

Utah State University

DigitalCommons@USU

---

All Graduate Theses and Dissertations

Graduate Studies

---

8-2020

## Indoor Source Localization of Radio Frequency Transmitters Using Blind Channel Identification Techniques

Madison L. Rose  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

---

### Recommended Citation

Rose, Madison L., "Indoor Source Localization of Radio Frequency Transmitters Using Blind Channel Identification Techniques" (2020). *All Graduate Theses and Dissertations*. 7810.

<https://digitalcommons.usu.edu/etd/7810>

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



INDOOR SOURCE LOCALIZATION OF RADIO FREQUENCY TRANSMITTERS  
USING BLIND CHANNEL IDENTIFICATION TECHNIQUES

by

Madison L. Rose

A thesis submitted in partial fulfillment  
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

---

Todd Moon, Ph.D.  
Major Professor

---

Jacob Gunther, Ph.D.  
Committee Member

---

Reyhan Baktur, Ph.D.  
Committee Member

---

Richard S. Inouye, Ph.D.  
Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY  
Logan, Utah

2020

Copyright © Madison L. Rose 2020

All Rights Reserved

## ABSTRACT

Indoor Source Localization of Radio Frequency Transmitters using Blind Channel  
Identification Techniques

by

Madison L. Rose, Master of Science

Utah State University, 2020

Major Professor: Todd Moon, Ph.D.

Department: Electrical and Computer Engineering

Source localization for audio and radio frequency (RF) signals is a well-researched topic that is increasingly important. Source localization is not trivial, especially in tricky indoor multipath environments. Multipath environments do not always yield a direct path, which many algorithms need for accurate results. The complexity of the problem is further increased when the transmitted signal, and the environment are unknown.

Of the common categories of source localization algorithms, time difference of arrival (TDOA) provides an effective group of source localization algorithms for unknown sources and environments. TDOA can be accurate, and robust to the blind setup. Several time delay estimation (TDE) methods estimate the received channel parameters; these techniques are referred to as channel identification, or blind channel identification (BCI) when the signal and/or environment is unknown. These methods exploit the cross-relation (CR) which utilizes the knowledge that each received signal is merely a convoluted version of the same transmitted signal, without the need of a statistical model. The CR algorithms provide an adaptive approach to solve for the channels iteratively.

The purpose of the research described in this thesis is to extend CR source localization algorithms that provide accurate results for indoor multipath environments without specifically knowing the environment or the transmitted source. This thesis introduces, derives, and tests multiple blind channel identification algorithms and some variations for source localization.

(394 pages)

## PUBLIC ABSTRACT

Indoor Source Localization of Radio Frequency Transmitters using Blind Channel  
Identification Techniques

Madison L. Rose

Locating transmitters is a research area that is becoming increasingly relevant as technology advances. It is especially useful for determining the location of livestock, drones, keys, phones, tablets, etc. As a result of this push for locating devices, many algorithms have been developed to determine source locations. Most source location algorithms and techniques rely on a “line of sight”, or a direct path between the source and the receivers to provide accurate results.

Indoor environments pose a challenge to locating transmitters due to the many surfaces that allow radio waves to interact (reflect, refract, and generally distort) with them. Because of the effects of the radio wave interactions, a direct path from the transmitter to the receivers may not be possible inside, increasing the difficulty. This problem is further augmented when the transmitter is transmitting an unknown signal in an unknown environment.

This research derives algorithms to address these issues. The algorithms are tested via simulations and real-world environmental testing.

## ACKNOWLEDGMENTS

I am so grateful for the tremendous help and support that I have been blessed with throughout the course of my schooling.

First, I would like to express my gratitude for my husband, Sterling, and for his endless patience. He has put his career on hold and supported me throughout the past three years so that I could finish. He has suffered through many of my late night breakdowns, helped me through all of my doubts, and allowed me to drag him across the country for this research. He is truly one in a million.

Next, for my classmate, friend, and research partner Joseph Ipson. He has spent many hours with me figuring out problems and sifting through code. He has provided many good ideas, edits, and laughs. He is one of the most intelligent and thorough people I have ever met and will go on to do great things with his life.

I am extremely grateful for Dr. Todd Moon under whose tutelage I have been able to gain more knowledge than I ever thought possible in the duration of my time working under him. His classes are some of the most interesting and excellent I have ever had. His enthusiasm has coaxed my love for learning back to life. Without his guidance and assistance in overcoming the hurdles, this project could not have come about as it did.

I am thankful for the team at the Laboratory of Telecommunication Sciences for listening to my stories about trash-pandas, and providing funding, tools, and some added knowledge to get this project off the ground.

Lastly, I would like to thank the many friends and family who have continuously supported me. I am especially grateful to the examples of my parents and grandparents for showing me that I can do anything I set my mind to. I am grateful for my friends and classmates, Colton, Russell, Zakk, Sam, Josh, Emily, Caleb, and many more for the laughter and commiseration which has kept me sane all these years.

Madison L. Rose

## CONTENTS

	Page
ABSTRACT . . . . .	iii
PUBLIC ABSTRACT . . . . .	v
ACKNOWLEDGMENTS . . . . .	vi
LIST OF TABLES . . . . .	x
LIST OF FIGURES . . . . .	xiii
ACRONYMS . . . . .	xxi
1 INTRODUCTION . . . . .	1
1.1 Thesis Outline . . . . .	2
2 LITERATURE REVIEW . . . . .	3
2.1 Source Localization . . . . .	3
2.2 Time Delay Estimation and Blind Channel Identification Algorithms . . . . .	6
2.3 Cross-Relation Method . . . . .	8
3 PROBLEM OVERVIEW . . . . .	9
3.1 Objective . . . . .	9
3.2 The Multipath Problem . . . . .	10
3.2.1 TDOA Calculations . . . . .	11
3.2.2 Time Delay Estimation Methods . . . . .	14
3.3 Simulations . . . . .	20
3.3.1 Different Simulation Parameters . . . . .	23
3.3.2 Coding Modifications . . . . .	26
3.3.3 Results Explanation . . . . .	26
3.4 Data Collection . . . . .	29
3.5 Data Sets . . . . .	30
3.5.1 HT Data Sets . . . . .	31
3.5.2 Gaussian White-Noise Data Sets . . . . .	31
4 PRE-EXISTING CROSS RELATION METHODS . . . . .	34
4.1 Pre-Existing Algorithms . . . . .	34
4.1.1 System Model . . . . .	34
4.1.2 Cross Relation . . . . .	35
4.1.3 A Least-Squares Approach According to Xu et al. . . . .	36
4.1.4 Adaptive Eigenvalue Decomposition . . . . .	37
4.1.5 Pre-Existing Algorithms Implementations, Advantages, and Disadvantages . . . . .	38



4.2	Modified Algorithms . . . . .	45
4.2.1	Modified System Model . . . . .	46
4.2.2	Complex Cross Relation . . . . .	49
4.2.3	Modified Adaptive Eigenvalue Decomposition . . . . .	52
4.2.4	Modified AED with Sparsity . . . . .	53
4.2.5	Modified Algorithms Implementations, Advantages, Disadvantages . . . . .	55
4.3	Algorithm Testing and Results . . . . .	59
4.3.1	Simulation . . . . .	60
5	THE ADAPTIVE CROSS-CHANNEL WITH SPARSE SHIFT-SUPPRESSION (AXIS) METHOD . . . . .	74
5.1	AXIS Derivation . . . . .	74
5.1.1	Algorithm Advantages, Implementations, and Modifications . . . . .	75
5.2	Variations . . . . .	79
5.2.1	Varied AXIS . . . . .	80
5.2.2	Dual Fixed-Peak AXIS . . . . .	83
5.2.3	Non-Sparse AXIS . . . . .	86
5.2.4	Single Constraint AXIS (SC) . . . . .	88
5.3	Data Verification . . . . .	95
5.3.1	Simulation . . . . .	95
5.4	Algorithm Comparison . . . . .	116
6	THE “BRUTE FORCE” METHOD . . . . .	117
6.1	Brute Force Derivation . . . . .	117
6.1.1	Algorithm Advantages, Disadvantages and Implementation . . . . .	118
6.2	Variations . . . . .	119
6.2.1	Adaptive Brute Force . . . . .	120
6.2.2	Varied Adaptive Brute Force . . . . .	124
6.2.3	Dual Fixed-Peak Brute Force . . . . .	126
6.3	Data Verification . . . . .	134
6.3.1	Simulation . . . . .	134
6.4	Algorithm Comparison . . . . .	148
7	RESULTS . . . . .	149
7.1	Comparison of Algorithms . . . . .	149
7.1.1	Algorithm Complexity . . . . .	149
7.1.2	Simulation Results . . . . .	151
7.1.3	Results with Real Observed Data . . . . .	154
7.2	Final Observations . . . . .	157
8	CONCLUSION . . . . .	159
8.1	Research . . . . .	159
8.2	Future Work . . . . .	160
	REFERENCES . . . . .	162

APPENDICES	167
A Math Derivations and Notation	168
A.1 Notation	168
A.2 Derivations	172
B Code Listings	192
B.1 Modifications	192
B.2 The Least Squares (SVD) Approach According to Xu et al.	195
B.3 The Adaptive Eigenvalue Decomposition	196
B.4 The Modified Adaptive Eigenvalue Decomposition	198
B.5 The Modified Adaptive Eigenvalue Decomposition with Sparsity	200
B.6 The Adaptive Cross-Channel Identification with Sparse Shift-Suppression	202
B.7 The Varied Adaptive Cross-Channel Identification with Sparse Shift-Suppression	204
B.8 The Dual Fixed-Peak Adaptive Cross-Channel Identification with Sparse Shift-Suppression	207
B.9 The Single Constraint Method	210
B.10 The Varied Single Constraint Method	212
B.11 The Dual Fixed-Peak Single Constraint Method	215
B.12 The Brute Force Method	218
B.13 The Varied Brute Force Method	220
B.14 The Dual Fixed-Peak Brute Force Method	222
B.15 The Cross Correlation Function	225
B.16 The Overhead File to Call the Algorithms	226
C Oversized Figures	237
D Further Results	260
D.1 Simulation	260

## LIST OF TABLES

Table	Page
3.1 Noise amount with corresponding SNR in dB. . . . .	24
3.2 Various model order estimates used for algorithm simulation . . . . .	25
4.1 Numerical results for pre-existing and modified algorithms in the “Ideal” case.	65
4.2 Numerical results for pre-existing and modified algorithms in the narrowband case. . . . .	66
4.3 Numerical results for pre-existing and modified algorithms in the noisy case.	67
4.4 Numerical results for pre-existing and modified algorithms in the smaller data size case. . . . .	68
4.5 Numerical results for pre-existing and modified algorithms in the wrong model order, $L/2$ , case. . . . .	69
4.6 Numerical results for pre-existing and modified algorithms in the wrong model order, $L - 10$ , case. . . . .	70
4.7 Numerical results for pre-existing and modified algorithms in the wrong model order, $2L$ , case. . . . .	71
4.8 Numerical results for pre-existing and modified algorithms in the wrong model order, $L + 10$ , case. . . . .	72
4.9 Numerical results for pre-existing and modified algorithms in the normalized random start case. . . . .	73
5.1 Numerical results for the AXIS algorithms in the “ideal” case for set IR-1. .	98
5.2 Numerical results for the AXIS algorithms in the “ideal” case for set IR-2. .	99
5.3 Numerical results for the AXIS algorithms in the narrowband case for set IR-1.	100
5.4 Numerical results for the AXIS algorithms in the narrowband case for set IR-2.	101
5.5 Numerical results for the AXIS algorithms in the noisy case for set IR-1. . .	102
5.6 Numerical results for the AXIS algorithms in the noisy case for set IR-2. . .	103

5.7	Numerical results for the AXIS algorithms in the smaller data size case for set IR-1. . . . .	104
5.8	Numerical results for the AXIS algorithms in the smaller data size case for set IR-2. . . . .	105
5.9	Numerical results of the AXIS algorithms in the wrong model order, $L/2$ , case for set IR-1. . . . .	106
5.10	Numerical results of the AXIS algorithms in the wrong model order, $L/2$ , case for set IR-2. . . . .	107
5.11	Numerical results of the AXIS algorithms in the wrong model order, $L - 10$ , case for set IR-1. . . . .	108
5.12	Numerical results of the AXIS algorithms in the wrong model order, $L - 10$ , case for set IR-2. . . . .	109
5.13	Numerical results of the AXIS algorithms in the wrong model order, $2L$ , case for set IR-1. . . . .	110
5.14	Numerical results of the AXIS algorithms in the wrong model order, $2L$ , case for set IR-2. . . . .	111
5.15	Numerical results of the AXIS algorithms in the wrong model order, $L + 10$ , case for set IR-1. . . . .	112
5.16	Numerical results of the AXIS algorithms in the wrong model order, $L + 10$ , case for set IR-2. . . . .	113
5.17	Numerical results of the AXIS algorithms in the normalized random start case for set IR-1. . . . .	114
5.18	Numerical results of the AXIS algorithms in the normalized random start case for set IR-2. . . . .	115
6.1	Numerical results for the BF algorithms in the “ideal” case. . . . .	139
6.2	Numerical results for the BF algorithms in the narrowband case. . . . .	140
6.3	Numerical results for the BF algorithms in the noisy case. . . . .	141
6.4	Numerical results for the BF algorithms in the smaller data size case. . . .	142
6.5	Numerical results for the BF algorithms in the wrong model order, $L/2$ , case. . .	143
6.6	Numerical results for the BF algorithms in the wrong model order, $L - 10$ , case. . . . .	144

6.7	Numerical results for the BF algorithms in the wrong model order, $2L$ , case.	145
6.8	Numerical results for the BF algorithms in the wrong model order, $L + 10$ , case. . . . .	146
6.9	Numerical results for the BF algorithms in the normalized random start case.	147
7.1	Comparison of the computational complexity of the algorithms. . . . .	150
7.2	Table of various variables used in simulation. . . . .	151
7.3	The distance in meters between antennas (receiver or transmitters). . . . .	156

## LIST OF FIGURES

Figure	Page
3.1 An example of the multipath problem, where each color of line denotes a different path from the transmitter to a receiver, and the black lines denote walls. The solid line paths are possible reflective paths, and the dashed lines are direct paths. . . . .	11
3.2 Simple pictorial TDOA method. . . . .	13
3.3 The results of CC TDOA using the LUT method for 250,000 data samples, and two receivers. . . . .	14
3.4 An example of the normalized delay between two channels, $\mathbf{h}_1$ and $\mathbf{h}_2$ . . . .	15
3.5 The environment used to simulate impulse responses including the location of the transmitter, and two of the potential receivers which were used for all testing. The black lines represent the “walls”, or rather boundaries, on which reflections and refractions may occur. . . . .	20
3.6 Ideal impulse response explanation . . . . .	21
3.7 Simulated impulse response explanation. . . . .	22
3.8 Power spectral density comparison for (a) wideband, versus (b) narrowband data at 20 MHz. . . . .	24
3.9 SANT building layout with the basic outline of the data collection. The circles either denote the location of a receiver or the location of the transmitter, whereas the lines depict the general layout of the major walls on the second floor of the building. Figure (a) shows the locations of the receivers in the ceiling and Figure (b) shows the various locations of the data transmitter used. . . . .	29
3.10 Transmitter locations for the data sets. . . . .	30
3.11 The respective locations of each channel . . . . .	31
3.12 Comparison of the normalized frequency bandwidths at each receiver for both data sets for the corner data set. . . . .	32
4.1 System Model . . . . .	35

4.2	Results of the SVD algorithm with 30,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . .	39
4.3	Results of the SVD algorithm with 300 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . .	40
4.4	Results of the SVD algorithm with 30,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . .	41
4.5	Results of the SVD algorithm with 30,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	42
4.6	Results of the SVD algorithm with 30,000 data samples, with SNR = 20 dB at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . .	43
4.7	Results of the SVD algorithm with 30,000 data samples, with SNR = 60 dB for wideband at 200 MHz sampling rate with incorrect channel orders, $L$ (a) $L/2$ , (b) $L - 10$ , (c) $L * 2$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0.	44
4.8	Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	45
4.9	Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0 using a normalized random start. . . . .	46
4.10	Results of the AED algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . .	47
4.11	Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	48
4.12	Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	49
4.13	Results of the AED algorithm with 250,000 data samples, with SNR = 20 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	50

4.14	Results of the AED algorithm with 30,000 data samples, with SNR = 60 dB for wideband at 200 MHz sampling rate with incorrect channel orders, $L$ (a) $L/2$ , (b) $L - 10$ , (c) $L * 2$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0.	51
4.15	Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	55
4.16	Results of the ModAEDS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	56
4.17	Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.	57
4.18	Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	58
4.19	Results of the modAED algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	59
4.20	Results of the ModAED algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	60
4.21	Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect estimates of transfer function length, $L$ , (a) $L/2$ , (b) $L - 10$ , (c) $L * 2$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0.	61
4.22	Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0, and normalized random initial estimates after 100 iterations.	62
4.23	Results of the ModAED and ModAEDS algorithms with 2,500 data samples, with SNR = 60 dB, for wideband simulation at 100 MHz sampling rate for impulse response set IR-1, seed 0.	63



4.24	Results of the ModAED and ModAEDS algorithms with 250,000 data samples, with SNR = 20 dB, for wideband simulation at 150 MHz sampling rate for impulse response set IR-1, seed 0. . . . .	64
4.25	Results of the ModAED and ModAEDS algorithms with 250,000 data samples, with SNR = 60 dB, for wideband simulation at 100 MHz sampling rate for incorrect estimates of model order, $L = L/2$ , for impulse response set IR-1, seed 0. . . . .	64
5.1	Results of the AXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	76
5.2	Results of the AXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	77
5.3	Results of the AXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	78
5.4	Results of the AXIS algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	79
5.5	Results of the AXIS algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	80
5.6	Results of the AXIS algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a) $L/2$ , (b) $L - 10$ , (c) $2L$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0. . . . .	81
5.7	Results of the VAXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	83
5.8	Results of the DFPAXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	87
5.9	Results of the NSAXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response IR-1 seed 0. . . . .	88

5.10	Results of the SC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . .	90
5.11	Results of the VSC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	92
5.12	Results of the DFPSC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.	93
5.13	Results of the Non-Sparse SC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	94
6.1	Results of the non-adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	119
6.2	Results of the non-adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	120
6.3	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	121
6.4	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	122
6.5	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	123
6.6	Results of the Adaptive BF algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	124

6.7	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	125
6.8	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a) $L/2$ , (b) $L - 10$ , (c) $2L$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0. . . . .	126
6.9	Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1 with a normalized random start. . . . .	127
6.10	Results of the VBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	128
6.11	Results of the VBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	129
6.12	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	130
6.13	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0. . . . .	131
6.14	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	132
6.15	Results of the DFPBF algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	133
6.16	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0. . . . .	134
6.17	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a) $L/2$ , (b) $L - 10$ , (c) $2L$ , and (d) $L + 10$ , for impulse response set IR-1, seed 0. . . . .	135

6.18	Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0 with a normalized random start. . . . .	136
7.1	The approximate locations of the antennas used during real-world data testing. The large blue circles correspond to the location of the receivers, and the smaller multi-colored circles the location of the transmitters. Note that this figure is not to scale. . . . .	155
7.2	Channel estimates for the Gaussian Data sets with the most “reasonable” results, using a model order of $L = 32$ . These plots show $\ln(\mathbf{h}_i)$ , for $i = 2, 3$ , as the values received by the receiver are very small. . . . .	156
7.3	Channel estimates for the HT Data sets, using a model order of $L = 32$ . These plots show $\ln(\mathbf{h}_i)$ , for $i = 2, 3$ , to match the results shown above in Figure 7.2. . . . .	157
C.1	Comparison of the 20 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	238
C.2	Comparison of the 40 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	239
C.3	Comparison of the 60 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	240
C.4	Comparison of the 80 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	241
C.5	Comparison of the 100 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	242
C.6	Comparison of the 120 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	243
C.7	Comparison of the 140 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	244
C.8	Comparison of the 150 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	245
C.9	Comparison of the 160 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	246
C.10	Comparison of the 180 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	247

C.11 Comparison of the 200 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000 . . . . .	248
C.12 Comparison of the 20 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	249
C.13 Comparison of the 40 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	250
C.14 Comparison of the 60 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	251
C.15 Comparison of the 80 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	252
C.16 Comparison of the 100 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	253
C.17 Comparison of the 120 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	254
C.18 Comparison of the 140 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	255
C.19 Comparison of the 150 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	256
C.20 Comparison of the 160 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	257
C.21 Comparison of the 180 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	258
C.22 Comparison of the 200 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000 . . . . .	259

## ACRONYMS

ADC	analog to digital converter
AED	adaptive eigenvalue decomposition
AEDS	adaptive eigenvalue decomposition with sparsity
AOA	angle of arrival (sometimes referred to as DOA)
AXIS	adaptive cross-channel identification with sparse shift-suppression
BCI/BCE	blind channel identification/estimation
BF	“brute” force
BSI/BSE	blind system identification/estimation
CC	cross-correlation
CCC	classical CC
CI	channel identification
CR	cross-relation
CRLB	Cramer Rao lower bound
DFPAXIS	dual fixed-peak AXIS
DFPBF	dual fixed-peak BF
DFPSC	dual fixed-peak single constraint
DOA	direction of arrival (sometimes referred to as AOA)
FDOA	frequency difference of arrival
FFT	fast Fourier transform
GCC	generalized cross-correlation
HT	handheld transceiver
IFFT	inverse FFT
LOB	line of bearing
LOS	line of sight
LPF	low-pass filter
LUT	look-up-table
ML	maximum likelihood

MIMO	multiple input multiple output
MUSIC	multiple signal classification
NaN	not a number
NLOS	no line of sight
PHAT	phase transform
RF	radio frequency
RSSI	received signal strength indicators
SC	single constraint
SCOT	smoothed coherence transform
SDR	software defined radio
SIMO	single input multiple output
SNR	signal to noise ratio
SVD	singular value decomposition
TDE	time delay estimation
TDOA	time difference of arrival
TOA	time of arrival
USRP	universal software radio peripheral
USU	Utah State University
VAXIS	varied AXIS
VBF	varied BF
VSC	varied SC

## CHAPTER 1

### INTRODUCTION

Identifying the physical location of a transmitter is a common problem generally referred to as source localization or geolocation (depending on the point of reference) [1–4]. Although source localization is a well known and researched problem, it is challenging in the applications involving unknown signals. The difficulty is further increased for applications specifically in high multipath environments, as a direct “line-of-sight” (LOS) is not always received [1, 2, 4–7].

