest from

the direction of the Milky Way as this corresponds to the densest concentration of stars observable form Earth (Ko, 1958). Cosmic noise is a low power noise source, with the exception of strong radio emissions from sources such as supernova remnants and radio galaxies.

4. **Self-Clutter:** Clutter can be defined as the signal from the transmitting radar waveform scattering from unwanted targets in the radar FoV (Richards et al., 2010). This may occur as signals scattering from different targets not just the range-Doppler bin of interest (e.g. an earlier transmitted signal scatted from a further range arrives at the same time as the current actual range of interest). It can also manifest as an artifact of the signal processing method, creating a noise floor in each range-Doppler bin relative to the strongest observed sources, causing the noise floor to increase for the whole radar (Hysell, 2018).

The dominant noise sources present in ICEBEAR varies based on ionospheric activity. The ICEBEAR data analysis technique has a peak power isolation of ∼28 dB for the Pseudo-Random Noise (PRN) code transmitted by the radar. This means that the power measured in a range-Doppler bin appears in the surrounding bins as a self-clutter noise source at a power 28 dB down from the source bin. As such, ICEBEAR therefore has a dynamic range of 28 dB, as all signals at powers less than 28 dB below the strongest measured signal will be lost under the self-clutter noise. The term noise floor is often used to describe the noise in the radar system, defined as the sum of all noise sources present in the radar. During active ionospheric periods when instabilities form, this self-clutter noise source will dominate (other noise signals will still be present just at lower powers) when the scattering targets are sufficiently strong. When the radar is operated during quiet ionospheric conditions during which instabilities cannot form, only background noise will be measured on the receiver and the ICEBEAR processing self-clutter noise floor will be 28 dB lower than the measured background noise. In quiet ionospheric conditions, it is expected that cosmic noise is the dominant source of noise. Due to the noise-like transmission from modulating the ICEBEAR transmission with the PRN code, ICEBEAR signals do not interfere with other radio systems. The match filtering performed with the PRN code allows ICEBEAR to strongly reject signals

from terrestrial noise sources when ionospheric scatter is detected (Sulzer, 1989). Thermal noise is always present but is expected to be weaker than the cosmic noise.

In examining the noise of a system it is important to understand how the radar collects data. ICEBEAR is a bi-static coherent radar, which means that its transmitter and receiver are spatially separated. As discussed in Chapter 2, ICEBEAR is designed to measure E-region ionospheric plasma instabilities. Plasma structures with wavelengths approximately half the wavelength of the transmitted radar waveform frequency will be coherently scattered and subsequently observed by the radar receiver.

4.3 Determining Noise

ICEBEAR noise is assessed after determining the cross-correlation between two receiving antennas; the process for this calculation is described in Chapter 2, Section 2.3. The cross-correlation output, referred to as the range-Doppler matrix, is a 2D matrix of complex values that determine the power and angle of arrival (AOA) for each range-Doppler bin in the radar field of view (FoV). Each row and column in this matrix represent the range and Doppler bins respectively. There are 2000 range bins, with 100 Doppler frequency bins per range bin (1.5 km range resolution and 10 Hz frequency resolution respectively), such that the matrix is 2000 rows by 100 columns, for a total of 200,000 elements.

The noise power levels calculated in this analysis are not absolute, but arbitrary in that they depend on the (relative) raw voltage magnitudes collected by the receiver. The receiver cannot make absolute measurements due to technical limitations, such as temperature fluctuations in the equipment, and physical limitations as the size of the scattering target or its scattering cross-section. Both of these limitations lead to fluctuations in the measured power. For this reason only the relative differences between measurements has physical meaning, while the absolute value is arbitrary. The noise power measurements are presented in decibels, dB. Calculation of the noise power is based on the following expression,

$$dB = 10\log(P) \tag{4.1}$$

where dB is the power in decibels, P is the calculated power from the ICEBEAR processing steps. Note that this is not SNR like in Chapter 2 Equation 2.12. Just the relative "power" measurements of the noise is still expressed in dB For power measurements made using equation 4.1, a 3 dB difference indicates that one measurement has twice the power.

As discussed in Chapter 1, coherent radars operate by collecting the transmitted signal after it coherently scatters off of plasma structures in the ionosphere. The scattering condition requires that the propagation of the radio waves must be nearly perpendicular to the magnetic field lines of the Earth (Hunsucker & Hargreaves, 2003). The angle between the magnetic field lines and the direction of radio wave propagation is defined as the aspect angle, with 0° indicating propagation directly perpendicular to the field lines. As the aspect angle increases, the signal strength of reflected signals will decrease. Classically, minimal scatter is expected from aspect angles greater that a few degrees (Kustov et al., 1994). Scatter from aspect angles more than 5° is expected to be at least 20 dB weaker than scatter from and aspect angle of 0° (Hall & Moorcroft, 1992). Following the aspect angle sensitivity, the ICEBEAR radar is configured so that at an altitude of ∼105 km, the aspect angle for the transmitted waveform propagation is 0° near the center of the radar FoV. While there have been observations of coherent ionospheric scatter measured at high aspect angles, this occurrence can be attributed to high E-region densities and signal refraction (Uspensky et al., 1994). At altitudes above and below the E-layer of the ionosphere, either no instabilities can form or the aspect angle will be too large for scatter. Therefore, the near and far ranges in the ICEBEAR range-Doppler matrix are not expected to contain any coherent scatter from the ionosphere and should contain only noise. The same can be said for the edges of the Doppler frequency spectrum, as the plasma motion is not expected to be fast enough to cause a frequency shift greater than 400 Hz except during very active ionospheric conditions.

Clutter Correction

For typical ICEBEAR processing, once the range-Doppler matrix is computed (Chapter 2.4 Section 2.3.2) a clutter correction is applied to the data. An average is taken of the first 100 range gates (150 km) across all Doppler frequencies and is subtracted from all the elements of the range-Doppler matrix to correct for bias introduced by self-clutter from the radar and

polarized cosmic noise (Conway, 1974).

The noise value is then calculated by taking the median of all 200,000 elements of the range-Doppler matrix. This FoV median is used instead of an average so that the noise value will not be biased by stray data points with high power. Sources of high power measurements, such as meteor echos and the ionospheric plasma, can bias the noise calculation when assessing the entire FoV, or if the signal appears in the data averaged together. Using the FoV median value will minimize the influence of high power measurements on the overall trend of the background noise, while still providing a large sample size to calculate the noise. This research is to determine if this median technique of calculating noise for ICEBEAR, which is computationally efficient, gives the same, or sufficiently similar, results as averages over the FoV where scatter is not expected (extremely close or far ranges, high Doppler velocity, etc.).

4.3.1 Analysis Methodology

There were two main objectives in the analysis of ICEBEAR noise:

1. Determine if the current method of calculating noise is representative of the noise in the radar measurements.

2. Determine if there is a pattern or cycle of the noise that repeats daily, or over the period of a year. Specifically, to identify if there is a diurnal trend or if thermal noise is the dominant noise source.

The current FoV median noise calculation can be verified by comparison to alternate methods of calculating the noise. To identify patterns and the dominant noise source, the noise for various days and months of the year were calculated and compared. Both objectives are subject to clutter from ionospheric scatter biasing the noise values during active periods. Determining the noise is most important during active ionospheric periods for the calculation of SNR, thus active periods were analyzed. However for pattern detection, the presence of ionospheric signals will overpower any underlying trend due to self-clutter, so observations during quiet periods is essential. Therefore a combination of active and quiet ionospheric

periods were evaluated at various times to ensure the noise was properly quantified.

Noise Calculation

The typical method of determining noise for a radar is by evaluating sections within the FoV that are not expected to contain the scattered radar signals as discussed above. As no ionospheric scatter is expected for aspect angles $> |\pm 5°|$, the ranges <400 km and >1500 km should only contain noise. Similarly, large Doppler frequencies are generally not expected in the radar measurements (except during extremely geomagnetically disturbed periods), so the highest frequencies calculated should also contain only noise. The low level data product, the range-Doppler matrix, allows for various combinations of range and Doppler noise calculations to be considered. In this analysis, five different averaging methods were used to compute the noise in the ICEBEAR radar and compare to the FoV median noise:

1. Average Doppler frequencies -500 to -400 Hz across all ranges (Negative Doppler)
2. Average Doppler frequencies 400 to 500 Hz across all ranges (Positive Doppler)
3. Average ranges 0 to 300 km for all Doppler frequencies (Close Range)
4. Average ranges 2700 to 3000 km for all Doppler frequencies (Far Range)
5. Average all range and Doppler frequencies measured (FoV average)

As only noise should be present in the different selective averaged ranges and frequencies (methods 1 through 4), the different averaging methods should produce equivalent noise results, as will be shown. The FoV average provided a comparative basis for the FoV median calculation as well as a reference for the other averages. During quiet periods with no ionospheric scatter the FoV average is expected to be the same as the other averaging methods, however, the FoV average is expected to be higher during active periods as it will include ionospheric scatter in the average. The different range and frequency noise averages were compared to the median and average of the whole FoV noise calculations to determine the best technique for determining the noise both computationally and scientifically. Once this basic noise calculation was confirmed, noise properties such as diurnal trends were investigated.

To examine the effect of the clutter correction on the noise values, all averages, as well as the FoV median calculation, were calculated twice. Once before the clutter correction was applied to the range-Doppler matrix, and once after. This provided details on the quality of the clutter correction with regard to the noise calculation.

ICEBEAR Data

Chapter 2, Section 2.3 described the procedure used by ICEBEAR to generate a range-Doppler matrix for every 0.1 seconds of measurement, and then coherently average matrices together to reduce noise. Nominal processing uses data averaged over 5 seconds, which is an element-wise average of 50 consecutive range-Doppler matrices, such that 720 noise values are generated over an hour period. In this study, noise values are generated from these 5 second averages using the median and averaging calculation methods discussed over the periods listed in Table 4.1 using the available ICEBEAR data. Both the ICEBEAR-linear and ICEBEAR-3D antenna configurations were examined to properly compare the noise and examine any changes that may have occurred when the antenna configuration was changed. The ICEBEAR-linear configuration time periods were calculated using 1λ as a common basis to compare results. Since the ICEBEAR-3D configuration does not have a 1λ baseline both ICEBEAR-3D and ICEBEAR-linear configurations were processed using an antenna pair with a 4λ baseline so that the two array configurations can also be compared to each other directly. This also provided some insight into how the noise values may change between different baselines by comparing the 1λ and 4λ results from the linear array.

The activity level of the ionosphere is based on the Kp index. This is a measure of global geomagnetic activity which is computed using measurements from 13 stations from around the world (Elliott et al., 2013). When the K$p\geq$ 3 the ionosphere is considered to be active and scatter is typically observed by ICEBEAR. Therefore, in this analysis when K$p\geq$3 it is considered an active ionospheric period, and while K$p\leq$2 it is considered a quiet ionospheric period. Kp values were retrieved from the GFZ-Postdam ftp server (GFZ-Helmholtz-Center-Postdam, 2020) for periods of ICEBEAR observation to identify quiet and active times. A balanced combination of quiet and active periods is ideal for this noise evaluation.

Antenna Configuration	Start Date (year-month-date)	Start Time, UTC (hr:min:sec)	Period (hr)
Linear	2018-01-19	19:00:00	70
Linear	2018-02-18	00:00:00	62
Linear	2018-03-07	00:00:00	84
Linear	2018-04-10	21:00:00	65
Linear	2019-07-10	18:00:00	26
3D	2019-10-23	22:30:00	57.5
3D	2019-10-26	22:30:00	41.5
3D	2019-10-29	19:00:00	26
3D	2020-05-29	00:00:00	48

Table 4.1: Data sets used in examination of the ICEBEAR radar noise

4.4 Processing Results

4.4.1 Mean Noise Value Comparison and Clutter Correction

For the data sets listed in Table 4.1, the averaging methods and the FoV median noise of the range-Doppler matrix has been calculated. Figure 4.1 shows the noise power for all processing methods calculated over a quiet period on 2018 March 7 and 8 which was a quiet ionospheric period with Kp<2. The baseline using antennas 1 and 2 (which have a 1λ spacing) from the ICEBEAR-linear antenna configuration was used to calculate a cross-correlation range-Doppler matrix. The top figure is the power before the clutter correction is applied to the range-Doppler matrix, and the bottom figure is the power calculated after the clutter correction is applied. Consider first the top figure, over the 2 days of data presented there is a repeating pattern observed. Starting around 2:00:00 UTC the noise power increases significantly to 5:00:00 UTC, where noise power peaks at a level 2 dB higher than the starting data. The noise power then decreases at a similar rate until around 8:00:00 UTC where the noise levels out at power similar to the starting values. At 15:00:00 UTC the power decreases again, this time by only 0.5 dB, reaching the a minimum around 18:00:00 UTC. The noise

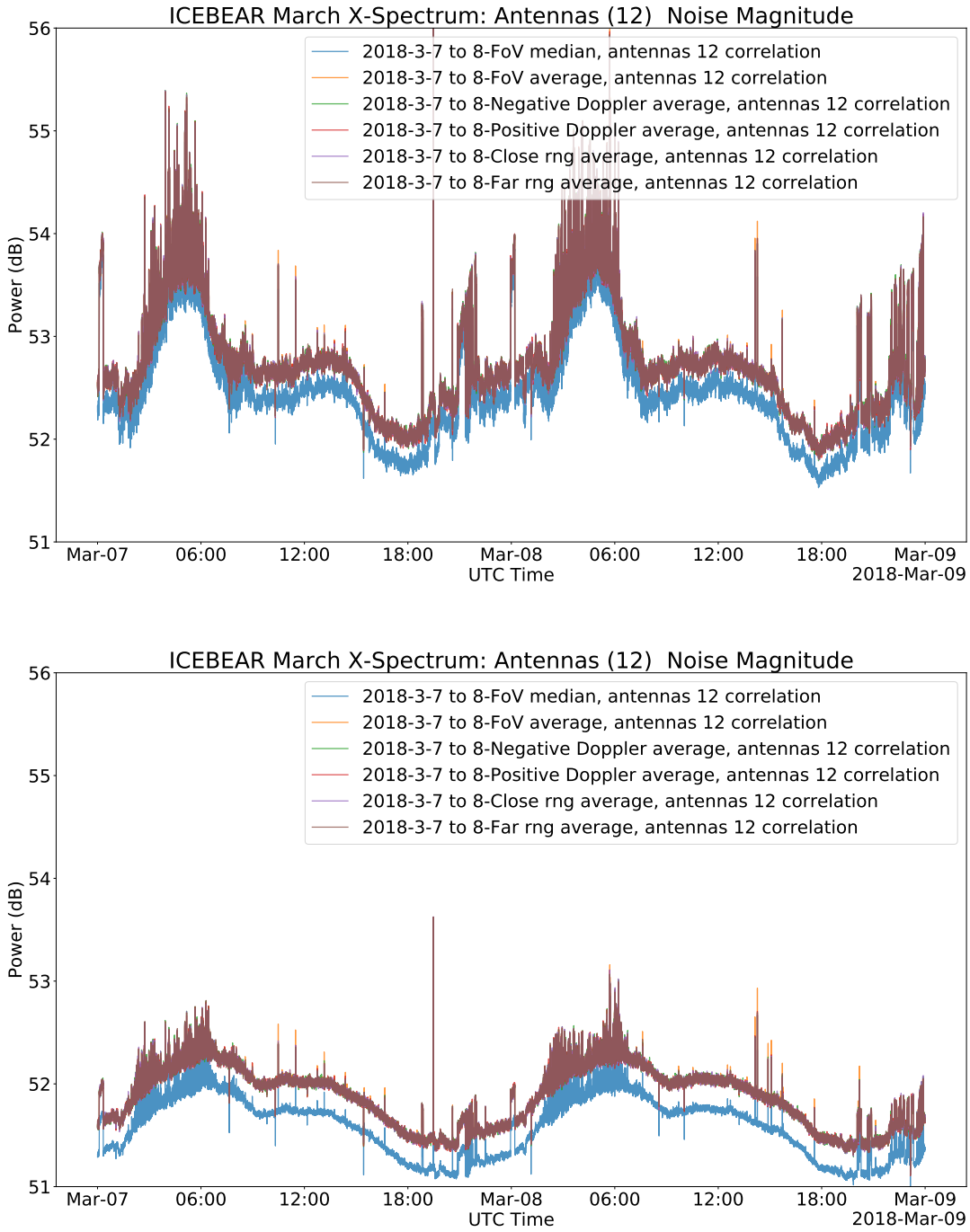


Figure 4.1: Raw noise power (top) and noise power after clutter correction (bottom) calculated using multiple averaging methods and the FoV median, on 7−8 March 2018. This is a quiet period with a 1λ antenna separation for the linear ICEBEAR antenna layout configuration. All averaging methods show essentially identical noise values. The FoV median noise power is weaker but follows the same trend. The clutter correction reduces biasing effects of strong clutter in the noise, causing the overall trend in the noise floor to become more pronounced.

power then gradually increases back to the starting level such that the pattern then repeats at 0:00:00 UTC of the next day. The bottom figure depicts a similar trend, only the clutter correction has reduced the total power and variation of the pattern. Starting from a base noise power at 0:00:00 UTC the noise gradually increases starting around 2:00:00 UTC to a peak at 5:00:00 UTC that is 1 dB higher than the starting value. Then noise power then beings to decrease at a similar rate until 8:00:00 UTC, where the power remains constant until 3:00:00 UTC at a power level 0.5 dB higher than the starting value. After 12:00:00 UTC the noise power slowly decreases to a minimum around 18:00:00 UTC at a level 0.25 dB lower than the starting noise. Finally the noise power increases until it has returned to the starting value at 0:00:00 UTC of the next day.

In both figures it is clear that all the averaging methods produce near identical results, so plotted noise values overlap. This is expected during a quiet period, as the range and frequency averages should only contain noise and so will the full FoV average as no ionospheric scatter is present, though some meteor trails may be observed. Since all the averages produce similar results, this confirms there are no variations in the noise parameters based on location within the FoV. The median value shows the same pattern as the averaged values, but has a consistently lower power value, due to statistical fluctuations in the noise. This shows the FoV median noise consistently trends lower than the averaging methods. While the average of the data is higher, there are more data points with lower power in the FoV.

Considering the clutter correction data in the bottom plot of Figure 4.1, the relationship between the median and average noise values is not changed compared to the top plot, however, the overall trend has changed. In the top plot, the strongest noise power occurs around 5:00:00 UTC, with the rest of the data showing a decreasing, then increasing trend between these peaks. Comparing this to the clutter correction data in the bottom plot, the relative strength of the 5:00:00 UTC peaks have been reduced by the clutter correction, causing the decreasing/increasing trends of the noise power to become more pronounced compared to the 5:00:00 UTC peak. This suggest the clutter correction is reducing the effect of strong biases in noise data by reducing the power for each measurement in time. Any bias that increased the noise floor at that time will also increase the magnitude of the clutter

	FoV Median	FoV Average	High Freq	Low Freq	Close Range
Quiet Uncorrected	0.059%	0.0033%	0.0048%	0.0048%	0.0049%
Quiet Corrected	0.070%	0.0032%	0.0045%	0.0045%	0.0047%
Active Uncorrected	0.013%	0.097%	0.0062%	0.0047%	0.0077%
Active Corrected	0.016%	0.22%	0.0070%	0.0054%	0.0056%

Table 4.2: Percent difference of noise power calculation methods compared against power calculated using the far range method. Results are presented for quiet and active ionospheric conditions, and for noise calculated before and after the clutter correction is applied. Demonstrates the similarity of the FoV median method to the other averaging methods, and that the median becomes more similar during active ionospheric conditions.

correction, so the data is scaled down relative to the noise power measured. The numeric comparison of these methods in quiet/active periods and for corrected/uncorrected results is shown in Table 4.2. The work by Adam Lozinsky has shown that far ranges are much less affected by meteors (Lozinsky, personal communication, December 20, 2020) and should contain only noise sources, so it is used as a reference for comparison. Using this reference, the percent difference of the other noise calculation methods is calculated using,

$$\% = \frac{1}{n} \sum_{i} \left| \frac{N(i) - N_r(i)}{N_r(i)} \right| * 100 \tag{4.2}$$

where % is the percent difference between the compared values, N is a data set with i elements that is compared against the reference data set N_r, and n is the total number of elements in the data sets. These comparisons further demonstrate the conclusions mentioned above, where the FoV median method calculates noise data that differs the most, but is still very similar to the noise calculated by the other methods.

Similarly, examining the noise power during an active period will show how the different noise calculation methods compare during times when ICEBEAR data contains ionospheric scatter. Figure 4.2 depicts the noise power calculated for a 30 minute segment of an active period (Kp>3) on 10 March 2018 using the 1λ baseline between antennas 1 and 2. The top and bottom figures, like in 4.1, show the power with and without the clutter correction

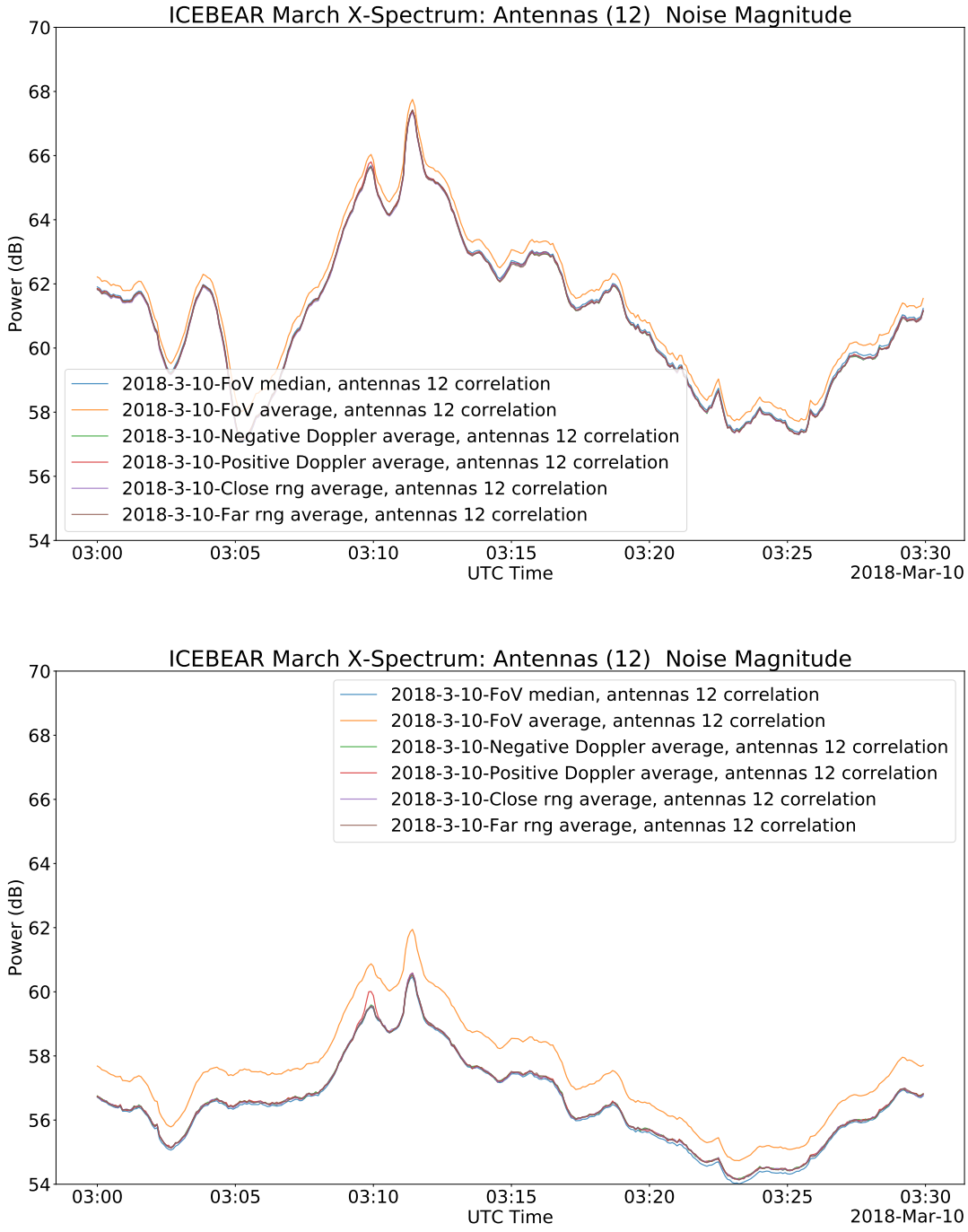


Figure 4.2: Raw noise power (top) and noise power after clutter correction (bottom) calculated using multiple averaging methods and the FoV median, on 10 March 2018. This is a 30 minute period of high ionospheric activity measured using the linear ICE-BEAR antenna layout configuration with a 1λ antenna separation. Before the clutter correction, the median and average powers match during the active period, with the FoV average trending very slightly larger than the other noise calculation methods. The FoV average is slightly more distinct from the other methods after the clutter correction is applied.

applied. Only 30 minutes of an active period is shown so that the variability of the noise can be clearly displayed, but increased power level compared to the quiet periods is persistent through the active period. The active period has much more variability in the noise power than the quiet period, with a maximum difference in power levels of 10 dB while the quiet period has a maximum difference of only 1.5 dB. During active periods when ionospheric scatter is present, the full FoV average power includes high power ionospheric scatter in the average (as it averages across all data in the measurement) so the noise power generated by this method is greater than the rest of the noise power generated by the other averaging methods which do not include ionospheric scatter, as is expected. The FoV median noise has also changed and now better agrees with the averaging methods that average over only a section of ranges or frequency shifts. As was mentioned in Section 4.2, ICEBEAR has a dynamic measurement range of 28 dB. When strong ionospheric scatter is present during active periods, the power levels measured by the radar are much higher than during quiet periods, so the self-clutter caused by the dynamic range comes into affect as the noise from quiet periods is lower than the 28 dB down noise floor generated by the signal processing technique. The self-clutter from the processing technique applies a very consistent noise floor across the FoV and is the dominant noise source during active ionospheric periods. This consistent noise floor is what causes the median FoV technique to better agree with the noise values calculated by the averaging techniques. Table 4.2 also includes a percent difference comparison of the noise techniques calculated during an active period.

The phase difference measured on the 1λ baseline between antennas 1 and 2 from the ICEBEAR-linear antenna configuration is shown in Figure 4.3 and Figure 4.4. These figures depict the same quiet period as shown in Figure 4.1 (7 March 2018, 0:00:00 UTC to 9 March 2018 0:00:00 UTC) and the active period shown in Figure 4.2 (10 March 2018, 3:00 to 3:30:00 UTC) respectively. The top and bottom plots in these figures depict the noise phase before and after the clutter correction is applied. In the top plot of Figure 4.3, a pattern is observed in the phase measurement that repeats each day. From an initial starting phase value at 0:00:00 UTC, the phase quickly decreases to a minimum about 50° below the starting value, then quickly increases to a maximum about 50° above the starting value, and finally returns to the starting value. This cosine-like pattern goes from 0:00:00 UTC to

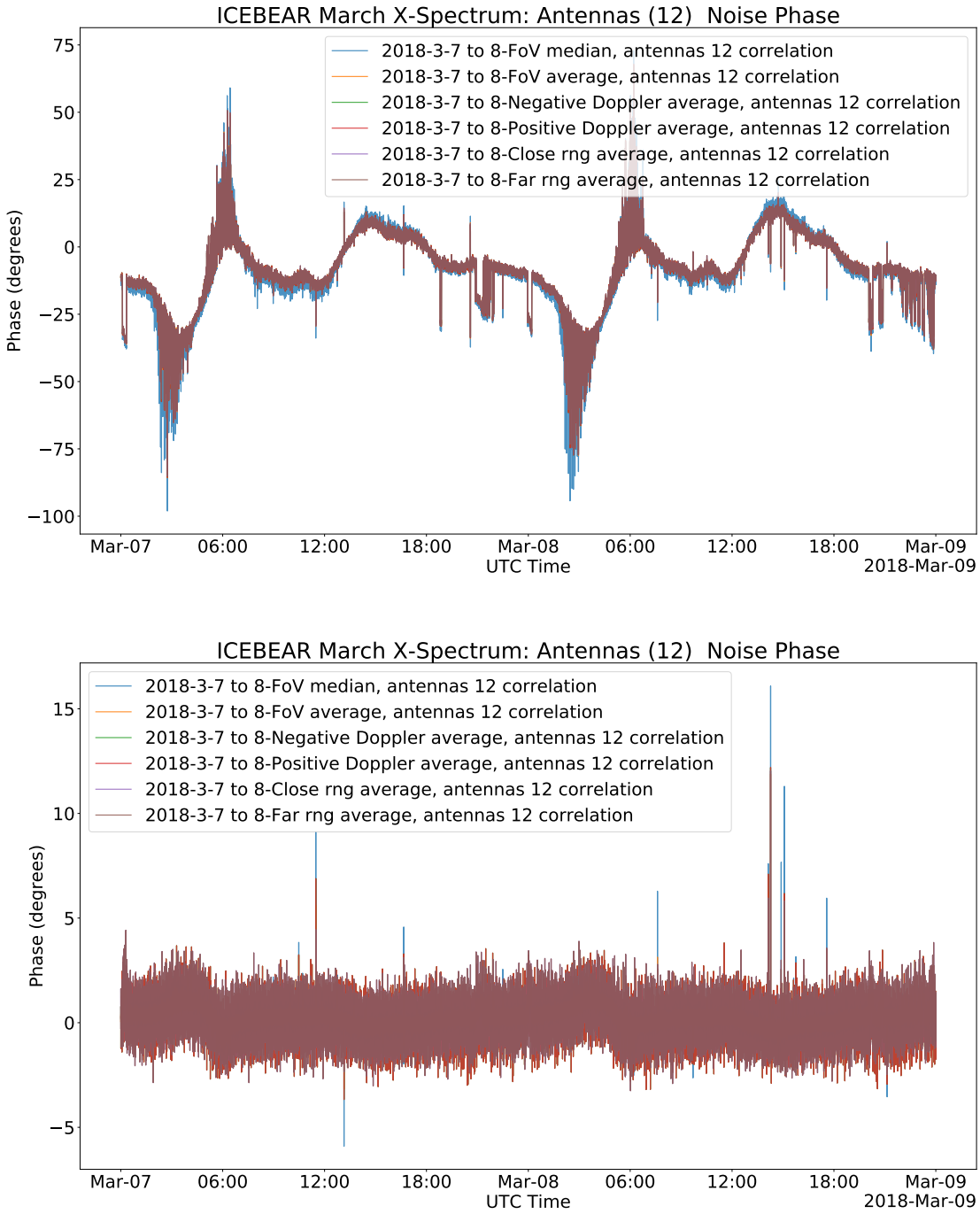


Figure 4.3: Raw noise phase (top) and noise phase after clutter correction (bottom) calculated using multiple averaging methods and a FoV median on 7−8 March 2018. This is the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The clutter correction suppresses the background phase variation from the noise sources. All noise averaging methods as well as the noise median method follow the exact same trend with comparable magnitudes. The clutter correction removes most of the phase bias.

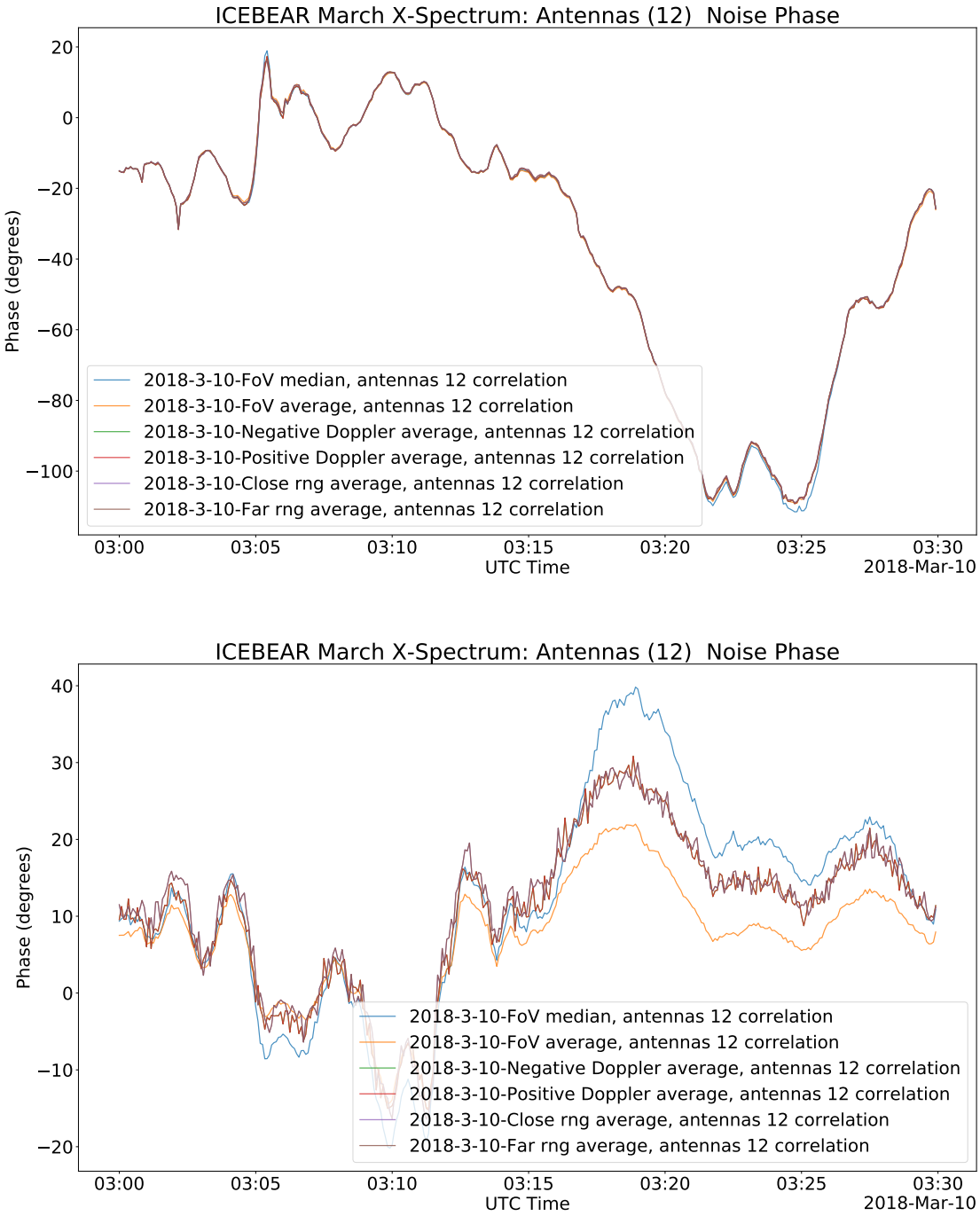


Figure 4.4: Raw noise phase (top) and noise phase after clutter correction (bottom) calculated using multiple averaging methods and a FoV median on 10 March 2018. This is a 30 minute period of high ionospheric activity measured using the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The clutter correction, unlike in the quiet data, does not suppress the phase activity, but changes it due to the presence of scattering signals shifting the clutter correction based on how they are interpreted in the noise calculation technique.

8:00:00 UTC. The phase is then steady around the staring value until 12:00:00 UTC when it increases by 20° over the course of an hour. The phase then slowly decreases until 0:00:00 UTC on the next day, returning to the starting phase value and the pattern repeats. The bottom plot of this figure has no discernible pattern and the phase varies randomly around 0° with a max offset of ±4°. This demonstrates that the clutter correction removes the phase effects of noise from quiet ionospheric periods. Figure 4.4 has a very different phase response as this data occurs during an active period. Since the self-clutter from the ICE-BEAR dynamic range is the dominant source there is no pattern to the phase measurements. Instead, the phase measurement depends on the ionospheric scatter. When the clutter correction is applied to the data, it causes the phase values to change significantly between the top and bottom plots, but there is still no pattern to the phase value.

The phase of the measured signals at the antennas are a summation of all the AOAs and phases from incoming scatter and noise, with signals that have higher powers biasing lower power phase measurements to the phase and AOA of those higher power signals. Consider first Figure 4.3 which displays a quiet ionospheric period. In the top plot with uncorrected phase, it is clear that all noise calculation methods produce identical results as all the data sets overlap. Once the clutter correction (see Section 4.3 for details) is applied, the majority of the phase fluctuation is removed and the noise becomes a near constant value over the entire interval. Table 4.3 shows the percent difference comparisons of the noise methods described in Section 4.3.1 against the close range average method for the phase values, as was presented in Table 4.2 for noise power using Equation 4.2. Here it again is shown that all these noise calculation methods generate nearly identical results. As is seen in the half hour period with a Kp>3 shown in Figure 4.4, the noise phase becomes much more chaotic during active periods compared to the trends observed in Figure 4.3. Scatter from ionospheric structures has very high signal power compared to other noise sources, and the noise floor phase is very sensitive to these high power signals. When this scatter is present, the AOA of the scattered signal will dominate the phase angles, so the usual daily variation is lost. Comparing the noise phase before and after the clutter correction, and comparing quiet and active periods, shows that the noise phase is very sensitive.

	FoV Median	FoV Average	High Freq	Low Freq	Far Range
Quiet Uncorrected	0.61%	0.11%	0.16%	0.16%	0.0%
Quiet Corrected	0.18%	0.17%	0.25%	0.25%	0.0%
Active Uncorrected	0.19%	0.12%	0.082%	0.082%	0.0%
Active Corrected	1.00%	1.12%	0.36%	0.36%	0.0%

Table 4.3: Percent difference of noise phase calculation techniques compared against phase calculated using the far range technique. Results are presented for quiet and active ionospheric conditions, and for noise calculated before and after the clutter correction is applied. This demonstrates the similarity of the FoV median technique to the other averaging techniques, and that the phase pattern is very slightly disturbed by active ionospheric conditions

In the calculation of the SNR only the power of noise measurements are used, as the SNR is a ratio of of the signal and noise power. Noise phase is not used in the calculation so the phase sensitivity does directly impact SNR. However, the phase sensitivity is still a very important aspect of ICEBEAR measurements. In the ICEBEAR data analysis cross-correlation range-Doppler results, phase is used to determine the AOA of a signal using interferometry (see Chapter 2 Section 2.3.1), so it is important to understand the effect that noise sources have on ICEBEAR measurements. The high phase sensitivity allows the radar greater resolving capability for determining AOA, but also means the radar is more sensitive to the effects of noise biasing phase measurements. Figure 4.3, demonstrated that there is a diurnal trend (described in detail the next section) in the noise phase measurements. During active periods, like the half hour period shown in Figure 4.4, this diurnal trend is overpowered by the ionospheric activity and self-clutter from the ICEBEAR data processing technique. Because there is no ionospheric scatter during quite ionospheric periods, the self-clutter noise will be much less than the other noise sources as it is always 28 dB less than the highest power measured, and the highest power measured during quiet periods is just noise from other sources. These sources are terrestrial, such as HAM operators and local FM and AM radio, and the polarized cosmic radio signals from outer space, most notably the Milky Way galaxy (Ko, 1958). Thermal noise has a random phase pattern, so it is clear from the top plot in Figure 4.3 that thermal noise is not the dominant source as there is a distinct pattern

in the phase measurements. Cosmic noise has a diurnal pattern (Ko, 1958) while terrestrial noise will have a random pattern depending on the source, so cosmic noise is clearly the dominant source of noise during quiet ionospheric periods.

During quite ionospheric periods cosmic radio signals are the dominant noise source, and during active ionospheric periods self-clutter is the dominant noise source. The clutter correction is used to correct phase measurements so that the phase affects of noise are removed from the data before AOAs are calculated. In the quiet period shown in Figure 4.3, the bottom plot demonstrates that when the clutter correction is applied to the data the phase effects from the noise sources seen in the top plot are removed. Leaving only a few degrees of variance in the phase measurement. ICEBEAR has a phase error tolerance of 5° (Lozinsky, personal communication, December 20, 2020), so the variation seen in the bottom plot is within the tolerance level of the radar. In the active period shown in Figure 4.4, the clutter correction is no longer removing all variation in the noise, but it does reduce the total range of variation seen in the data. However, the clutter correction is designed to remove the effects of self-clutter from the data measurements, not flatten the noise phase variation. Since self-clutter is the dominant source in active periods, that means the phase of this noise is added to all measurements in the radar. Application of the clutter correction subtracts this phase effect from all the data, so while the noise may still have notable variation in the phase, the effect of the self-clutter has still been reduced from the the signal measurements as desired.

4.4.2 Diurnal Trends

Consider again the quiet period power and phase measurements from Figure 4.1 and Figure 4.3, specifically the top figure before the clutter correction is applied. Over the two day period shown there is a clear repeating pattern seen in the noise. The noise phase also observes a trend that has features that occur at the same time features occur in the noise power. This diurnal repeating pattern is seen in all quite data sets analyzed from Table 4.1. Figure 4.5 displays a 24 hour period of quiet ionospheric conditions ($Kp<2$) starting 2018 April 11 21:00:00 UTC. This figure is an example demonstrating that the diurnal trend seen

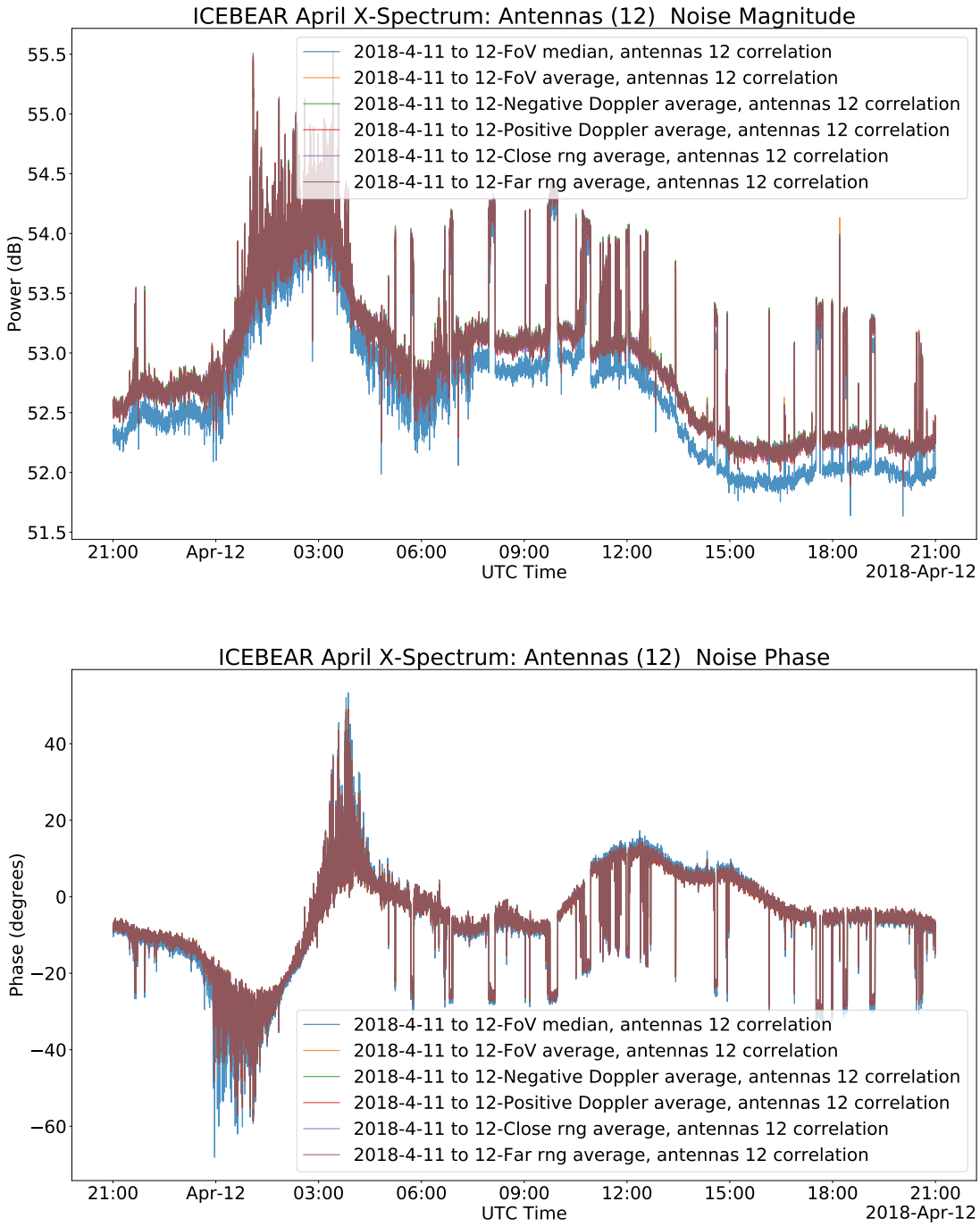


Figure 4.5: Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median on 12 April 2018. This is a quiet period for the linear ICEBEAR antenna layout configuration with a 1λ antenna separation. The FoV median noise follows the average noise powers at a slightly lower power, while the phase of all methods is nearly identical.

in the March data shown in Figure 4.1 and Figure 4.3 is observable in other months. The top figure depicts the noise power before the clutter correction is applied using data from the 1λ baseline between antennas 1 and 2 in the ICEBEAR-linear antenna configuration, while the bottom figure depicts the noise phase before the clutter correction using the same data set. In this April data, a number of short term enhancements occur in power throughout the data set lasting on the order of minutes, with corresponding shifts in the phase. The occurrence of these events is sporadic, which suggests that they are the result of some man-made terrestrial noise source. These short term enhancements occur in other data sets analyzed from Table 4.1, but have no consistency in the duration, frequency, or time of occurrence. These effects have a power level averaging $\sim$1 dB$>$ than the cosmic noise detected during quiet periods when K$p<$ 2.

The diurnal trend that is observed in the March data presented in Figures 4.1 and 4.3 is seen again in the April data set presented in Figure 4.5, only shifted 2 hours earlier. So the pattern that starts at 0:00:00 UTC in March starts at 22:00:00 UTC in April. This temporal shifting of the observed noise trend appears in the other months analyzed (see Table 4.1), with the trend shifting approximately 2 hours earlier each month. As the Earth rotates around the sun, the part of the universe observable at a given location and time will change. It is observed that the diurnal noise pattern varies over time and shifts with the months. This is consistent for cosmic noise as the dominating noise source for ICEBEAR during quiet ionospheric periods. The strongest cosmic noise is observed from the Milky Way (Ko, 1958), which coincides with when the noise power measured by ICEBEAR is highest. Over a 1 year period, the diurnal trend shifts a full 24 hours, so observation times of the diurnal trend in ICEBEAR will repeat yearly.

Further examination of this diurnal trend has shown that the strongest observed peak in noise power trend and the corresponding phase variation is a result of a specific galactic source within the Milky Way. The radio galaxy Cygnus A is observed by ICEBEAR on a daily basis as the radio galaxy passes through the ICEBEAR FoV. This phenomenon, and a radar calibration technique using the effect of Cygnus A on ICEBEAR noise measurements, is discussed in the next chapter (Chapter 5).

4.4.3 Baseline Dependence

Finally, the effect of different baselines is studied by using a 4λ baseline as a example for comparison against the 1λ baselines shown previously in this chapter. For the ICEBEAR-3D antenna array configuration, Figure 4.6 presents the noise power and phase from the baseline between antennas 0 and 3, in the top and bottom figures respectively, over a 48 hour period starting on 29 May 2020 0:00:00 UTC. This is contrasted by Figure 4.7 which presents the noise power and phase for a 4λ antenna separation using the ICEBEAR-linear antenna configuration. This uses antennas 4 and 8 from the linear array and covers a 48 hour period starting on 7 March 2018 0:00:00 UTC. Both data sets were measured during a quiet ionospheric period and display a sequence of four strong peaks (enhancements in the power 2 dB stronger than the rest of the noise in the data set) in noise power. Occurring simultaneously with the power peaks are large oscillations of the noise phase characterized by a sharp decrease in the phase value, followed by a sharp increase, then returning to the starting phase value. Consider the times of occurrence of the four strong peaks in noise power of Figure 4.7 and the time of occurrence of the strong noise enhancement seen in the noise power shown in the top figure of the 1λ baseline from the same time period shown in Figure 4.1. Both data sets have the four peaks (4λ baseline), or single peak (1λ baseline), occur between 2:00:00 and 8:00:00 UTC indicating the increase in noise power is resulting from the same source and is observed differently based on the size of the baseline. The single peak from the 1λ baseline is observed as four distinct peaks in the 4λ baseline, and is a result of grating lobes (the equivalent of fringes in optics nomenclature) in the radar beam pattern of the larger baseline. This can be described by treating the two receiving antennas as a Young's double slit experiment. The grating (or fringe) equation is (Hecht, 2016),

$$a\ \sin(\theta) = m\lambda \tag{4.3}$$

where a is the separation between antennas, θ is the AOA measured perpendicular from a plane through the measuring antennas, m is the grating lobe (fringe) order, and λ is the radar wavelength. For an antenna spacing of 1λ, the first grating lobe peaks will be located at $0°$,

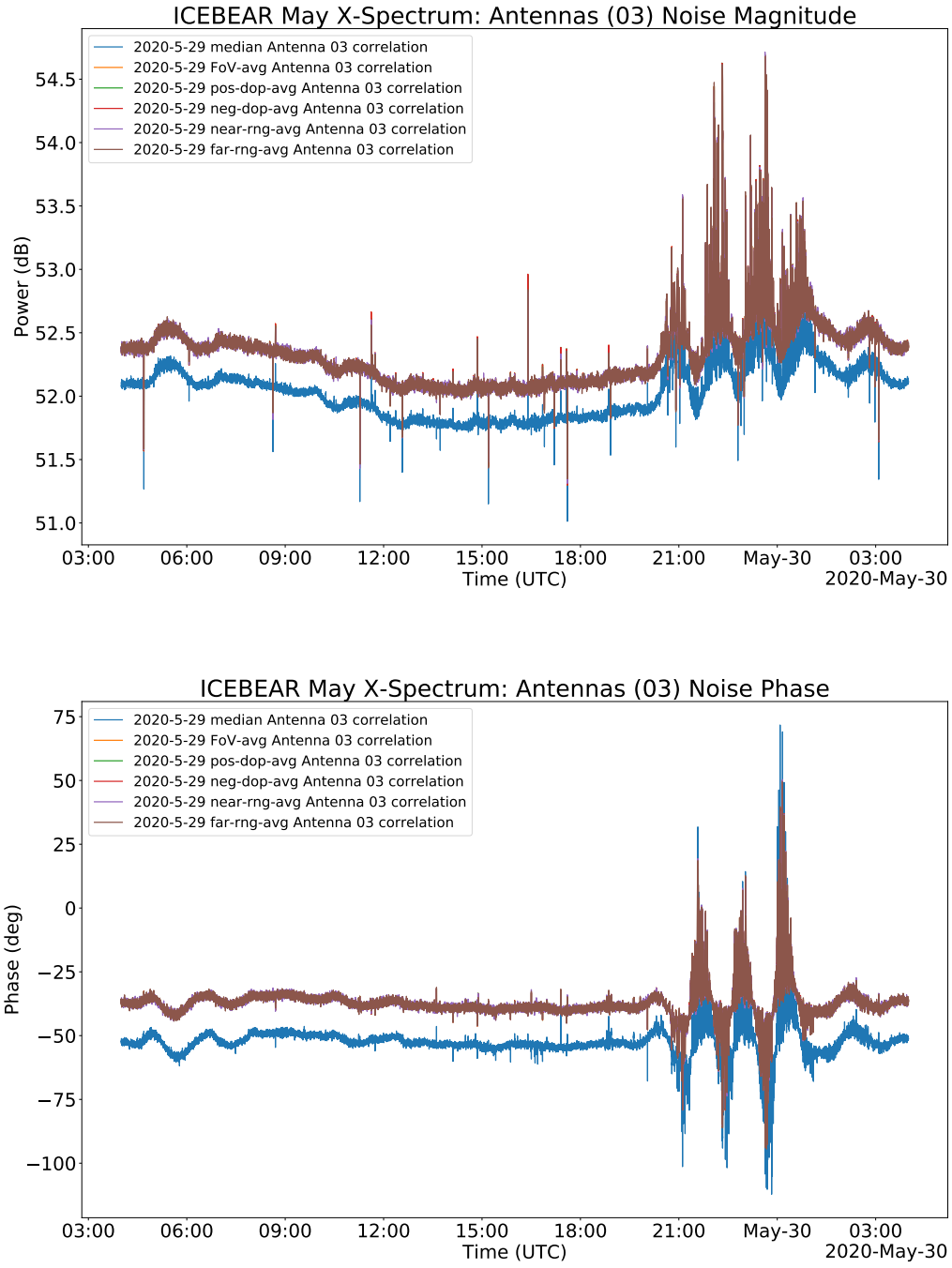


Figure 4.6: Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median starting on 29 May 2020 3:00:00 UTC for 24 hours. This is a quiet period for the ICEBEAR-3D configuration with a 4λ antenna separation using antennas 0 and 3. The FoV median power and phase values follow the trend of the average noise methods, but are shifted down. Baselines larger than 1λ can have a phase offset for the FoV median not seen in the 1λ noise data due to the grating lobes.