Source localization has become a very useful and important task. Source localization is used to locate wireless phones [8], locate livestock using RFID [9], locate drones [10], etc. Source localization in multipath environments is also useful in various security situations, such as determining if and where a signal is being transmitted from in secure facilities, office buildings, residences, banks, museums, and other places of interest.

The most common methods for source localization are: time of arrival (TOA), time difference of arrival (TDOA), direction of arrival (DOA), and received signal strength (RSS). These different algorithms provide different information, either distance or direction [1, 2, 4, 11]. But as there are costs and benefits to each algorithm, the challenge arises with respect to balancing the issues, and obtaining the most accurate algorithm for the environment.

Although direction information is useful, the method for acquiring the information is expensive, and relies heavily on LOS. Of the four common methods, distance can be determined using TOA, RSS and TDOA. For a blind situation (unknown source, and unknown environment), TDOA would be the only viable technique, as TOA needs time synchronization between the source and the receivers, and RSS is not as accurate, although some algorithms have been developed to combat these issues [1, 2, 4, 11]. There are various TDOA algorithms. Time delay estimation (TDE) in some literature is synonymous with TDOA, however, it is most commonly described as a way to measure TDOA [12–14].



This thesis details research involving TDOA algorithms via TDE for multipath environment source localization. Multiple algorithms are proposed, derived and simulated with unknown transmitted data. The promising algorithms are then tested with real world data collected via software defined radios (SDRs).

## 1.1 Thesis Outline

The rest of this thesis will delve further into source localization techniques and algorithms, further explain the problem, offer possible solutions, and provide and discuss results.

Chapter 2 offers a more detailed explanation on the various source localization techniques, and details prior algorithms and techniques that have been previously done relating to this problem.

Chapter 3 discusses the objectives of the research. Simulation situations, assumptions and settings are discussed and how data was collected is presented.

Chapters 4 – 6 address the algorithms explored in this research via derivation, simulation, and some real-world testing. Chapter 4 explores previous cross-relation algorithms via derivation and simulation for a basis on which to compare the other algorithms. Chapter 5 introduces the main algorithm, and its variations. Chapter 6 investigates a different approach.

Chapter 7 compares all algorithms, and their results. The effects of various parameters on each algorithm is also discussed in detail. Final observations are also examined.

Chapter 8 concludes the thesis.

Complete mathematical derivations, code listings, figures, and results are located in the appendices at the end of this thesis.

## CHAPTER 2

### LITERATURE REVIEW

Source localization methods, time delay estimation techniques and blind channel identification algorithms are well researched. A brief review of the existing literature is presented in this chapter.

#### 2.1 Source Localization

The term source localization is in a way self-explanatory: finding and identifying the position of a source. A source could refer to an audio speaker, radio-frequency transmitter, or any other signal emitting object. As mentioned in the introduction, there are four common techniques of source localization: time of arrival (TOA), time difference of arrival (TDOA), direction of arrival (DOA) or angle of arrival (AOA), and received signal strength (RSS) [1,2,4]. Frequency difference of arrival (FDOA) is also common [2]. These algorithm categories provide either distance, or direction data. TOA, TDOA, FDOA, and RSS provide distance information, and DOA provides a bearing or direction [1, 2, 4, 11]. Each of the algorithms have varying degrees of accuracy, and complexity.

Although many algorithms have been developed for source localization, the act of finding the source location is not insignificant due to the nonlinear relationship between the data measurements and the source [1,2,4,5,7,12,13,15–21].

#### TOA, RSS, DOA, FDOA

Time of Arrival (TOA) and Received Signal Strength<sup>1</sup> (RSS) both provide an estimate of distance from the receiver to the transmitter. As no direction is received, a circle with radius equivalent to the estimated distance centered at the receiver provides 360° of possible positions for the transmitter. This estimated transmitter position can be narrowed down by

---

<sup>1</sup>Sometimes referred to as Received Signal Strength Indicators (RSSI).

using multiple receiver nodes for triangulation. At least three measurements are required for localization. The area where the distance circles overlap is the best estimate of the transmitter. These methods assume that position of the multiple receivers are all known [1, 2, 4, 11].

TOA uses the time a signal propagates to estimate distance. As signals propagate through space at the speed of light, the propagation can be calculated if the time the signal left the transmitter is known. Thus, in order for the triangulation of the transmitter location to occur, a synchronization of the transmitter and the receiver clocks is required. Even a small timing error can cause large errors in the distance calculation. This also means that the transmitted signal either must be known by the receivers, or the signal must possess a time stamp. This makes the method very accurate, however, for unknown data, this method is not viable [1, 2, 4, 11].

RSS uses the strength or power of the received signal to determine the location of the transmitter. The power received is assumed to follow a decay model determined by a loss coefficient, allowing for the distance to be estimated assuming the environment the signal travels through is known. This method is simple, allowing for inexpensive hardware, and unlike TOA, no synchronization is required between the transmitter and the receivers. Unfortunately, this method has lower accuracy than the other methods, and typically assumes that the loss coefficient,  $\theta$ , is known (although there has been research in this area to combat this uncertainty). If the loss coefficient is unknown, and accuracy is important, this method may require many more measurements to achieve accuracy [1, 2, 4, 11].

Direction of Arrival<sup>2</sup> (DOA) provides an estimate of the direction or bearing of the signal being transmitted. This is done by implementing a phased antenna array at each receiver. This antenna array is then steered such that the main lobe of the antenna is pointed in the direction of the peak energy received from the transmitter. A line of bearing (LOB) is drawn from the receiver in the direction of the transmitter. At least two receivers are needed for triangulation. Where the two LOBs cross is the estimated position of the transmitter. The position of the receivers is assumed to be known for this method [1, 2, 4, 11].

---

<sup>2</sup>Sometimes referred to as Angle of Arrival, or AOA.

DOA only requires two receivers for triangulation, less than the three needed for the TOA and RSS methods. And unlike TOA, no time synchronization is required. However, each receiver needs a complex front-end system for the precise angle calculations. DOA relies heavily on LOS, which in high multipath environments this method is not reliable. Although some methods have been taken to address the multipath issue using a uniform circular array [1, 2, 4, 5, 11].

Frequency difference of arrival (FDOA), commonly referred to as differential Doppler in the literature, yields a frequency measurement that can be used for either range or for velocity. As the name implies, FDOA looks at the difference in frequency between receivers. For FDOA to be possible, either the receivers or the transmitter must be in motion. Due to the movement, a Doppler shift should occur between the receivers, returning a change in frequency. Two receivers are needed for one FDOA measurement. The differential Doppler shift between the two receivers must be greater than the measurement error for accurate processing. The receivers also need to be able to measure very small frequency accurately such that FDOA can be detected over the noise. FDOA measurements are used to compute surfaces on which the transmitter must lie. Where two surfaces intersect is the estimated location of the transmitter, thus at least three receivers are needed for source localization [2, 17–24].

FDOA is most commonly used in tandem with TDOA to improve accuracy for moving transmitters or moving receivers [17–23]. In these instances, only two receivers are needed (although more may be helpful), producing one TDOA measurement and one FDOA measurement. Where these two measurements (the hyperbola and the surface) intersect is the transmitter location estimate. According to a few sources, TDOA serves as a bearing in this case, whereas FDOA acts as the range estimate [2, 17, 22]. Most of the other sources refer to TDOA as the position estimate, and FDOA the velocity estimate for a moving transmitter. [18–20].

## **TDOA**

Time Difference of Arrival (TDOA) is a variation on TOA which looks at the time a

signal arrives at each receiver to measure the difference between times. TDOA uses hyperbolae to estimate the position of the transmitter. The location of the receivers corresponds to location of the foci of the hyperbolae. Thus, two receivers are needed for one hyperbola. Two (noise-free) TDOA measurements are needed for source localization, and hence at least three receivers are needed, although up to four hyperbolae are possible. Where the hyperbolae intersect is the estimated location of the transmitter. With noisy measurements, more TDOA measurements are needed for better accuracy [1–4, 11, 17, 19, 22].

To implement TDOA, all the receiver clocks must be synchronized. This differs from TOA in that the receiver clocks need not be synchronized with the transmitter, and no timestamp is needed. This means that the method of TDOA can be used with an unknown signal. TDOA also has the advantage of being more robust with respect to multipath error [1, 2, 4, 21].

## 2.2 Time Delay Estimation and Blind Channel Identification Algorithms

Time Delay Estimation (TDE) is a technique that estimates the delay between receivers. One of the uses of TDE is for TDOA source localization [12–14]. There are various TDE algorithms that exist, such as

- Generalized Cross-Correlation (GCC) [12–16, 25–28]
- Neural Networks (NN) or Machine Learning [29–33]
- Maximum Likelihood (ML) [12, 16, 26, 34–37],
- Cross-Relation (CR) [12, 15, 25, 26, 28, 34, 35, 38–40].

Of these algorithms, GCC, ML and CR are historically the most common [12, 26], although some recent research has looked into developing machine learning or neural networks for TDE [29–33]. GCC computes the frequency spectrum of the signal channel, while the CR algorithm employs channel identification techniques, and ML can do both (depending on application) [12].

The maximum likelihood method is a statistical optimal parameter estimation technique, employing a probabilistic model of the observed data [12, 26, 34–36]. This method can be optimal if signals have infinite length [26], however, there is no closed form solution, and a probabilistic model of the data is needed [12, 34]. The general cross-correlation technique determines the time-lag that maximizes the weighted cross-correlation between channels. GCC is widely used, but in very reverberant environments, the algorithm deteriorates [12, 14, 15, 26, 28]. Thus, for unknown signals in a multipath environment, the cross-relation channel identification technique is favored [13, 15, 25].

Channel identification (CI) is a fundamental method of estimating the system a signal is transmitted through using the data transmitted and received, and previous statistical, and temporal environmental knowledge. Blind channel identification (BCI) differs from the standard CI in that the transmitted signal is unknown [34–37, 39, 39, 40]. BCI is no trivial task, as the system and the transmitted signal are unknown for every received signal. Mathematically, this means that for every output signal received, there are multiple unknown variables.

There are a couple of techniques used to combat the unknowns [34, 39]:

1. Exploiting the statistical and temporal characteristics of the environment the signal is transmitted and received through.
2. Utilizing the knowledge that each received signal is merely a convoluted version of the same transmitted signal(s).

Using this knowledge, adaptive (self-recovering) blind channel identification algorithms have been implemented, and perform fairly accurately [15, 25, 35, 38].

### 2.3 Cross-Relation Method

Assuming an unknown transmitted signal,  $s(n)$ , in a multipath environment, detected by multiple receivers, the single input, multiple output (SIMO) system can be modelled as:

$$\begin{cases} x_1(n) = s(n) * h_1(n) + w_1(n) \\ \vdots \\ x_M(n) = s(n) * h_M(n) + w_M(n). \end{cases} \quad (2.1)$$

In the above equation,  $*$  denotes linear convolution,  $s(n)$  the signal transmitted,  $h_i(n)$  denotes the individual channel impulse responses mapping from the source to the  $i^{\text{th}}$  receiver,  $x_i(n)$  the value received from the transmitted signal at the  $i^{\text{th}}$  receiver, and  $w(n) = [w_1(n) w_2(n) \dots w_M(n)]$  is additive white noise, and  $n$  denotes the  $n^{\text{th}}$  sample.  $M$  denotes the number of channels.

Cross-relation for a pair of receivers can be seen by neglecting noise [12, 13, 15, 25, 28, 34, 35, 38–40]. The previously described signal model for a pair of signals can be expressed individually as  $x_i(n) = s(n) * h_i(n)$ . Convolution  $x_i(n)$  with  $h_j(n)$ , and  $x_j(n)$  with  $h_i(n)$ , where  $i \neq j$ , gives

$$h_j(n) * x_i(n) - h_i(n) * x_j(n) = 0. \quad (2.2)$$

The CR technique uses (2.2) to iteratively (or adaptively) solve for the impulse responses,  $h_i$  and  $h_j$  in a constrained or unconstrained update equation, depending on the algorithm.

CR without further constraints such as sparsity and subspace to limit the search area, is ill-posed. The results of the unconstrained CR technique are inaccurate [28, 40–43]. Overestimating the channel order of  $h_i$  can cause CR to suffer. CR can also show bias with finite samples [34].

This thesis explores some existing CR algorithms and introduces a few new CR algorithms for multipath TDOA source localization.

## CHAPTER 3

### PROBLEM OVERVIEW

This chapter reviews the problem researched in this thesis, and how the solution is approached. It details the assumptions used, and lists characteristics of the simulation and testing environments. Data collection methods are also explained.

#### 3.1 Objective

The main purpose of this research is to locate a transmitter in an unknown location, transmitting unknown data in a multipath environment using multiple software defined radios (SDRs). The transmitter is assumed to be stationary, and within the outer walls of the testing environment. The center frequency is assumed to be known, and the system is assumed to be single input multiple output (SIMO).

In these conditions, line of sight (LOS) is not guaranteed, due for example to multiple reflections off the walls. As direction of arrival (DOA) relies heavily on LOS, it is not a viable option. Time synchronization between the receivers and the transmitter may not be possible either, as the purpose of the research is for unknown signals, thus time of arrival (TOA) is also not a viable option. Received signal strength (RSS), although cheaper, is typically not as accurate. Consequently, time difference of arrival (TDOA) techniques propose the most logical source localization technique. For these reasons, TDOA results are not compared to results of other source localization techniques as that is not the purpose of this research.

In order to estimate the position of the transmitter, time delay estimation (TDE) will be used. As the situation is truly blind—unknown source, unknown environment, and unknown location—blind channel estimation<sup>1</sup> (BCE) techniques have shown promise [25, 34–42]. Of the BCE techniques, the cross-relation (CR) technique is favored in multipath environments as it assumes a reverberation signal model, over the generalized cross-correlation (GCC)

---

<sup>1</sup>Sometimes referred to as blind channel identification (BCI) or blind system identification/estimation (BSI/BSE).



technique due to deterioration [12–15, 26, 28]. Thus, all the CR techniques derived in this research are compared to the GCC as a baseline.

### 3.2 The Multipath Problem

Source localization techniques tend to rely on LOS, or rather a direct path for accurate results [1, 6]. Ideally, in free space, when a signal is transmitted, a single peak is received at each transmitter. There should be a relative time delay between these peaks which corresponds to the distance between the receivers. In enclosed environments, reflections, refractions and general distortion can occur when the transmitted waves interact with walls, furniture and other obstructions. Due to this interaction, multiple attenuated and delayed versions of the signal can reach each transmitter. Thus, the resulting signal should contain the direct path plus multiple attenuated and delayed paths. This is referred to as the multipath problem [16].

As mentioned previously, most source localization techniques rely on the assumption that a direct path exists and is received. As can be seen in Figure 3.1, a direct path may or may not be received at the receiver. The case that a direct path is not received is referred to as no line of sight (NLOS) conditions. In NLOS conditions, the received signal must cover more distance to reach the receiver, resulting in larger estimation errors [1, 6, 7, 16]. When NLOS paths are received, it is recommended by some sources, to identify and remove the NLOS measurements from the localization process. This is because NLOS paths can cause issues, making them undesirable [1, 2, 16]. Identifying NLOS conditions presents unique challenges which are too involved to be included in this research [1, 2, 7].

Ideally, the first peak received is the largest peak, as it signifies the shortest path the transmitted signal took from transmitter to receiver, i.e. the direct path. However due to the wave interactions, the signal can either be amplified or attenuated in its path to, and at the receiver. This means that the first peak may not be the largest peak. First peaks are challenging to distinguish as they could be distortions from side lobes [12, 16].

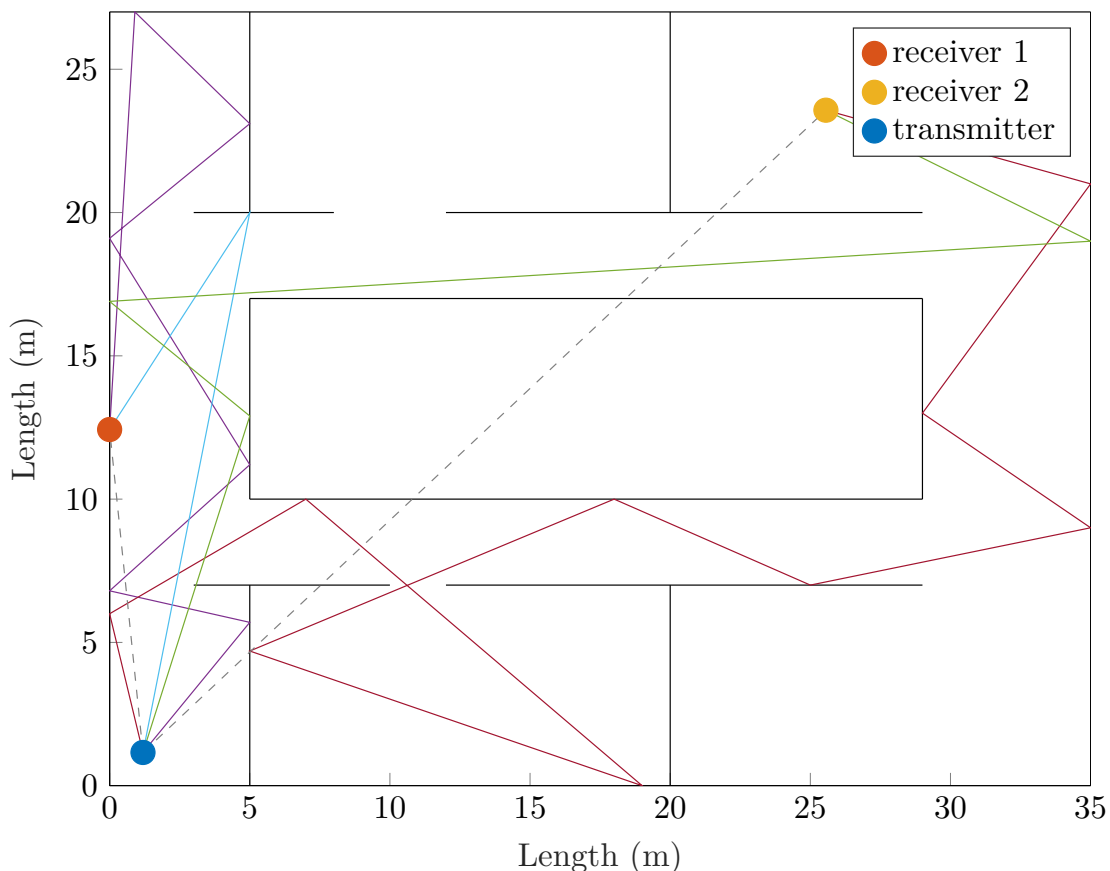


Fig. 3.1: An example of the multipath problem, where each color of line denotes a different path from the transmitter to a receiver, and the black lines denote walls. The solid line paths are possible reflective paths, and the dashed lines are direct paths.

### 3.2.1 TDOA Calculations

TDOA uses intersecting hyperbolae to show the possible locations of the transmitter, with the receiver positions corresponding to the foci points. TDOA measurements present a challenge due to high nonlinearity. Nonlinearity can occur not only when computing the hyperbolae, but also with respect to the relationship between the source location and the data. This results in a nonconvex maximum likelihood (ML) function. There have been multiple approaches proposed for solving the TDOA equations.

To solve the ML function, the equations can be modified, and a ML method can be used. In ideal situations, this can prove optimal. However, it does not have a general closed form solution, can be computationally expensive, and can experience convergence

difficulties. Closed form solutions have been proposed, but are suboptimal, and are limited by the amount of receivers, location of the receivers, and/or the signal to noise ratio (SNR) [2, 18, 19, 44, 45].

One common way is to linearize the equations using a Taylor series approximation. This approach to solving the TDOA equations is popular as it is simple, involving an iterative calculation. However, an initial estimate for the transmitter location is needed for calculation, and there is no guarantee of convergence, especially if little to nothing is known about the transmitter [2, 18, 19, 44, 45].

The third way typically used is a geometric way, using hyperbolic asymptotes. This way is not as computationally intensive, using simplified linear equations corresponding to the hyperbolic asymptotes. Unfortunately this way tends to be less accurate [2, 17, 44].

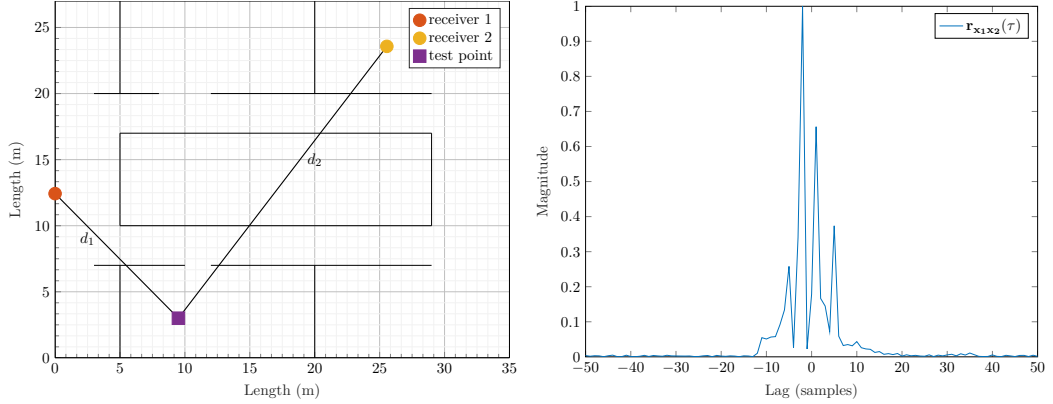
A less common way, referred to as a grid-search method or look-up-table (LUT) method, is used for a pictorial representation of the likelihood of the transmitter location. Although computationally expensive, there is no need to deal with the nonlinear TDOA computations. The LUT method computes a type of heat map which shows a rough probability likelihood of the location of the transmitter. This method is used in this research for the probability aspect.