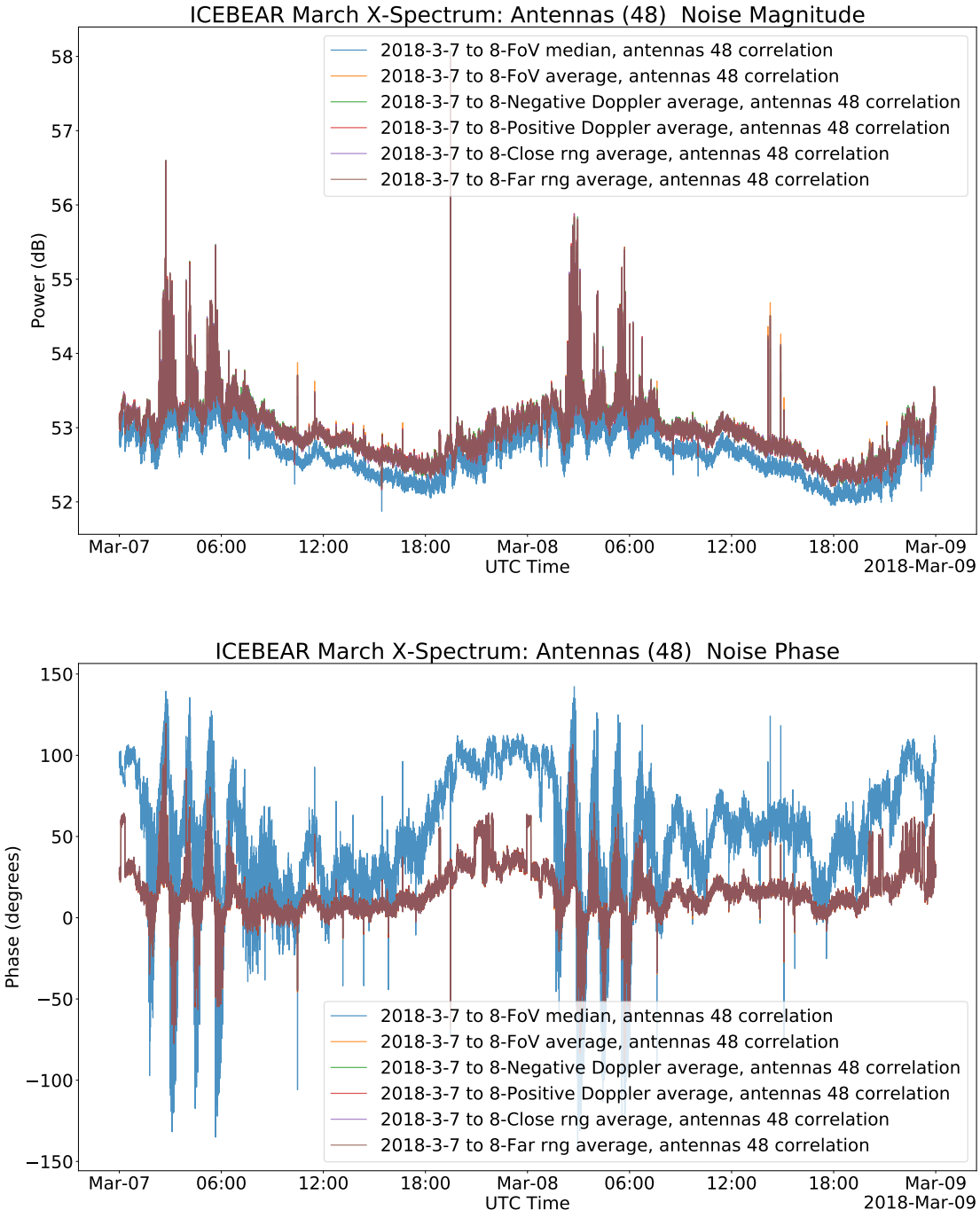


Figure 4.7: Raw noise power (top) and raw noise phase (bottom) calculated using multiple averaging methods and the FoV median on 7−8 March 2018. This is a quiet period using the ICEBEAR-linear configuration with a 4λ antenna separation. As expected, the FoV median noise power follows the same trend as the average noise methods at a slightly lower power. Due to grating lobes in larger baselines and the phase sensitivity of the ICEBEAR instrument, the FoV median noise phase is greatly influenced by phase variations in noise sources. Nevertheless, the general phase trend in the average noise data can still be seen in the FoV median data.

and ±90°. However, at 4λ they will be located at 0°, ±14.5°, ±30°, ±48.6°, and ±90°. As is clear from comparing the top plot of Figure 4.1 to the top plot of Figure 4.7, a larger baseline will result in more grating lobes within the ICEBEAR FoV as described by Equation 4.3, hence the detection of multiple peaks instead of one in the noise power. The exact beam patterns of the radar will be more complex then Equation 4.3 outlines as it does not take into account ground reflections and the antenna design, but the beam patterns will follow this same concept. Figure 4.8 depicts the modeled beam patterns for ICEBEAR antennas with 1λ and 4λ antenna spacing. The differences in the resulting beam patterns between the antenna pairs displays how the observable region of the ICEBEAR FoV changes based on antenna positions. Consider a source moving through the ICEBEAR FoV from East to West. In the 1λ baseline this will appear as single source while it moves through the main lobe of the beam pattern, but the 4λ baseline can observe the source instead as 5 distinct occurrences as it moves through each of the grating lobes of the beam pattern.

Looking at the power and phase presented in Figure 4.6 and Figure 4.7 over the 1 day period shown in each figure, there is a diurnal trend to the noise data. For the 4λ baseline, the trend differs from the 1λ baselines shown previously due to the grating lobes present in longer baselines. In the 4λ baseline, there are four peaks that occur in the noise power over a 6 hour period, after which the noise power decreases by 0.5 dB over a 12 hour period. Over the last 6 hours of a 24 hour period, the noise power increases back up to the same level as the start of the period and the pattern repeats. The phase measurements also show a diurnal pattern, however unlike with the 1λ baselines described earlier, the FoV median method of calculating the noise no longer generates a noise phase that agrees with the different averaging methods. In the same way as the noise observed during active ionospheric conditions was biased by signal strength, so too is the noise observed during quiet periods, with the grating lobes altering the trend in the total observed power due to the different "regions" of the FoV observed. The phase shown in Figure 4.7 depicts a FoV median noise phase that is drastically different compared to the other noise phase value calculated using averaging methods. This suggests that the FoV median is not as suited to determining noise for large baselines, and may not be the most suitable noise calculation technique. It underestimates the noise during quiet ionospheric periods and generates noise phase values that are notably different from

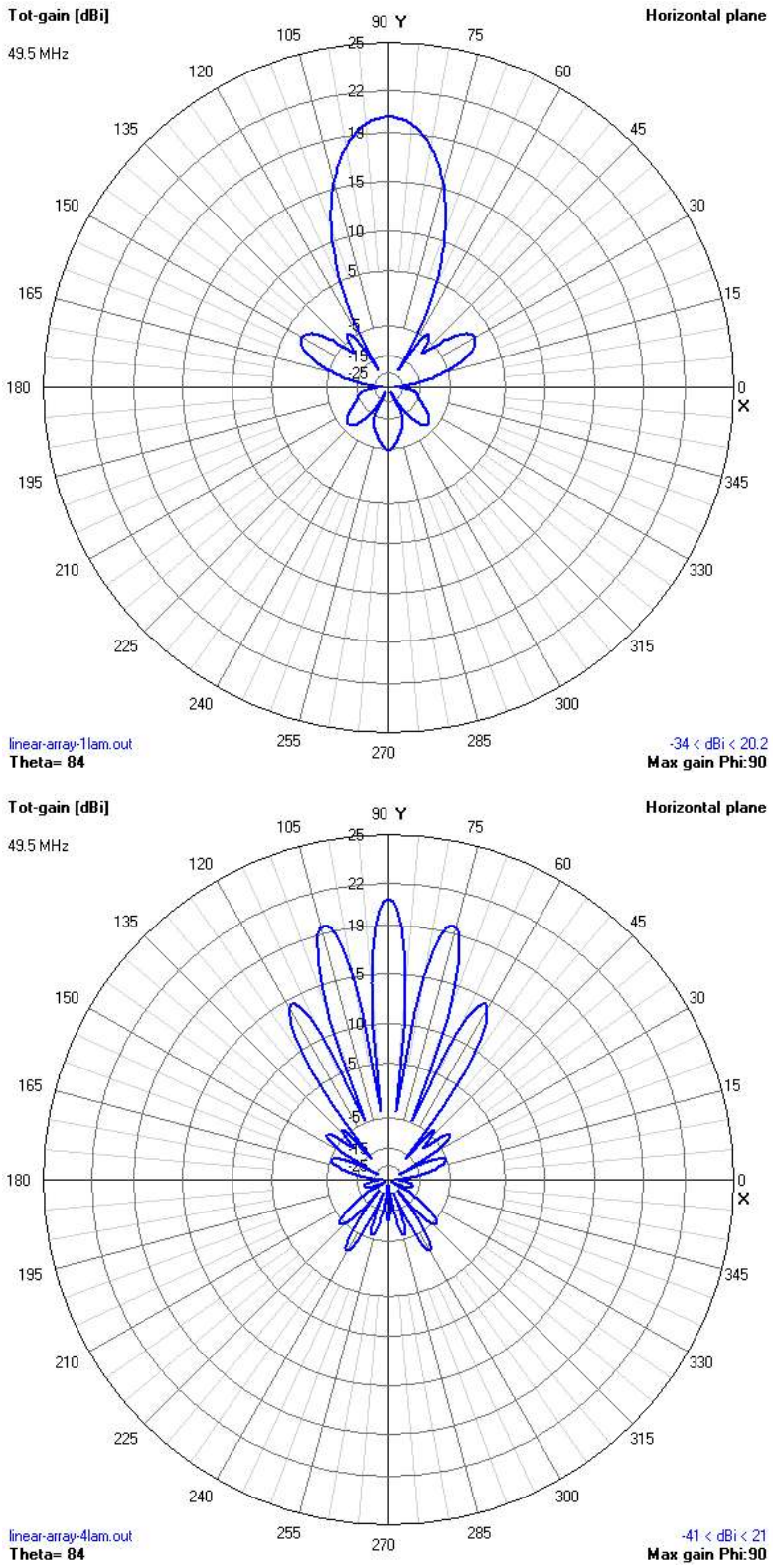


Figure 4.8: 1λ (top) and 4λ (bottom) baseline azimuth beam patterns generated using a NEC model of the ICEBEAR antennas (Voors, 2020). This demonstrates that baselines with a larger antenna spacing develop more grating lobes within the same FoV.

the averaging methods. It is therefore recommended that ICEBEAR use the far range average technique outlined in Section 4.3.1 to calculate noise in the radar. The far range average technique is not as affected adversely by different baseline lengths, and will only contain radar noise as the far ranges are beyond where ICEBEAR is capable of observing E-region instabilities and observes less meteor trails than the close range average.

4.5 Summary

Understanding the noise in a radar system is crucial to interpreting measured signals. For this reason the noise in the ICEBEAR radar has been investigated. At the present, ICEBEAR determines noise of the observed signal during a particular time period by taking the median of all power measurements within the range-Doppler matrix, or FoV, calculated from data measured on multiple baselines. This method was compared to noise calculated by taking an average of signal measured in regions of the FoV where no ionospheric scatter is expected, as well as an average across the entire FoV. Comparison of these methods over multiple days has shown that both the FoV median and average methods display the same trend, with the median power at a consistent offset below the average power by less than 1 dB. During active periods in the ionosphere, the noise is dominated by the clutter floor of the signal processing technique, and the FoV median noise and small range or Doppler shift averaging methods produce near identical values. These observations are also preserved for observations using antenna pairs of differing antenna separations, and is also preserved between both the previously implemented ICEBEAR-linear receiver configuration and the new ICEBEAR-3D receiver configuration.

The noise phase value, calculated as the FoV median or average of the phase difference between an antenna pair, for a 1λ antenna spacing shows no notable differences. When a longer antenna spacing is used, the FoV median noise phase and average data can be significantly different. Since besides the change in antenna separation there is no other differences between the measurements, this phase difference is a result of the beam patterns of longer baselines. The different beam patterns change which parts of the sky are observed for each baseline, introducing this difference in the final result.

Examining the noise trend over the period of a day has shown that there is a clear and repeating cycle to the measured noise in both power and phase. This diurnal trend slowly drifts backward in time by a rate of about 2 hours per month. Some of the ICEBEAR data analyzed have shown some sporadic signal measurements which causes sudden and short lived spikes in the power and shifts in the phase, but the overall trend remains the same. These short lived spikes have power levels a few dB greater than the background cosmic noise.

The radar noise determined by the noise calculation techniques presented in this chapter are used by ICEBEAR in calculation of the SNR, which is based on the power measurements. Considering the current observation of the median noise power and phase over all the data sets used, it is reasonable to conclude that the median is generally a good estimate of the radar noise. The median also reduces the effect of stray measurements from biasing the noise value. However, the averaging techniques presented have shown consistency in both noise power and phase measurements not observed in the median data baselines longer than 1λ. Consistency is important in calculating a reliable noise parameter, so it is recommended that ICEBEAR implement an averaging technique to calculate the noise instead of using the FoV median technique. The most reliable choice would be using the far ranges in the ICEBEAR FoV where no ionospheric scatter is expected due to the high aspect angle of the wave propagation in those ranges, and the low number of meteor trails observed in the far ranges. Based on the observed effects of the clutter correction (correction term used to remove phase bias in the radar measurements described in Section 4.3) on the measured noise data, it has been shown to be effective at removing the phase bias introduced by noise sources and self-clutter from ICEBEAR signal processing. Study of ICEBEAR noise over multiple days and months has shown clear diurnal patterns that are observed during quiet ionospheric periods. This demonstrates that ICEBEAR noise during quiet periods is dominated by cosmic noise, and that during active periods it is dominated by self-clutter from the signal processing technique.

5 Cygnus A Phase Calibration

5.1 Introduction

From the analysis of ICEBEAR noise in Chapter 4, a distinct peak in the diurnal noise pattern (demonstrated for 1λ and 4λ spacing) is detected daily in the power and phase measurements, indicating there is an external source passing through the radar field of view (FoV). The source lies within the Milky Way, however, this power enhancement corresponds to a strong phase response, and has a dependency on the radar beam pattern that is not expected from basic cosmic noise. This signature appears in the ICEBEAR noise measurements so a signal to noise ratio (SNR) cannot be determined for this signal. Instead the noise power and phase are used to study this signature. This chapter presents the source of this enhanced signal to be the radio galaxy Cygnus A and discusses how Cygnus A may be used to calibrate the phase of the ICEBEAR radar. At the present, phase corrections have been made for each antenna in the radar using a spectrum analyzer. The spectrum analyzer is connected to the signal path on either side of the bulkhead (the connection point between the indoor RF equipment and the external coaxial cables and antennas), measuring the phase path to the antennas from the bulkhead and to the Ettus X300 transceivers from the bulkhead. These two measurements are combined to make the current phase corrections. It is possible that the extra path length through the bulkhead or the digitization in the X300 transceivers could introduce additional phase changes not accounted for by the spectrum analyzer. Cygnus A will provide an alternative method of phase calibration for the radar, and will also act as a check to verify the spectrum analyzer measurements.

First the properties of the distinct peak in the noise power and the corresponding phase changes will be described, along with how this signature can be caused by a radio galaxy.

Radio galaxies will then be defined and the method of determining the position of a stellar body will be outlined. Next it will be shown that the signature in the noise is clearly a result of the radio galaxy Cygnus A.

The concept of how to use a stellar source to perform a phase calibration will be presented based on the work from Palmer et al, (1996). This includes how to determine the theoretical phase difference between two antennas, and how this value relates to the measured phase differences to create phase correction terms. The procedure used to create phase corrections using Cygnus A will be presented, as well as the determined corrections. Finally, possible issues with the analysis will be discussed, including potential improvements.

5.2 Cygnus A Detection

5.2.1 The Noise Signature

Chapter 4 presented the analysis of the ICEBEAR background noise, which discovered a strong signature from the Milky Way galaxy where the cosmic noise is at its strongest (Ko, 1958). For the analysis preformed in this chapter, the ICEBEAR data analysis was altered to generate 1 s averages instead of the usual 5 s average that was used in Chapter 4 to provide more detail of the observed noise signature over time. Figure 5.1 presents the measured noise power and Figure 5.2 presents the phase of this signature for two separate months, January and April. Both of these figures show a seven hour period over which the average noise power gradually increases to a peak about 1.5 dB above the starting noise power then gradually decreases back to the starting level. This multiple hour long enhancement in the noise is referred to as a peak in the noise data as it reaches a noise power notably larger than the rest of the noise measured over the day. In the phase data shown, the phase demonstrates a cosine-like pattern as the measured phase decreases over 30° below the starting phase value, increases to over 30° above the starting value, then returns to the starting value. Figure 5.3 again presents the noise power signature from April, but now for two different baselines, 1λ and 4λ. In the 4λ noise the peak observed in the 1λ baseline separates into multiple peaks in the noise power, each with a corresponding to a cosine-like pattern in the phase due to the

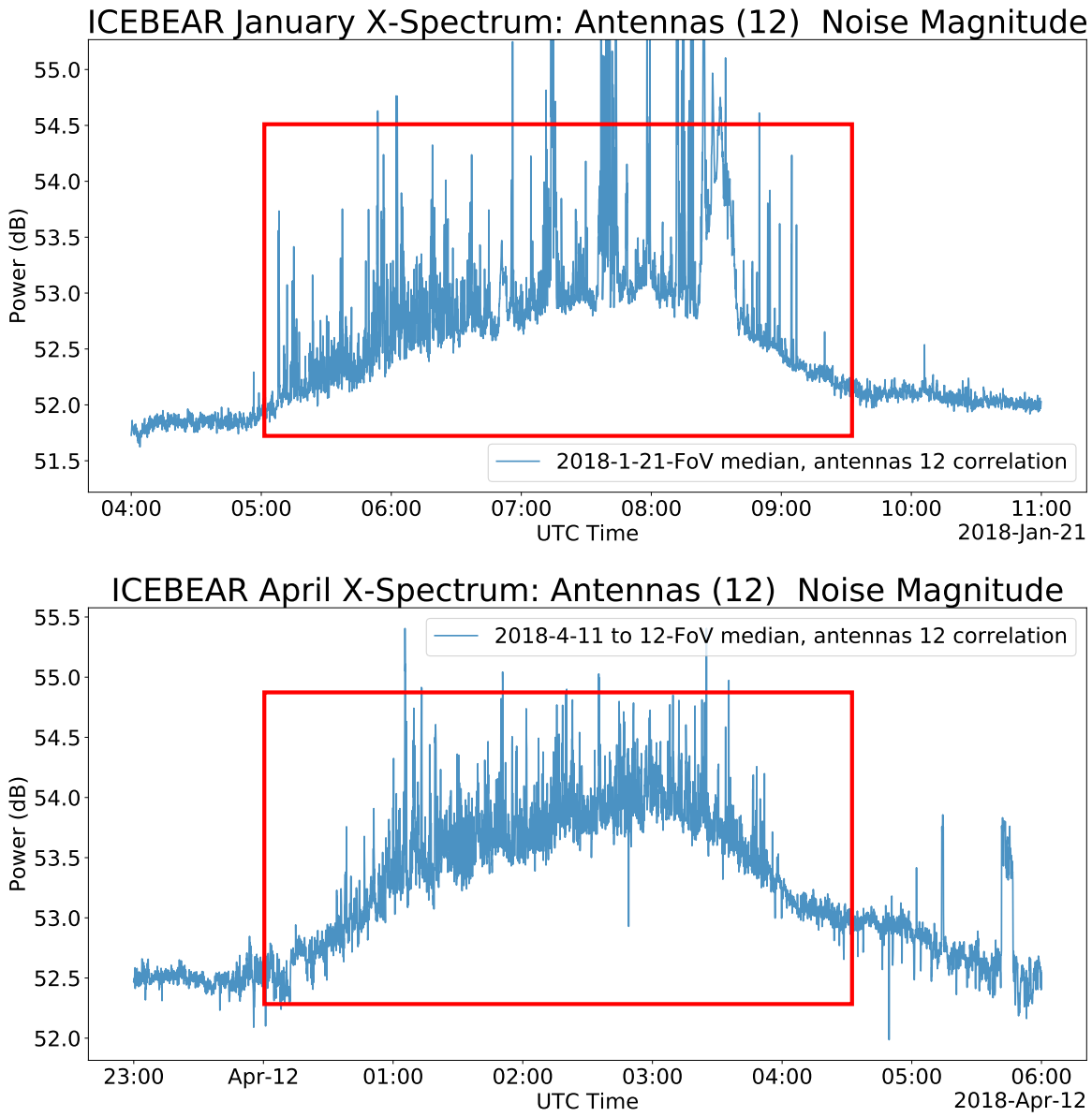


Figure 5.1: Distinct peak in the noise power indicated by the red rectangle, occurring in 21 January 2018, (top) and 12 April 2018, (bottom) using the ICEBEAR linear antenna layout configuration. The noise power (magnitude) is in dB for the noise signature from the cross-correlation between antennas 1 and 2, which have a separation of 6 m (or 1λ).

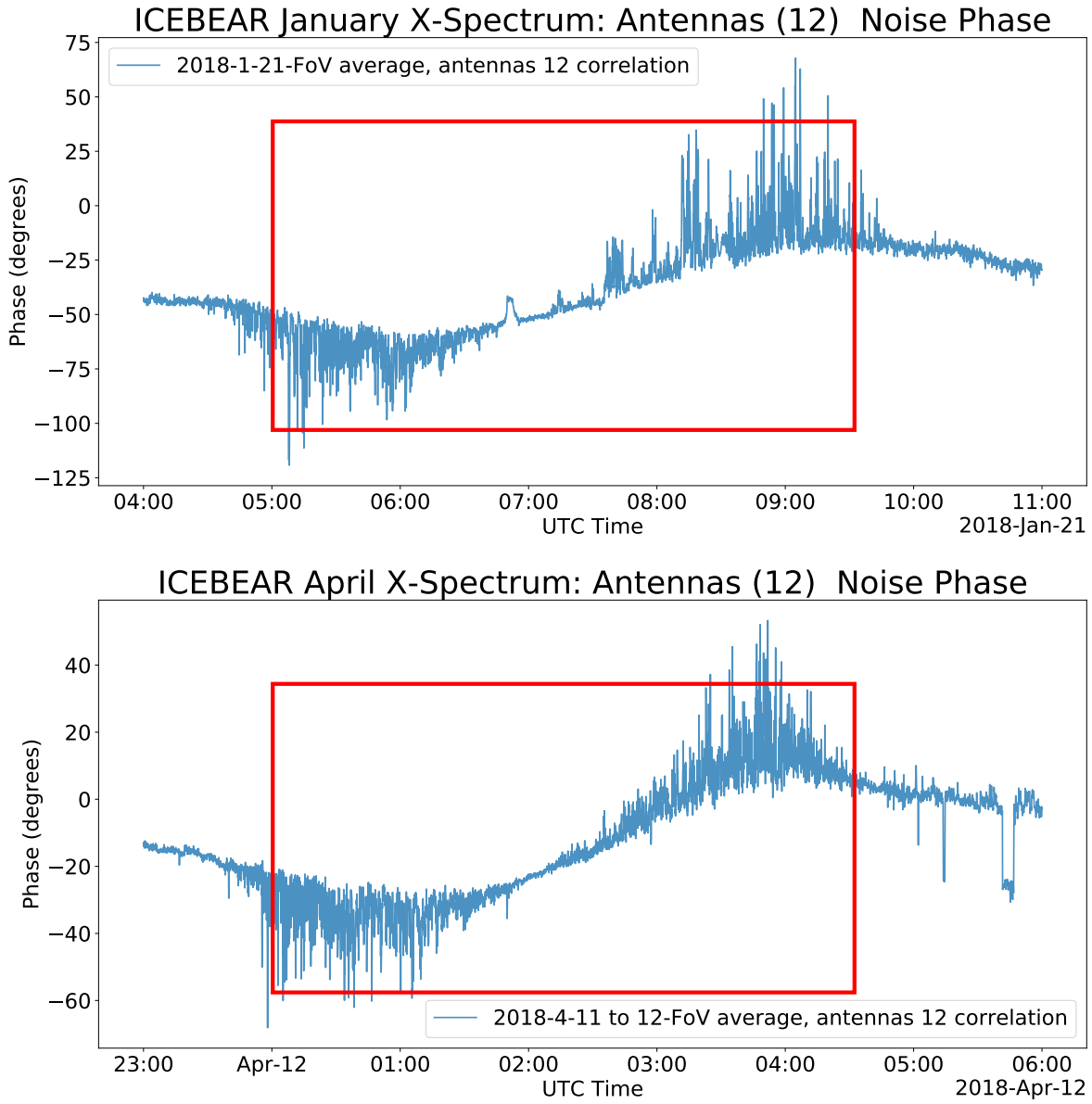


Figure 5.2: Oscillation of the noise phase indicated by the red rectangle, occurring in 21 January 2018, (top) and 12 April 2018, (bottom) using the ICEBEAR linear antenna layout configuration. The phase is in degrees for the noise signature from the cross-correlation between antennas 1 and 2, which have a separation of 6 m (or 1λ).

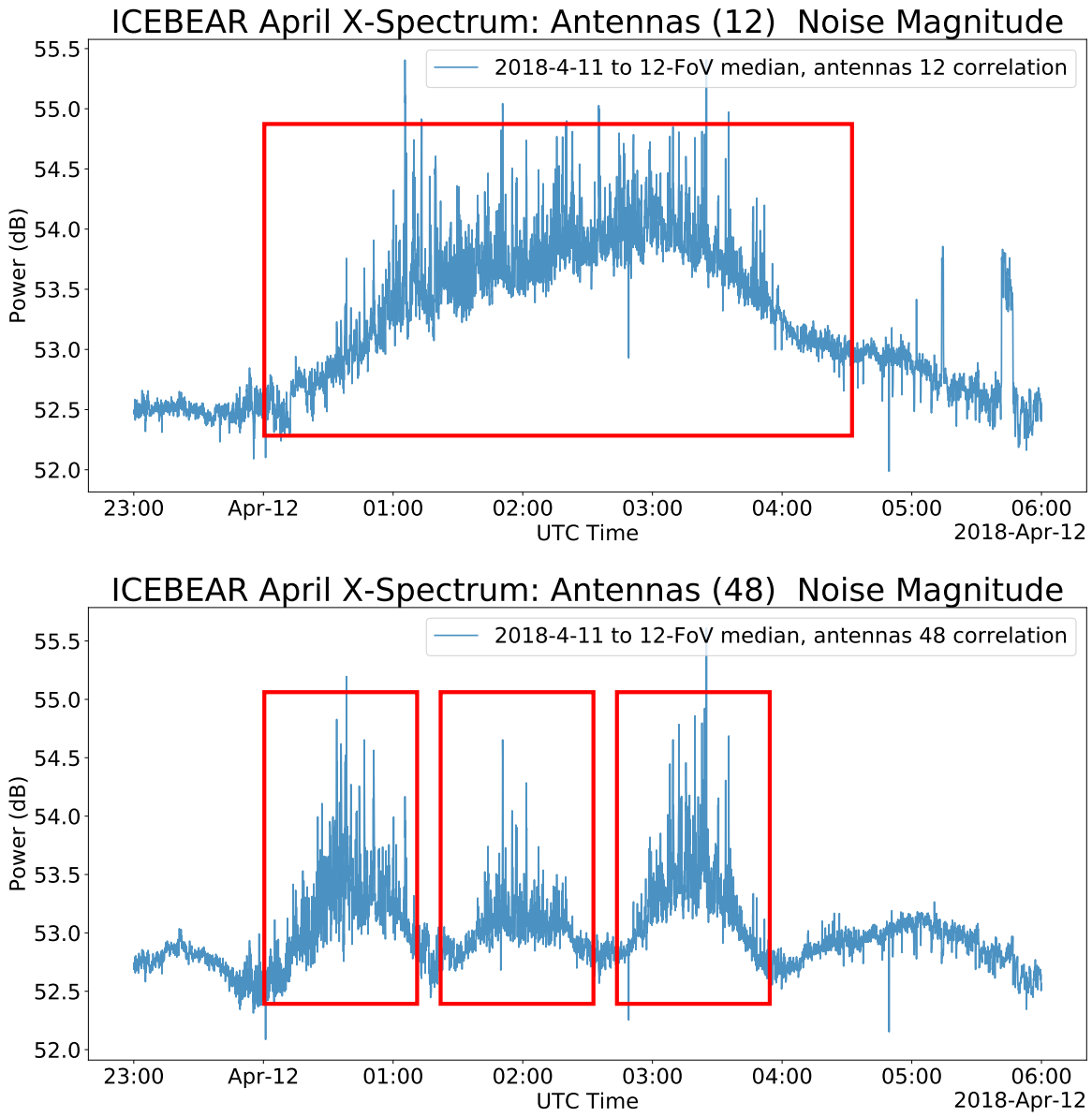


Figure 5.3: Distinct peaks in the noise power indicated by the red rectangles, occurring in 12 April 2018, using the ICEBEAR linear antenna layout configuration. The power is in dB for the noise signature from cross-correlation between antennas 1 and 2 (top) with a separation of 6 m (1λ), and between antennas 4 and 8 (bottom) with a separation of 24 m (4λ). Note that the peak from the 1λ baseline appears as three peaks in the 4λ baseline.

multiple grating lobes in the 4λ baseline beam pattern. More examples of this signature, for both 1λ and 4λ baselines, can be seen in the figures presented in the previous chapter. Red boxes have been used in these figures to highlight the peaks and cosine-like phase patterns in the data. This signature in the noise has a daily occurrence so it is determined to be a result of cosmic noise. This data is a measurement of ICEBEAR noise, so it is subject to influence from other noise sources in the radar. Noise power and phase values for this signature are not absolute, but no notable variation in other background noise sources is expected over the time period that the signature occurs (a few hours). The temperature of the exterior and interior radar components, and therefore the thermal noise, will notably fluctuate month to month (especially between winter and summer), but over short periods can be expected to be relatively consistent. When comparing observations from different time periods, it must be recognized that variation in the cosmic and thermal noise sources may result in a difference in noise power and phase measured, but overall trends are expected to remain constant across all measurements.

Based on the change in the measured response when using different baselines (one peak in the 1λ baseline but three peaks in the 4λ baseline), the source of the noise signature must be distinct enough that its position within the radar FoV can be resolved. Examining the noise power measurements there are distinct peaks that occur, and the number of peaks that occur increases with the antenna spacing, as does the number of grating lobes in the beam pattern. Therefore, the multiple peaks of the signature are a result of a source moving through multiple grating lobes in the beam pattern of the antenna pair. Examining the phase changes over the period of these signatures, there is a clear oscillation about the base phase value at the time of each peak in the noise power. This oscillation is often very pronounced to the point where the signal wraps within the $\pm\pi$ phase boundaries.

From these observations of the noise power and phase, it is clear that a source is moving through the beam pattern. The phase difference tends negative (East) as a source enters a beam pattern lobe from the East (or right in azimuth), then tends positive (West) as the source traverses through the lobe and exits on the West (left) side. This causes the observed oscillation in the phase measurement as the source moves through the grating lobes in the

beam pattern. The daily occurrence of the signature suggests the source must be in orbit about the Earth or a celestial body observable in the sky. Satellites and planes move too fast to be the cause of this noise signature. Analysis showed that the occurrence of the signal progressed in time with the background cosmic noise and in the same direction, and therefore was celestial in origin. The source of this signature is therefore a radio galaxy, as these are stellar bodies with high power radio emissions which are observable by the ICEBEAR radar.

5.2.2 Tracking Radio Galaxies

Using astronomical calculations and the geographical position of the ICEBEAR receiver, the angular location of any stellar body in the night sky with respect to the local horizon of ICEBEAR can be determined (Duffett-Smith, 1988). From these calculations, it can be easily determined if a radio galaxy is within the ICEBEAR FoV during the observation of the enhanced noise signal.

Radio Galaxies

Radio galaxies are galaxies that have a strong electromagnetic emissions in the radio spectrum, often a result of synchrotron radiation (Hey, 1983). There are many sources of galactic radio emissions, but radio galaxies are one of the strongest and most documented radio sources. If the signature detected in the ICEBEAR noise is a result of a radio galaxy, it must have strong emissions around 49.5 MHz frequency and must be visible in the northern hemisphere. Four strong emitting radio galaxies that meet this condition are, in order of the strongest to weakest power flux density: Cygnus A, Virgo A, Hercules A, and Perseus A (Hey, 1983).

Equatorial to Horizon Coordinate Conversion

Determining the angular position of these radio galaxies with respect to the ICEBEAR receiver is an involved, but straight forward, process. The Python (Python-Software-Foundation, 2020) code used to determine the azimuth and zenith angle of a stellar body with respect to the local horizon is presented in Appendix B. The position of a celestial body is usually

given in equatorial coordinates, defined by the declination of the body and the hour angle of the observation. For the purposes of this analysis it is desirable to know the position of the celestial body in terms of an azimuth and zenith angle in the sky with respect to the point of observation on Earth, known as the horizontal coordinates. The general process of the conversion from equatorial to horizontal coordinates will be outlined here and is taken from *Practical Astronomy with Your Calculator* by Duffett-Smith (1988).

The following analysis steps assume the right ascension and declination of the celestial body, as well as the latitude and longitude of the observer, are known including the time of observation in Universal Time (UT). While ICEBEAR measurements are made using Coordinate Universal Time (UTC), UTC is always within 0.9 s of UT (Duffett-Smith, 1988), so an assumption is made that UTC is equivalent to UT over the times scales of interest to ICEBEAR.

For calculation of the Horizontal Coordinate position of the stellar body, the declination, δ is known, but the hour angle, H, must be determined from the UT observation time. This is done by calculating the Julian Date, JD, which is used to convert UT to the Greenwhich mean Sidereal Time (GST). GST is then shifted to Local Sidereal Time (LST), where finally H can be calculated (Duffett-Smith, 1988). First, the UT observation time is defined using the Gregorian calendar system such that,

$$\begin{aligned} y &= \text{the year of observation} \\ m &= \text{the month of observation (0-12)} \\ d &= \text{the day of observation} \\ h &= \text{the hour fraction of observation} \end{aligned} \tag{5.1}$$

where the hour fraction represents the hour, minute, and second of the observation, computed as,

$$h = hour + \frac{minute}{60} + \frac{second}{3600} \tag{5.2}$$

If m is equal to 1 or 2, then 1 must be subtracted from the year y, and 12 added to the month m. The Julian Date, JD can then be defined by,

$$
\begin{aligned}
A &= \mathrm{INT}(y/100) \\
B &= \begin{cases} 2 - A + \mathrm{INT}(A/4), & \text{if observation time >= 15 October, 1582} \\ 0, & \text{otherwise} \end{cases} \\
C &= \begin{cases} \mathrm{INT}(365.25 * y - 0.75), & \text{if } y \text{ is negative} \\ \mathrm{INT}(365.25 * y), & \text{otherwise} \end{cases} \\
D &= \mathrm{INT}(30.6001 * [m + 1]) \\
JD &= B + C + D + d + 1720994.5
\end{aligned} \tag{5.3}
$$

where INT( ) indicates taking the integer part of the result within the brackets, and A, B, C, D are parameters that adjust the UT year, month, and day values to the Julian Date reference point. Now the GST can be calculated from the Julian Date by using the following expressions,

$$
\begin{aligned}
S &= JD - 2451545.0 \\
T &= S/36525.0 \\
T0 &= 6.697374558 + 2400.051336 * T + 0.000025862 * T^2
\end{aligned} \tag{5.4}
$$

where S and T are more conversion factors, and T0 is the GST start time for the UT day being converted. Therefore if $T0$ is not $0 < T0 \leq 24$, 24 must be added or subtracted until $T0$ falls within this range. All that remains is to shift the starting GST time by the hour fraction of the UT measurement,

$$GST = T0 + h * 1.002737909 \tag{5.5}$$

where once again the resulting value must be shifted by a multiple of 24 so that the final value lies between 0 and 24. This is the GST equivalent day fraction of the UT measurement.

LST shifts the GST value based on the longitude, LON, position of the observer. Here Western longitudes are considered as negative values, and Eastern longitudes are considered as positive values. So the LST is,

$$LST = GST + LON/15 \tag{5.6}$$

where once again the LST value must be shifted by multiples of 24 so that it falls between 0 and 24. The hour angle H, can finally be determined by,

$$H = LST - \alpha \tag{5.7}$$

where α is the right ascension of the celestial body. Now that the hour angle for the observation has been defined, the position of the celestial body at the UT observation is described in equatorial coordinates (H and δ). This can be converted to horizontal coordinates (azimuth angle and zenith angle) by,

$$\sin\theta = \sin(\delta)\sin(LAT) + \cos(\delta)\cos(LAT)\cos(H) \tag{5.8}$$

$$\cos\beta = \frac{\sin(\delta) - \sin(LAT)\sin(\theta)}{\cos(LAT)\cos(\theta)} \tag{5.9}$$

where θ is the zenith angle with respect to the horizon, β is the azimuth angle with respect to North, and LAT is the latitude of the observer (Duffett-Smith, 1988).

5.3 Identifying the Radio Galaxy Source

Using the process outlined in Section 5.2.2, the path of the four radio galaxies outlined in Section 5.2.2 travel with respect to the ICEBEAR receiver can be calculated for any time period. As an example, the time period on 21 January 2018 when the noise signature was present is used. This period is shown in Figure 5.1 where in the January data set the signature appears to last from approximately 5 UTC to 9 UTC. The altitude and zenith angles of the radio galaxies during these times can be compared to the beam pattern of the ICEBEAR

receiver to determine if the radio galaxy is within the radar beam pattern during this time period. Using Numerical Electromagnetics Code (NEC) models generated using the 4nec2 software, the beam pattern of the 10 antenna ICEBEAR-3D receiver array was calculated (Voors, 2020). The azimuth cross-section of the beam pattern is shown in Figure 5.4 along with the azimuth positions of 4 radio galaxies plotted over a 12 hour period on 21 January 2018. The vertical cross-section of the beam pattern is shown in Figure 5.5 along with the zenith angle positions of 4 radio galaxies plotted over the same period on 21 January 2018.

From the NEC models, the 10 antenna ICEBEAR-3D receiver array beam pattern width was determined to have a 54° azimuthal extent centered at 7° East of North, and a 6° elevation extent centered 6° above the horizon. The position of the radio galaxies at any time can then be compared to the beam pattern to determine if the radio galaxy falls within the observable FoV of the receiver. Figures 5.4 and 5.5 present the azimuth angle and zenith (elevation) angle plots respectively for the radio galaxy positions, with the position plotted in green if the radio galaxy was within the ICEBEAR receiver FoV. Of the radio galaxies assessed, Cygnus A was within the ICEBEAR beam pattern from approximately 4 UTC to 9 UTC, which overlaps with the time period that the noise signature was observed by ICEBEAR.

5.3.1 Cygnus A

From the evaluation of the noise signatures (during the periods listed in Table 4.1) and comparison of the occurrence times to the location of various radio galaxies, it was concluded that the presence of Cygnus A within the ICEBEAR receiver beam pattern causes a clear, reoccurring effect on the ICEBEAR noise.

Cygnus A is the strongest emitting radio galaxy that can be observed from Earth, with emissions on comparable levels to the Cassiopeia supernova remnant. This radio galaxy has emissions on the order of 10^8 flux unit densities within the 50 MHz range (Hey, 1983). Other radar systems have used Cygnus A as a calibration source, such as the Middle and Upper atmosphere (MU) radar in Japan which was shown to be capable of observing Cygnus A and used the radio galaxy to perform phase calibrations of the radar (Palmer et al., 1996). Since the presence of Cygnus A causes repeatable and measurable effects on the ICEBEAR noise,

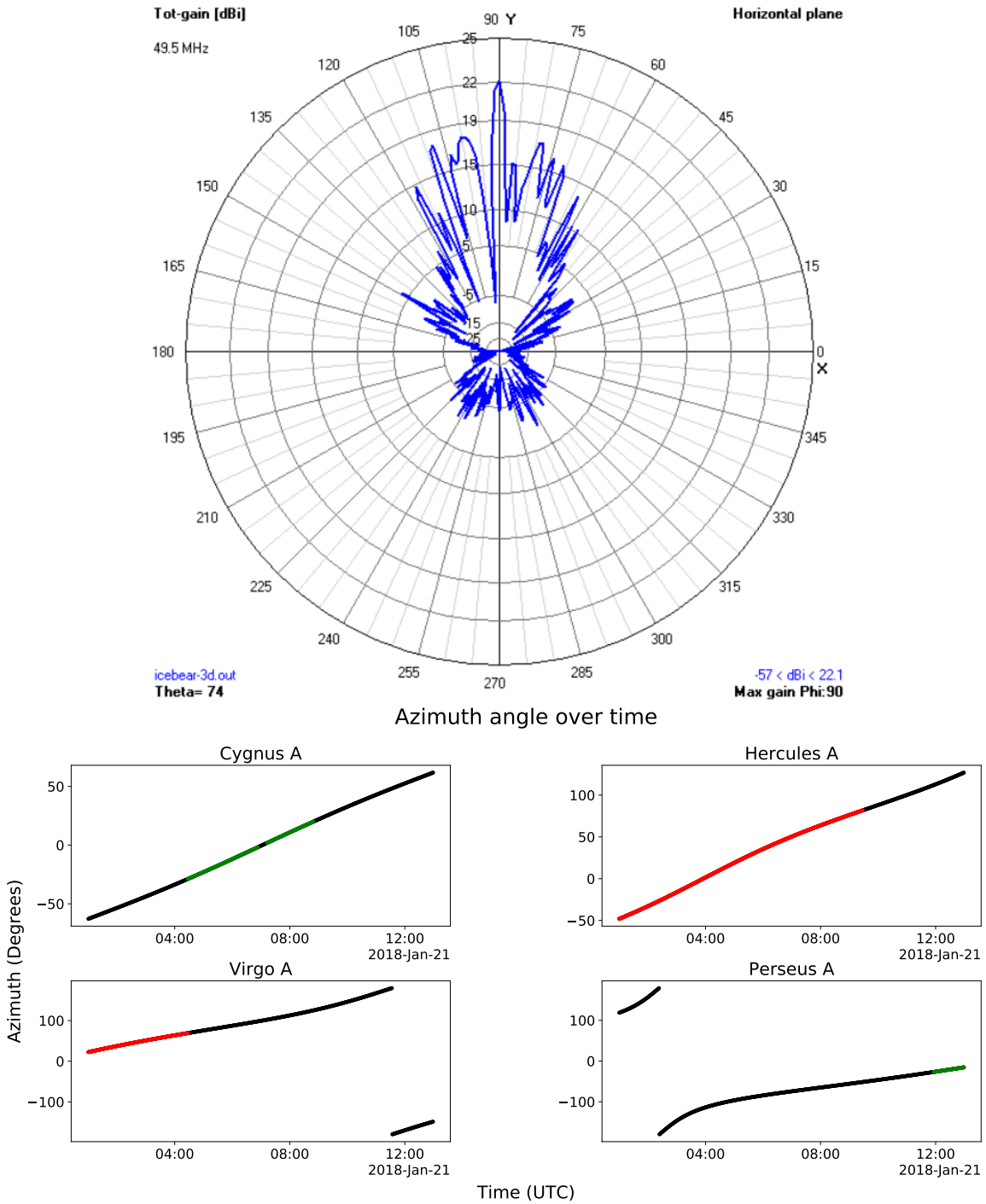


Figure 5.4: (Top) Horizontal cross-section of the 10 antenna ICEBEAR-3D receiver array beam pattern generated using the 4nec2 modeling software (Voors, 2020), North is at 90°. (Bottom) Azimuth angles of 4 radio galaxies with respect to the ICEBEAR receiver location over a 12 hour period on 21 January 2018, North is at 0°. Red regions of the azimuth indicate that the radio galaxy is below the horizon, black indicates above the horizon, and green indicates that the radio galaxy azimuth and zenith angles are within the ICEBEAR receiver beam pattern FoV.

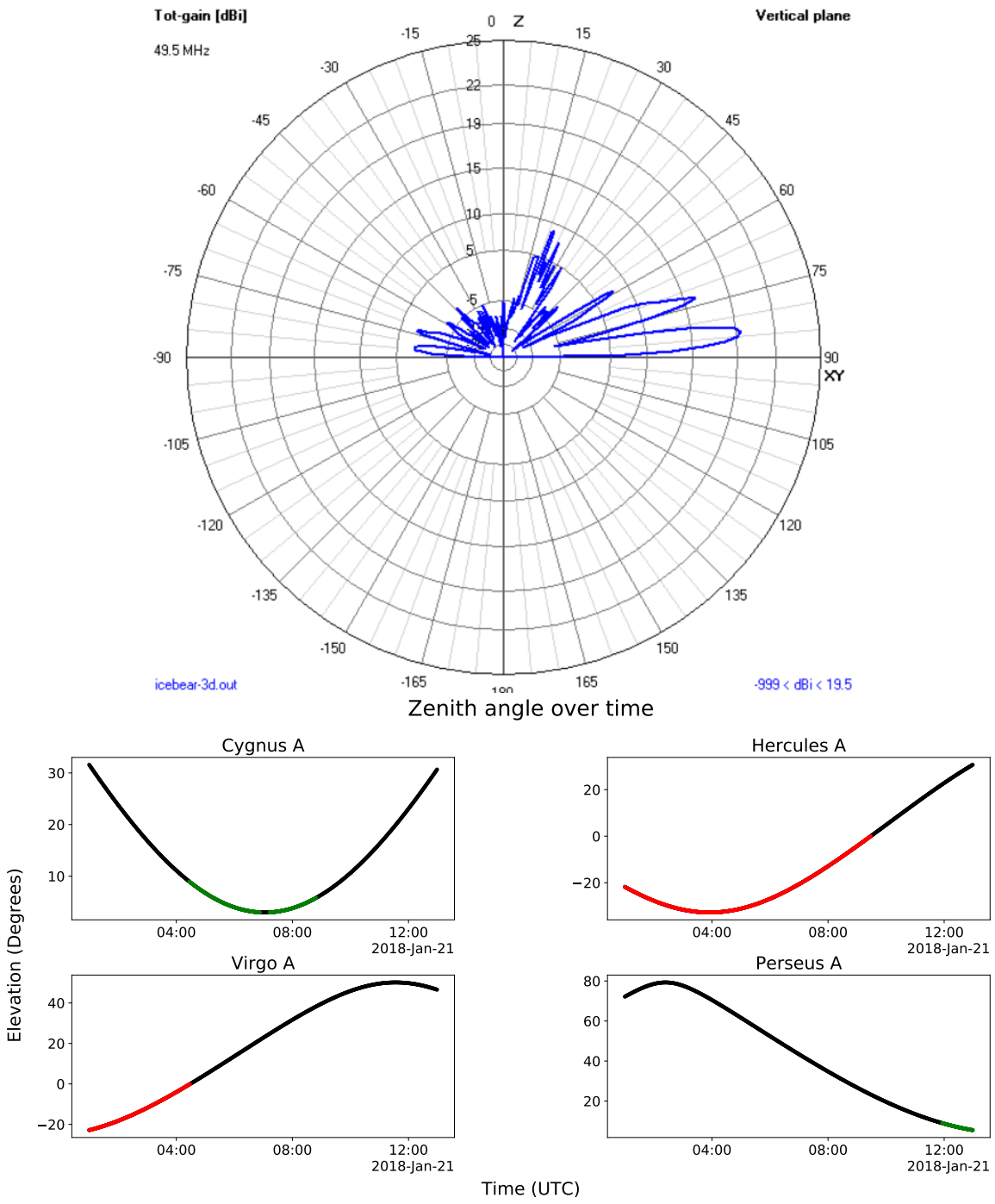


Figure 5.5: (Top) Vertical cross-section of the 10 antenna ICEBEAR-3D receiver array beam pattern generated using the 4nec2 modeling software (Voors, 2020), horizon is at 90°. (Bottom) Zenith angles of 4 radio galaxies with respect to the ICEBEAR receiver location over a 12 hour period on 21 January 2018, horizon is at 0°. Red regions of the zenith indicate the radio galaxy is below the horizon, black indicated above the horizon, and green indicates that the radio galaxy azimuth and zenith angles are within the ICEBEAR receiver beam pattern FoV.

it is possible to calibrate the phase of ICEBEAR measurements using Cygnus A.

5.4 Phase Calibration of ICEBEAR

ICEBEAR receiver antennas have been calibrated by taking phase measurements along the signal path using a spectrum analyzer. Measurements are made from an antenna to the bulkhead, and then from the bulkhead to an X300 transceiver. These phase values are then combined to generate a phase correction for each antenna, which is applied to the raw antenna data before being used in ICEBEAR data analysis. However, there are two potential issues with this calibration technique. The first is that due to physical constraints of this calibration technique. Since the spectrum analyzer cannot measure through the bulkhead, which is only antenna cable connectors between the interior and exterior radar equipment, a small part of the signal chain has been neglected in the phase measurements. Furthermore, this calibration technique requires a spectrum analyzer to be taken to the radar site to perform the calibration, so degradation of the signal path and digital equipment may cause phase errors that will not be detected until the next time calibration measurements are made.

Design of a calibration technique that could be applied to ICEBEAR measurements remotely and with greater frequency would provide a means to validate the ICEBEAR phase calibrations that are currently in use (spectrum analyzer), as well as keep the calibrations up to date. Palmer et al. (1996) used the radio galaxy Cygnus A to generate phase corrections for the cross-correlation results of multiple baselines of the MU radar in Japan. This radar operates at 46.5 MHz and has a 3.5 MHz bandwidth which overlaps with the frequency range of ICEBEAR (Hashiguchi et al., 2018). Using the calibration technique presented by Palmer et al. (1996), Cygnus A was used to perform phase calibrations of the ICEBEAR radar baselines.

5.4.1 3D Interferometry

Figure 5.6 depicts the coordinate system used to relate two measuring antennas to the radio galaxy position. The origin of the coordinate system in this analysis is chosen to be the

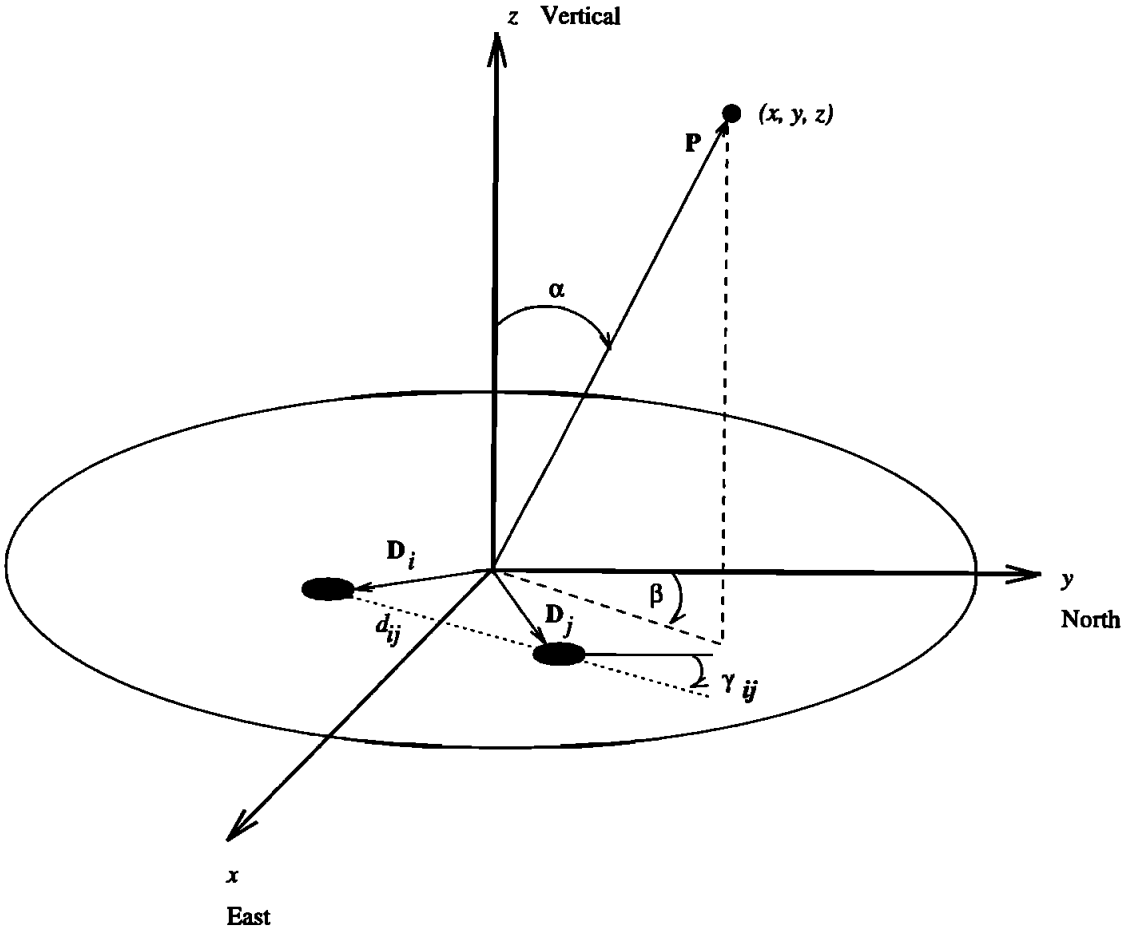


Figure 5.6: After Palmer et al., (1996). General coordinate system relating the positions of two antennas (i and j) and a stellar source (x, y, z) with respect to a frame of reference. The antenna vectors $\boldsymbol{D}$ represent the distance of the antennas from the origin, d is the distance between the two antennas, $\boldsymbol{n}$ is the unit vector pointing along vector $\boldsymbol{P}$ towards the stellar source, α is the elevation angle measured with respect to the zenith, β is the azimuth angle measured from North, and γ is the angle between North and the baseline.

geographical location of the ICEBEAR receiver (52.24319°, −106.450191°) to match the horizontal coordinate location of Cygnus A. The relative location of the antennas are measured with respect to the North-Eastern most antenna, denoted as antenna 0, so this antenna is defined as the origin of the chosen coordinate system. Positioning of the antennas in the ICEBEAR receiver array was shown in Chapter 2 (Figure 2.3). The distance from the measuring antennas to Cygnus A is much greater than the distance between the measuring antennas, so the Cygnus A signal is a plane wave.