Prior to computations, the area over which the locator may be present is divided into a grid or array pattern, as shown in Figure 3.2. The method can be simply explained in four steps. Cycling through the squares on the grid, the following steps must be taken:

1. Calculate the distance from the test point square to the receivers as shown in Figure 3.2(a). This is done using the Euclidean distance

$$\begin{aligned} d_1 &= \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2}, \\ d_2 &= \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2}, \end{aligned} \tag{3.1}$$

where  $(x_t, y_t)$  denote the coordinates for the test point, and  $(x_1, y_1)$  and  $(x_2, y_2)$  the coordinates for receivers 1 and 2 respectively.



(a) Simulation environment with two receivers and test point, with an example grid overlaid. (b) An example of a time likelihood curve via the magnitude plot for the CC for channels 1 and 2.

Fig. 3.2: Simple pictorial TDOA method.

- Determine the relative difference from the distance measurements in step 1,

$$\Delta d_{1,2} = d_1 - d_2. \quad (3.2)$$

- Calculate the time difference using the distance speed time formula,

$$\Delta t = \Delta d_{1,2} \cdot \frac{F_s}{c}, \quad (3.3)$$

where  $F_s$  is the sample rate associated with the impulse response, and  $c$  is the speed of light, approximately  $3 \times 10^8$  m/s. Note that  $\Delta t$  corresponds to values on the x-axis for a time likelihood plot, such as the CC magnitude plot, as shown in Figure (3.2(b)).

- Each square in the grid has a corresponding heatmap value that can be found from the time likelihood plot. The heatmap value is the CC magnitude that corresponds the delay found in step 3.

The values are quantized and then color-coded. This method shows the most likely place for the transmitter, as well a level of uncertainty. The results of the example situation are shown in Figure 3.3, where a brighter color indicates a higher likelihood of the transmitter being at that location.

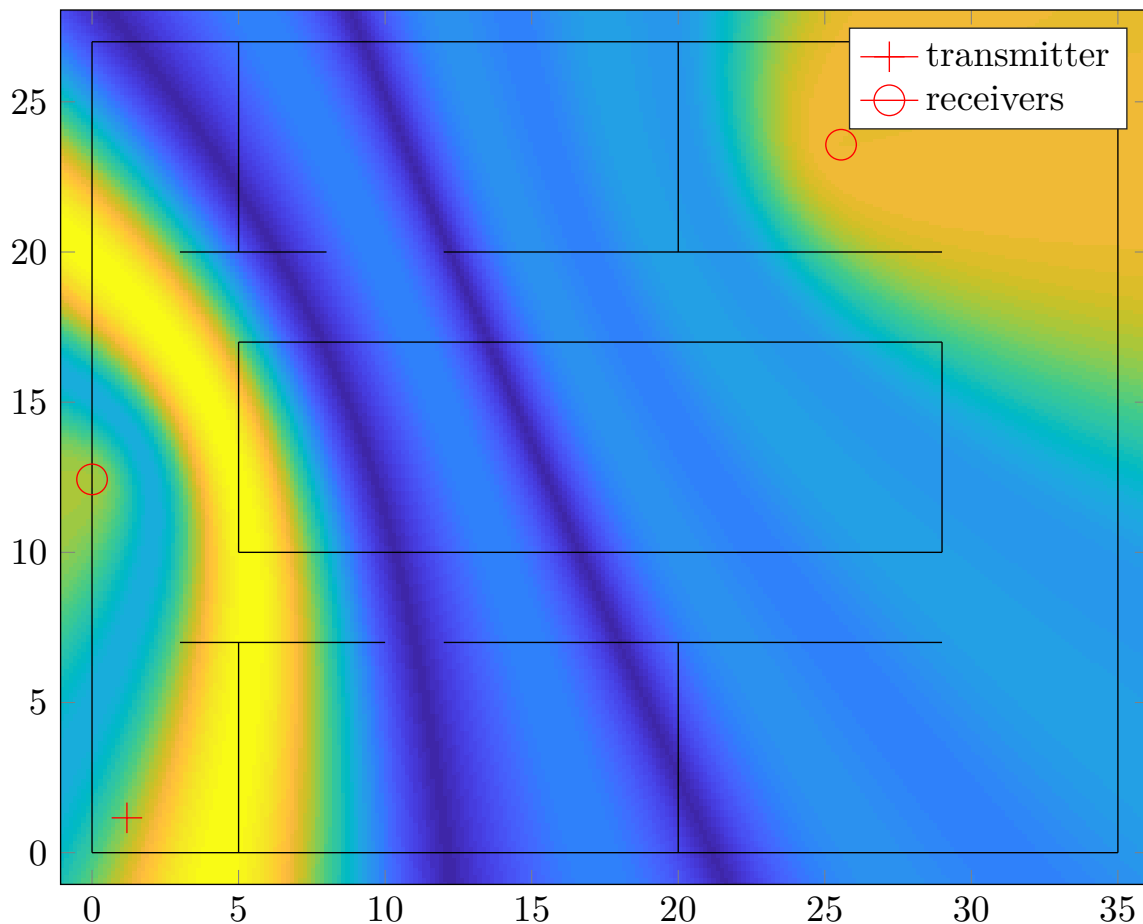


Fig. 3.3: The results of CC TDOA using the LUT method for 250,000 data samples, and two receivers.

### 3.2.2 Time Delay Estimation Methods

In order to compute TDOA, a time delay estimation (TDE) is needed. There are two ways that the TDE are computed in this thesis. The first method is looking at the relative delay between channels. The second is using the time-lag which maximizes the cross-correlation.

As the first peaks are more difficult to discern, the relative delay between channels is estimated using the greatest peaks between transfer functions, with the assumption that they are the first peaks. An example of delay estimation is shown in Figure 3.4. The relative time delay between channel 1 and channel 2 in Figure 3.4 can be explained mathematically

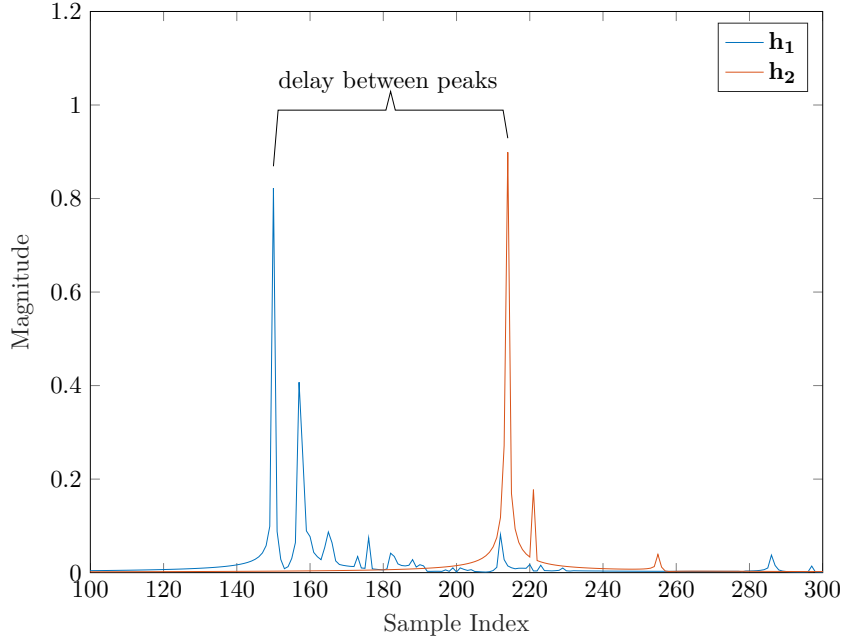


Fig. 3.4: An example of the normalized delay between two channels,  $\mathbf{h}_1$  and  $\mathbf{h}_2$ .

as

$$\tau_{1,2} = \tau_1 - \tau_2, \quad (3.4)$$

where  $\tau_i$  is the delay estimated from the transmitter to the  $i^{\text{th}}$  receiver. Distance is computed using the wavelength<sup>2</sup> and the delay from (3.4),

$$\begin{aligned} d_{1,2} &= \tau_{1,2} \cdot \frac{c}{F_s} \\ &= \tau_{1,2} \cdot \lambda. \end{aligned} \quad (3.5)$$

Computing the cross-correlation is explained in detail below. But as was shown in Figure 3.2(b), the relative time delay corresponds to the x-value of the largest peak.

It should also be noted that all the calculations in this thesis are discrete.

### Generalized Cross-Correlation TDOA Approach

<sup>2</sup>Note that  $\lambda$  here denotes wavelength, but in the following chapters,  $\lambda$  denotes a Lagrange multiplier.

The Generalized Cross Correlation (GCC) algorithm computes the time-lag that maximizes the cross-correlation (CC). The CC can be expressed as

$$\begin{aligned}\hat{\tau} &= \arg \max_{\tau} R_{x_k, x_l}(\tau) \\ &= \arg \max_{\tau} E[x_k(n)x_l(n + \tau_{k,l})].\end{aligned}\tag{3.6}$$

In (3.6),  $x_k(n)$  and  $x_l(n)$  represent received signals of the form

$$\begin{aligned}x_k(n) &= h_k(n) * s(n - \tau_k) + w_k(n) \\ x_l(n) &= h_l(n) * s(n - \tau_l) + w_l(n),\end{aligned}\tag{3.7}$$

where  $h_p(n)$  is the impulse response,  $s(n - \tau_p)$  is the signal and  $w_p(n)$  is the additive noise at the  $n^{\text{th}}$  sample for  $p = k, l$ . The expectation for the CC<sup>3</sup> can be approximated as

$$\hat{R}_{x_k, x_l}(\tau) = \sum_{n=0}^{N-|\tau|-1} x_k(n)x_l^*(n + \tau),\tag{3.8}$$

where  $N$  is the total number of samples [2, 12, 26, 27, 46, 48]. However, it is common practice to convert from the time (sample) domain to the frequency domain (using the Fast Fourier Transform (FFT), and inverse FFT (IFFT) respectively) for the CC computation as it can be implemented more efficiently. Converting it to the frequency domain is referred to as the cross-spectrum

$$\hat{R}_{x_k, x_l}(\tau) = \frac{1}{2N} \sum_{v=0}^{N-1} X_k(f_v)X_l^*(f_v)e^{j2\pi f_v \tau},\tag{3.9}$$

where  $f$  is the frequency in hertz (Hz), and

$$X_p(f) = \sum_{u=0}^{N-1} x_p(u)e^{-j2\pi f_v u}\tag{3.10}$$

[12, 15, 25, 26, 46, 49].

---

<sup>3</sup>Note that most versions of the CC functions involve a scaling value [12, 15, 26, 46], but the default for the `xcorr` function in MATLAB, which is used as the baseline for comparing the results of the algorithms researched in this thesis, does not [47].

The GCC is a more generalized version of the CC, weighting the cross-spectrum estimate. The weighting function should be chosen such that the CC function results in a sharp peak at the time delay [12, 26, 46]. The common equation used to portray the GCC is

$$\hat{R}_{x_k, x_l}(\tau) = \sum_{v=0}^{N-1} \Phi(f_v) X_k(f_v) X_l^*(f_v) e^{j2\pi f_v \tau}, \quad (3.11)$$

where  $\Phi(f_v)$  is a weighting function, which is determined by the application. There are many common GCC methods including

- the classical CC (CCC) method, shown in (3.9), [2, 12, 15, 25, 27, 46, 49]
- the phase transform (PHAT) [12, 13, 15, 25, 26, 40, 46],
- the maximum likelihood (ML) method, sometimes referred to as Hannan-Thomson (HT) method [2, 12, 15, 16, 26, 46],
- the smoothed coherence transform (SCOT) method [12, 15, 46] ,
- the Eckart filter method [12, 46],
- the Roth processor method [46], and
- the maximum *a posteriori* (MAP) method [26, 50].

Of the common methods, the CCC, PHAT and ML GCC methods are most used as can be seen from their collective sources.

The CCC method is typically used for its straight forward simple approach, however it may lead to a large bias in the estimate [25, 46, 49]. The PHAT method was created to avoid the delta function spreading that was observed with both the SCOT and Roth methods, although some spreading may still occur when nonideal situations. It is to be used “ad hoc”, and can enhance small errors, have erratic phase estimates, and may need additional weighting to combat signal power, [25, 26, 46]. The ML method is asymptotically optimal to achieve the Cramer Rao lower bound (CRLB), but in order to achieve optimality, the sample space needs to be large, the environment free of multipath, with little to no noise,



and the spectra of the signals, and attenuation factors must be known *a priori* [12,15,46,49]. In general, although GCC approaches are commonly used for TDOA they still struggle with the presence of multipath [15,25,27,40,49] In multipath environments, more peaks are likely to occur, making the delay peak spread, and can even become distorted by merging with other peaks [46].

As most of these methods require more information known prior to processing, only the CCC method is used in this thesis as a baseline to compare the algorithms that make up this research.

### Other Algorithms

The majority of this thesis is devoted to the derivation and testing of several cross-relation (CR) algorithms. A few of the algorithms discussed offer a brief explanation of preexisting work. And the rest are new algorithms that attempt to use channel identification to improve performance in the presence of multipath.

The preexisting work discussed in this thesis include the Adaptive Eigenvalue Decomposition [12,15,25], and the channel identification algorithm proposed by Xu et al. [38,39], which is referred to as the SVD algorithm in this thesis to differentiate it from the other algorithms. These explanations can be found in Chapter 4.

There are ten original algorithms and modified versions that are introduced by this thesis including:

- Modified Adaptive Eigenvalue Decomposition (ModAED), found in Chapter 4,
- Sparse Modified Adaptive Eigenvalue Decomposition (ModAEDS), found in Chapter 4,
- Adaptive Cross Channel Identification with Sparse Shift-Suppression (AXIS), found in Chapter 5,
- Varied Adaptive Cross-Channel Identification with Sparse Shift-Suppression (VAXIS), found in Chapter 5,

- Dual Fixed-Peak Adaptive Cross-Channel Identification with Sparse Shift-Suppression (DFPAXIS), found in Chapter 5,
- Non-Sparse Adaptive Cross-Channel Identification with Shift-Suppression (NSAXIS), found in Chapter 5
- Single Constraint Adaptive Cross-Channel Identification with Sparse Shift-Suppression (SC), found in Chapter 5,
- Varied Single Constraint (VSC), found in Chapter 5,
- Dual Fixed-Peak Single Constraint (DFPSC), found in Chapter 5, and
- Non-Sparse Single Constraint (NSSC), also found in Chapter 5.

A few other possible approaches were explored, which are also included in this thesis. The results of these algorithms were disappointing, and cannot claim to improve performance of channel identification in multipath environments due to their unreliability. These algorithms are discussed in Chapter 6.

All the algorithms explored in this thesis take (either by collecting or simulating the data) a received signal as input, and use the CR relationship to do channel identification. The received signal is assumed to be a transmitted signal convolved with an impulse response. Channel identification is then done to recover the impulse response without any knowledge concerning the transmitted signal. The resulting estimated impulse response is the output of the algorithms.

The algorithms presented in the following chapters are least-squares (LS) algorithms, and all but one of the algorithms are adaptive, using gradient descent for channel identification. For channel identification, the data is collected and then organized into a matrix, of size  $N - (L + 1) \times (L + 1)$ , where  $N$ , is the total number of data samples, and  $L$  is the channel order. The algorithms iterate through the data row by row for the adaptive update.

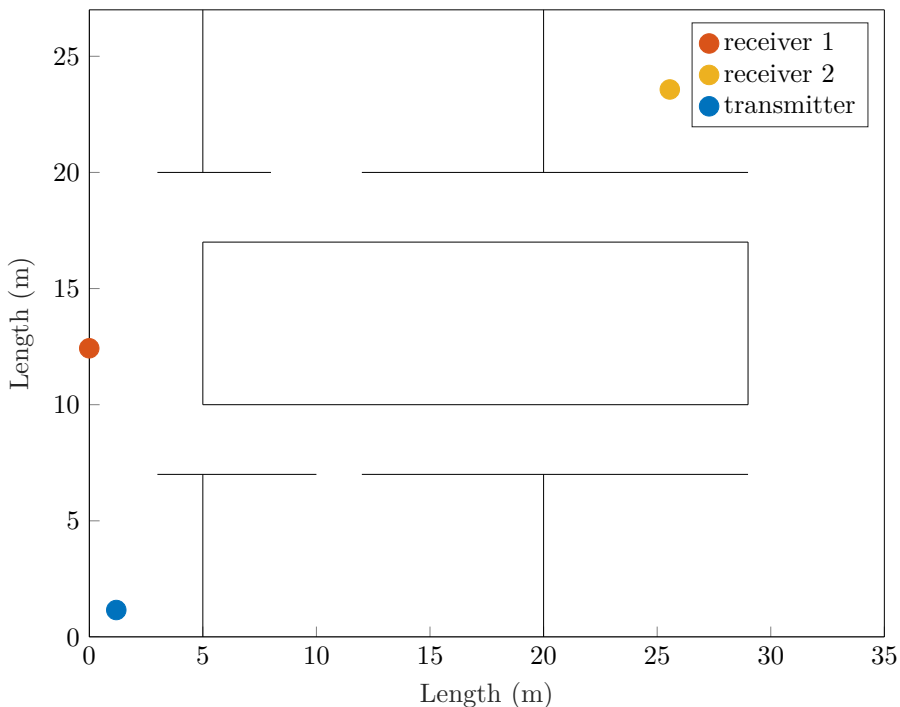


Fig. 3.5: The environment used to simulate impulse responses including the location of the transmitter, and two of the potential receivers which were used for all testing. The black lines represent the “walls”, or rather boundaries, on which reflections and refractions may occur.

### 3.3 Simulations

All simulations were done in MATLAB<sup>®</sup>. Realistic impulse responses were derived by applying a ray-tracing algorithm to a simulated indoor environment. An example of the simulated environment used for ray tracing can be seen in Figure 3.5. These pre-simulated impulse responses were then fed complex band limited white-noise to approximate a communication signal.

Multiple impulse responses of different sample rates (all in megahertz, shown as MHz) were used for simulation testing at sample rates

- 20 MHz,
- 40 MHz,
- 80 MHz,
- 100 MHz,
- 120 MHz,
- 140 MHz,
- 150 MHz,
- 160 MHz,
- 180 MHz,
- 200 MHz.

The ray tracing algorithm yields a total of 4096 path combinations; these paths were used to generate discrete impulse responses corresponding to the frequencies listed above. There are 21 variations of the 4096 paths, which used differing seed values corresponding to realistic phase shifts in reflections due to the inconsistencies of wall materials. It should also be noted that these paths are rough models for impulse responses inside buildings. The paths use hypothesized refractive indices values, and, as can be seen in Figure 3.5, there are only walls used in the model environment, no furniture, doors, windows, or people were simulated for simpler implementation.

In an ideal free-space situation, an ideal impulse would be sent through a system, and collected at each receiver. When the impulse would be collected at the receivers, the signal would first go through an ideal low-pass filter (LPF). The LPF bandlimits the signal, as an ideal impulse has an infinite spectrum, transforming the impulse into a sinc function as shown in Figure 3.6. After the signal has been filtered, it would then be converted to a digital signal using an analog to digital converter (ADC).

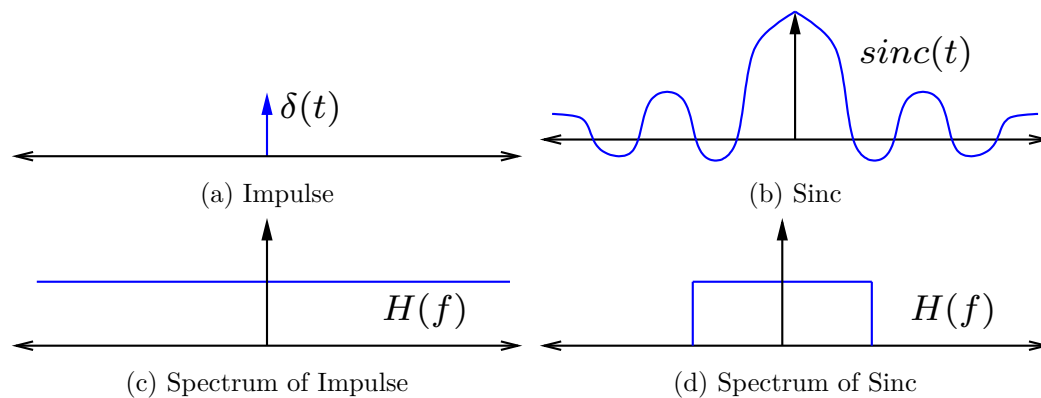


Fig. 3.6: Ideal impulse response explanation

In actuality (and especially in this research), free-space is not possible. Walls (the only part of the building simulated) can cause reflections, or refractions, which then can cause delays. In these multipath environments, like shown in Figure 3.1, that multiple impulses will be acquired by the receiver over time. Thus, there will be multiple delayed impulses that are acquired by the receiver.

To generate the discrete impulse responses as realistically as possible, an ideal impulse was generated in MATLAB, and “sent” through the indoor environment to determine the corresponding delays from transmitter to receivers. Once the delays had been determined, the received signal was put through an LPF. This replaced each delayed impulse with a sinc function. Then, the filtered impulse response was then sampled at the desired sampling rate. This process corresponds to Figure 3.7.