The relationship between the measured phase differences between any two antennas, denoted by i and j, and the real phase difference is provided by (Palmer et al., 1996),

$$\phi_{ij} = k(\boldsymbol{D_i} - \boldsymbol{D_j}) \cdot \boldsymbol{n} + \Delta\phi_{ij} \tag{5.10}$$

where ϕ_{ij} is the phase difference measured between antennas i and j, k is the wave number of the radar, $\boldsymbol{D}$ is the vector from the coordinate system origin to an antenna, $\boldsymbol{n}$ is the unit vector from the coordinate system origin pointing toward the radio galaxy, and $\Delta\phi_{ij}$ is the error in the radar measurement. The expression $k(\boldsymbol{D_i} - \boldsymbol{D_j}) \cdot \boldsymbol{n}$ is the 3D version of the 2D interferometry equation described in Chapter 2 (Equation 2.1) and represents the theoretical phase difference between the two measuring antennas.

From the coordinate system in Figure 5.6, the location of the radio galaxy is defined as an elevation angle α measured from the zenith ($90° - \theta$, where θ is defined by Equation 5.8), and an azimuth angle β measured from North. The baseline between the two measuring antennas is related to the coordinate system origin by the distance between the two antennas d_{ij} and the angle of the baseline with respect to North, γ_{ij}, or by the vectors $\boldsymbol{D_i}$ and $\boldsymbol{D_j}$. Therefore, Equation 5.10 can be rewritten as,

$$\boldsymbol{n} = [\sin(\beta)\sin(\alpha), \quad \cos(\beta)\sin(\alpha), \quad \cos(\alpha)] \tag{5.11}$$

$$k(\boldsymbol{D_i} - \boldsymbol{D_j}) = kd_{ij}[\sin(\gamma_{ij}), \quad \cos(\gamma_{ij}), \quad 0] \tag{5.12}$$

$$\phi_{ij} = kd_{ij}(\sin(\gamma_{ij})\sin(\beta)\sin(\alpha) + \cos(\gamma_{ij})\cos(\beta)\sin(\alpha)) + \Delta\phi_{ij} \tag{5.13}$$

The only value that is not known in Equation 5.13 is the error term $\Delta\phi_{ij}$. ϕ_{ij} is the phase

difference measurement, and the expression derived from $k(\boldsymbol{D_i} - \boldsymbol{D_j}) \cdot \boldsymbol{n}$ is the theoretical phase difference that should be measured between the two antennas i and j. Rearranging Equation 5.13 gives,

$$\Delta\phi_{ij} = \phi_{ij} - kd_{ij}(\sin(\gamma_{ij})\sin(\beta)\sin(\alpha) + \cos(\gamma_{ij})\cos(\beta)\sin(\alpha)) \tag{5.14}$$

where the error $\Delta\phi_{ij}$ in the measurement ϕ_{ij} is now calculated from known values. This error in the phase difference can then be used as the phase correction. Correction terms can be generated for each data point used in the analysis and then averaged to create a final correction value expressed as,

$$\Delta\overline{\phi_{ij}} = \frac{1}{N}\sum_{m=1}^{N}\Delta\phi_{ij}^{(m)} \tag{5.15}$$

where $\Delta\overline{\phi_{ij}}$ is the averaged phase correction term, N is the number of data points used to generate the correction, and m references the individual phase corrections that are averaged together into the final result. The variance, σ_{ij}, of the phase correction is then expressed as,

$$\sigma_{ij}^2 = \frac{1}{N}\sum_{m=1}^{N}(\Delta\phi_{ij}^{(m)} - \Delta\overline{\phi_{ij}})^2 \tag{5.16}$$

5.4.2 Generating Phase Corrections with Cygnus A

While Cygnus A is observable in the ICEBEAR FoV every day, the phase corrections must be from a quite period to be valid. If there is ionospheric activity, the calibration would be biased by the E-region coherent backscatter from plasma irregularities. Using only quite periods measured on 20 November 2020 and 20 December 2020, corrections were generated for each antenna pair or baseline, each providing a unique measurement of Cygnus A. In the analysis by Palmer et al. (1998), beam forming was used to track Cygnus A, such that the radar beam pattern maximum followed the position of the radio galaxy. The ICEBEAR detection of Cygnus A has been performed passively without beam forming. As a result, Cygnus A passes through peaks and nulls of the beam pattern. This results in the measured

phase difference changing as Cygnus A passes through each grating lobe of the beam pattern. The maximum possible phase difference between two antennas is 2π so wrapping occurs, with each 2π rotation of the phase difference corresponding to a grating lobe of the beam pattern. While the theoretical phase difference can wrap continuously as the radio galaxy position changes, the measured values only change while the radio source is within the ICEBEAR FoV. This again confirms that Cygnus A is the source responsible for the noise signatures detected in the ICEBEAR noise, and that calibration of the radar baselines is possible by comparing the measured radio source phase response to the expected signal phase response.

In selecting what data to use for the phase correction, a strong response to the radio galaxy is needed. If Cygnus A is in a null of the beam pattern, the background noise will be dominant over the Cygnus A signal. For data to be considered a strong response, the noise power measured must be >3 dB higher than the average noise power before/after Cygnus A is within the ICEBEAR beam pattern so that the effect of Cygnus A is the dominant source during the period of comparison. Figure 5.7 displays a comparison between the measured noise power for a baseline and the theoretical gain profile of the beam pattern Cygnus A passes through for that baseline. The data has been normalized so the two trends can be overlapped. It shows how the peaks and nulls of the noise power measurements of Cygnus A aligns with the theoretical beam pattern that Cygnus A passes through.

Power alone is not sufficient to identify a good response in the data set, and since the phase is the property being corrected it should also be used to locate and determine good response points. The point where a change in slope of the phase difference occurs is a good point to compare the measured and theoretical values. Slope change in the phase difference occurs as a result of the vertical progression of Cygnus A and the baseline orientation; baselines with a North-South component will measure this phase change, but it will not be present in East-West baselines that cannot measure the zenith angle change in the Cygnus A path. If the noise power at the phase change meets the >3 dB power condition, this point is used to calculate the phase error. When there is no slope change, or the power at that point is too low, the next best method of choosing data points to use in the analysis is to identify where the strongest affect of Cygnus A occurred. The theoretical calculation of the phase

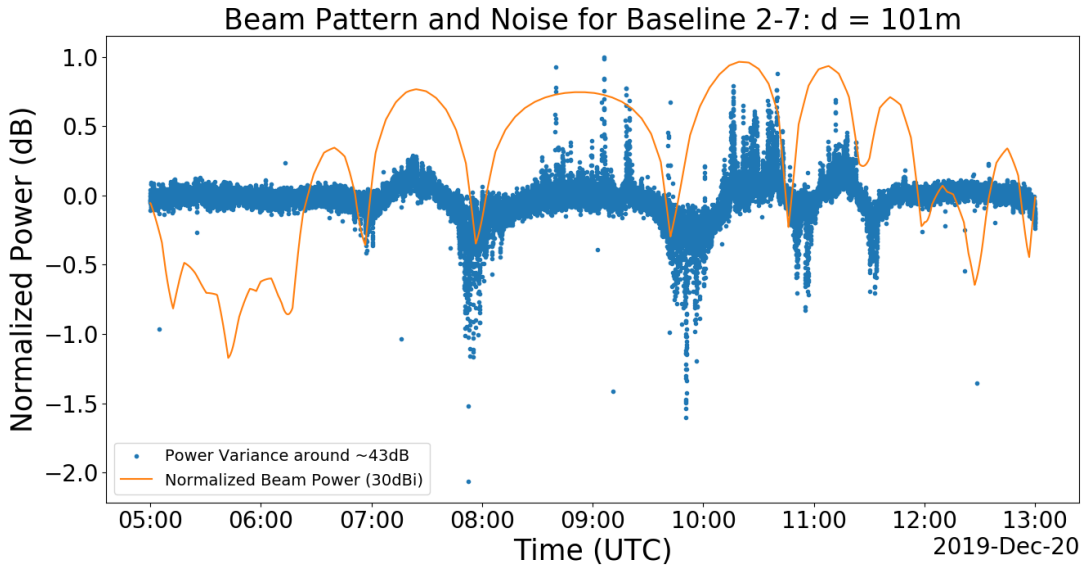


Figure 5.7: Comparison of measured noise power on antenna baseline 2-7 and the corresponding theoretical beam pattern gain of the baseline along the Cygnus A path using data measured on 20 December 2019. The beam pattern has been normalized by 30 dBi and the noise power has been shifted down by the average power (∼42.5 dB) and then normalized by the max power after shifting (∼5 dB). Note that the power peaks and nulls in the measurements align with the bean pattern of the baseline along the Cygnus A path.

difference between two antennas does not take into account the beam pattern, so it is not used to determine points of peak response. Instead, the measured data is smoothed using a running average, and points of stronger response are identified from the result. The running average is performed on the real and imaginary components of the data, and then converted to power and phase. Figure 5.8 shows an example of the smoothed power and phase data sets. For this example, baseline 1-3 does not contain a North-South baseline component, so there is no slope transition to compare against for the phase. Once the data points to be used in the analysis were selected, a range of 200 data points centered around the chosen response points were taken and used to calculate an average phase correction for the baseline.

Some baselines have a very poor response to Cygnus A and the reasons for this are discussed in Section 5.5.2. In these cases, the baseline cannot be used to generate a reasonable correction as the phase has little variation, so there is not enough of a pattern to compare to. This typically occurs when the background noise in the baseline is stronger than the signal from

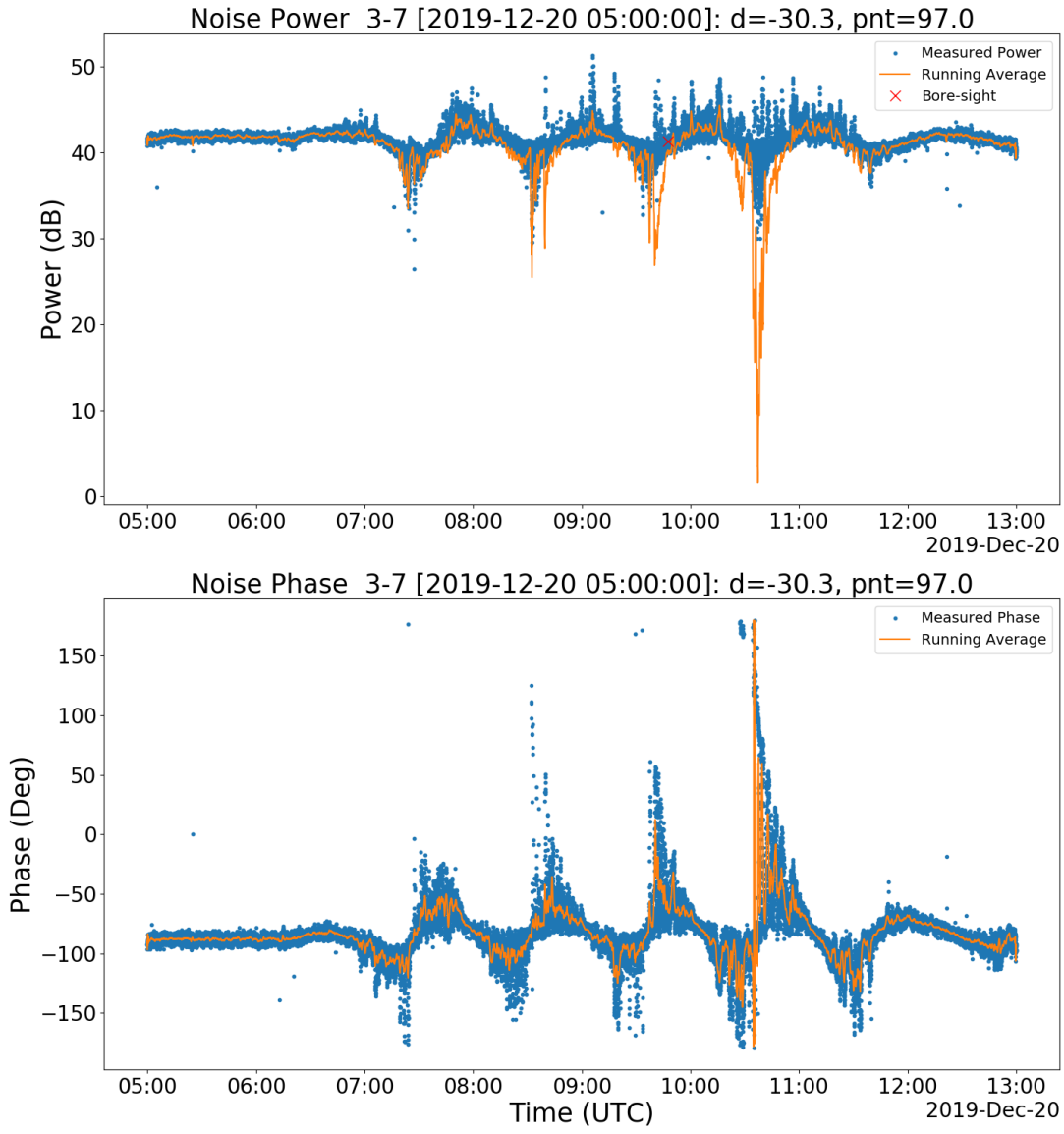


Figure 5.8: Noise power and phase measurements (blue) when Cygnus A was within the ICEBEAR FoV from 20 December 2019. The orange line is the data set smoothed by a running average. The running average was applied to the real and imaginary components, and then converted to power and phase. The overall behaviour of the data is preserved through the average while noise is reduced.

Cygnus A. If an antenna pair does not have a strong enough power response (>3 dB higher than the background noise level), or if the phase does not vary by more than 90°, it is tagged as a bad baseline; however, correction values are still generated for these baselines for the sake of consistency.

Phase corrections and variances were generated for each baseline using Equations 5.14, 5.15, and 5.16. These corrections were recorded so they could be used by the ICEBEAR analysis code. Bad baselines are tagged so they could be excluded from the data processing. Removing a baseline from the ICEBEAR imaging analysis will reduce the processing resolution capability, but it allowed for only valid corrections using the Cygnus A technique to be compared to phase corrections measured using the spectrum analyzer technique outlined in Section 5.4.

5.5 Calibration Results

Phase corrections generated by the Cygnus A technique were used to correct the measured phase difference for each baseline. The effect of the phase correction is shown in Figure 5.9, where the phase response of two baselines has been corrected using a Cygnus A observation. The corrected phase (orange) has been shifted by the phase correction and better aligns with the theoretical phase. Note that the measured phase curve always has a shallower slope than the theoretical phase curve for the Cygnus A observations. This effect is most prominent for short baselines, where there are less grating lobes in the beam pattern so there is less wrapping in the phase. This was determined to be a result of Cygnus A passing into the lower gain regions of the beam pattern grating lobes, so the background noise has a stronger bias on the measured phase difference. The measured patterns are not always phase shifted to perfectly match the theoretical prediction, but are always much closer after the correction is applied.

The Cygnus A phase corrections were applied to the ICEBEAR image processing technique. ICEBEAR image processing combines the 45 ICEBEAR receiver baselines (each unique antenna pair for the 10 receiver antennas in Figure 2.2) to determining the location, SNR, and

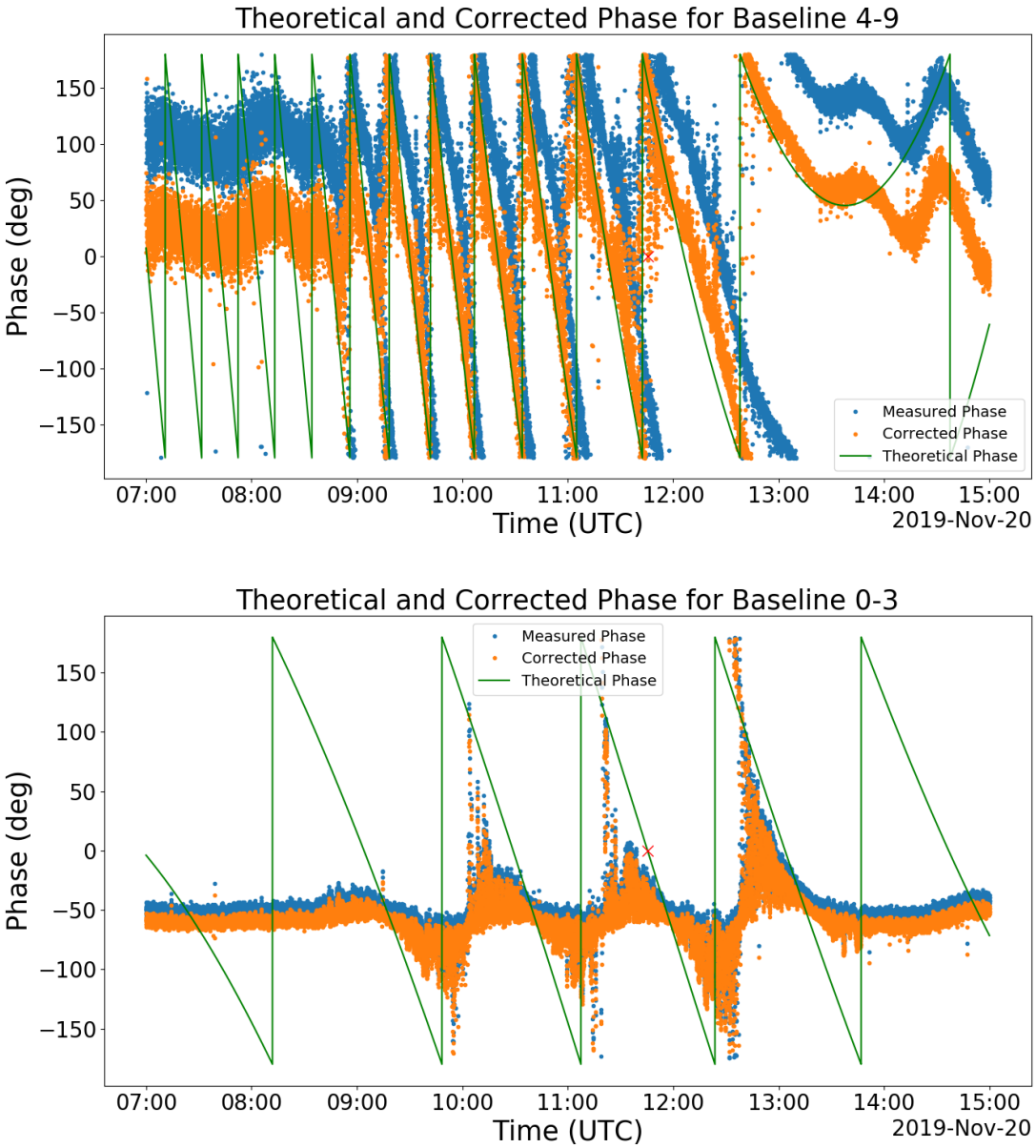


Figure 5.9: Measured phase (blue), corrected phase (orange), and theoretical phase (green) for baselines 4-9 (top) and 0-3 (bottom) using data from 20 November 2019. After the correction is applied, the data more closely aligns with the theoretical phase values.

Doppler velocity of E-region scatter. The range-Doppler matrices of each baseline (cross-spectra pair) are combined using spherical harmonic imaging (Lozinsky, personal communication, December 20, 2020). It is at this stage that baselines with a bad response to Cygnus A are excluded from the processing by zeroing (removing) the respective range-Doppler baseline matrix. For the corrections generated using the Cygnus A observation from 20 November 2019, baselines 1-5, 1-6, 2-5, 2-8, 4-5, 4-6, 5-6, 5-8, and 6-8 were marked as bad responses. To examine how these corrections affect the ICEBEAR imaging technique, they need to be applied to data collected during an active ionospheric period. Ionospheric scatter was detected on 19 December 2019, so the calibrations generated from the Cygnus A observation from 20 November 2019 were applied to this data. Figure 5.10 displays data processed from 5:00-6:00 UTC on 19 December 2019, using the Cygnus A phase corrections with the specified bad baselines removed. This figure has four panels which outline the range, position, and elevation of measured signals. The left two panels have plotted ICEBEAR measurements with respect to the West-East distance (x-axis) and the South-North (y-axis) distance from the ICEBEAR receiver by combining range and azimuth angle of arrival (AOA) of measured signals. The right two panels have plotted ICEBEAR measurements with respect to the altitude (x-axis) and South-North distance (y-axis) from the ICEBEAR receiver by combining range and elevation AOA of measured signals. The color scale used in the top two panels shows the Doppler velocity of the measured signals, and the color plot of the bottom two panels shows the SNR. This figure depicts where ICEBEAR measurements are mapped to, with the AOAs used affected by the phase calibration. With the Cygnus A phase corrections applied to the ICEBEAR data analysis, they could then be compared to the spectrum analyzer phase corrections.

5.5.1 Comparison to Spectrum Analyzer Phase Correction

To evaluate the Cygnus A calibration technique, it must be compared to the current spectrum analyzer calibration technique. However, the calibration results of the two techniques cannot be directly contrasted. The spectrum analyzer technique provides calibrations for each antenna signal chain, and is applied to the raw data measurements before processing. The Cygnus A technique generates phase calibrations for each baseline (antenna pair) af-

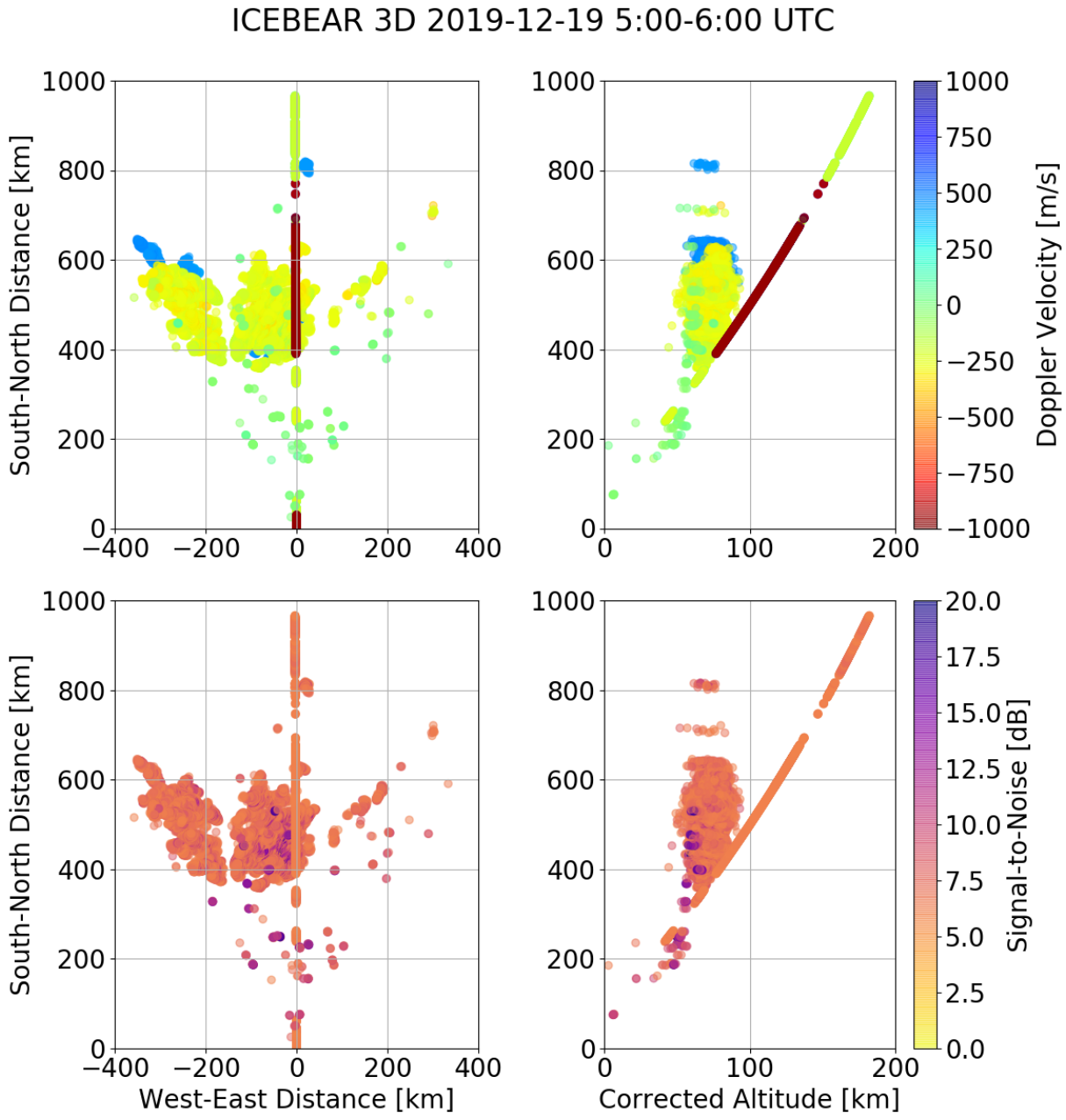


Figure 5.10: FoV Doppler velocity distribution (aziumithal top-left and altitude top-right) and FoV SNR distribution (azimuthal bottom-left and altitude bottom-right) of measured E-region scatter by the ICEBEAR receiver between 5:00 and 6:00 UTC, 19 December 2019. Generated using Cygnus A phase corrections on each baseline. Baselines 1-5, 1-6, 2-5, 2-8, 4-5, 4-6, 5-6, 5-8, and 6-8 were not used in processing this data.

ter a cross-correlation of the data from the 2 antennas has been performed (as described in Chapter 2, Section 2.3), and is therefore applied to the the data processing between the cross-correlation and imaging steps. While these two techniques calibrate the radar phase at different stages along the analysis, the final data processing outputs for each technique can be compared to evaluate the effect of the phase calibration on the AOAs determined by the data processing. Figure 5.11 displays the ICEBEAR imaging results when the phase of the ICEBEAR receiver antennas were corrected using the spectrum analyzer technique for the same period presented in Figure 5.10. To match the Cygnus A calibration technique, the same baselines that were removed in processing Figure 5.10 have also been removed from the spectrum analyzer results.

Both the Cygnus A and spectrum analyzer data sets have 145,372 data points measured over a 1 hour period. Since the techniques being evaluated here are phase calibrations, only the AOA measurements should be affected in the data products. To ensure that the other data products are not affected by the correction, the SNR in both data sets was also assessed. The average difference of the SNR, azimuthal angle of arrival, and the elevation angle of arrival between the two calibration techniques is listed in Table 5.1. It was found that between the two data sets, SNR had an average difference of $2.75\text{x}10^{-08}$ dB, azimuth AOA had a difference of 0.0242°, and elevation AOA had a difference of 0.956°. The SNR difference is attributed to floating point errors in the calculation, thus the SNR between the two data sets is unchanged by the phase corrections as would be expected.