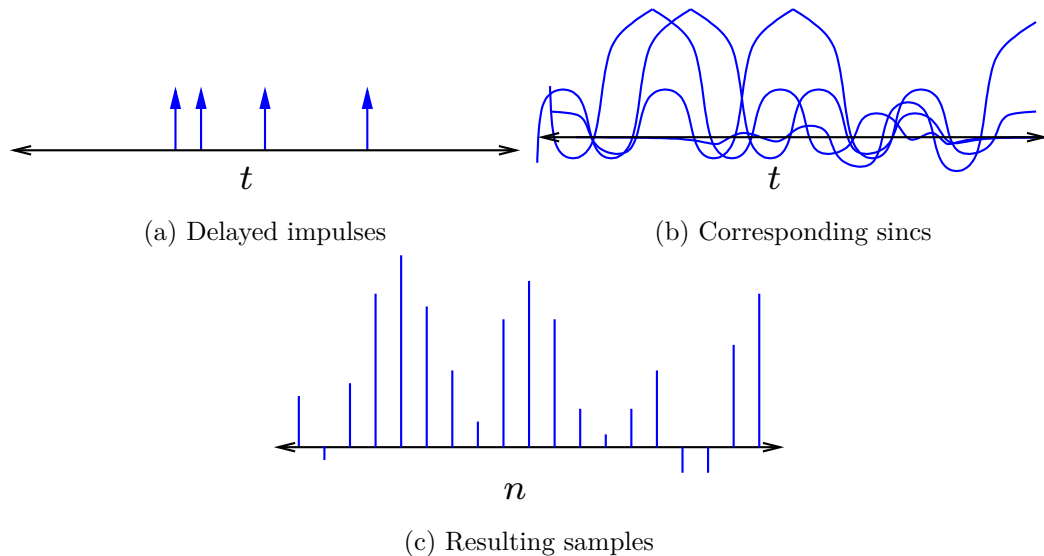


Fig. 3.7: Simulated impulse response explanation.

Two versions of impulse responses were simulated. The first version, referred to as set IR-1, assumes the power of the received signal is greater than the power of the transmitted signal. This might occur if the signal is amplified by some material as it is transmitted. The second set, referred to as set IR-2 assumes the opposite, and is the more likely of the two due to natural attenuation that occurs when refracted and reflected. For the two receivers shown in Figure 3.5, both impulse response sets can be seen in Appendix C, in Figures C.1– C.11 and in Figures C.12– C.22 respectively. These two channels were used for all simulation testing. This wide spread of pre-simulated impulse responses allows for thorough testing of the algorithms. (The ray-tracing algorithm will not be discussed further in this document as it is not the focus of the research.)

Simulated data is run through each of the 21 variations of the impulse responses for each sample rate. The channel estimation results for the 21 variations are averaged together to prevent anomalies. The algorithm performances are all compared using these averaged results.

### 3.3.1 Different Simulation Parameters

For more thorough testing, assorted simulation parameters were varied.

#### Signal Bandwidth

The generated white noise fed into the transfer functions was wideband (full bandwidth). Narrowband (approximately 5 MHz) data was also simulated to provide an accurate range of data. To simulate narrowband data, the generated white-noise was filtered to 5 MHz bandlimited and then decimated.

The data is filtered using a 10<sup>th</sup> order Chebyshev Type I filter with a peak-to-peak passband ripple of 0.5 dB, and cutoff frequency,

$$f_c = 2 \frac{5 \cdot 10^6}{F_s}, \quad (3.12)$$

via the *cheby1* command in MATLAB. Then the data is filtered normally with the impulse response before being decimated down to 5 MHz. See Figure 3.8 for visual comparison of the signal bandwidth at 20 MHz sampling rate.

It should also be noted that the wideband bandwidth corresponds to the sampling rate, i.e. a signal filtered with an impulse response which has a 20 MHz sampling rate is assumed to have a bandwidth of 20 MHz.

#### Noise

In ideal environments, no noise will interfere with the signal, however ideal environments are not realistic, especially with multipath. Additive noise is common. Thus, for simulation, two different noise amounts were tested.

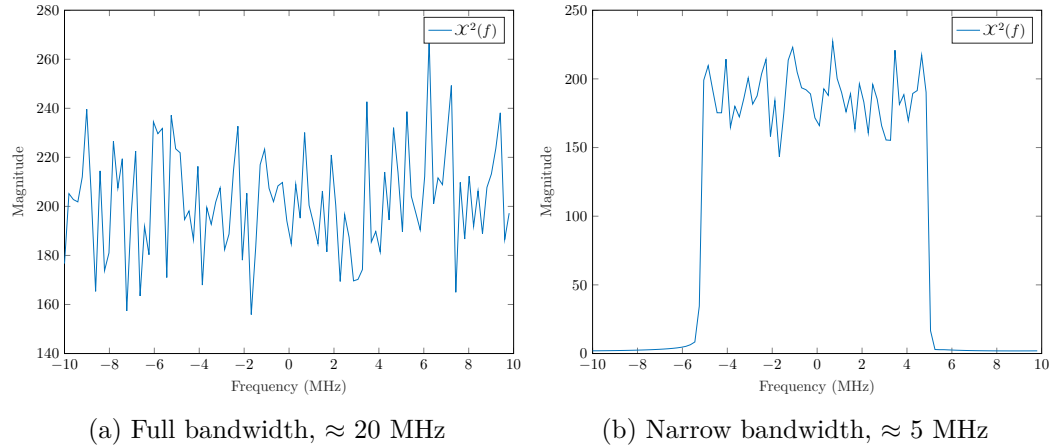


Fig. 3.8: Power spectral density comparison for (a) wideband, versus (b) narrowband data at 20 MHz.

SNR (dB)
60
20

Table 3.1: Noise amount with corresponding SNR in dB.

The signal to noise ratio (SNR) described here is calculated using the variance for the power comparison

$$SNR = 20 \cdot \log_{10}\left(\frac{\sigma_s}{\sigma_n}\right), \quad (3.13)$$

where  $\sigma_s^2$  is the variance of the signal, and  $\sigma_n^2$  is the variance of the noise.

### Sample Size

All the algorithms, except for one (which had memory issues due to processing) were tested using data sets of two sizes: 250,000 samples, and 2,500 samples. The algorithm which has memory issues was tested with 30,000 and 300 samples instead. The smaller amount is  $\frac{1}{100}^{th}$  the size of the larger amount. Theoretically, all the algorithms should perform better with more data, although it should still execute acceptably with fewer samples.

### Model Order Estimates

As explained in the literature, many algorithms and techniques depend heavily on knowing/having the correct model order [34, 35, 38]. The model orders in this thesis correspond to the sampling rate of each impulse response. For example, an impulse response with a 20 MHz sampling rate has model order  $L = 20$ , etc. The algorithms explored in this thesis also experiment with different model orders in simulation to determine if model order is a breaking point for the algorithms.

Four model orders,  $L$ , were tested, with some under-model order and some over-model order estimates

Model Order	Variant 1	Variant 2	Variant 3	Variant 4
$L$	$L/2$	$L - 10$	$2L$	$L + 10$

Table 3.2: Various model order estimates used for algorithm simulation

### The Ideal Simulation

For this research, a makeshift ideal case is used, comprising of a combination of the “ideal” simulation parameters explained above. The “Ideal Case” includes

- wideband,
- $SNR = 60$  dB,
- large data size,
- correct channel order,  $L$ .

This “Ideal Case” includes favorable conditions in which the channels should be easier to identify, and thus the delay more correct.



### 3.3.2 Coding Modifications

Some alternative programming modifications have been implemented that can make the proposed algorithms faster and/or more accurate. Most, if not all, the algorithms derived in this thesis are iterative update algorithms. These adaptive algorithms typically process a single row of data per update. Instead of updating the impulse response estimates by an individual row, the programs are modified such that blocks of multiple rows of data are used per update. This modification takes and computes multiple row calculations simultaneously. The resulting step (update) is averaged from these calculations, as explained in Appendix B.1.

Another modification is upsampling the received data to compensate for the limited sample rate. Upsampling provides a fractional bin delay estimate instead of a whole bin delay estimate. This allows for more accurate delay estimate. The estimated impulse responses do need to be appropriately modified to compensate for this upsampling as well if implemented.

### 3.3.3 Results Explanation

Simulation results are shown in two ways in this thesis. The first way is by plot, and the second way is by table. Only results for sampling rates  $F_s = 20, 100, 150,$  and  $200$  MHz are shown in the tables and the plots for channels 30 and 3000 in each chapter. Channel 30 and channel 3000 correspond to the receiver locations shown in Figure 3.5. Although there are 4096 possible channels, channels 30 and 3000 were picked arbitrarily. Each chapter also only shows the results for the algorithms discussed in that chapter. A complete table of all sampling rates and all of the algorithms is shown in Appendix D, but more plots are not shown, as they are deemed unnecessary.

The plots are similar to what is shown in Figure 3.4. Two plots are shown side-by-side, showing both what the channel estimates should look like, and what the estimates do look like after processing. This shows how well the channel estimation methods work. Each plot shows the estimates for the sample rate version seed 0 (the first of the 21 variants).

The table show results averaged over the 21 variations of the impulse responses. The simulation tables contain 9 columns. These tables show information about

- simulation state,
- sample rate,
- algorithm,
- actual distance between receivers
- averaged estimated distance between receivers
- minimum estimated distance between receivers
- median estimated distance between receivers,
- maximum estimated distance between receivers, and
- averaged error between estimated distance and the actual distance.

The *Simulation State* column is used to describe the parameter situation and the impulse response set used during simulation. Possible parameter situations include different signal bandwidths, noise rates, sample sizes, and model orders, as explained in more detail above in Section 3.3.1.

The *Sample Rate* and *Algorithm* columns are fairly straight forward. The second column details the impulse response sample rate used for simulation. The third column details the algorithm for which results reside in that row of the table.

The fourth column,  $d_{1,2}$ , represents the true distance between receivers in meters. It is the same for all of the tables, as the distance between receivers does not change. The distance between receivers is calculated using precalculated distance estimates. These precalculated distance estimates were computed during generation of the impulse responses (via the ray-tracing algorithm). They compute the distance from transmitter to the receivers. The relative distance between receivers is then calculated using (3.2).

Column five is used to show the averaged estimated distance between receivers,  $\hat{d}_{avg_{1,2}}$ , in meters. The distances are calculated using (3.4) and (3.5). They are then averaged together over the 21 variants as shown in (3.14).

$$\begin{aligned}\tilde{d}_{avg_{1,2}} &= \sum_{v=0}^{20} \hat{d}_{1,2}^{(v)} \\ \hat{d}_{avg_{1,2}} &= \frac{\tilde{d}_{avg_{1,2}}}{21},\end{aligned}\tag{3.14}$$

where  $^{(v)}$  denotes which variant of the 21 impulse responses was used.

The next three columns show the range of the estimates giving the minimum, median and maximum delay estimates in meters. These are shown in the table using  $\hat{d}_{min_{1,2}}$ ,  $\hat{d}_{med_{1,2}}$ , and  $\hat{d}_{max_{1,2}}$  respectively. These values are calculated using the built-in functions in MATLAB.

The last column shows the error in meters between the true distance and the estimated distance averaged. This is calculated using the typical error calculations averaged together

$$\begin{aligned}\tilde{\epsilon}_{avg_{1,2}} &= \sum_{v=0}^{20} d_{1,2} - \hat{d}_{1,2}^{(v)} \\ \epsilon_{avg_{1,2}} &= \frac{\tilde{\epsilon}_{avg_{1,2}}}{21}.\end{aligned}\tag{3.15}$$

It should be noted that there is the possibility for there to be no maximum peak recorded, for one or both channels. In this instance, the resulting distance estimate in the table will be a *NaN*, which stands for not a number. NaNs could occur when there is not enough room within the model order for the correct delay to establish or if the estimate degenerates into a normalized random signal before iterations were finished. There are probably a few other situations which can occur, but the two explained above are some that have been observed during this research.

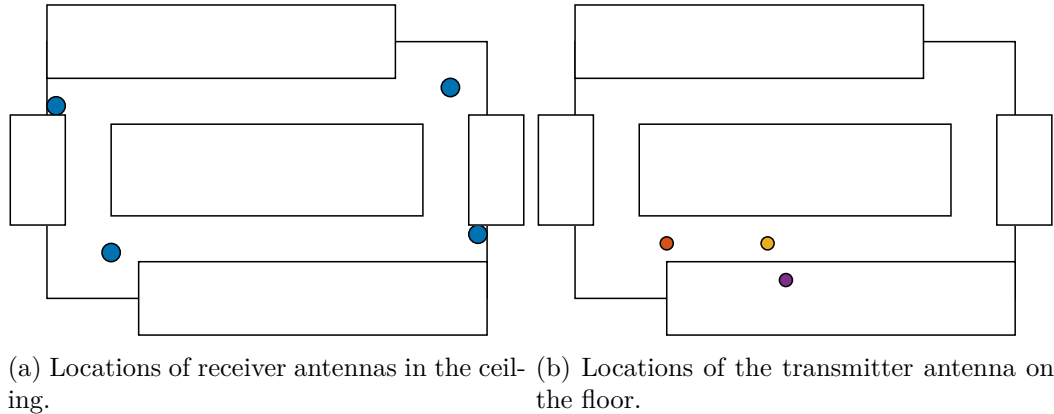


Fig. 3.9: SANTS building layout with the basic outline of the data collection. The circles either denote the location of a receiver or the location of the transmitter, whereas the lines depict the general layout of the major walls on the second floor of the building. Figure (a) shows the locations of the receivers in the ceiling and Figure (b) shows the various locations of the data transmitter used.

### 3.4 Data Collection

The data being transmitted is assumed to be an RF communication signal in nature, and was modeled this way in environmental testing.

Actual data was collected from the second floor of the Engineering Innovations (SANTS) building on the Utah State University (USU) campus in Logan, Utah. This location has time-synchronized antenna nodes spread throughout the ceiling of the second floor connected to a corresponding SDR. Data was collected at the receivers from a transmitter of a known frequency, placed at a known location. A mockup of the general building layout is shown in Figure 3.9, showing both the approximate locations of the receivers, and the known location of the transmitters. Refer to Section 3.5 for the specific data sets.

Two different transmission signals were collected. The first signal collected was a signal from a handheld portable two-way radio transceiver, referred to as a HT. The center frequency that the HT transmits at is  $f_c = 462$  MHz. The second signal collected was transmitted from a SRH779 omnidirectional antenna connected to an Ettus Universal Software Radio Peripheral (USRP) B200 mini SDR. The transmission was Gaussian distributed white noise, transmitted at 16 MHz sampling rate, with a center frequency of  $f_c = 147$  MHz. A QPSK communication signal was proposed, however the computation in GNURadio was

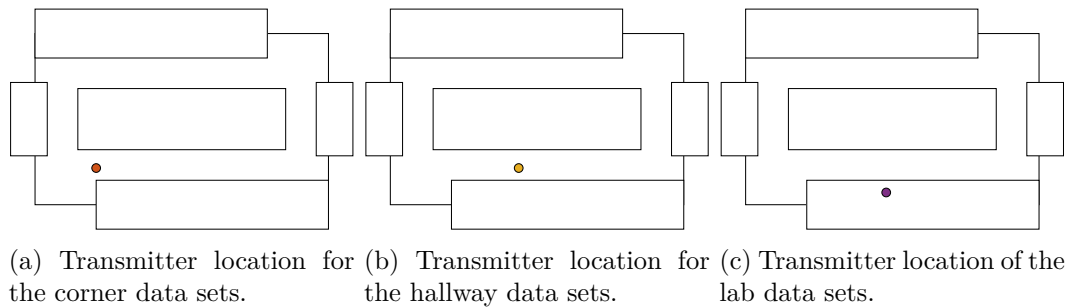


Fig. 3.10: Transmitter locations for the data sets.

too slow to keep up with the transmission at the 16 MHz sampling rate. Both transmitters were tested at 3 different locations, as shown in Figure 3.9(b).

Four VERT2450 omnidirectional antennas spread through the ceiling were connected to four Ettus USRP E320 SDRs. The SDRs were tuned to the known center frequency, and sampled at 16 MHz. Whilst the radios did perform synchronized sampling, their data collection start times were not time-aligned. Because of this misalignment, it was necessary to “calibrate” the received signals post-collection, to relative time-alignment using a separate signal fed directly into the receivers. It should be noted that the radios do have the capability to synchronize, however during the course of this research it was not able to be successfully implemented.

### 3.5 Data Sets

Six data sets were used for testing the algorithms. The data sets were collected on the second floor of the SANT building at USU. Three datasets were taken using the HT, and three datasets were taken using the B200 mini. The locations for the transmitters corresponding to each data set can be seen in Figure 3.10.

The ceiling nodes correspond to four channels; the northwest corner corresponds to channel 3, the northeast corner to channel 1, the southwest corner to channel 2, and the southeast corner to channel 4, as shown in Figure 3.11

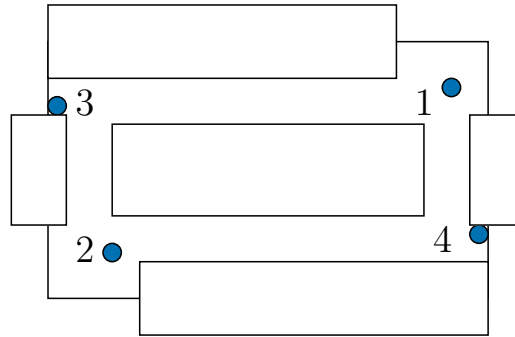


Fig. 3.11: The respective locations of each channel

### 3.5.1 HT Data Sets

Three HT data sets were collected:

1. Corner HT Data Set,
2. Hall HT Data Set,
3. Lab HT Data Set.

These data sets were collected by all four ceiling receivers, tuned to  $f_c = 462$  MHz with a 16 MHz sampling rate over 5 ms periods. The HT transmits a powerful signal as it is meant to transmit up to about 37 kilometers (km), but for inside of the SANT building, only about 50 meters (m), maximum are needed. Because the HT transmits such a powerful signal, it is very narrowband, as can be seen in Figure 3.12.

### 3.5.2 Gaussian White-Noise Data Sets

The other three data sets were taken with generated Gaussian white-noise through GNURadio to the B200 mini SDR:

1. Corner Gaussian Noise Data Set,
2. Hall Gaussian Noise Data Set,
3. Lab Gaussian Noise Data Set.

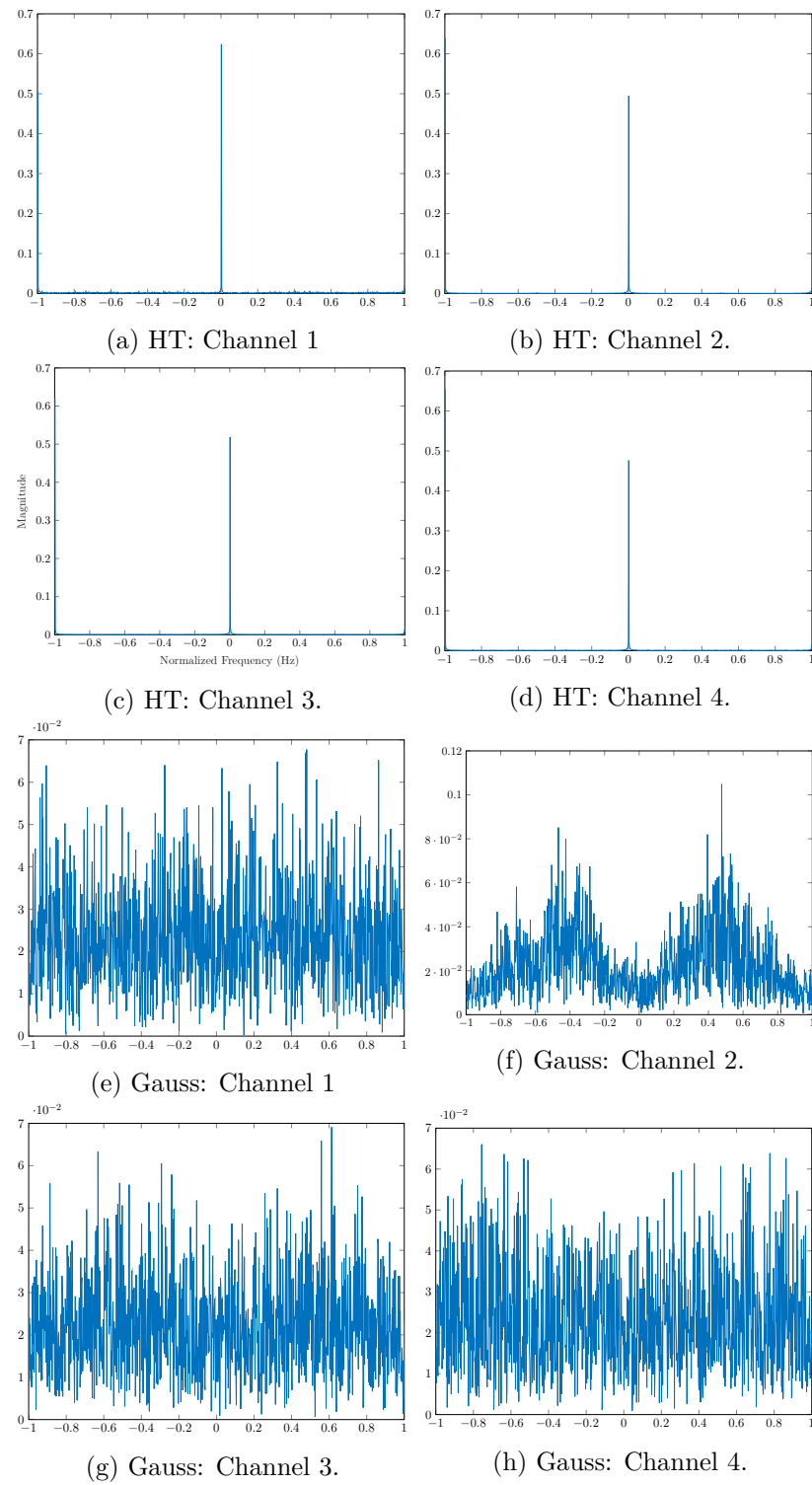


Fig. 3.12: Comparison of the normalized frequency bandwidths at each receiver for both data sets for the corner data set.

These data sets were collected by all four ceiling receivers, tuned to  $f_c = 147$  MHz with a 16 MHz sampling rate over 5 ms periods. The B200 mini is not as powerful a transmitter as the HT, and was set up to transmit a wideband signal, as can be seen in Figure 3.12.