SNR difference	$2.75\text{x}10^{-08}$ dB
Azimuth AOA difference	0.0242°
Elevation AOA difference	0.956°

Table 5.1: Average difference in SNR, azimuth AOA, and elevation AOA between ICEBEAR imaging outputs using the Cygnus A calibration technique and the spectrum analyzer calibration technique. The difference in the AOAs is small and shows agreement between the to calibration techniques. Note that the SNR difference between techniques is due to floating point errors in the data processing.

The azimuth AOA differences are small enough to conclude that the two techniques match within the angular resolution of ICEBEAR, but the elevation AOAs are not as well aligned.

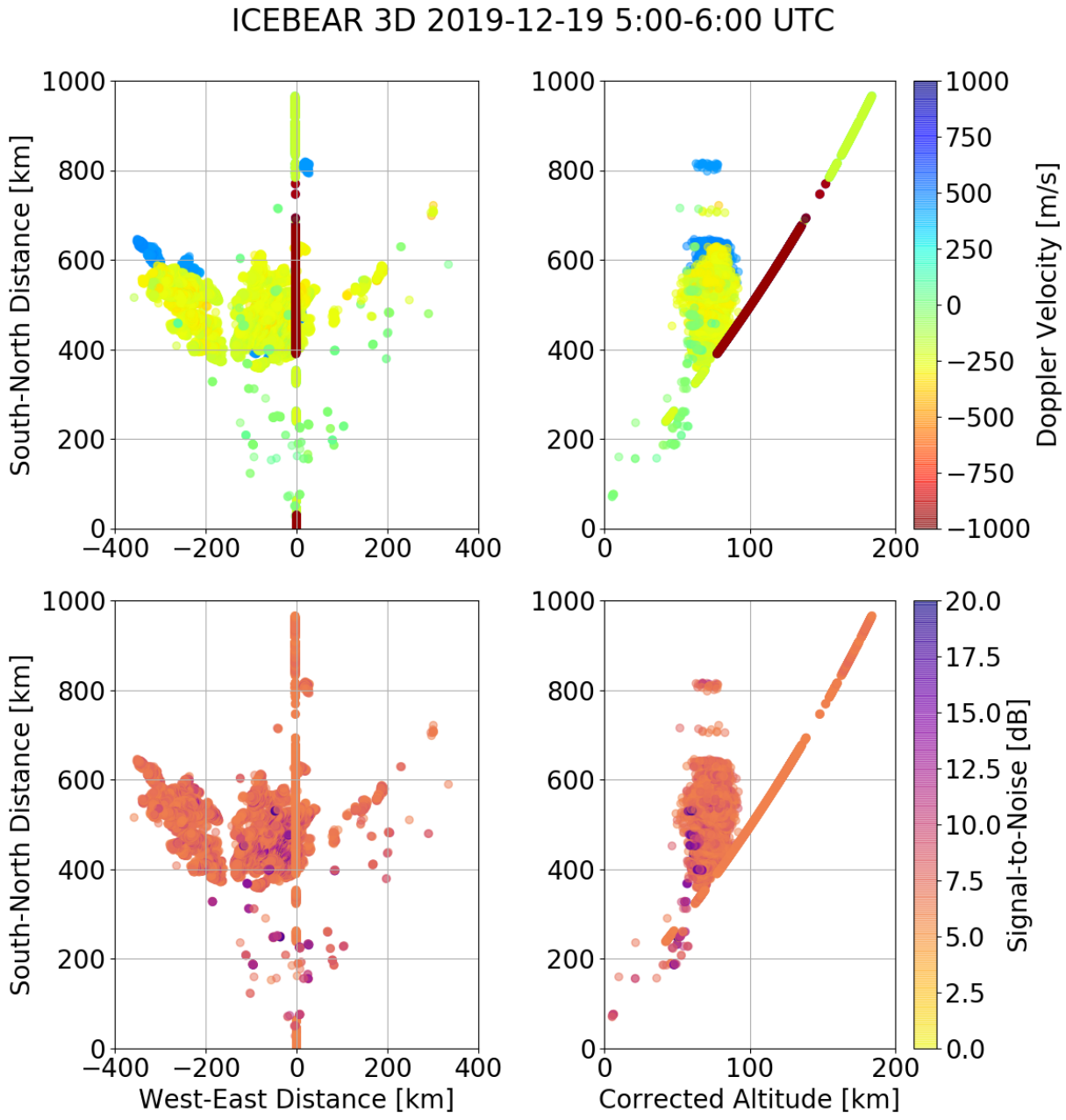


Figure 5.11: FoV Doppler velocity distribution (aziumithal top-left and altitude top-right) and FoV SNR distribution (azimuthal bottom-left and altitude bottom-right) of measured E-region scatter by the ICEBEAR receiver between 5:00 and 6:00 UTC, 2019, December 19. Generated using antenna phase corrections measured by a spectrum analyzer. Baselines 1-5, 1-6, 2-5, 2-8, 4-5, 4-6, 5-6, 5-8, and 6-8 were not used in processing this data to compare with Cygnus A phase correction results.

Elevation angles are more sensitive in ICEBEAR, and the baselines that were removed all had vertical components, which may have contributed to the greater error between techniques. Furthermore, diffraction of the radio signal from Cygnus A as it passes through the ionosphere to the ICEBEAR receiver could also contribute to the larger difference in AOA measurements as this diffraction would not be accounted for in calculation of the expected phase difference in the Cygnus A calibration technique. However, the elevation angles still fall within the angular resolution of the radar and are considered to be in agreement. Further work is being done refining ICEBEAR-3D elevation determination, and if proper corrections can be generated for the baselines that had low responses to Cygnus A, then the two techniques may better agree. Considering the spectrum analyzer and Cygnus A phase corrections take different approaches to correcting phase measurements, they have generated processed ICEBEAR data that is in strong agreement.

The strong agreement between the two techniques indicates the validity of using Cygnus A as a means to calibrate the ICEBEAR phase measurements. It conversely demonstrates the spectrum analyzer technique correctly quantifies and corrects the phase differences in the signal paths for each antenna. Using the Cygnus A technique is advantageous as it can be recalculated after any quiet measurement of Cygnus A to keep phase calibrations up to date, where as the spectrum analyzer requires making a site trip to perform the measurements. Implementing Cygnus A corrections will help monitor the ICEBEAR system, keep the phase calibrated at all times, and provide a check to, or be checked by, spectrum analyzer measurements when they are made.

5.5.2 Cygnus A Detection Issues

There is a varying level of response from Cygnus A on different baselines. Figure 5.12 displays a responsive and unresponsive baseline to Cygnus A. While these two examples show the extremes, a number of baselines have such a poor response to Cygnus A that they cannot be used in the Cygnus A phase calibration technique. Looking at various properties of the measured data, it was noted that there was a correlation between the noise power and phase response. Baselines that had a higher average noise power relative to the other baselines have

consistently poorer phase responses to Cygnus A. This indicates that strong background noise on certain baselines can overpower the signal from the radio galaxy. Figure 5.13 displays the noise power of the same responsive and unresponsive baselines as Figure 5.12, demonstrating the higher average noise power in the unresponsive baseline. In this particular example, the noise on the less responsive baseline (bottom plot) is $\sim$10 dB higher than the power on the responsive baseline (top plot). The responsiveness of a baseline, or the phase spread which is the difference between the maximum and minimum measured phase difference (maximum possible phase spread is 2π), was compared to the average noise power on the baseline for all baselines calculated using data on 20 November 2019 and 20 December 2019. The results of this comparison are shown in Figure 5.14, where the phase spread decreases exponentially with increasing average noise power. As the phase response decreases, it will be more difficult to compare the expected and measured phase for the Cygnus A calibration technique. It is expected that noise sources influencing these measurements are external and appear to only affect certain baselines, which may be a result of the unique beam pattern of the baseline measuring a stronger response to a noise source not observed by the other baselines. However, the noise power discrepancy could also be a result of internal factors in the signal path and system temperature, increasing the radar noise.

Some baselines appear to have a response that has a small temporal shift compared to the theoretical phase difference. The modeled beam pattern which represents how the baseline should respond to Cygnus A assumes both antennas are pointing 7° East of North, but if one of the antennas is rotated by 5°, the beam pattern shifts by approximately 2° and the power distribution is slightly altered. Therefore, small errors in the antenna pointing direction and positioning may compound into notable changes from the expected beam pattern and may contribute to misalignment of theory and measurements. If any support lines are loose, the antennas will twist in the wind which may cause this effect. While the general beam pattern maintains the same shape and small rotations will have little effect on the regular operation of ICEBEAR, these errors may be seen in the noise power and phase observations.

To find the cause of the power difference between baselines, multiple quiet days of Cygnus A data were examined. In addition to certain baselines having poor responses, there was also

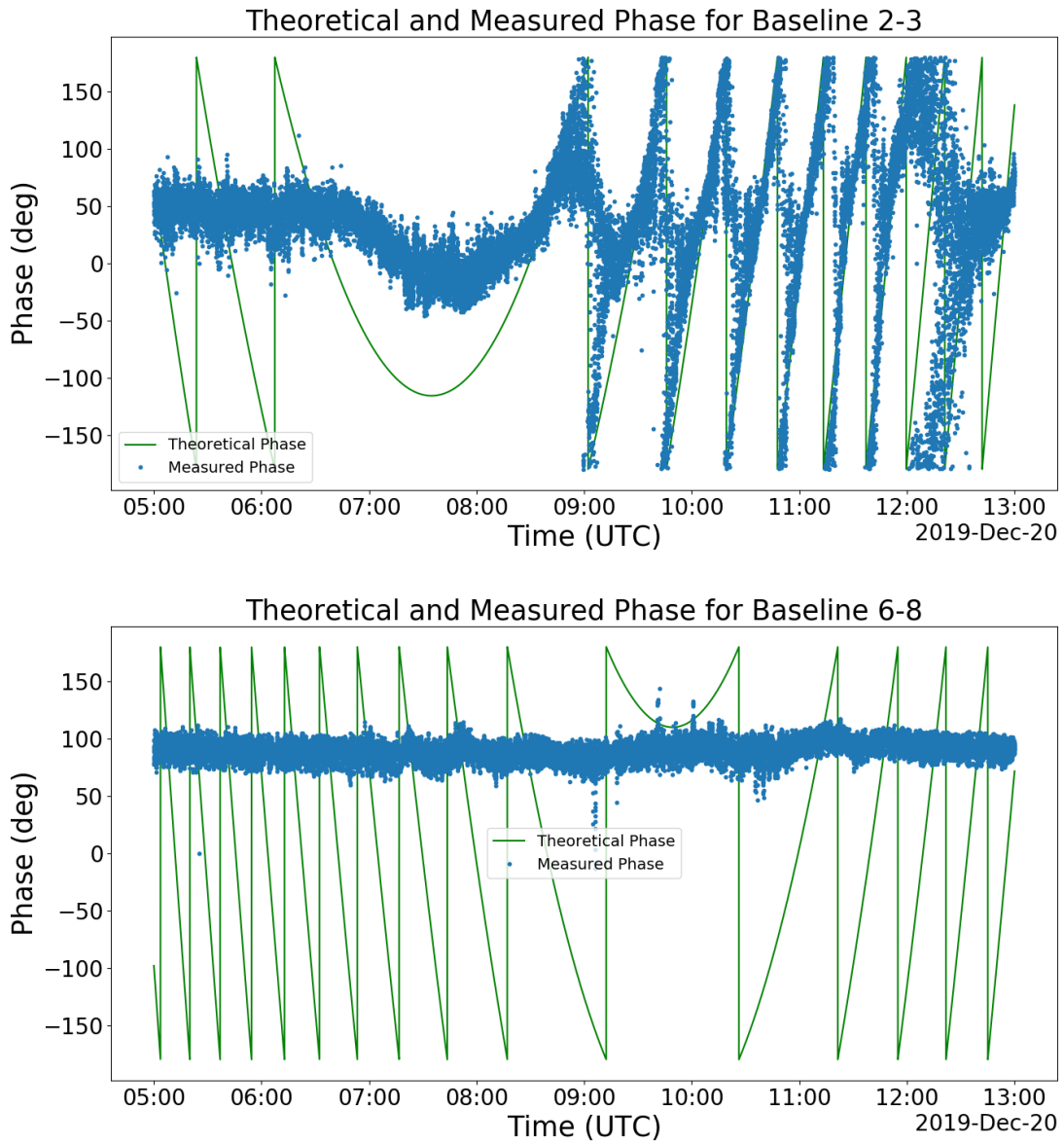


Figure 5.12: Comparison of measured phase difference (blue) and the theoretical phase difference (green) over an 8 hour period on 20 December 2019. The top plot is a responsive baseline (2-3) and the bottom plot is an unresponsive baseline (6-8) from the ICEBEAR-3D antenna layout configuration. Baseline 6-8 shows no notable pattern and therefore cannot be matched to the theoretical values, and corrections generated for baseline 6-8 will be meaningless. When there is a clear pattern in the phase response as in baseline 2-3, comparison of the measured and theoretical values will generate valid phase calibrations.

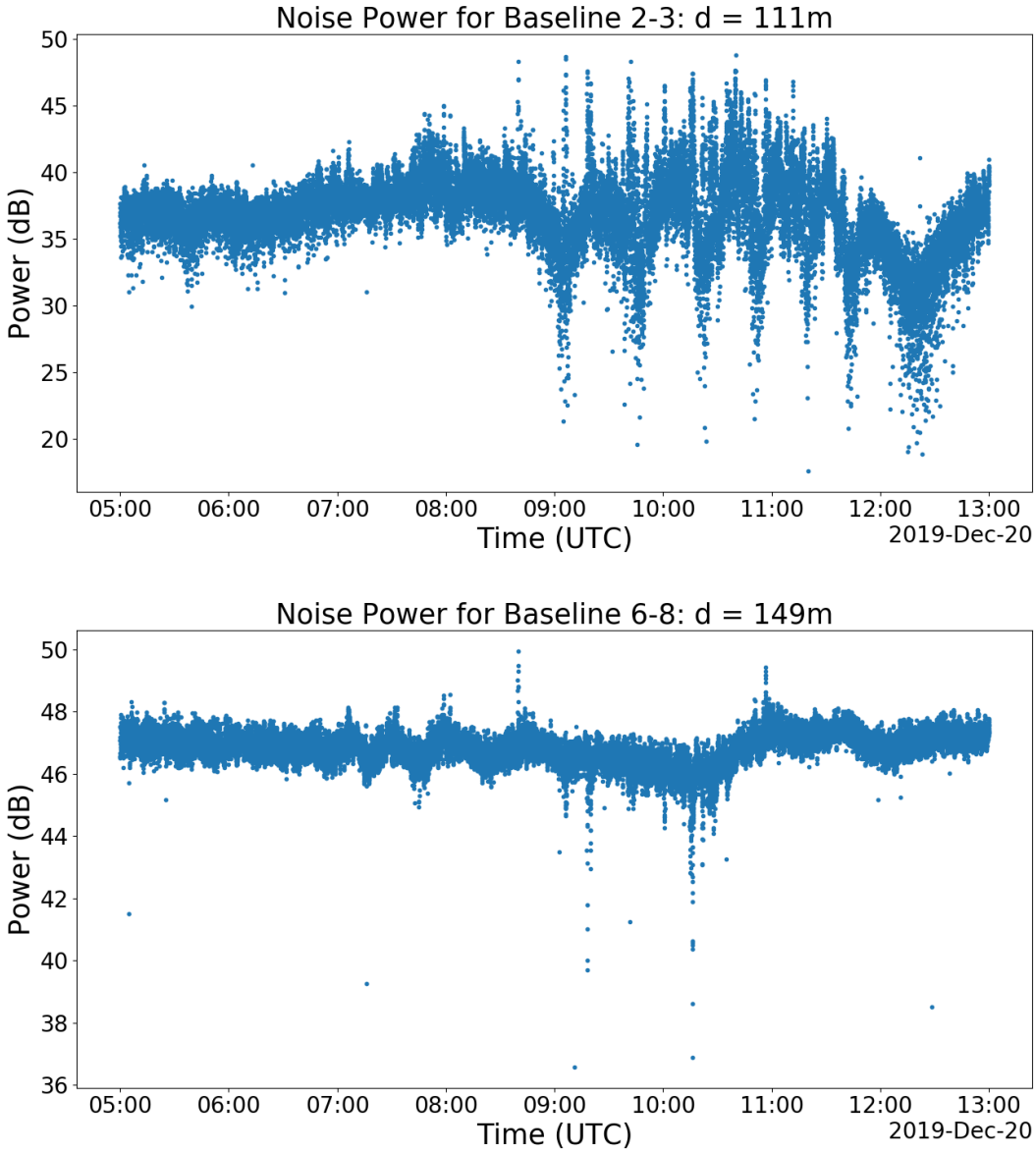


Figure 5.13: Noise power measured over the same 8 hour period as in Figure 5.12 on 20 December 2019. The top plot is a responsive baseline (2-3), and the bottom plot is an unresponsive baseline (2-6) in the ICEBEAR-3D antenna layout configuration. The distance between the antennas used in the baseline d, is shown in meters. Baselines that have a higher average noise power display less response to the presence of Cygnus A. In this example, the more responsive baseline has a power level $\sim$10 dB lower than the unresponsive baseline.

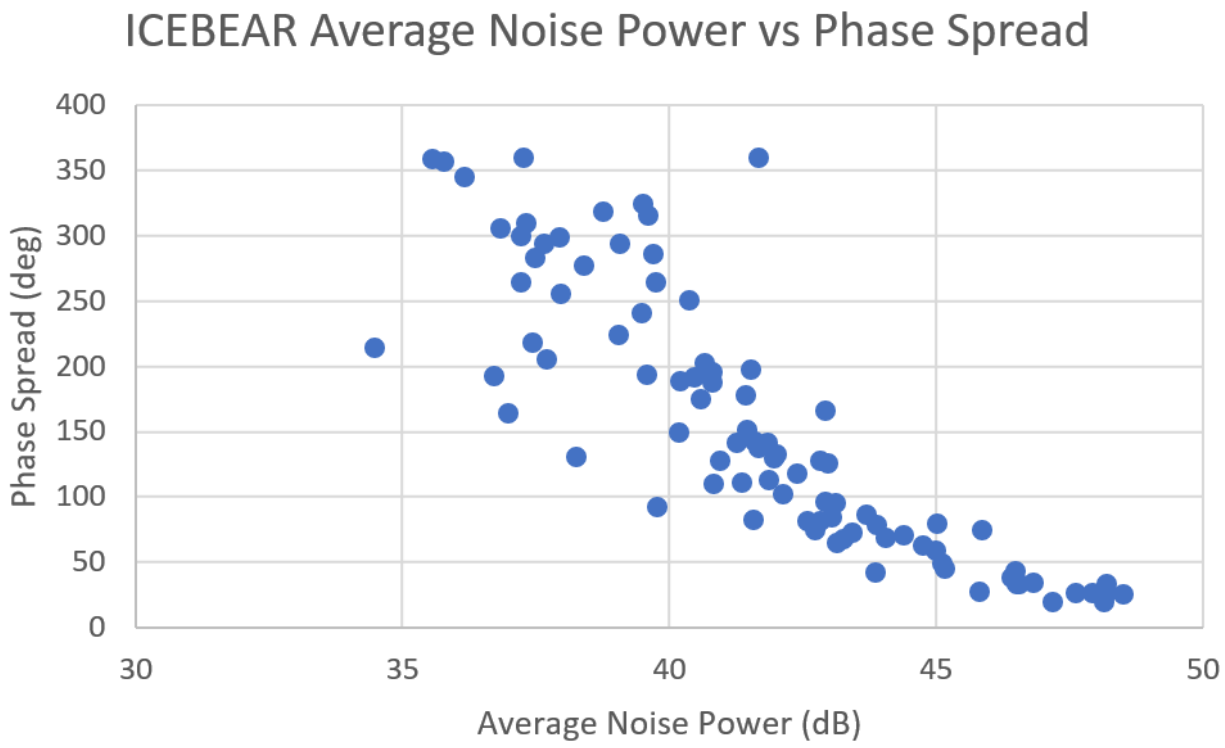


Figure 5.14: Each data point represents a baseline in the ICEBEAR radar assessed when Cygnus A was visible on either 20 November 2019 or 20 December 2019. There is a clear exponential decrease in the phase spread as the noise power increases, indicating a poorer response to the presence of Cygnus A

a varying level of response between the different months of observation that were compared.

First the local environment around the receiver antennas was considered. Positioning of the baselines in the receiver array were examined, looking at relative positions of other antennas, possible reflection off of nearby water, and the presence of obstacles such as trees, wires, and buildings was also considered. Other nearby antennas or wires may generate their own electromagnetic emissions that could influence a baseline measurement, or effect the signal propagation of Cygnus A. Water or changes in the ground (moisture) around the antennas could also effect the signal propagation. The presence of trees and buildings near the radar may block signal propagation which could lower power measurements on certain antennas. However, there was no clear consistency between these environmental factors and the measured signals, so no correlation could be drawn from the antenna positioning and the surrounding environment.

Riometer data from the GO-RIO project at the University of Calgary (Donovan & Spanswick, 2020) was examined to determine if differences in atmospheric absorption could attribute to the changes measured between baselines and changes measured on different days. Fort Smith, Gilliam, and Lucky Lake riometers were compared as they bound the edges of the ICEBEAR FoV. Absorption levels did change between times of observation and position, but none of these effects could be correlated to the changes detected on the baselines.

The raw antenna data was also examined during the Cygnus A period to determine if the varying responses to Cygnus A were a result of a processing artifact. Short samples of 6 s duration were examined in the ICEBEAR data and it was found that some noise sources correlated between all the antennas. Figure 5.15 displays an example of raw data cross-correlation between two antennas. Cygnus A was not present at the time of this analysis and no ionospheric scatter was occurring. The noise power of this correlation varies between baselines, and is stronger on baselines that were measured to have a higher average power during observation of Cygnus A, so a correlation between power, phase and, the correlation strength between data sets is present. The source of this noise could be interference from terrestrial radio sources or even other parts of the background cosmic noise being more prominent on different baselines due to their geometry (both the Rx receiver antenna array and the noise source). Regardless of the cause of this added noise, it is clearly present and hampers the observation of Cygnus A in the ICEBEAR radar.

5.6 Future Considerations

It has been shown in this chapter that ICEBEAR can use observations of Cygnus A to generate phase corrections. However, for some baselines these corrections cannot be generated due to the high background noise levels on the antenna pair masking signals from Cygnus A, making it impossible to get clear readings of the radio galaxy from that baseline. Even baselines with a strong response that clearly observe the transit of Cygnus A through the ICEBEAR beam pattern only have usable data during quite ionospheric periods and when Cygnus A is within the ICEBEAR FoV.

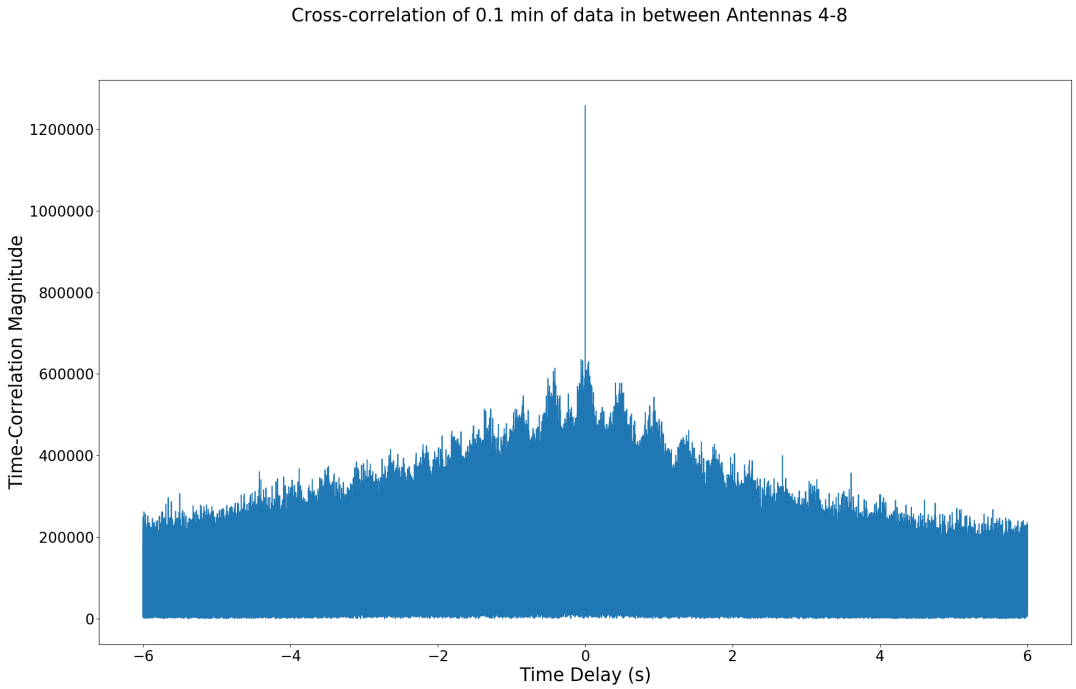


Figure 5.15: Cross-correlation in time of 6 s of raw data between antennas 4 and 8 taken around 2 UTC 21 January 2021. Cygnus A was not present at the time of this correlation analysis and no ionospheric scatter was occurring. This indicates that some noise source, such as the background cosmic noise or a terrestrial signal, was correlating across this baseline in the ICEBEAR data.

Implementation of beam forming in this analysis would allow the radar to track the predicted position of Cygnus A by manipulating its beam pattern to trace the path using a beam pattern grating lobe peak. This would ideally provide a stronger response to the presence of Cygnus A at a consistent level of power, which would provide a more robust correction factor. This could also help determine if the impact of other factors are influencing the ICEBEAR measurements of Cygnus A.

Another consideration is that the calculation of the theoretical phase difference for a baseline does not take into account possible height differences of the measuring antennas. Each antenna in ICEBEAR is currently mounted at 15 m above the ground, but future experiments may make use of antennas at differing heights. Therefore, the phase correction analysis may need to be expanded to include antenna altitude as well. The analysis steps will be the same, but will need to be derived again to properly take antenna height into account.

Further investigation into the cause of different baselines measuring varying noise powers and phase responses could be conducted. While the noise variations are small enough to have no effect on processing of regular ICEBEAR data, baselines with high noise power (from other currently unidentified sources) limit the usefulness of using Cygnus A to generate phase corrections. If the cause of this issue can be discovered, Cygnus A phase corrections could be generated for all baselines.

5.7 Summary

This chapter has presented a technique to perform phase calibrations of ICEBEAR measurements using the radio galaxy Cygnus A as it passes through the radar FoV. First it has been shown that Cygnus A is a detectable and unique source in the ICEBEAR measurements by examining the changes in the noise power and phase. Cygnus A has the strongest electromagnetic emissions of any radio galaxy and is within the ICEBEAR FoV precisely when the noise is enhanced. Using astronomical calculations, it was possible to determine the horizontal coordinates of Cygnus A with respect to the ICEBEAR receiver array at any time. This was used to calculate the phase difference that should be measured on a baseline that is observing the radio galaxy. By comparing the measured phase to the theoretical phase, the difference can be attributed to the phase error of the baseline measurement. This error can then be used to correct the phase on the baseline. Corrections were generated for each baseline using Cygnus A observations collected during quiet ionospheric periods. Noise data was selected from periods when the strongest response to the radio galaxy occurred, and the corrections were then applied to the ICEBEAR image data processing. The Cygnus A phase corrections were shown to agree with antenna phase corrections measured using the spectrum analyzer technique.

The Cygnus A signal is detectable in the ICEBEAR noise because it generally has a stronger signal than the other cosmic noise sources. However, Cygnus A is a weak source and its presence is lost during active periods when ionospheric scatter is observed, as the noise is dominated by cosmic background noise and/or self-clutter due to the radar waveform analysis technique. The sky has random noise sources with varying power and phases and as a whole

is randomly polarized. Changes in the ICEBEAR noise from cosmic or terrestrial origin could introduce a noise source with a stronger signal than Cygnus A, making the presence of Cygnus A undetectable. Due to higher noise power measurements on certain baselines resulting from some additional noise source stronger than Cygnus A, some baselines are unable to clearly detect the radio galaxy and cannot be used for the generation of phase corrections. While these baselines can be removed from higher level ICEBEAR analysis, this is a pervasive issue within the Cygnus A technique. At the present, the radio galaxy has been used to verify that the spectrum analyzer phase corrections are correct, but this radio source calibration technique cannot be used as a complete means of ICEBEAR phase calibration until a strong response to Cygnus A can be measured on all baselines at once. When this measurement can be made, all the baselines can be calibrated and Cygnus A will provide a reliable and adaptable technique of calibrating the ICEBEAR radar. Nonetheless, it can be used as a check to verify future spectrum analyzer measurements.

6 ICEBEAR Observations

6.1 Introduction

Previous chapters have presented the optimization of ICEBEAR analysis, the study and observation of the ICEBEAR noise floor, and a phase calibration method with the goal to improve ICEBEAR operations and data processing. This chapter will present some example ICEBEAR observation results. These results were generated using the GPU data processing speed improvement presented in Chapter 3, while improved understanding of the noise floor (Chapter 4) and phase calibrations (Chapter 5) have justified confidence in the results. ICEBEAR is used to study the E-region of the ionosphere, with plasma instabilities and meteor trails as the two main scattering sources.

First, observations of instability echos will be presented. The scattering parameters will be outlined and results from the ICEBEAR-linear receiver array (pre-summer 2019 receiver array configuration), as well as from the new ICEBEAR-3D receiver array configuration will be presented. Next detection of meteor trails by ICEBEAR will be covered. Standard occurrence of the meteors and typical measured values will be discussed. Example meteor measurements will be presented along with a comparison to SuperDARN measurements, which gives an example of collaborative studies that ICEBEAR can perform with other instruments.

6.2 Instability Echos

ICEBEAR observes instabilities in the E-region as radar echos. Instabilities form along the magnetic field lines of the Earth and are driven by energetic solar ionization and particle

precipitation of charged particles in the the upper atmosphere. These oscillating plasma structures coherently scatter radar signals if the radar wave propagation is perpendicular to the magnetic field lines and the instability has an oscillation frequency meeting the Bragg scattering condition of the radar signal wavelength. Measured ionospheric echos can provide insight into E-region dynamics by detecting structure formation and movement. This data can be further compared and combined with data sets from other near-Earth space observation instruments such as the SuperDARN radar network, the E-POP satellite mission, or the T-REX all-sky imagers to enhance the understanding of E-region plasma physics and space weather.

6.2.1 ICEBEAR-Linear Receiver Observations

The linear receiver array provided high resolution range and Doppler measurements in the azimuthal plane, but being a linear East-West oriented antenna array could not provide any elevation data of the instabilities. Multiple baselines with the same antenna separation in the array provided redundancy in the radar measurements. Figure 6.1 displays an example measurement taken during an active ionospheric period on 10 March 2018, mapped onto the geographical ICEBEAR field of view (FoV) and displays the Doppler velocity of the instabilities along with their azimuthal structure. Figure 6.2 shows the same data set, but now as a range-Doppler plot, displaying the signal to noise ratio (SNR) for each range and Doppler measurement. With the high time resolution capability of the radar, the formation, growth, and decay of instabilities can be investigated. This was a very active day, and Figure 6.2 shows a good example of ICEBEAR detecting the 4 classic categorizations of ionospheric echos that were outlined in Chapter 1.

6.2.2 ICEBEAR-3D Observations

With the reconfiguration of the ICEBEAR receiver antennas into a 2D layout, along with the implementation of new spherical harmonics processing methods making use of the GPU processing presented in this thesis, the ICEBEAR radar can now determine the elevation angle of arrival (AOA) of measured signals along with the azimuthal AOA (Lozinsky, per-

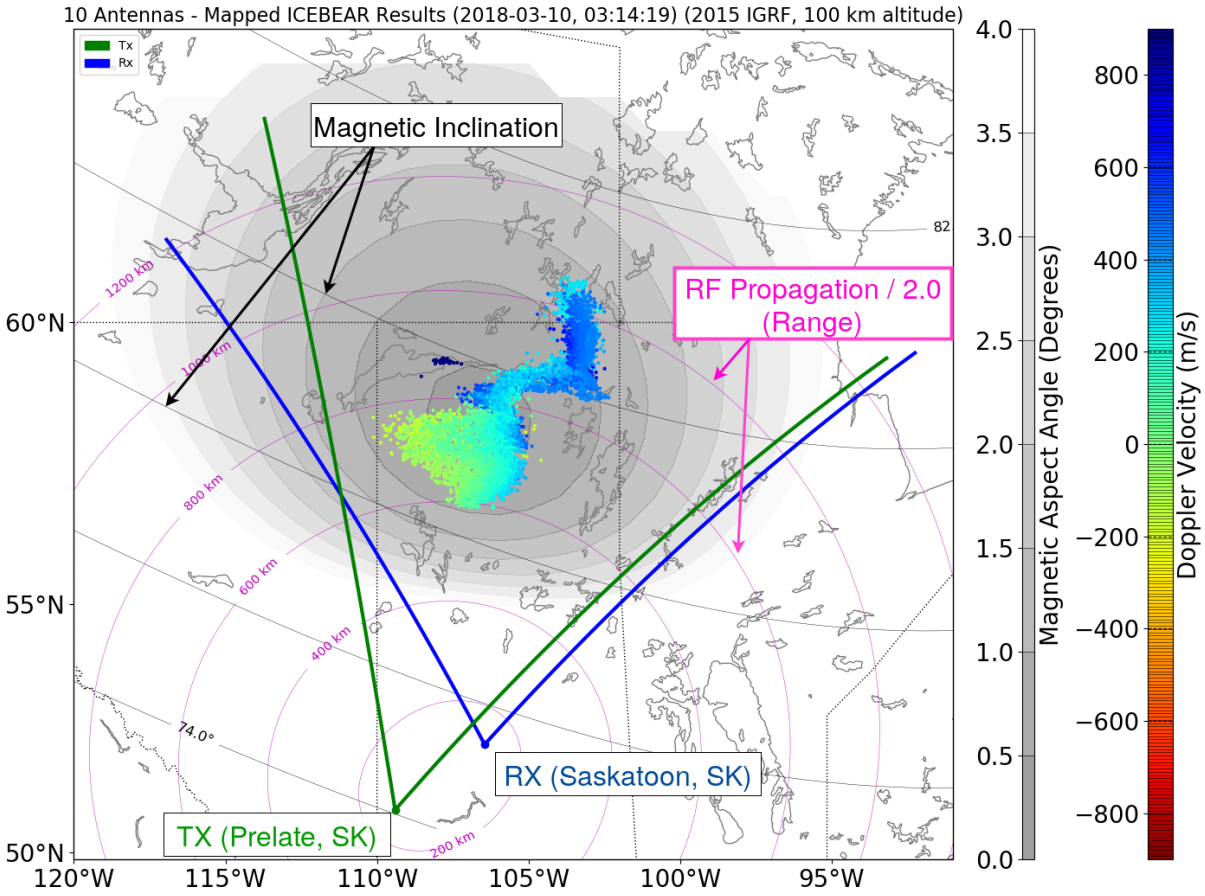


Figure 6.1: ICEBEAR data averaged over 5 seconds on 10 March 2018, at 3:14:19 UTC, mapped over the FoV using all 10 antennas of the ICEBEAR-linear receiver array. This plot depicts the Tx and Rx locations and FoVs, along with the magnetic inclination angle and Radio Frequency (RF) propagation distance. Doppler velocity of the measured signals is depicted using a color scale. The gray scale contour represents the magnetic aspect angle.

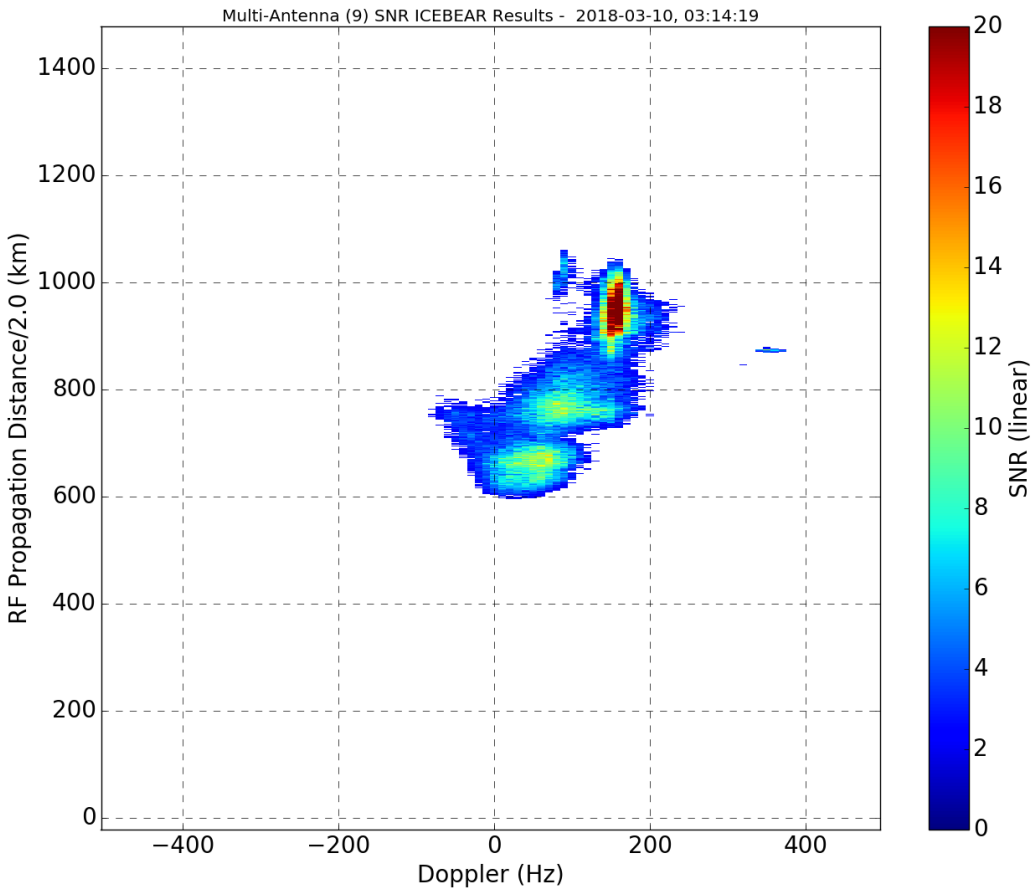


Figure 6.2: ICEBEAR data averaged over 5 seconds on 10 March 2018, at 3:14:19 UTC, depicted in a range-Doppler plot made using all 10 antennas (9 identical baselines) of the ICEBEAR-linear receiver array. The color scale represents the SNR measurements for each range-Doppler value.

sonal communication, December 20, 2020). This new radar configuration has been dubbed ICEBEAR-3D, as ICEBEAR can now generate 3D measurements using azimuth AOA, elevation AOA, and range. Calibration and refinement of the analysis process is still in progress, but preliminary processed results of measurements made with ICEBEAR-3D can now be generated and used for investigating E-region dynamics. Figure 5.11 in Chapter 5 depicts the mapped azimuth and elevation location of scatter collected between 5 and 6 UTC on 19 December 2019. The top row of images shows a color map of the Doppler velocity, and the bottom row shows a color map of the SNR, for both the azimuth and elevation extent of detected scatter. Just as with the linear receiver array, the high spatial and temporal resolution of the radar allows for detailed study of E-region dynamics.

6.3 Meteor Trail Echos

Meteors have long been studied and characterized in the space physics field. They enter the atmosphere at velocities ranging from $30-70$ km/s and leave trails of quasi-neutral ionized trails made from vaporized metallic ions from the meteor and ionized atmospheric particles along the meteor path. Radio waves reflect off electrons within the trail, allowing for radar observation of these trails, which typically form between 80-115 km (Bronshten, 1983). These trails can drift with the neutral winds in the upper atmosphere, and are most often detected around zero Doppler velocity. Meteor trails are categorized as underdense, transitional, and overdense. Categorization is based on the electron line density of the trail, as this impacts the scattering mechanism for radio waves, with overdense trails defined as electron densities $\alpha > 2.4 \times 10^{12}$ el/cm, underdense as $\alpha << 2.4 \times 10^{12}$ el/cm, and transitional is defined as densities between the two conditions (Bronshten, 1983). Trail decay is directly related to the electron density, as trails with a greater electron concentration take longer to decay, thus overdense meteor trails are the longest lived occurrence. The reflected radar signal from these trails is referred to as meteor trail echos and are detected by the ICEBEAR radar as strong signals that are localized within a couple of range gates.

6.3.1 ICEBEAR Observation and SuperDARN Comparison

To demonstrate the capability of ICEBEAR to detect meteor trail echos, a period of quiet ionospheric activity when Kp< 2 was assessed to find meteor trail echos and the results were then compared to Saskatoon SuperDARN measurements from the same time period. Quiet periods were examined so that no ionospheric scatter would be present in the ICEBEAR measurements, allowing for meteor trails to be easily identified from the data analysis. SuperDARN is known to detect meteor echos, the occurrence of which is often used as a means of calibration for the radar interferometers (Burrell et al., 2016; Chisham, 2018; Hussey et al., 2000). The ICEBEAR receiver and Saskatoon SuperDARN radar are both located just outside of the city of Saskatoon so measurements between the two locations can be compared through range and heading (East of North). The Eastern half of the ICEBEAR FoV overlaps with Saskatoon SuperDARN beams 0 through 9, so meteor echos that occur in this overlapping region will be detected by both radars.

ICEBEAR data between 12−18 UTC on 7 March 2018 was assessed using a 0.5 s averaging interval to detect meteor events. Approximately 200 meteor echos were detected, with events lasting for less than 0.5 s, to as long as 3 s. SuperDARN has 16 beams, each with a 3.24° beam width, and was operating in a mode that generated a full FoV plot every 60 s during this observation period by scanning through each beam (Chisham et al., 2007). Each beam collects data for a 3.75 s period, therefore to maximize the chance of overlap between the ICEBEAR measurement of a meteor echo with SuperDARN, longer lasting overdense meteors were the focus of this comparison. The meteor events detected by ICEBEAR were manually searched through. Three events that corresponded to signals detected in both radars are highlighted in Table 6.1, which shows the velocity, range, heading, and power measured by each radar. Comparing the two radar measurements, velocity, range, and heading of the detected meteor echos between the two radars match within error. The power measurements cannot be directly compared as they are not absolute, but the relative powers between the three events is in agreement between the two radars.

An example FoV comparison is presented in Figure 6.3 for meteor trail 3 from Table 6.1. The ICEBEAR data shown is a 0.5 s measurement of the ICEBEAR FoV, while the Saskatoon

Date: (2018-03-07)	Trail 1	Trail 2	Trail 3
UTC Start Time	12:34:11	12:34:32	16:09:11
Duration (s)	0.5	2.5	2.5

ICEBEAR			
Velocity (m/s)	30.3–60.6	30–60.6	−30.3–0
Range (km)	360	500	680
Heading (° E of N)	−5.3	30.6	9.3
Power (dB)	15	18	12

SuperDARN			
Velocity (m/s)	0–50	0–50	−50–0
Range (km)	405	495	675
Heading (° E of N)	3.7	32.8	10.1
Power (dB)	22–26	>26	18–22

Table 6.1: Comparison of 3 meteor trail events measured by ICEBEAR and SuperDARN. Velocity, range, and heading all agree between the two radars. As the power measurements are not absolute they are different, but the relative power between the events is the same.

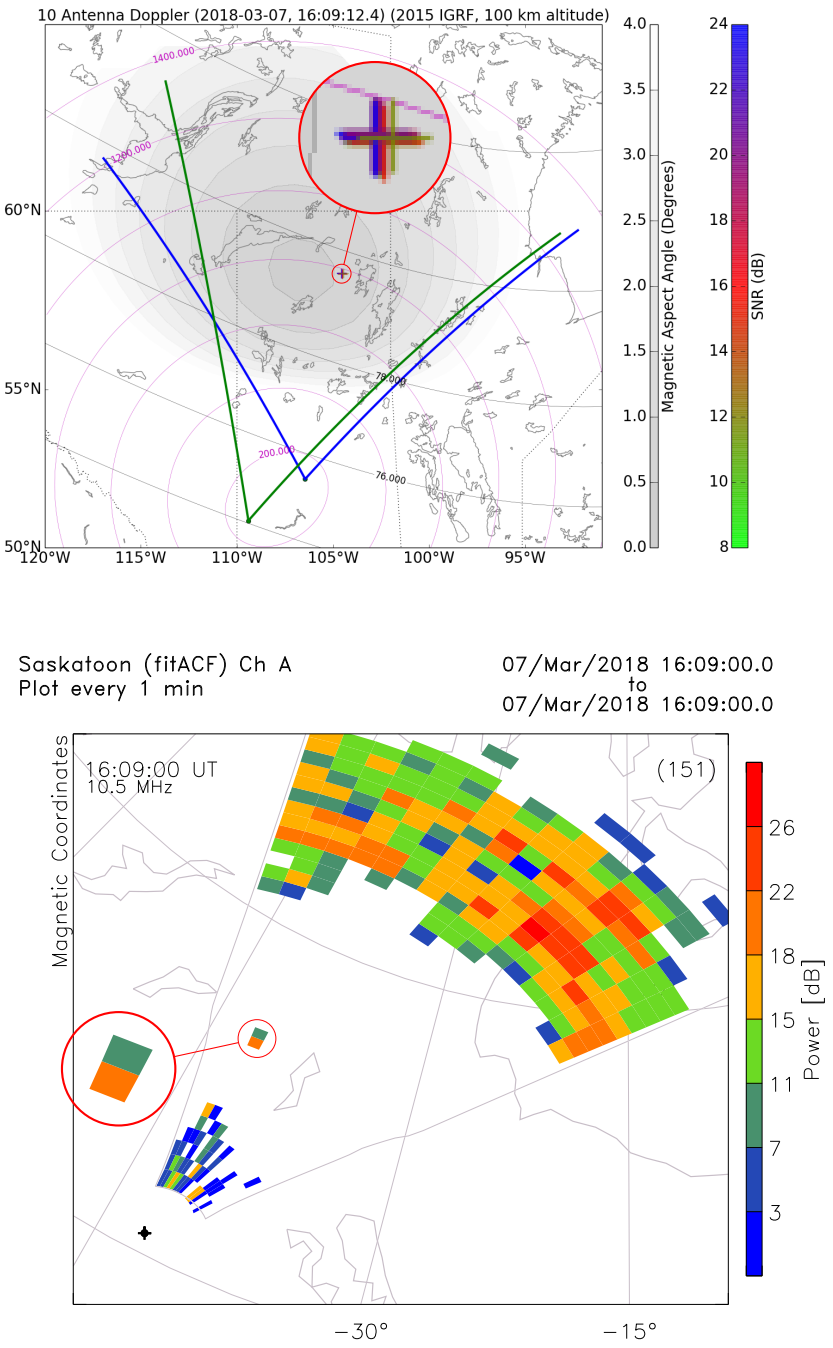


Figure 6.3: Comparison ICEBEAR and SuperDARN measurements on 7 March 2018, at 16:09 UTC. The meteor trail detected by the two radar is identified by a red circle on the FoV plot. While the two maps do not use the same projection, examining the location of the trail with respect to near by lake features shows the signal is located in the same general location by both radars. For ICEBEAR a 0.5 s measurement of the FoV is shown, and for the Saskatoon SupderDARN a 1 minute scan of the FoV is shown when beam 3 was scanning at the same time as the ICEBEAR measurement.

SuperDARN data shows a full FoV scan that occurred over the course of 1 minute. ICEBEAR measured the trail to last from 16:09:11 UTC to 16:09:13.5 UTC, which overlaps with when SuperDARN beam 3 and 4 were active. In the FoV plot of ICEBEAR data a single trail was detected and is marked by a red circle on the plot. In the SuperDARN FoV plot, a signal is detected in 2 range bins near the middle of the FoV and is marked by a red circle on the plot. While the projections used by the two images is not the same, the heading and range of the signal (shown in Table 6.1) measured by ICEBEAR is within 1° of the of the signature detected by SuperDARN highlighted in Figure 6.3 and only has a 5 km difference in range which is within the range resolution of the radars. This comparison between the two radars shows strong agreement in the measurement of a distinct and isolated signal. Future scientific collaborations between the radars will allow for multi-frequency observations of E-region activity.

Meteor trail observations by ICEBEAR can be used to validate the radar measurements as altitudes that meteor trails form at are well understood (Bronshten, 1983). They can also be used for scientific study, for example making measurements of the neutral wind at E-region altitudes.

6.4 Summary

The ICEBEAR radar is a unique radar capable of generating high spatial and temporal measurements, providing unique insight into the dynamics of the E-region. The research performed in this thesis has improved ICEBEAR analysis capability to process measured data in real-time. Calibration using Cygnus A presented in Chapter 5, along with the increased understanding of the ICEBEAR background noise, presents verification of the ICEBEAR power and phase measurements. When used in conjunction with other instruments, ICEBEAR will play an important roll in improving our understanding of space physics. Preliminary meteor trail echo comparisons with the Saskatoon SuperDARN radar has shown the reliability of ICEBEAR measurements and the roll ICEBEAR can play in multi-instrument observations.

7 Conclusion

The E-region of the ionosphere is a dynamic layer of the atmosphere where enhanced solar and magnetosphere activity can drive currents through the conductive plasma layer forming localized plasma instabilities and dynamics. The ICEBEAR instrument was designed to provide high resolution observations of the E-region, enabling further study of the dynamics in this region to improve our understanding of space weather. This thesis has presented research that has been done to improve and verify the functionality of the ICEBEAR instrument to define its role in space weather observations.

Chapter 1 presented the main goals of this thesis: present improvements to the computation speed of ICEBEAR data analysis using graphical processing unit (GPU) acceleration, study the ICEBEAR noise floor to provide better understanding of the ICEBEAR noise environment and verify the signal to noise ratio (SNR) data product, and present a new technique of phase calibration for ICEBEAR using observations of the radio galaxy Cygnus A. The basics of plasma physics was outlined, with a focus on the parameters defining the E-region of the ionosphere and the driving forces of instabilities in that region.

The Ionospheric Continuous wave (CW) E-region Bi-static Experimental Auroral Radar (ICEBEAR) instrument was outlined in Chapter 2. This chapter presented the hardware components of the ICEBEAR antennas, transmitter, and receiver systems. ICEBEAR uses a 49.5 MHz transmission frequency modulated by a binary Pseudo-Random Noise (PRN) code. The current configuration of the system transmits a 10 μs symbol length for the 10,000 element PRN code. ICEBEAR nominally has a 10 Hz frequency resolution, a 0.1s temporal resolution, and a 1.5 km range resolution. The radar has a bandwidth of 100 kHz, and a maximum range of 30,000 km. Samples are measured in-phase and quadrature to create complex voltage measurements of the detected power and phase of radio signals. Interfer-

ometry between antennas is used to determine the angle of arrival (AOA) of these measured signals. The original ICEBEAR receiver antenna array could produce only aziumthal AOA measurements, but the new ICEBEAR-3D 2D receiver antenna configuration enables measurement of both the azimuth and elevation AOA of a signal. Using the PRN code, the noise like scattered signal can be decoded from the background noise in the voltage measurements. The analysis process of retrieving ICEBEAR data was presented in Chapter 2, displaying how ICEBEAR determines the SNR, Doppler velocity, and position of each scattering volume in the field of view (FoV).

Chapter 3 presented how the data analysis outline in the previous chapter was accelerated by performing the data processing on a Graphical Processing Unit (GPU). GPUs are capable of much higher computational throughput when dealing with instructions or commands that are repetitive or can be run in parallel. The basic structure of the GPU was outlined along with how parallel processing utilizes that structure. ICEBEAR uses a NVIDIA GeForce GTX 1080 Ti GPU, and the architecture of the GPU used influences the structure of the parallel programming. As such, the detailed layout of the 1080 Ti was outlined before presenting how the analysis code was parallelized. The parallel implementation of the presented analysis utilized only 49.6% of the computational ability of the GPU, but resulted in improved calculation speeds. The parallel analysis was able to compute 5 s of data in only 0.0774 s, which is 4793.3% (48 times) faster than the original sequential analysis. ICEBEAR data processing is now capable of computing a 5 s average for all 45 unique receiver baselines in 3.48 s which is faster than real-time.

Chapter 4 presented an analysis of the background noise in ICEBEAR measurements. It outlined the methods used to process and determine ICEBEAR noise for the determination of the SNR data product. Based on the results of processing data from both receiver antenna array configurations, it is recommended that ICEBEAR switch to using an average of the furthest range gates to calculate noise estimates. The current FoV median technique is subject to greater variability on long baselines. Far ranges in the ICEBEAR radar will contain only noise values as these ranges are far beyond where E-region scatter occurs and are above where meteor trails form. Satellites and planes could appear in this data, but

they will be short lived signals that will be averaged out in calculating the noise. Further examination of the noise was performed to examine the diurnal trends and dominating noise sources in the radar FoV. The repeating noise pattern, along with the temporal drift of the pattern over time, demonstrated that ICEBEAR noise is dominated by cosmic sources. In addition, it was shown that the clutter-correction used to remove phase bias from ICEBEAR measurements from signal processing and self-clutter was effective.

Chapter 5 outlined a method of performing phase calibrations for the ICEBEAR radar using the radio galaxy Cygnus A. It was first shown that Cygnus A was observable by the ICEBEAR radar by examining the position of various strong cosmic radio emission sources in the sky and comparing this to regions of the sky observed by ICEBEAR as the Earth rotates. Expressions were then presented that could be used to calculate the phase error of a baseline measurement based on the baseline orientation and the radio source position. Based on these relationships, phase error corrections were created for each baseline in the ICEBEAR receiver antenna array configuration. Using these Cygnus A generated corrections, processed data was compared with results using phase corrections generated using the spectrum analyzer technique. Considering only the baselines which the Cygnus A technique was able to generate phase corrections for, the two calibration techniques were shown to be essentially identical. This demonstrated the validity of the stellar phase calibration method, and conversely supported the accuracy of the spectrum analyzer technique. Further research is required to investigate why Cygnus A was undetectable in certain baselines, but it is expected to be due to the presence of other noise sources in the sky (likely terrestrial). Nonetheless, the the Cygnus A cosmic radio source calibration technique has been shown to be a usable and reliable phase calibration technique.