Because the B200 mini is a low-power transmitter, it was very challenging to see the signal on the ceiling receivers, even when there was LOS. The transmitted power was also spread to transmit the wideband signal, making the SNR very small. Unfortunately, equipment to amplify the transmission power of the B200 mini was unavailable during research. As a result, not all the results are shown in this thesis, but a brief discussion can be found in Chapter 7.



CHAPTER 4  
PRE-EXISTING CROSS RELATION METHODS

For time difference of arrival (TDOA), blind channel identification (BCI) is a well studied subject, as explained in Chapter 2. Two algorithms developed by Xu et. al in [38] and by Benesty in [15] serve as the basis for the algorithms developed in this thesis. Their work, and similar work developed during this research are expounded in this chapter.

#### 4.1 Pre-Existing Algorithms

The least squares (LS) BCI algorithm put forth by Xu et al. in [38, 39] uses the cross-relation (CR), and the assumption that the transfer functions are in the nullspace of the data matrix, as shown in the following sections. Xu's algorithm is labeled the SVD algorithm, as all the algorithms utilize a least-squares approach, and Xu's algorithm uses the Singular Value Decomposition (SVD) for channel identification. The Adaptive Eigenvalue Decomposition (AED) as referred to in [12, 15, 25] is also a least-squares approach using the same nullspace assumptions.

##### 4.1.1 System Model

Consider a source at some unknown location transmitting an unknown signal,  $s(n)$ , in a multipath environment. The transmission is detected by multiple receivers. The system is modeled as

$$\begin{cases} x_1(n) = s(n) * h_1(n) + w_1(n) \\ \vdots \\ x_M(n) = s(n) * h_M(n) + w_M(n), \end{cases} \quad (4.1)$$

where  $*$  denotes convolution,  $h_i(n)$  denotes the individual channel mapping from the source to the  $i^{\text{th}}$  receiver,  $x_i(n)$  the value received from the transmitted signal at the  $i^{\text{th}}$  receiver, and  $w(n) = [w_1(n) w_2(n) \dots w_M(n)]$  is additive white noise, and  $n$  denotes the  $n^{\text{th}}$  sample.

$M$  denotes the number of receivers. This model assumes a finite impulse response, and is a single input, multiple output (SIMO) system, as is shown in Figure 4.1.

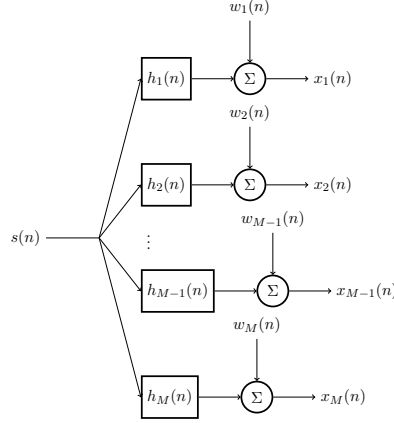


Fig. 4.1: System Model

#### 4.1.2 Cross Relation

Neglecting noise, the signal model, (4.1), can be expressed individually as  $x_i(n) = s(n) * h_i(n)$ . Convolving  $x_i(n)$  with  $h_j(n)$ , where  $i \neq j$  produces a CR,

$$\begin{aligned} h_j(n) * x_i(n) &= h_j(n) * [s(n) * h_i(n)] \\ &= h_i(n) * [s(n) * h_j(n)] \end{aligned} \quad (4.2)$$

$$h_j(n) * x_i(n) = h_i(n) * x_j(n) \quad (4.3)$$

$$\boxed{h_j(n) * x_i(n) - h_i(n) * x_j(n) = 0.} \quad (4.4)$$

The CR is the key to the BCI algorithms. Equation (4.4) can be rewritten for two receiver measurements as the matrix equation<sup>1</sup>

$$\begin{bmatrix} X_i(L) & \vdots & -X_j(L) \end{bmatrix} \begin{bmatrix} h_j \\ h_i \end{bmatrix} = \mathbf{0}, \quad (4.5)$$

<sup>1</sup>For multiple receivers ( $M > 2$ ), (4.5) can be modified as seen in Appendix A.1.

where  $\mathbf{X}_m(L)$  is a Toeplitz matrix of the form

$$\mathbf{X}_m(L) = \begin{bmatrix} x_m(L) & x_m(L-1) & \cdots & x_m(0) \\ x_m(L+1) & x_m(L) & \cdots & x_m(1) \\ \vdots & \ddots & \ddots & \vdots \\ x_m(N) & x_m(N-1) & \cdots & x_m(N-L-1) \end{bmatrix},$$

and  $\mathbf{h}_m$  is the impulse response vector

$$\mathbf{h}_m \triangleq [h_m(0), \dots, h_m(L)]^T.$$

For brevity, (4.5) can be rewritten as

$$\mathbf{X}\mathbf{h} = \mathbf{0}. \quad (4.6)$$

The size of  $\mathbf{X}$  is  $(N - (L + 1)) \times M(L + 1)$ , and  $\mathbf{h}$  is of size  $M(L + 1) \times 1$ , where the amount of channels,  $M = 2$ ,  $L$  is the channel order, and  $N$  is the data size for all of the calculations in this research. Equation (4.6) is the basis for the approach derived by Xu et al. [38], referred to as the SVD algorithm, and the AED algorithm. Equation (4.6) implies that the  $\mathbf{h}$  is in the nullspace of the data matrix  $\mathbf{X}$  (and vice versa).

### 4.1.3 A Least-Squares Approach According to Xu et al.

When there is noise, (4.6) is not exact. Indeed

$$\mathbf{X}\mathbf{h} \approx \mathbf{0}. \quad (4.7)$$

The approach of Xu frames the problem as,

$$\begin{aligned} & \underset{\mathbf{h}}{\text{minimize}} && \|\mathbf{X}\mathbf{h}\|_2^2 \\ & \text{subject to} && \|\mathbf{h}\|_2^2 = 1, \end{aligned} \quad (4.8)$$

where  $\mathbf{h}$  is an approximation of the true impulse responses.

The solution is found by taking the singular value decomposition (SVD) of the matrix  $\mathbf{X}$ . As the data matrix  $\mathbf{X}$  does not actually have a nullspace due to the noise, an approximate nullspace is derived by taking the right singular vector of  $\mathbf{X}$ , which corresponds to the smallest singular value of  $\mathbf{X}$ . The constraint ensures that a nonzero filter vector is selected.

This algorithm is referred to as the SVD algorithm. The code for the SVD algorithm can be seen in the code listing in Appendix B.2.

#### 4.1.4 Adaptive Eigenvalue Decomposition

The Adaptive Eigenvalue Decomposition (AED) [12, 15, 25], as the name implies, takes an adaptive approach, which minimizes the difference between update estimates. The algorithm attempts to solve this minimization equation for  $\mathbf{h}^{(n)}$

$$\begin{aligned} \min_{\mathbf{h}^{(n)}} \quad & \mathbf{h}^{(n)\text{T}} \mathbf{R}_{1,2} \mathbf{h}^{(n)} \\ \text{subject to} \quad & \mathbf{h}^{(n)\text{T}} \mathbf{h}^{(n)} = 1, \end{aligned} \tag{4.9}$$

where  $\mathbf{h}^{(n)}$  is the current iteration of the impulse response estimate, and  $\mathbf{R}_{1,2}$  is the covariance matrix between channels 1 and 2. The covariance matrix is computed by taking the expectation of the  $L_2$ -norm of the data matrix  $\mathbf{X}$ , which is computed as

$$\begin{bmatrix} \mathbf{R}_{x_1,x_1} & \mathbf{R}_{x_1,x_2} \\ \mathbf{R}_{x_2,x_1} & \mathbf{R}_{x_2,x_2} \end{bmatrix}, \tag{4.10}$$

where  $\mathbf{R}_{x_i,x_j} = E[x_i x_j^{\text{T}}]$ . Thus (4.9) can equivalently be written as

$$\begin{aligned} \min_{\mathbf{h}^{(n)}} \quad & E[\|\mathbf{X}\mathbf{h}\|_2^2] \\ \text{subject to} \quad & \|\mathbf{h}\|_2^2 = 1, \end{aligned} \tag{4.11}$$

assuming that the data is not complex. Equation (4.11) is very similar to the objective

function used by the SVD approach shown in (4.8), showing a logical connection between these two algorithms. The large difference is just in how they are calculated.

To compute the solution for the AED algorithm using gradient descent, the adaptive constrained LMS algorithm, found in [51], is used for the update equation. The update equation is

$$\begin{aligned}\tilde{\mathbf{h}}^{(n+1)} &= \mathbf{h}^{(n)} - \eta \frac{\mathbf{h}^{(n)H} \mathbf{x}_r}{\|\mathbf{h}^{(n)}\|_2^2} \mathbf{x}_r \\ \mathbf{h}^{(n+1)} &= \frac{\tilde{\mathbf{h}}^{(n+1)}}{\|\tilde{\mathbf{h}}^{(n+1)}\|_2},\end{aligned}\tag{4.12}$$

a normalized update where  $\mathbf{x}_r$  is a row of the data matrix  $\mathbf{X}$ , transposed as all vectors are assumed to be column formatted. Note that [12, 15, 25] assume that the signal is acoustic, and thus not adjusted for the complex RF signals, (4.12) makes the necessary adjustments.

The code for the AED algorithm can be seen in the code listing in Appendix B.3.

#### 4.1.5 Pre-Existing Algorithms Implementations, Advantages, and Disadvantages

Both pre-existing algorithms show promise, but there are issues associated with both.

The SVD algorithm can work rather well in the “ideal” simulation<sup>2</sup>, as can be seen in Figure 4.2. The channel estimates are very similar to the actual channels, with peaks in the correct corresponding places. The authors of [38] note in their paper that high SNR, and more data perform better, and discuss necessary and sufficient conditions (such as rank of data matrix  $\mathbf{X}$ , coprime nature of polynomials and nullspace dimensionality) in which the channels are identifiable.

However it must be noted that in nonideal conditions (small amount of data, narrow-band, low SNR, or incorrect  $L$ ), as seen in Figures 4.3, 4.5, 4.6, and 4.7 the algorithm does noticeably poorly. Figure 4.3 shows the results with fewer data points. Impulse response estimates for the lower sampling rates still perform well. For the higher sampling rates, the model order,  $L$ , is too close to the number of data points, and so performs poorly.

<sup>2</sup>Refer to section 3.3 for an explanation of the “Ideal Case”.

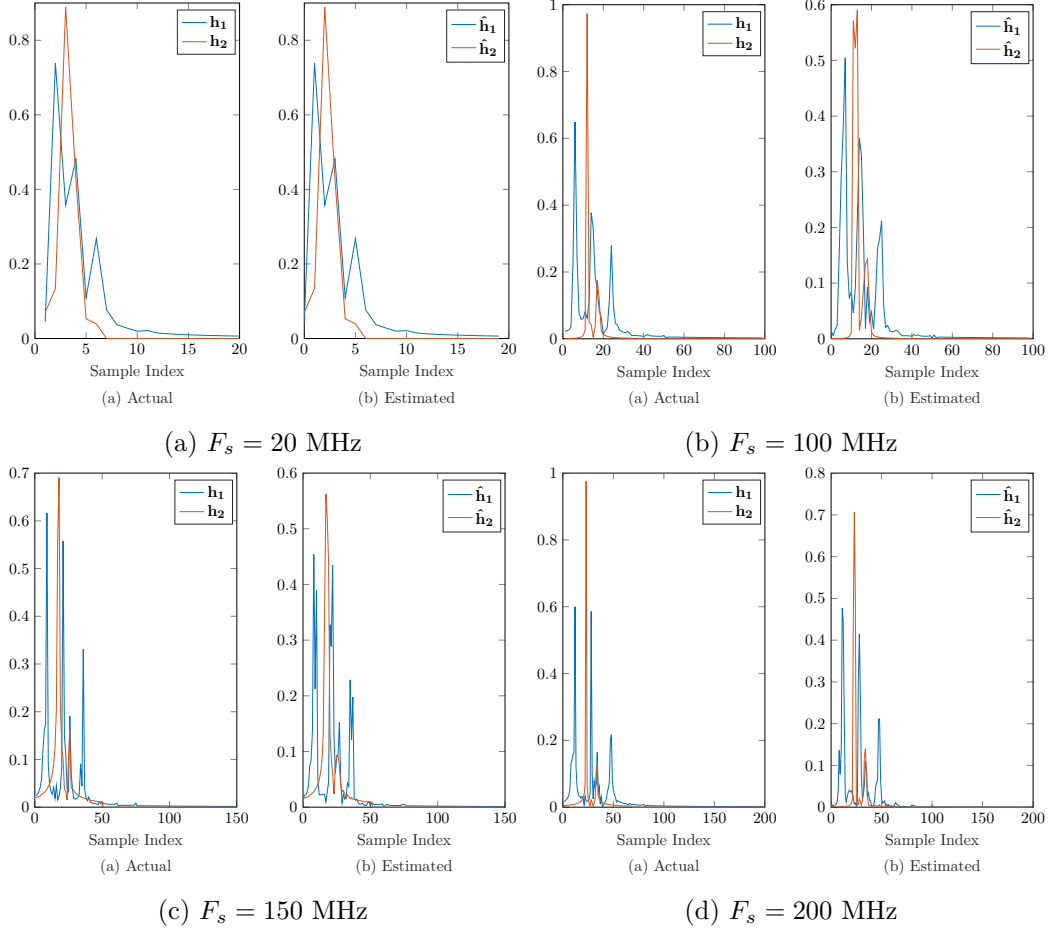


Fig. 4.2: Results of the SVD algorithm with 30,000 data samples, with  $SNR = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

This is because the data essentially becomes an underdetermined system, with too little information to effectively solve for all the model parameters. Figure 4.5 shows the results of the SVD approach for the simulated narrowband data. The narrowband estimates do not appear to resolve the two channels at all. The results seem to plot the same estimates, with small variations, for both channels. The plots are intuitive as narrowness in frequency generally results in broadness in the time domain. Figure 4.6 shows the results for lower SNR. Once reasonable noise is added ( $SNR = 20$  dB), the algorithm appears to deteriorate to random results. If the impulse response length,  $L$ , is incorrect, the estimate also deteriorates, as seen in Figure 4.7. Note that the correct model order for 200 MHz sampling rate

simulation is  $L = 200$ .

Figure 4.4 shows the results when the first peak is not the largest peak. Some of the estimates are decent (such as  $F_s = 20$  MHz plot), but most of the estimates do not identify the channels correctly.

Besides the reliability on an ideal setup, the SVD approach also suffers with respect to computational costs. The SVD is well known to be a costly operation, resulting in a slow processing, and utilizes a large amount of memory. The data matrix  $X$  of size  $(N - L) \times M(L + 1)$ , has complexity  $\mathcal{O}((N - L)^2 M(L + 1))$ , or more specifically there are  $4((N - L)^2 M(L + 1)) + 8((N - L)(M(L + 1))^2) + 9(M(L + 1))^3$  computations involved

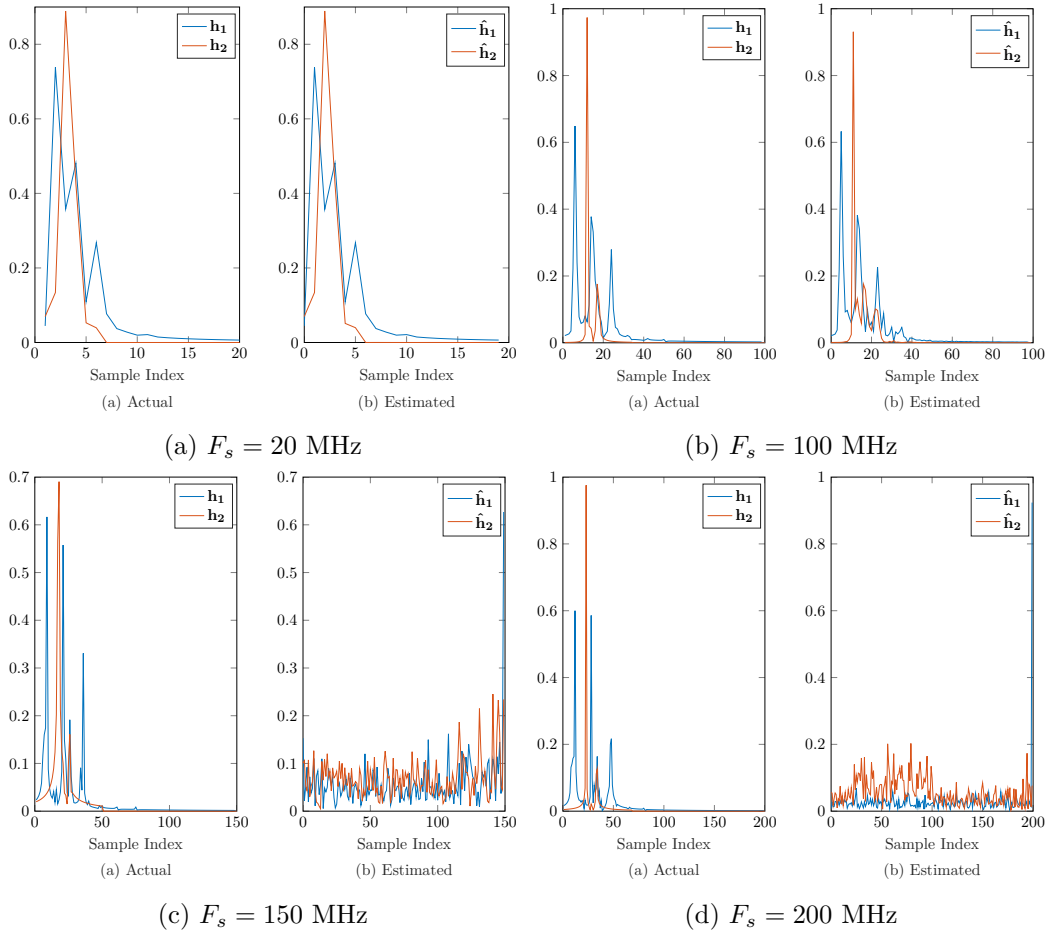


Fig. 4.3: Results of the SVD algorithm with 300 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

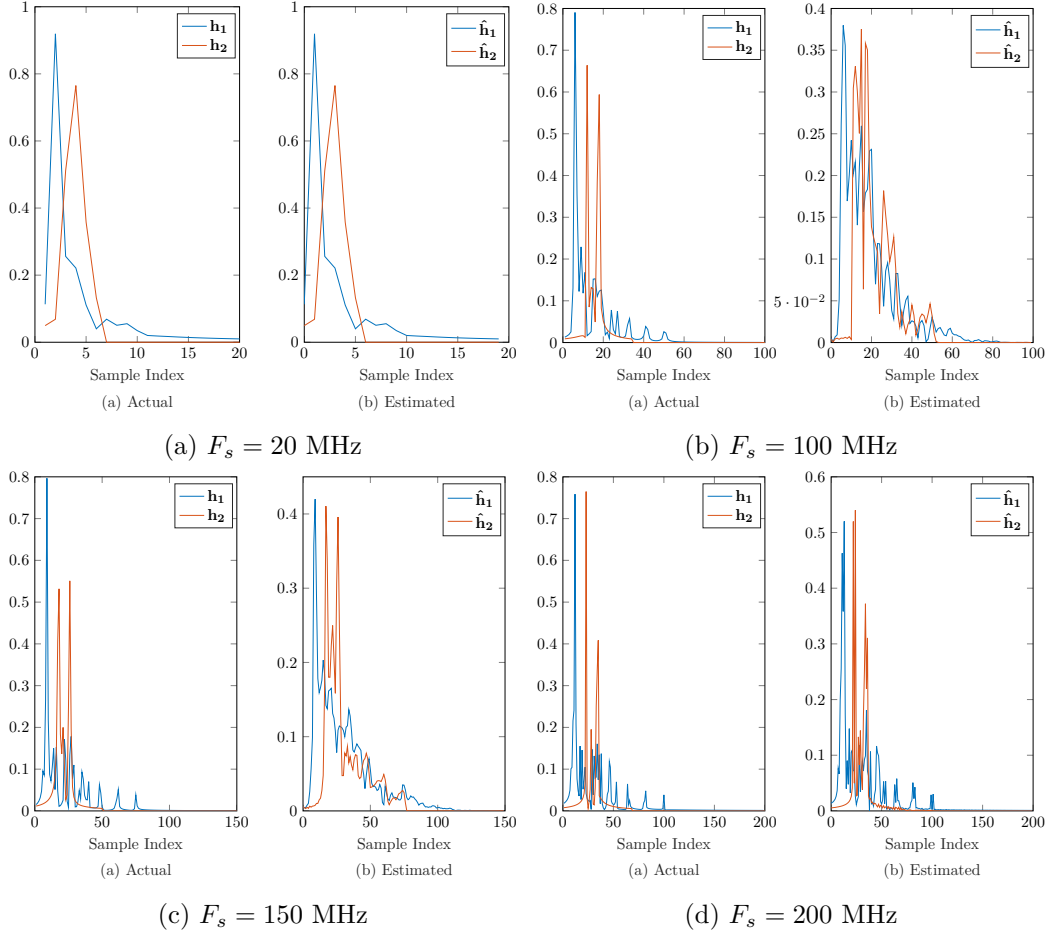


Fig. 4.4: Results of the SVD algorithm with 30,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

in the calculation for the SVD [50, 52]. Thus, the viability of this option is also limited to the system memory capacity. Due to this limitation, the SVD algorithm can only process approximately 30,000 data points before running out of memory in MATLAB.