Finally, Chapter 6 provided an outline of the observational capabilities of the ICEBEAR radar. The improvements and analysis performed on the ICEBEAR instrument presented in this thesis has focused on ensuring that ICEBEAR is an effective and reliable device for studying the E-region and space weather. This chapter presented range-Doppler plots and FoV maps of E-region activity from both the original linear receiver antenna configuration, and the new 2D configuration (ICEBEAR-3D). It also demonstrated some meteor trail ob-

servations and a comparison to measurements from SuperDARN, highlighting the potential for collaborative studies of ionospheric plasma dynamics between multiple instruments.

7.1 Future Work

As technology and the ICEBEAR data processing algorithms improve, the GPU accelerated code can be expanded and improved upon to provide further improvements to data processing speeds. New aspects of the data processing can be parallelized and implemented in the GPU environment, further utilizing processing power improvements possible using GPU systems. Significant technological advances have almost become expected from the technology industry. New GPUs will be better designed for the complex data analysis required by ICEBEAR, and the radar will be able to take advantage of these technology improvements to further enhance E-region plasma physics understanding. When these new devices become available, they can be integrated into the ICEBEAR processing systems and current CUDA algorithms can be adapted to utilize the new available hardware.

As was discussed in Chapter 5 Section 5.6, there are improvements that must be made to the Cygnus A phase calibration technique before it can be used routinely for ICEBEAR phase calibrations. Future projects can investigate the causes of inconsistent measurements of Cygnus A. This could be a result of ionospheric activity influencing the Cygnus A signal, temperature influences on radar equipment, nearby radio sources influencing particular baselines, or perhaps a combination of multiple influences. Statistical analysis of the ICEBEAR noise and Cygnus A detection will provide further insight by providing a larger data base of ICEBEAR phase calibrations.

References

Alabaster, C. (2012). *Pulse doppler radar: Principles, technology, applications.*

Baumjohann, W., & Treumann, R. (1997). Basic space plasma physics. *Imperial College Press*.

Bronshten, V. (1983). Physics of meteoric phenomena. *Springer, Geophysics and Astrophysics Monographs*.

Burrell, A., Yeoman, T., Milan, S., & Lester, M. (2016). Phase calibration of interferometer arrays at high-frequency radars. *Radio Science*, *51*, 1445-1456.

Chau, J., & St.-Maurice, J.-P. (2016). Unusual 5 m e region field-aligned irregularities observed from northern germany during the magnetic storm of 17 march 2015. *Journal of Geophysical Research: Space Physics*, *121*, 10,316-10,340.

Chisham, G. (2018). Calibrating superdarn interferometers using meteor backscatter. *Radio Science*, *53*, 761-764.

Chisham, G., Lester, M., Milan, S., Freeman, M., Bristow, W., Grocott, A., ... Walker, A. (2007). A decade of the super dual auroral radar network (superdarn): scientific achievements, new techniques, and future directions. *Surv. Geophys*, *28*, 33-109.

Conway, R. (1974). Radio measurements in polarization. *Planets, Stars, and Nebulae studied with Photopolarimetry*, 352-358.

Cooper, J. (2006). A 50 mhz fmcw radar for the study of e-region coherent backscatter. *University of Saskatchewan*, *Masters Thesis*.

Cushcraft, C. (2018a). 617-6b 6 meter boomer antenna. *Radio Manual*. Retrieved 2020, May 21, from `http://www.radiomanual.info/schemi/ACC_antenna/Cushcraft_617-6B_user.pdf`

Cushcraft, C. (2018b). A50-5s. *Radioworld*. Retrieved 2020, May 21, from `https://www.radioworld.ca/cus-a505s`

Donovan, E., & Spanswick, E. (2020). Go-rio project. *University of Calgary's Auroral Imaging Group (AIG)*. Retrieved 2020, September 21, from `https://data-portal.phys.ucalgary.ca/`

Duffett-Smith, P. (1988). *Practical astronomy with your calculator.*

Elliott, H., Jahn, J., & McComas, D. (2013). The kp index and solar wind speed relationship: Insights for improving space weather forecasts. *Space Weather*, *11*, 339-349.

Ettus, R. (2018a). Octoclock. *Ettus Research*. Retrieved 2020, May 25, from `https://www.ettus.com/all-products/octoclock/`

Ettus, R. (2018b). Usrp x300. *Ettus Research*. Retrieved 2020, May 26, from `https://www.ettus.com/all-products/x300-kit/`

Farley, D. (2009). The equatorial e-region and its plasma instabilities: a tutorial. *Annales Geophysicae*, *27*, 1509-1520.

Fejer, B., & Kelly, M. (1980). Ionospheric irregularities. *Reviews of Geophysics*, *18(2)*, 401-454.

GFZ-Helmholtz-Center-Postdam. (2020). Kp index records. Retrieved 2020, October 15, from `ftp://ftp.gfz-potsdam.de/pub/home/obs/kp-ap/`

Google. (n.d.). Icebear receiver site. Retrieved 2021, April 20, from `https://www.google.com/maps/@52.2430508,-106.4502157,204m/data=!3m1!1e3`

Greenwald, R., Baker, K., Dudeney, J., Pinnock, M., Jones, T., Thomas, E., ... Yamagishi, H. (1995). Darn/superdarn. *Space Science Reviews*, *71*, 761-796.

Hall, G., & Moorcroft, D. (1992). Magnetic aspect angle effects in radar aurora at 48.5 mhz, corrected for refraction. *Journal of Geophysical Research*, *97*, 19,471-19,488.

Hariharan, P. (2007). *Basics of interferometry second edition*.

Harris, M., & NVIDIA. (2019). Optimizing parallel reduction in cuda. Retrieved 2018, March 7, from `https://developer.download.nvidia.com/assets/cuda/files/reduction.pdf`

Hashiguchi, H., Manjo, T., & Yamamoto, M. (2018). Development of middle and upper atmosphere radar real-time processing system with adaptive clutter rejection. *Radio Science*, *53*, 83-92.

Hecht, E. (2016). *Optics*.

Hey, H. (1983). The radio universe, 3rd edition. *Pergamon Press, Oxford*.

Hunsucker, R., & Hargreaves, J. (2003). The high latitude ionosphere and its effects on radio propagation. *Cambridge University Press*, *Atmospheric and Space Science Series*.

Hussey, G. (1994). The polarization of 50 mhz auroral backscatter. *University of Saskatchewan*, *Ph. D. Thesis*.

Hussey, G., Meek, C., Andre, D., Manson, A., Sofko, G., & Hall, C. (2000). A comparison of northern hemisphere winds using superdarn meteor trail and mf radar wind measurements. *Journal of Geophysical Research*, *105*, 18,053-18,066.

Huyghebaert, D. (2019). The ionospheric continuous-wave e-region bistatic experimental auroral radar (icebear). *University of Saskatchewan*, *Ph.D. Thesis*.

Huyghebaert, D., Hussey, G., Vierinen, J., McWilliams, K., & St-Maurice, J. (2019). Icebear: An all-digital bistatic coded continuous wave radar for studies of the e region of the ionosphere. *Radio Science*, *54*, 349-364.

Hysell, D. (2018). Antennas and radar for environmental scientists and engineers. *Cambridge University Press*.

Hysell, D., Larsen, M., & Sulzer, M. (2016). Observational evidence for new instabilities in the midlatitude e and f region. *Annales Geophysicae*, *34*, 927-941.

Jackson, T., Labs. (2018). Fury gpsdo. *Jackson-Labs*. Retrieved 2020, May 25, from `http://www.jackson-labs.com/index.php/products/fury`

Kivelson, M., & Russell, C. (1995). Introduction to space physics. *Cambridge University Press*.

Ko, H. (1958). The distribution of cosmic radio background radiation. *Proceedings of the IRE*, *46.1*, 208-215.

Koehler, J., Sofko, G., Maguire, M., Osterried, R., McKibben, M., Mu, J., ... Ortlepp, A. (1995). The sapphire auroral radar system. *Canadian Journal of Physics*, *73*, 211-266.

KR, E. (2018). 3333-sma bpf. *KR Electronics*. Retrieved 2020, May 26, from `http://krfilters.com/filter-docs/3333-SMA.pdf`

Kustov, A., Uspensky, M., Sofko, G., Koehler, J., & Mu, J. (1994). Aspect angle dependence of the radar aurora doppler velocity. *Journal of Geophysical Research*, *99*, 2131-2144.

Memeti, S., Li, L., Pllana, S., Kolodziej, J., & Kessler, C. (2017). Benchmarking opencl, openacc, openmp, and cuda: programming productivity, performance, and energy consumption. *Arxiv*. Retrieved January 14th, 2018, from `https://arxiv.org/pdf/1704.05316.pdf`

Mini, C. (2018). Zfl-500ln/zfl-500ln+. *Mini Circuits*. Retrieved 2020, May 26, from `https://www.minicircuits.com/pdfs/ZFL-500LN.pdf`

Nickolls, J., & Dally, W. (2010). The gpu computing era. *IEEE Micro*, *30*, 56-69.

NVIDIA. (2009). Nvidia's next generation cuda compute architecture: Fermi v1.1. Retrieved 2020, June 8, from `https://www.nvidia.com/content/PDF/fermi\_white\_papers/NVIDIA\_Fermi\_Compute\_Architecture\_Whitepaper.pdf`

NVIDIA. (2016). Geforce gtx 1080 whitepaper. Retrieved 2020, June 4, from `https://international.download.nvidia.com/geforce-com/international/pdfs/GeForce\_GTX\_1080\_Whitepaper\_FINAL.pdf`

NVIDIA. (2017). Geforce gtx 1080 ti. Retrieved 2020, June 15, from `https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-1080-ti/specifications`

NVIDIA. (2019a). Cuda toolkit documentation v10.1.243. Retrieved 2020, June 15, from `https://docs.nvidia.com/cuda/archive/10.1/`

NVIDIA. (2019b). Cuda toolkit download. Retrieved 2020, June 15, from `https://developer.nvidia.com/cuda-toolkit`

Palmer, R., Vangal, S., Larsen, M., Fukao, S., Nakamura, T., & Yamamoto, M. (1996). Phase calibration of vhf spatial interferometry radars using stellar sources. *Radio Science*, *31*, 147-156.

Python-Software-Foundation. (2020). Python language e-download. Retrieved 2020, June 23, from `https://www.python.org/downloads/`

R., V., Rideout, W., Swoboda, J., Vierinen, J., & Lind, F. (2021). Digital rf (version 2.6.6). Retrieved 2021, March 4, from `https://github.com/MITHaystack/digital\_rf`

Richards, M., Scheer, J., & Holm, W. (2010). Principals of modern radar: Basic principles. *Edison, NJ: Scitech Publishing*.

Sahr, J., & Fejer, B. (1996). Auroral electrojet plasma irregularity theory and experiment: A critical review of present understanding and future directions. *Journal of Geophysical Research: Space Physics*, *101*, 26,893-26,909.

Schlegel, K. (1996). Coherent backscatter from ionospheric e-region plasma irregularities. *Journal of Atmospheric and Terrestrial Physics*, *58*, 933-941.

Schrijver, C., Kauristie, K., Aylward, A., Denardini, C., Gibson, S., Glover, A., ... Vilmer, N. (2015). Understanding space weather to shield society: A global road map for 2015-2025 commissioned by cospar and ilws. *Advances in Space Research*, *55*, 2745-2807.

St.-Maurice, J., & Chau, J. (2016). A theoretical framework for the changing spectral properties of meter-scale farley-buneman waves between 90km and 125km altitudes. *Journal of Geophysical Research: Space Physics*, *121*, 10341-10366.

Sulzer, M. (1989). Recent incoherent scatter techniques. *Advanced Space Research*, *9*, (5)153-(5)162.

Thompson, A., Moran, J., & Jr., G. S. (2017). Interferometry and synthesis in radio astronomy. *Springer Open*.

Trobec, R., Slivnik, B., Bulic, P., & Robic, B. (2018). Introduction to parallel computing: From algorithms to programming on state-of-the-art platforms. *Springer*.

Uspensky, M., Williams, P., Romanov, V., Pivovarov, V., Sofko, G., & Koehler, J. (1994). Auroral radar backscatter at off-perpendicular aspect angles due to enhanced ionospheric refraction. *Journal of Geophysical Research*, *99*, 17,503-17,509.

Voors, A. (2020). Nec based antenna modeler and optimizer. *4nec2*. Retrieved 2020, June 29, from `https://www.qsl.net/4nec2/`

Appendix A

GPU Accelerated CUDA Code

A.1 CUDA code

```
#include <iostream>
#include <stdio.h>
#include <cuda_runtime.h>
#include <cufft.h>
#include <math.h>
/*
Conversion of Spread Spectrum matched filter and decimation (ssmf.c)
into CUDA code to increase analysis speed.
Draven Galeschuk - June 19/2018
*/

/*
To compile, run the following command:
nvcc -Xcompiler -fPIC -shared -o libssmf.so ssmf.cu -I/usr/local/cuda-9.1/
    include -L/usr/local/cuda-9.1/lib64 -lcufft
*/


/* Kernel - This function is run on GPU
ssmf code loops over three variables, based on the block and thread dimensions
    . Decimation rate
is assumed to be 200. Result stores real and imaginary combinations of
    measurement and code
data in a result matrix.
@meas - complex type array of measurements (np.complex64 in python)
@code - " " " "     (must be np.comlex64, may change to float 32 if .cu file
    is adjusted)
@result = a 2000 x 100 complex value matrix
*/


// Decimation and filtering function. Computes a decimation at a rate of 200
__global__ void ssmf_kernel(cufftComplex *meas, cufftComplex *code,
    cufftComplex *result, int a_shift)
{
    // Allocates shared memory arrays inside the block
    __shared__ cufftComplex smeas[100];

    // indexing Yay!
    int rg = blockIdx.z;
    int cid = threadIdx.x;
    int ti = blockIdx.y;
```

```
38    int di = threadIdx.x + blockIdx.y * 200;
39    int shift = gridDim.y * blockDim.y;
40
41    // Store a set of values to be decimated in shared memory
42    // Perform a reduction on the values held in smeas
43    // Multiply each measurement by the corresponding code value
44    // Sum all values in smeas and store in the first element
45    // summed by sequential addressing.
46    if(cid<100){
47       smeas[cid].x = meas[di+rg+100+a_shift*20000].x*code[di+100].x + meas[di+
      rg+a_shift*20000].x*code[di].x;
48       smeas[cid].y = meas[di+rg+100+a_shift*20000].y*code[di+100].x + meas[di+
      rg+a_shift*20000].y*code[di].x;
49    }
50
51    __syncthreads();
52
53    if(cid<50){
54       smeas[cid].x += smeas[cid+50].x;
55       smeas[cid].y += smeas[cid+50].y;
56    }
57
58    __syncthreads();
59
60    if(cid<25){
61       smeas[cid].x += smeas[cid+25].x;
62       smeas[cid].y += smeas[cid+25].y;
63    }
64
65    __syncthreads();
66
67    if(cid<5){
68       smeas[cid].x = smeas[cid*5].x + smeas[cid*5+1].x + smeas[cid*5+2].x +
      smeas[cid*5+3].x + smeas[cid*5+4].x;
69       smeas[cid].y = smeas[cid*5].y + smeas[cid*5+1].y + smeas[cid*5+2].y +
      smeas[cid*5+3].y + smeas[cid*5+4].y;
70    }
71
72    __syncthreads();
73
74    if(cid<1){
75       smeas[cid].x += smeas[cid*5+1].x + smeas[cid*5+2].x + smeas[cid*5+3].x +
       smeas[cid*5+4].x;
76       smeas[cid].y += smeas[cid*5+1].y + smeas[cid*5+2].y + smeas[cid*5+3].y +
       smeas[cid*5+4].y;
77    }
78
79    // Store reduced value in appropriate element of the result matrix
80    if(cid == 0){
81       result[rg*shift + ti].x = smeas[0].x;
82       result[rg*shift + ti].y = smeas[0].y;
83    }
84
85 }
```

```
86
87
88
89 // Perform conjugate multiplication between two matrices ie: arr1 * conj(arr2)
90 // This is a cross correlation if arr1 and arr2 are Fourier transforms
91 __global__ void conj_kernel(cufftComplex *arr1, cufftComplex *arr2,
       cufftComplex *res){
92
93    //Define thread variable i and intermediate float variables
94    int i = blockIdx.x * blockDim.x + threadIdx.x;
95    float xr;
96    float xi;
97    float yri;
98    float yir;
99    float x;
100   float y;
101
102   //Required if statement to exclude threads of index greater than number of
       elements (result of base 2 thread requirement)
103   if(i<200000){
104      xr = arr1[i].x * arr2[i].x; //part of new real computed from real parts
       of arr
105      xi = arr1[i].y * arr2[i].y; //part of new real computed from imag parts
       of arr
106      yri = arr1[i].x * arr2[i].y;  //part of new imag computed from arr1 real
        and arr2 imag
107      yir = arr1[i].y * arr2[i].x;  //part of new imag computed from arr1 imag
        and arr2 real
108
109      x = xr + xi;  //New real value for index
110      y = yir - yri;  //New imaginary value for index
111
112
113      //Floating point errors cause non-zero results where a zero is expected,
        ie: arr1 == arr2
114      //The following if statements attempt to correct for these errors by
       checking ratios
115      if(abs(x/xr)<0.000001 && abs(x/xi)<0.000001){
116         res[i].x += 0;
117      }else{
118         res[i].x += x;//arr1[i].x * arr2[i].x + arr1[i].y * arr2[i].y;
119      }
120
121      if(abs(y/yir)<0.000001 && abs(y/yri)<0.000001){
122         res[i].y += 0;
123      }else{
124         res[i].y += y;//arr1[i].y * arr2[i].x - arr1[i].x * arr2[i].y;
125      }
126   }
127 }
128
129
130 /* Main - Prepares CUDA memory and launches Kernel */
131
```

```
132 extern "C" {
133 void ssmf(cufftComplex *meas1, cufftComplex *meas2, cufftComplex *code,
    cufftComplex *result, size_t measlen, size_t codelen, size_t size, int avg,
     int check)
134 {
135    // Memory management for the kernnels
136    cufftHandle plan;
137
138    // define sizes for each array type
139    size_t size_m = measlen * sizeof(cufftComplex);
140    size_t size_c = codelen * sizeof(cufftComplex);
141    size_t size_out = size * sizeof(cufftComplex);
142    int n[1];
143
144    n[0] = 100;
145
146    // Build device pointers
147    cufftComplex *d_meas1, *d_meas2, *d_temp1, *d_temp2, *d_code, *d_res, *
    smeas;
148    cudaMalloc((void **) &d_meas1, size_m);
149    cudaMalloc((void **) &d_meas2, size_m);
150    cudaMalloc((void **) &d_temp1, size_out);
151    cudaMalloc((void **) &d_temp2, size_out);
152    cudaMalloc((void **) &d_code, size_c);
153    cudaMalloc((void **) &d_res, size_out);
154    cudaMalloc((void **) &smeas, 100*sizeof(cufftComplex));
155
156    // Assign device pointer values
157    cudaMemcpy(d_meas1, meas1, size_m, cudaMemcpyHostToDevice);
158    cudaMemcpy(d_meas2, meas2, size_m, cudaMemcpyHostToDevice);
159    cudaMemcpy(d_code, code, size_c, cudaMemcpyHostToDevice);
160    cudaMemcpy(d_res, result, size_out, cudaMemcpyHostToDevice);
161
162    // Threads must be base 2, 128 is fist factor greater than 100
163    // For thread block limits, see CUDA Toolkit Documentation
164    dim3 threadsPerBlock(128, 1, 1);
165    // Each block calculates an element of the result matrix
166    dim3 blocksPerGrid(1, 100, 2000);
167    cufftPlanMany(&plan, 1, n, NULL, 0, 0, NULL, 0, 0, CUFFT_C2C, 2000);
168
169    // If passed check == 0, performing single ssmf - ie: decimation and filter
170    // If passed check == 1, performing cross correlation ssmfx
171    if(check==0){
172       for(int i=0; i<avg; i++){
173          ssmf_kernel<<<blocksPerGrid, threadsPerBlock>>>(d_meas1, d_code,
    d_res, i);
174          cufftExecC2C(plan, d_res, d_res, CUFFT_FORWARD);
175       }
176    }else{
177       for(int i=0; i<avg; i++){
178          // Launch Kernel: Each block will perform 1 decimation, so 200
    threads are needed per block
179          // threadsPerBlock is the first power of 2 greater than loop
    parameter halved (since fist
```

```
180         // command does a read and sum)
181
182         ssmf_kernel<<<blocksPerGrid, threadsPerBlock>>>(d_meas1, d_code,
    d_temp1, i);
183         ssmf_kernel<<<blocksPerGrid, threadsPerBlock>>>(d_meas2, d_code,
    d_temp2, i);
184
185         //Perform fft
186         cufftExecC2C(plan, d_temp1, d_temp1, CUFFT_FORWARD);
187         cufftExecC2C(plan, d_temp2, d_temp2, CUFFT_FORWARD);
188
189         //Perform Cross Correlation
190         conj_kernel<<<512,391>>>(d_temp1, d_temp2, d_res);
191       }
192    }
193
194    // Retrieve result data
195    cudaMemcpy(result, d_res, size_out, cudaMemcpyDeviceToHost);
196
197
198    // Free up Device memory
199    cudaFree(d_meas1);
200    cudaFree(d_meas2);
201    cudaFree(d_temp1);
202    cudaFree(d_temp2);
203    cudaFree(d_code);
204    cudaFree(d_res);
205    cudaFree(smeas);
206    cufftDestroy(plan);
207 }
208 }
```

A.2 Python Wrapper

```
1  def func():
2     """
3        Python wrapper for GPU CUDA code in libssmf.so
4     """
5     dll = C.CDLL('./libssmf.so', mode=C.RTLD_GLOBAL)
6     func = dll.ssmf
7     func.argtypes = [C.POINTER(C.c_float), C.POINTER(C.c_float), C.POINTER(C.
      c_float), C.POINTER(C.c_float), C.c_size_t, C.c_size_t, C.c_size_t, C.c_int
      , C.c_int]
8     return func
9
10 __fmed = func()
11
12 def ssmfx(meas0, meas1, code, averages, nrang, fdec, codelen):
13    """
14       Retrieves signal from measurement, performs cross-correlation and
      decimates data using CUDA code
15    """
16
```

```
    # Define sizes for arrays
    measlen0 = len(meas0)
    measlen1 = len(meas1)
    nt = int(codelen/fdec)
    size = nt * nrang

    code = code.astype(np.complex64)
    result0 = np.zeros((nrang, nt), dtype=np.complex64)

    # Create pointers to convert python types to C types
    m_p0 = meas0.ctypes.data_as(C.POINTER(C.c_float))
    m_p1 = meas1.ctypes.data_as(C.POINTER(C.c_float))
    c_p = code.ctypes.data_as(C.POINTER(C.c_float))
    r_p0 = result0.ctypes.data_as(C.POINTER(C.c_float))

    # Runs ssmf.cu on data set using defined pointers
    __fmed(m_p0, m_p1, c_p, r_p0, measlen0, codelen, size, averages, 1)

    S = result0

    return(S)
```

Appendix B

Local Horizon Position Code

```
import numpy as np
import datetime

# Takes in Right Ascension, Declination of star, lat and lon of observer, and a time array. Computes elevation and azimuth angles of star for each time passed
def star_track(RA, DEC, LAT, LON, TIME):
    """
       RA, DEC, LAT, and LON are expected to be in decimal degree format.
       TIME is expected to be a datetime array.Even if only one time is passed, must be a numpy array.
    """
    t_len = len(TIME)
    JD = np.zeros(t_len)
    h_fraction = np.zeros(t_len)
    ALT = np.zeros(t_len)
    AZ = np.zeros(t_len)

    # Computes Days since beginning of the year for each time (DAY_BEG_YEAR)
    # as well as the days since J2000 and the time fraction of the current TIME
    for i in range(t_len):
        h_fraction[i] = TIME[i].hour + TIME[i].minute/60.0 + TIME[i].second/3600.0
        y = int(TIME[i].year)
        m = int(TIME[i].month)
        d = int(TIME[i].day)

        if (m==1) or (m==2):
            y = y - 1
            m = m + 12
        if (y+m/12+d/365.25>=1582+10/12+15/365.25):
            A = int(y/100)
            B = 2-A+int(A/4)
        else:
            B = 0
        if (y<0):
            C = int(365.25*y-0.75)
        else:
            C = int(365.25*y)
        D = int(30.6001*(m+1))

        JD[i] = B+C+D+d+1720994.5

    S = JD - 2451545.0
    T = S / 36525.0
```

```
    T0 = 6.697374558+2400.051336*T+0.000025862*(T**2)

    wrap = np.floor(T0/24)
    T0 = T0 - wrap*24

    GST = T0 + h_fraction*1.002737909

    wrap = np.floor(GST/24)
    GST = GST - wrap*24

    LST = GST + LON/15

    wrap = np.floor(LST/24)
    LST = LST - wrap*24

    # Convert LST from time to angular
    LST = LST*360/24

    # Compute Hour Angle, then correct so that HA is within 0 and 360 deg
    HA = LST - RA

    for i in range(t_len):
        if (HA[i] < 0):
            shift = abs(int(HA[i]/360))+1
            HA[i] = HA[i] + shift*360
        if (HA[i] > 360):
            shift = abs(int(HA[i]/360))
            HA[i] = HA[i] - shift*360

    # Compute Altitude and azimuth factor A using trig relationships
    ALT = np.arcsin(np.sin(DEC*np.pi/180)*np.sin(LAT*np.pi/180) + np.cos(DEC*np
     .pi/180)*np.cos(LAT*np.pi/180)*np.cos(HA*np.pi/180))
    A = np.arccos((np.sin(DEC*np.pi/180) - np.sin(ALT)*np.sin(LAT*np.pi/180))/(
     np.cos(ALT)*np.cos(LAT*np.pi/180)))

    # Convert to degrees
    ALT = ALT*180/np.pi
    A = A*180/np.pi

    # Determine the azimuth value from A and HA
    for i in range(t_len):
        if (np.sin(HA[i]*np.pi/180) < 0):
            AZ[i] = A[i]
        else:
            AZ[i] = 360 - A[i]

    # Shift data from 0 deg at North rotating east 360 deg to
    # 0 deg at North and rotate east (pos) and west (neg) 180 deg
    for i in range(t_len):
        AZ[i] = AZ[i] - 360
        if AZ[i] < -180:
            AZ[i] = AZ[i] + 360

    return ALT, AZ
```