The AED algorithm is a less computationally complex algorithm which iterates through the data matrix row by row to update the channel estimates, requiring only  $(N - L)(9M(L + 1) + 2)$  computations. The AED algorithm also differs from the SVD algorithm in that it focuses on estimating the time delay between the channels, rather than trying to perfectly estimate the channels themselves. The AED algorithm estimates the largest peaks (which are ideally the direct paths) of the impulse responses, and with the assumptions stated in



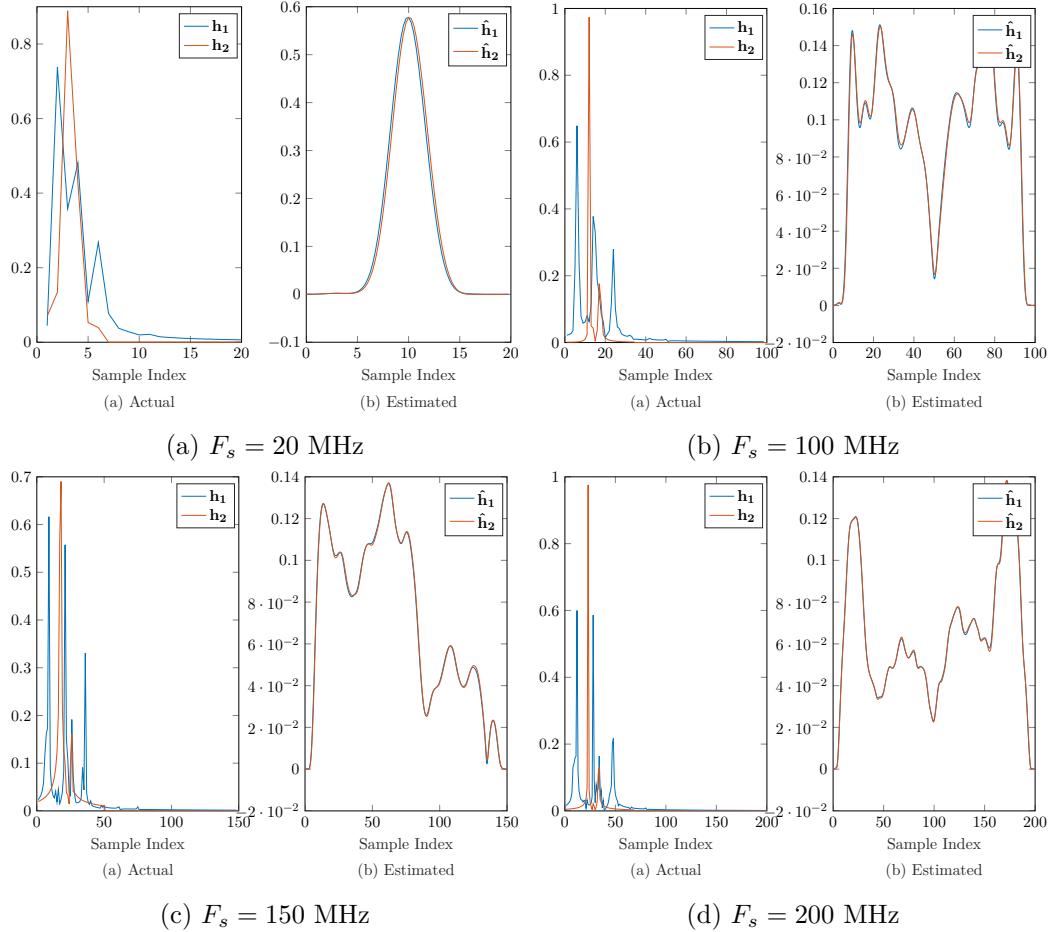


Fig. 4.5: Results of the SVD algorithm with 30,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Chapter 3, estimates the time delay between peaks. For TDOA, the minimal information is enough, however if identifying the channel profile exactly is important, this algorithm may be unacceptable.

In [15, 25], Benesty specifies that in order to accurately estimate the time delay between channels the transfer function approximation must initially start with an unit vector,

$$\mathbf{e}_d = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^T \quad (4.13)$$

where the value of the  $d^{\text{th}}$  index is equal to 1. The  $d^{\text{th}}$  index should fall somewhere in the middle of the first or second impulse response,  $L$ . Due to the mathematical relationship

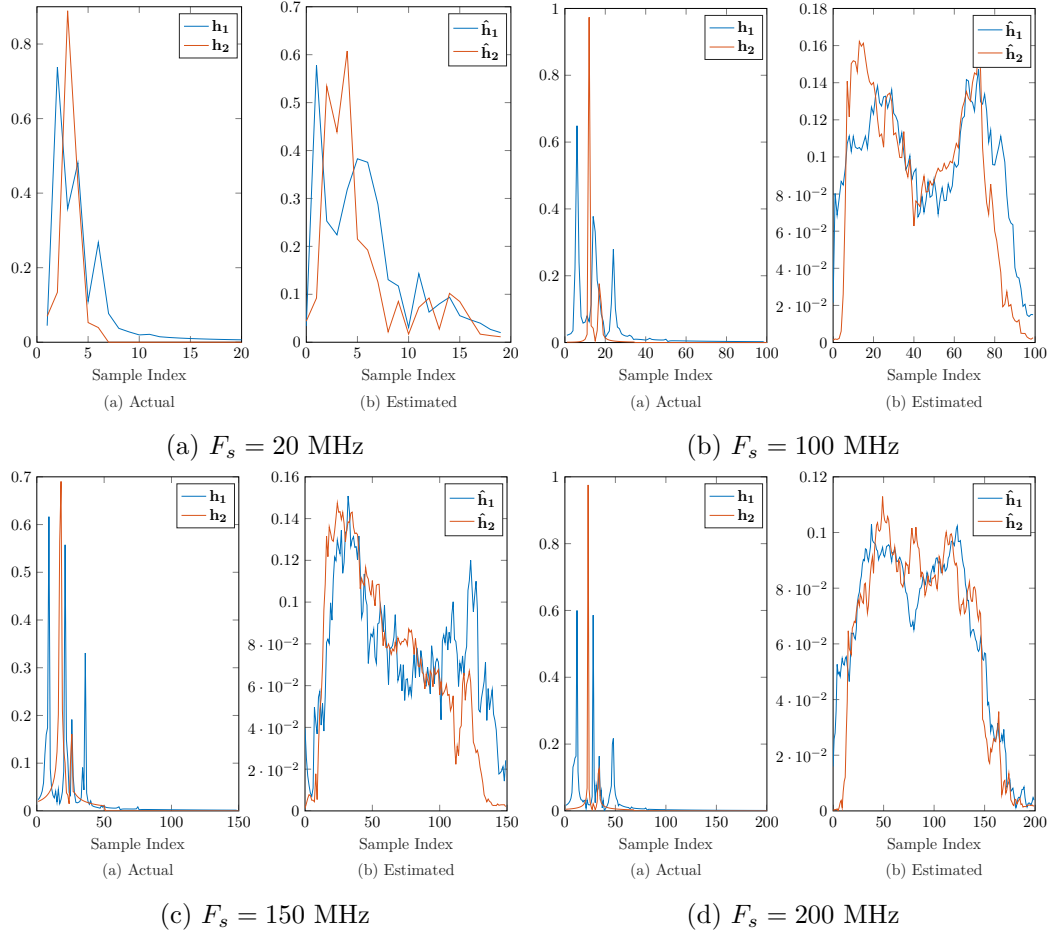


Fig. 4.6: Results of the SVD algorithm with 30,000 data samples, with  $\text{SNR} = 20$  dB at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

between the two impulse responses, a corresponding peak will appear at the delay amount.

With the initial estimate,  $\mathbf{h}^{(0)} = \mathbf{e}_d$ , the AED algorithm performs fairly well in ideal conditions as can be seen in Figure 4.8. It should be observed that the estimates are centered in the middle of the sample index due to the position of the  $d^{\text{th}}$  index in the initial impulse response. It should also be observed that the estimates are not quite as complete as the SVD estimates shown.

However, not all the sources for the AED algorithm explain what the initial transfer function estimate should be [12]. The only stipulation for the initial estimates for the transfer function approximation is that the  $\|\mathbf{h}^{(n)}\|_2^2 = 1$ . So, with this stipulation, a normalized

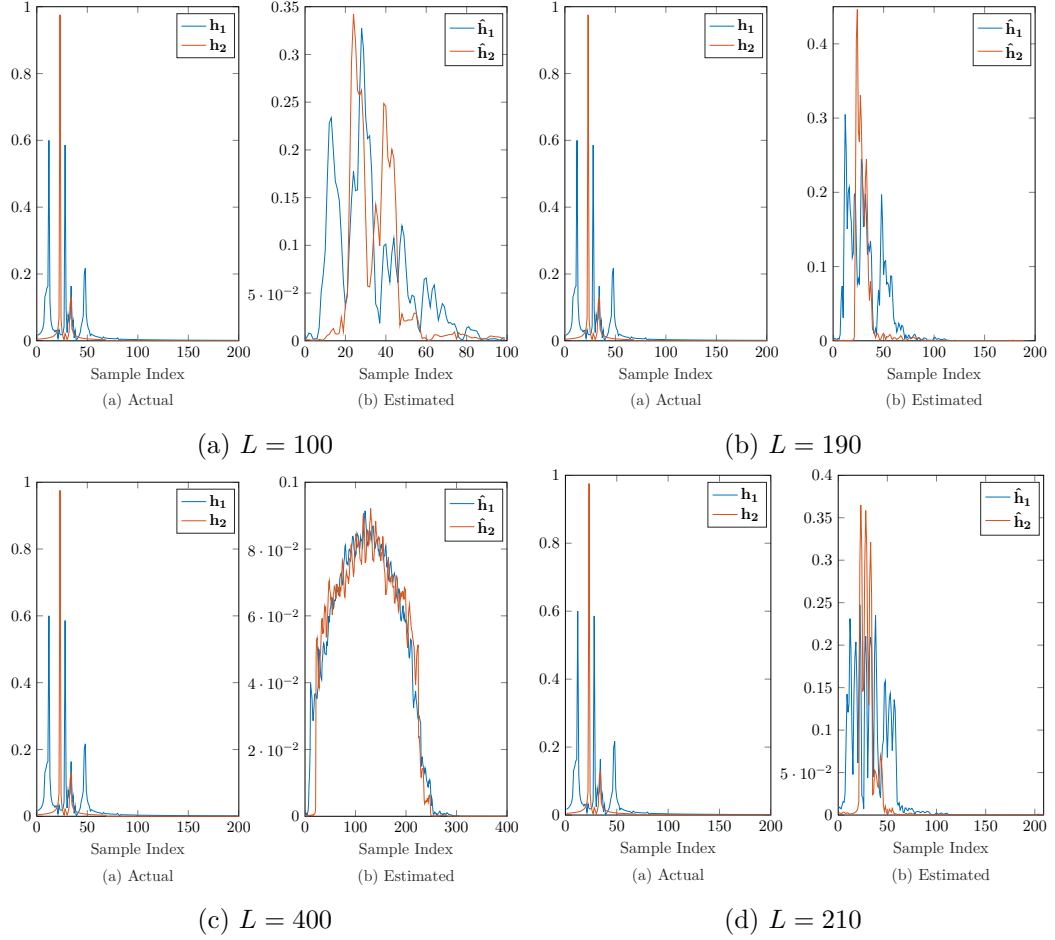


Fig. 4.7: Results of the SVD algorithm with 30,000 data samples, with  $\text{SNR} = 60$  dB for wideband at 200 MHz sampling rate with incorrect channel orders,  $L$  (a)  $L/2$ , (b)  $L - 10$ , (c)  $L * 2$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

random start, requires more updates, and iterations. However, this typically results in a random normalized result with no true single maximum peak as is shown in Figure 4.9.

Other nonideal conditions can be seen in Figures 4.10, 4.11, 4.13, and 4.14. Figure 4.10 shows the results with  $\frac{1}{100}$ <sup>th</sup> the data sample size. As with the SVD algorithm, the narrowband data cannot appear to resolve the peaks of the two channels, see Figure 4.11. The AED algorithm does not appear to be affected greatly by more noise as shown in Figure 4.13, producing similar results to the ideal case in Figure 4.8. Also, the model order does not greatly affect the channel estimates as seen in Figure 4.14.

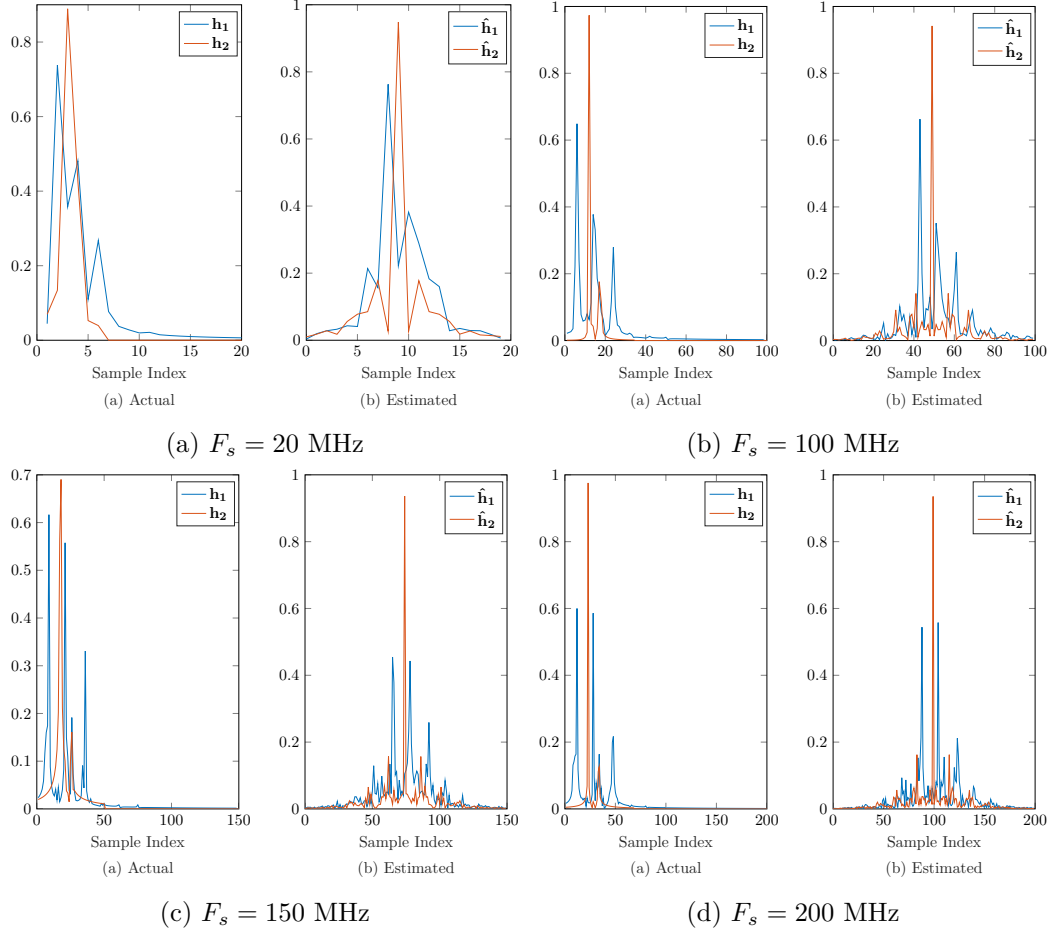


Fig. 4.8: Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Figure 4.12 shows the results of the AED algorithm with IR-2 set. The channel estimates produced are fairly good. The interesting thing to note is that the channels appear to be flipped in order, and  $\hat{\mathbf{h}}_2$  is more reflective of  $\mathbf{h}_1$ , rather than  $\mathbf{h}_2$  (which makes  $\hat{\mathbf{h}}_1$  reflective of  $\mathbf{h}_2$  in turn).

Tables 4.1–4.9 shows a numerical comparison of the estimates for both of these pre-existing algorithms, and for the modified algorithms introduced in the next section. A discussion of the numerical results can be found in Section 4.3.

## 4.2 Modified Algorithms

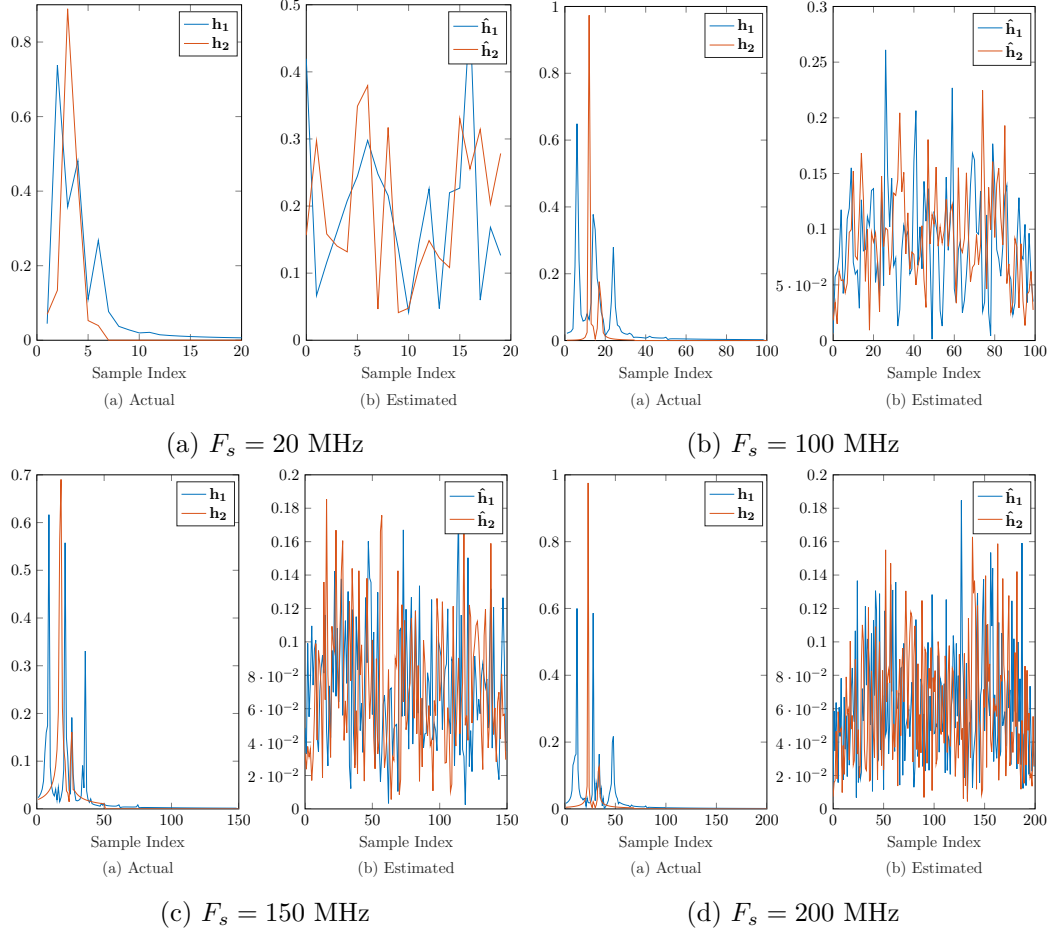


Fig. 4.9: Results of the AED algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0 using a normalized random start.

Using the AED algorithm as a basis, two similar complex algorithms were derived during the course of this research. These two algorithms are then used as a basis for the algorithms developed in Chapter 5.

#### 4.2.1 Modified System Model

The system model explained here is like the model used for the AED and SVD algorithms, shown in Figure 4.1. It should be noted that the AED algorithm was originally created for acoustic signals, whereas the data transmitted in this research are assumed to be complex RF signals. The SVD algorithm generalizes naturally for complex signals and

needs no modification, but for ease of computation, a slightly different complex model is used for all the other algorithms in this thesis. This affects the cross-relation equations on which these algorithms are based on, and so is explained in detail here.

As before, consider a source, at some unknown location, transmitting an unknown signal,  $s(n)$ , in a multipath environment. The transmission is detected by multiple receivers.

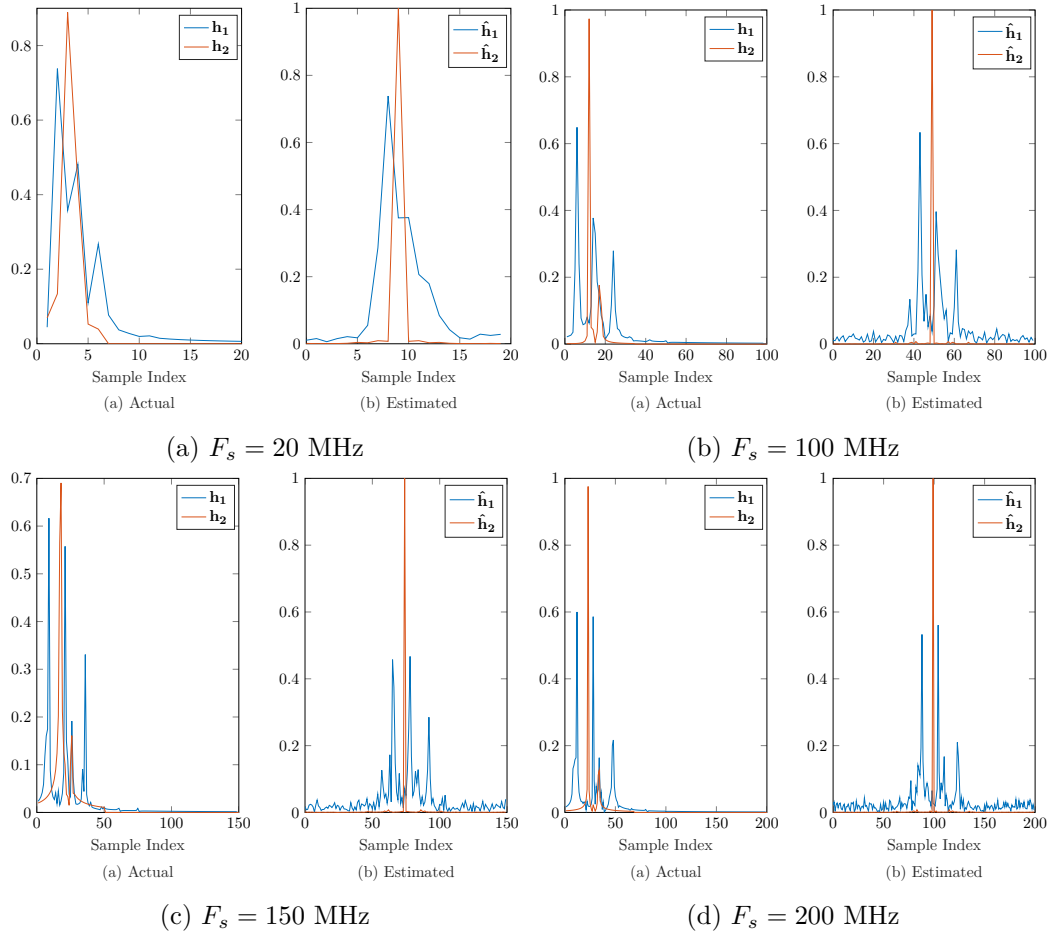


Fig. 4.10: Results of the AED algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

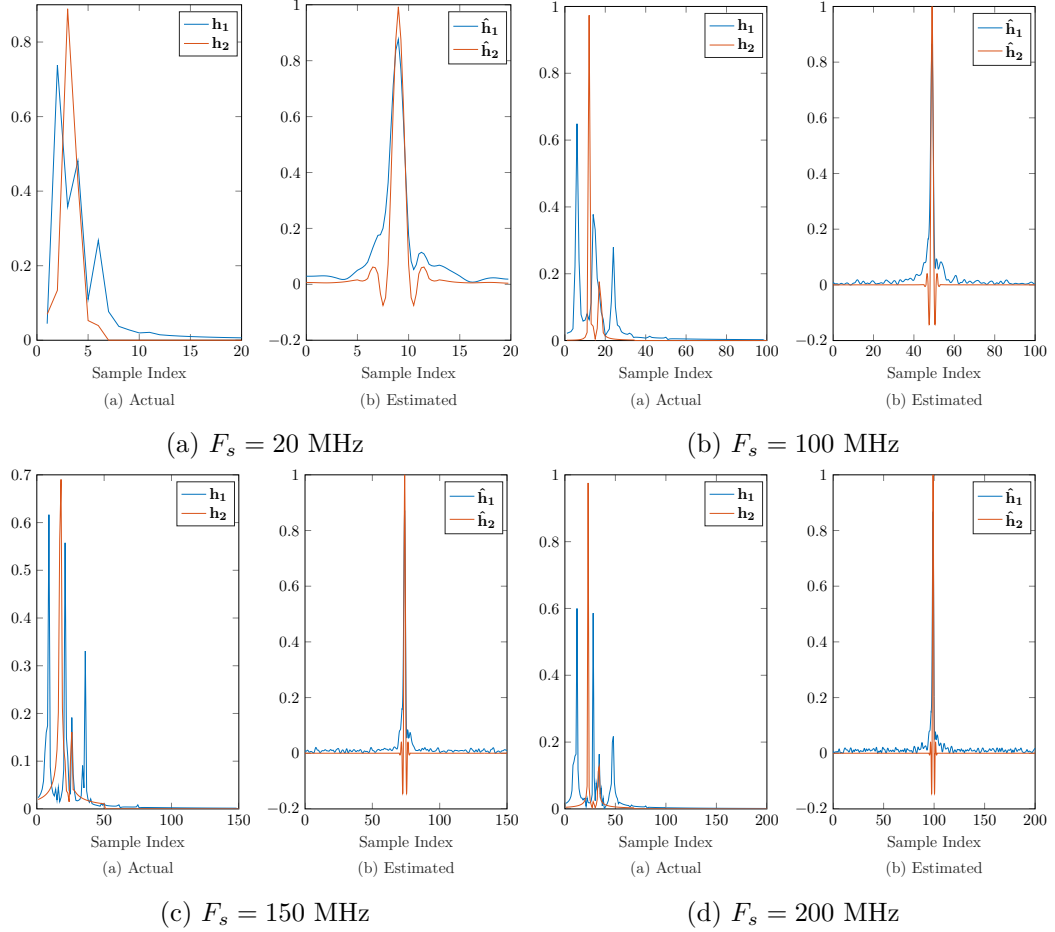


Fig. 4.11: Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

The system is modeled as

$$\begin{cases} x_1(n) = s(n) * h_1^*(n) + w_1(n) \\ \vdots \\ x_M(n) = s(n) * h_M^*(n) + w_M(n), \end{cases} \quad (4.14)$$

where  $*$  denotes convolution,  $h_i^*(n)$  denotes the complex conjugate of each individual channel mapping from the source to the  $i^{\text{th}}$  receiver,  $x_i(n)$  the value received from the transmitted signal at the  $i^{\text{th}}$  receiver, and  $w(n) = [w_1(n) w_2(n) \dots w_M(n)]$  is additive white noise, and  $n$

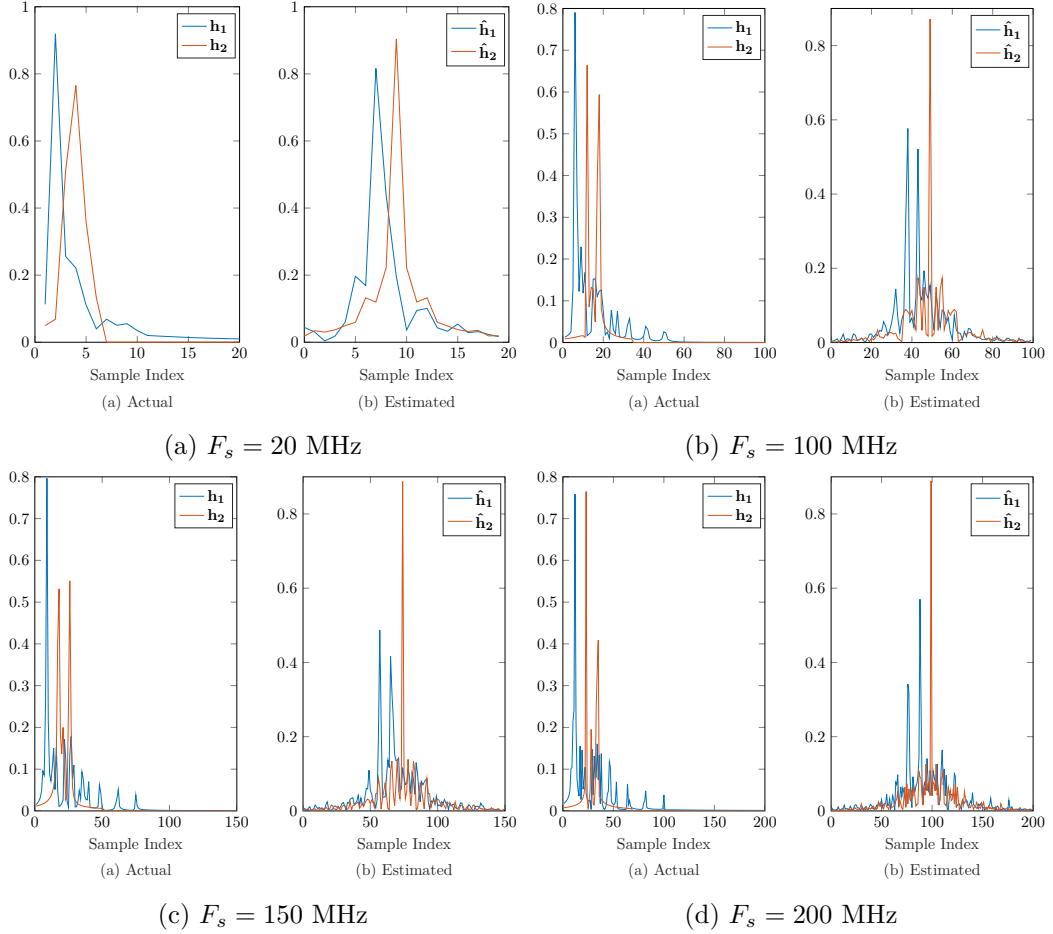


Fig. 4.12: Results of the AED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

denotes the  $n^{\text{th}}$  sample.  $M$  denotes the number of receivers. This model also assumes a finite impulse response and is a SIMO system.

#### 4.2.2 Complex Cross Relation

As done previously, neglecting noise, the signal model, (4.14), can be expressed individually as  $x_i(n) = s(n) * h_i^*(n)$ . Convolution  $x_i(n)$  with  $h_j^*(n)$ , where  $i \neq j$  produces a



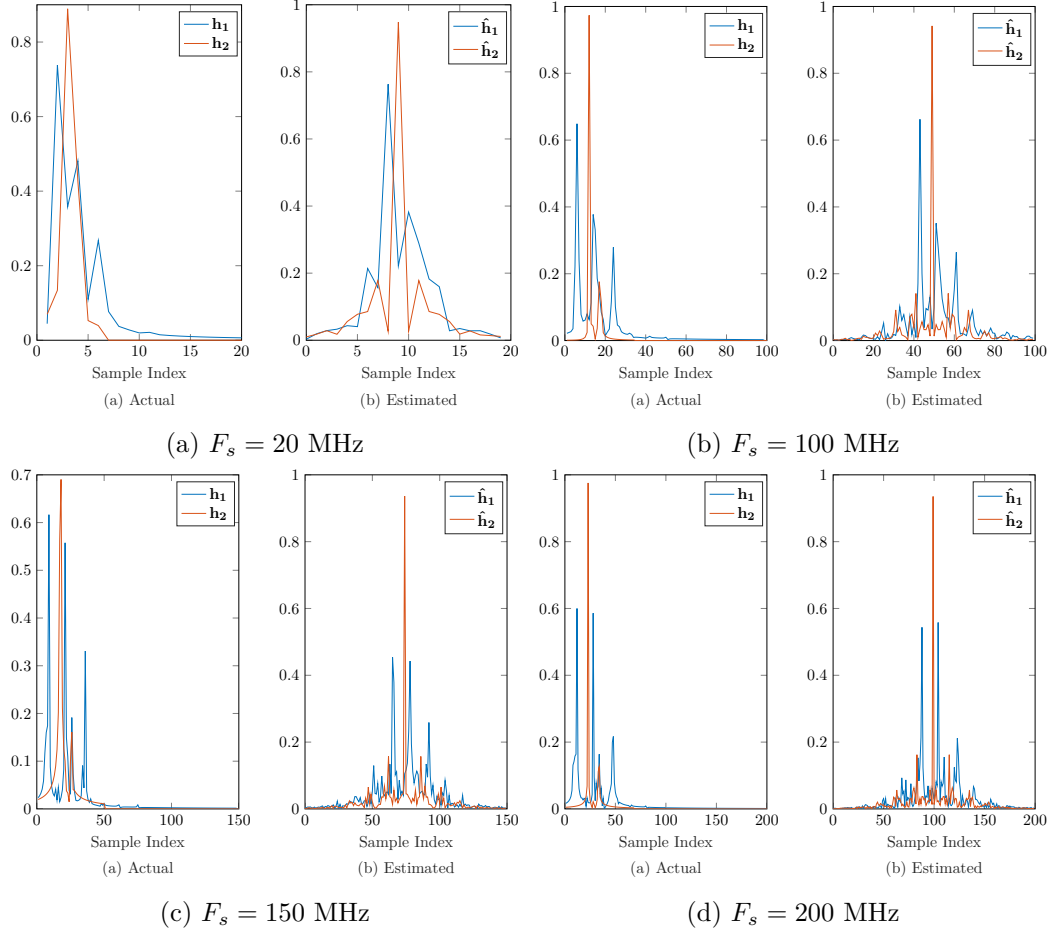


Fig. 4.13: Results of the AED algorithm with 250,000 data samples, with SNR = 20 dB for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

complex CR,

$$h_j^*(n) * x_i(n) = h_j^*(n) * [s(n) * h_i^*(n)] \quad (4.15)$$

$$h_j^*(n) * x_i(n) - h_i^*(n) * x_j(n) = 0.$$

Equation (4.15) can also be expanded for two receiver measurements as the matrix equation<sup>3</sup>

$$\begin{bmatrix} X_i(L) & \vdots & -X_j(L) \end{bmatrix} \begin{bmatrix} \mathbf{h}_j^* \\ \mathbf{h}_i^* \end{bmatrix} = 0, \quad (4.16)$$

<sup>3</sup>For multiple receivers ( $M > 2$ ), Equation (4.16) can be modified as seen in the appendix, A.1.

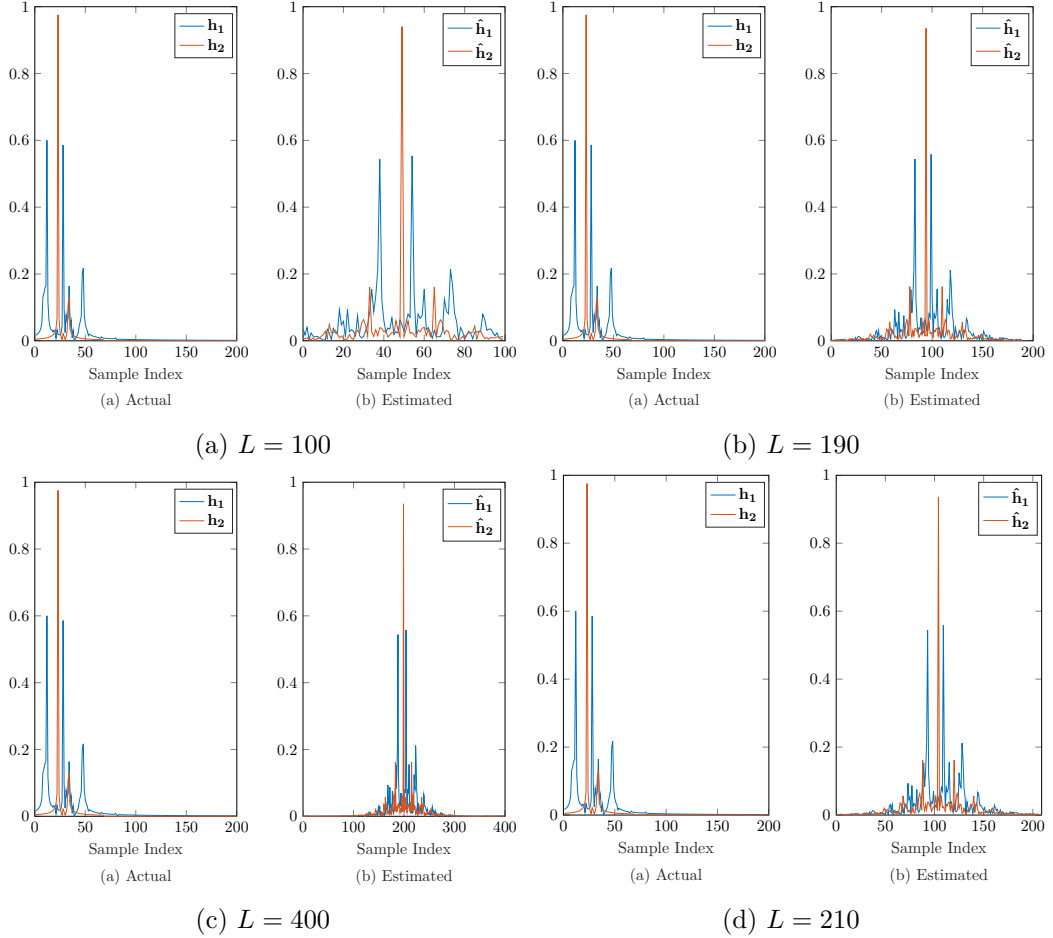


Fig. 4.14: Results of the AED algorithm with 30,000 data samples, with SNR = 60 dB for wideband at 200 MHz sampling rate with incorrect channel orders,  $L$  (a)  $L/2$ , (b)  $L - 10$ , (c)  $L * 2$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

where  $X_m(L)$  is a Toeplitz matrix of the same form described in Section 4.1.2, and  $\mathbf{h}_m^*$  is the impulse response vector

$$\mathbf{h}_m^* \triangleq [h_m(0), \dots, h_m(L)]^H.$$

For brevity, (4.16) can be rewritten as

$$\mathbf{X}\mathbf{h}^* = \mathbf{0}. \quad (4.17)$$

Equation (4.17) is the basis for the approaches derived and explored in this research. As explained in Section 4.1.2, (4.17) implies that  $\mathbf{h}^*$  is in the null space of  $\mathbf{X}$ .

### 4.2.3 Modified Adaptive Eigenvalue Decomposition

The Modified Adaptive Eigenvalue Decomposition (ModAED) also takes an adaptive least-squares approach like, and produces a similar update to the AED algorithm. This algorithm uses a different objective function, using (4.17) as a constraint. The ModAED algorithm focuses on minimizing the change between updates, resembling an error minimization

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0, \end{aligned} \quad (4.18)$$

where  $\mathbf{x}_r$  is a row of the data matrix  $\mathbf{X}$ , transposed, as all vectors are assumed to be column formatted. Equation (4.18) is related to the criterion giving rise to the normalized least-mean-squares (NLMS) algorithm [40, 48].

The update equation which solves this objective function is derived using typical minimization techniques. Initially setting up the Lagrangian based on (4.18) produces

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r. \quad (4.19)$$

The update is then obtained by taking the gradient derivative of (4.19), with respect to the updated transfer function estimate,  $\mathbf{h}^{(n+1)}$ . Then, setting the derivative equal to 0 and solving for the updated transfer function estimate forms

$$\begin{aligned} \frac{d}{d\mathbf{h}^{(n+1)}} \mathcal{L} &= 2\mathbf{h}^{(n+1)} - 2\mathbf{h}^{(n)} + \lambda \mathbf{x}_r \\ \mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \lambda \mathbf{x}_r. \end{aligned} \quad (4.20)$$

The value for  $\lambda$  is found by substituting the results of (4.20) into the constraint listed in (4.9) as

$$\tilde{\lambda} = \frac{\mathbf{h}^{(n)\text{H}} \mathbf{x}_r}{\|\mathbf{x}_r\|_2^2}. \quad (4.21)$$

Note that  $\tilde{\lambda} = \lambda^H$ . Replacing the  $\lambda$  in (4.20) with the conjugate of (4.21) provides the update used in the ModAED algorithm,

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \eta \frac{\mathbf{x}_r^H \mathbf{h}^{(n)}}{\|\mathbf{x}_r\|_2^2} \mathbf{x}_r. \quad (4.22)$$

Note,  $\eta$  in (4.22) was added (as a step size) to make the equation more adaptive, limiting the change between steps.

The complete derivation and the code listing for the ModAED algorithm are found in Appendix A.2, and B.4 respectively.

#### 4.2.4 Modified AED with Sparsity

A sparsity constraint may be helpful to accurately estimate the channels [40–42]. The second algorithm derived during this research implemented the sparsity constraint with the Modified AED algorithm, changing the objective function only slightly to

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)H} \mathbf{x}_r = 0. \end{aligned} \quad (4.23)$$

Sparsity is typically enforced by utilizing an  $L_1$ -norm [13, 41–43, 49].

The update equation is derived, as before, by first setting up the Lagrangian

$$\mathcal{L} = \mathbf{h}^{(n+1)H} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)H} \mathbf{h}^{(n)} - \mathbf{h}^{(n)H} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)H} \mathbf{h}^{(n)} + \alpha \|\mathbf{h}^{(n+1)}\|_1 + \lambda \mathbf{h}^{(n+1)H} \mathbf{x}_r. \quad (4.24)$$

Taking the derivative of (4.24) is more complicated due to the  $L_1$  norm, as the  $L_1$  norm contains a discontinuity at 0,

$$\begin{aligned} \frac{d}{d\mathbf{h}^{(n+1)*}} \|\mathbf{h}^{(n+1)*}\|_1 &= \frac{d}{d\mathbf{h}^{(n+1)*}} m \sum_{i=0}^{M(L+1)-1} |\mathbf{h}_i^{(n+1)}| \\ &= \sum_{i=0}^{M(L+1)-1} \begin{cases} 1 & \mathbf{h}_i^{(n+1)} > 0 \\ -1 & \mathbf{h}_i^{(n+1)} < 0 \end{cases}. \end{aligned} \quad (4.25)$$

It is common practice to use a subgradient approximation [53, 54] in place of the true  $L_1$ -norm derivative to cover all possible cases. The subgradient for the  $L_1$  norm in (4.24) is

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \|\mathbf{h}^{(n+1)*}\|_1 \approx \sum_{i=0}^{M(L+1)-1} \begin{cases} \text{sign}(\mathbf{h}_i^{(n+1)}) & \mathbf{h}_i^{(n+1)} \neq 0 \\ 0 & \mathbf{h}_i^{(n+1)} = 0, \end{cases} \quad (4.26)$$

which encompasses all the cases. As the data is complex, it is more mathematically correct to isolate the phase of the impulse response estimates to determine which direction the coefficients are (positive or negative),  $\text{sign}(\mathbf{h}^{(n+1)}) = \mathbf{e}^{j\angle\mathbf{h}^{(n+1)}}$ ,

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n+1)}} + \lambda \mathbf{x}_r \quad (4.27)$$

An assumption is made in order to compute the transfer function, as only a small change is assumed to happen each iteration,  $\text{sgn}(\mathbf{h}^{(n+1)}) \approx \text{sgn}(\mathbf{h}^{(n)})$ . This assumption allows simply solving for the update,  $\mathbf{h}^{(n+1)}$ , without the complexity of an added sign term.

Using this approximation and assumption of the  $L_1$  norm, the update can be found by setting (4.27) equal to 0, and solving for the update,

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \lambda \mathbf{x}_r. \quad (4.28)$$

The value of  $\lambda$  is found, as before, by substituting the results of (4.28) into the constraint listed in (4.23) resulting in a value similar to (4.21), with the sparsity constraint added in,

$$\lambda = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})^H \mathbf{x}_r}{\|\mathbf{x}_r\|_2^2}. \quad (4.29)$$

Replacing the value for  $\lambda$  in (4.28) with the results in (4.23) produces the update used in the sparse ModAED (ModAEDS) algorithm,

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \eta \frac{(\mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})^H \mathbf{x}_r}{\|\mathbf{x}_r\|_2^2} \mathbf{x}_r. \quad (4.30)$$

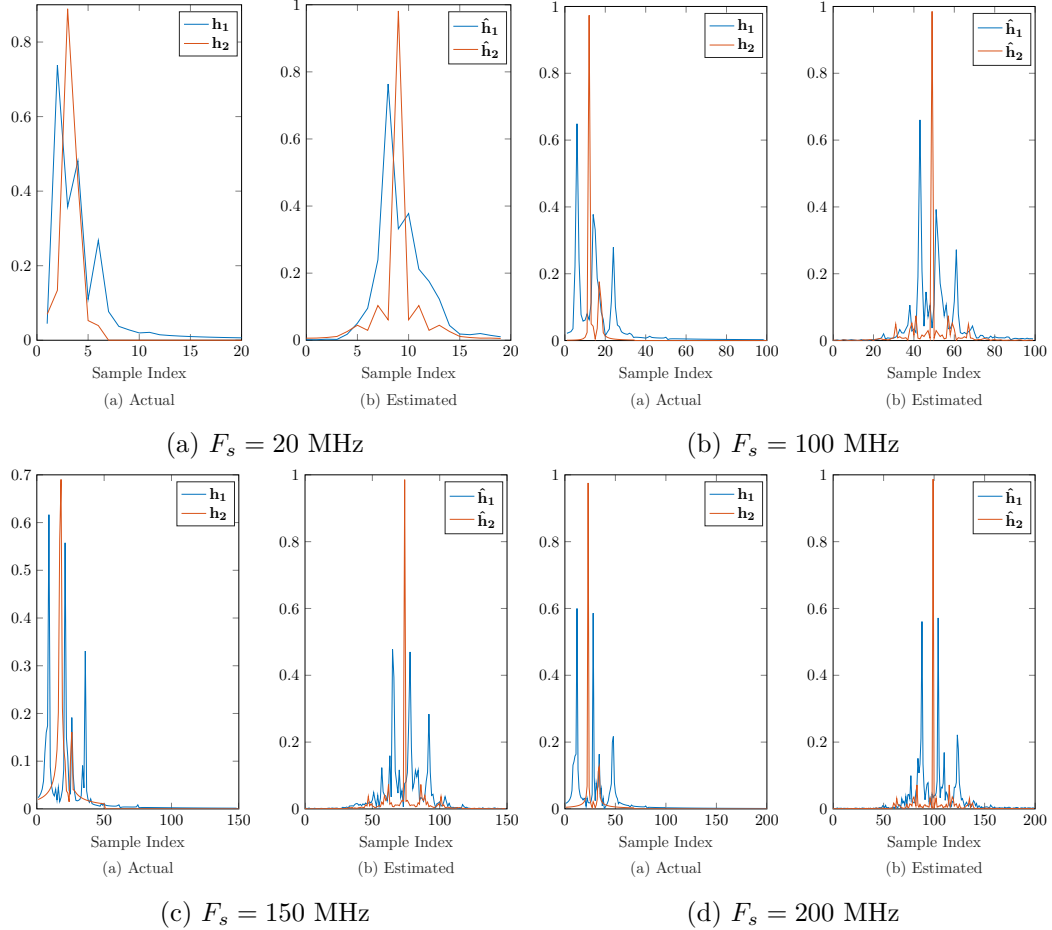


Fig. 4.15: Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

The complete derivation and code listing for the ModAEDS algorithm are found in Appendices A.2 and B.5 respectively.

#### 4.2.5 Modified Algorithms Implementations, Advantages, Disadvantages

As with the AED algorithm, the ModAED and ModAEDS algorithms focus on estimating the time delay between the channels, rather than estimate the channels entirely. These algorithms also estimate the largest peaks (which ideally are the direct paths) of the impulse responses, and as a result produce similar channel estimates.

Like the AED algorithm, these algorithms require (4.13) as the initial start value for

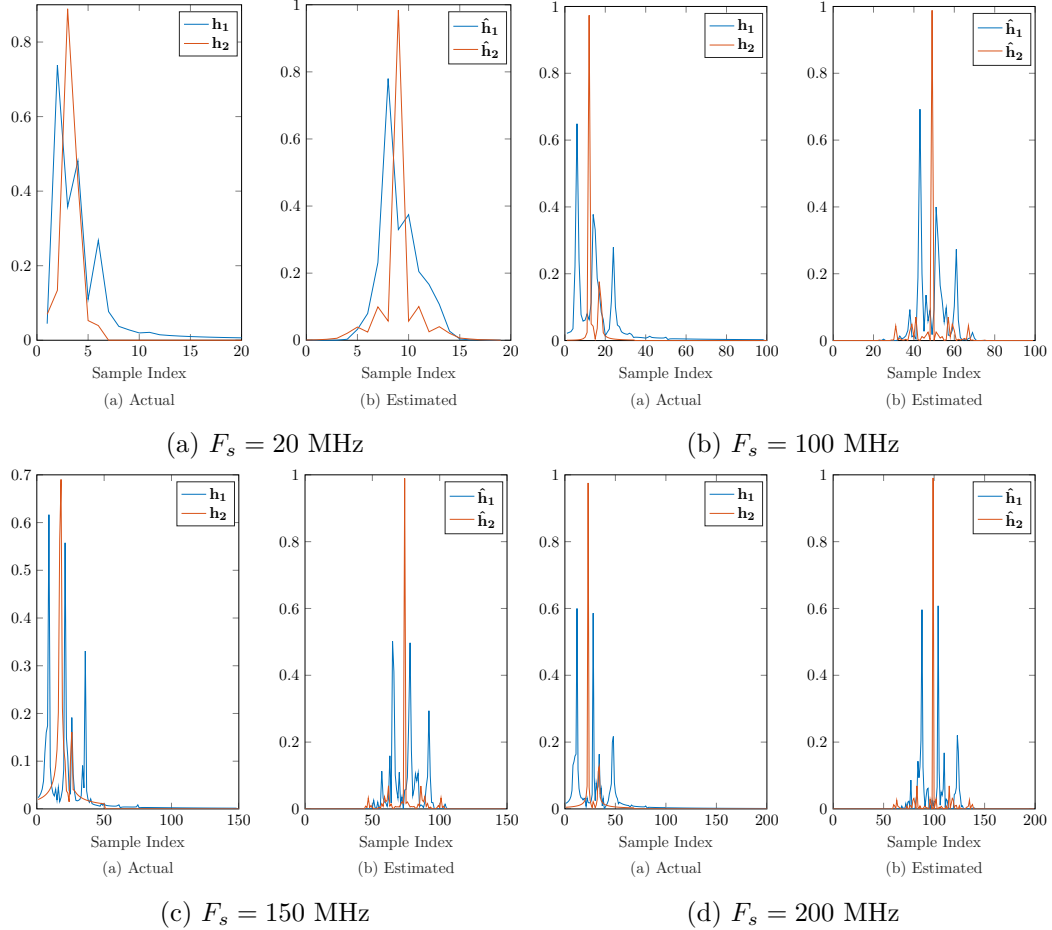


Fig. 4.16: Results of the ModAEDS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

the channel estimates. With the initial estimate,  $\mathbf{h}^{(0)} = \mathbf{e}_d$ , the ModAED and ModAEDS algorithm performs fairly well in ideal conditions as can be seen in Figures 4.15, 4.16, and 4.17. The estimates are centered in the middle of the sample index due to the position of the  $d^{\text{th}}$  index in the initial estimate, and the estimates are not quite as complete as the SVD estimates shown, like the AED algorithm. It should be noted that the ModAED algorithm switches the estimates for the impulse responses for set IR-2, and places them in the reflective position of the  $d^{\text{th}}$  index. These figures also show that there is minimal observable differences between the ModAED and ModAEDS algorithms, with the ModAEDS being slightly more sparse in the smaller peaks.

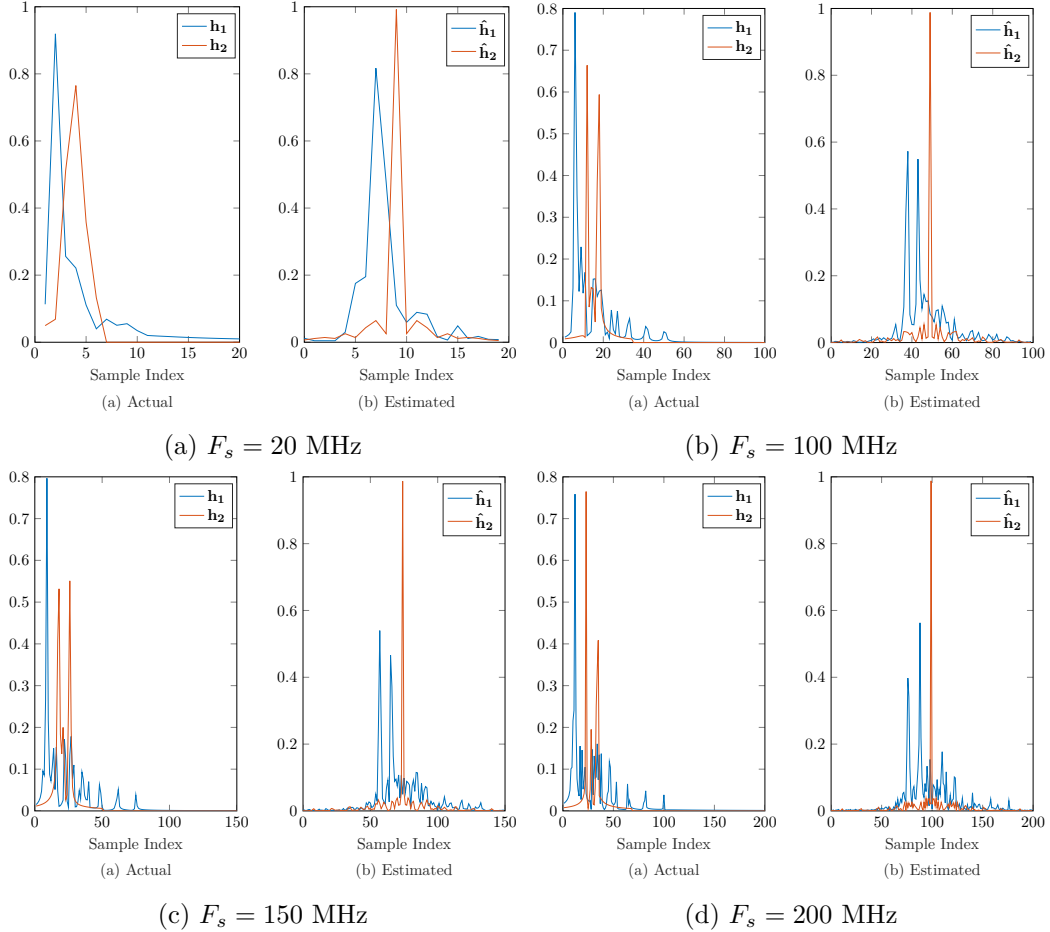


Fig. 4.17: Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

Nonideal conditions were also tested as shown in Figures, 4.18, 4.19, 4.20, and 4.21. Figure 4.18 displays the estimates for narrowband simulation. As with the SVD algorithm, the narrowband signal suffers due to the relationship between the frequency and time domain. Figure 4.19 reflects that more noise does not significantly impact the estimates, . Figure 4.20 shows that considerably fewer samples for computation do not severely alter the algorithm either. The model order,  $L$ , does not need to be known for the impulse response estimates either, as shown in Figure 4.21. Using a normalized random start for the initial channel estimate typically results in a random normalized result with no true single maximum peak as is shown in Figure 4.22.



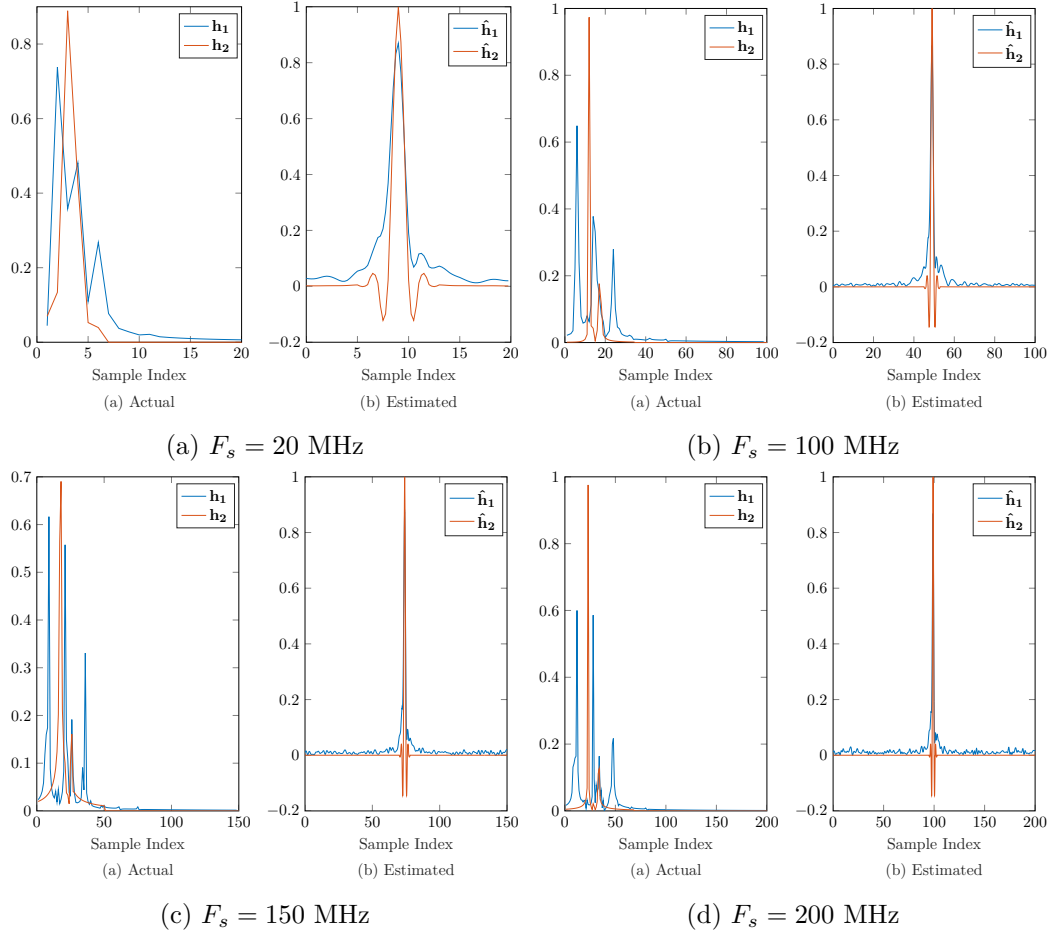


Fig. 4.18: Results of the ModaED algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

There is also not much (if any) difference between the performance of the ideal, and some of the nonideal scenarios (lower SNR, less data samples, and incorrect  $L$ ) tested, and so the individual results for the ModaEDS algorithm are not shown, although Figures 4.23, 4.24 and 4.25 are shown to compare the results for both the ModaED and ModaEDS algorithms. Unfortunately, the narrowband simulation, still performs terribly, but slightly sparser than the results for the ModaED algorithm narrowband simulation shown in Figure 4.18. Note that this method is similar to the one explained in [40].

One advantage that these modified algorithms is that they require less computation. The ModaED and ModaEDS algorithms require only  $(N - L)(6M(L + 1) + 2)$  and  $(N -$

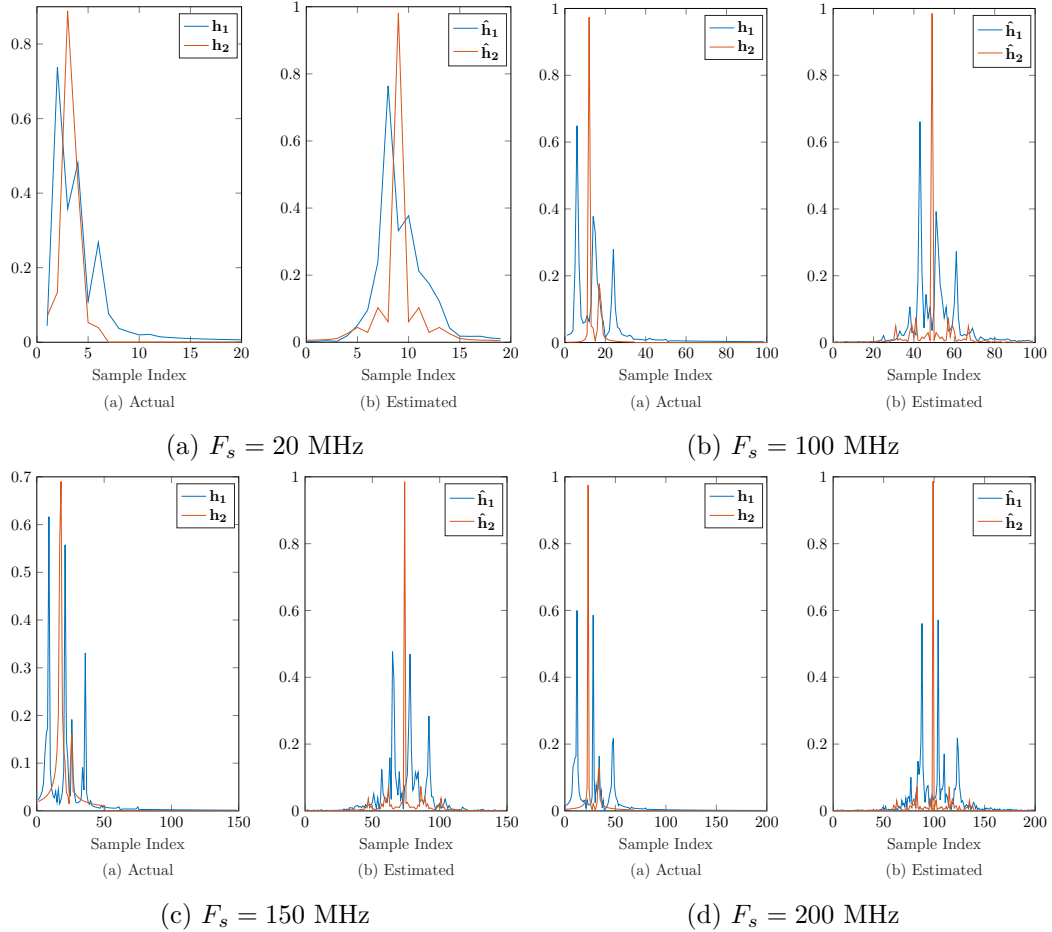


Fig. 4.19: Results of the modaED algorithm with 250,000 data samples, with  $\text{SNR} = 20$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

$L)(8M(L + 1) + 2)$  computations respectively; the ModAEDS also computes a complex exponential for each of the  $N - L$  lines. This is due to the ModAED and ModAEDS algorithm not requiring a secondary normalization.

As explained previously, Tables 4.1–4.9 shows a numerical comparison of the estimates for both of the pre-existing algorithms, and for these modified algorithms introduced. A discussion of the numerical results can be found in in the next section, Section 4.3.

### 4.3 Algorithm Testing and Results

Both sets of algorithms were thoroughly tested in simulation. Due to the complexity

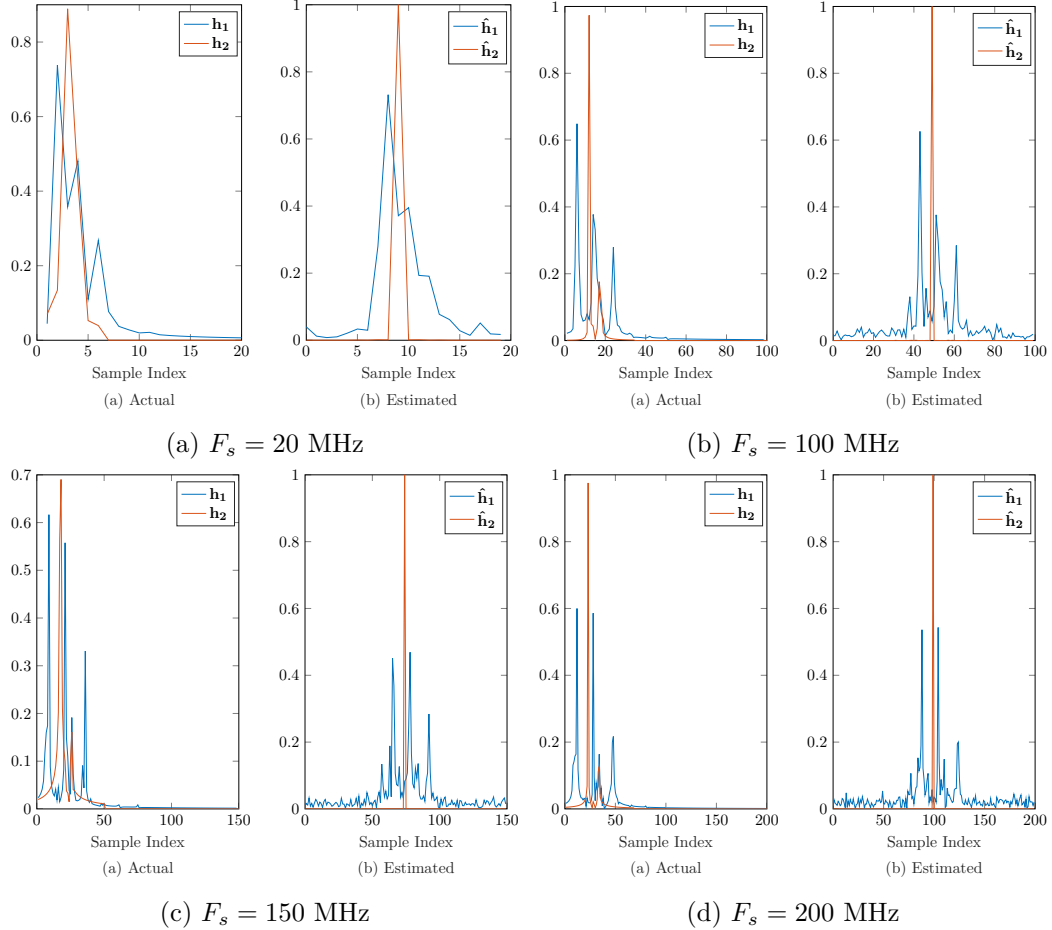


Fig. 4.20: Results of the ModAED algorithm with 2,500 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

involved in computing the SVD, and the finicky nature of the results in the presence of any of the nonideal conditions, the SVD approach was not tested with real-world data. The simulation results are discussed and analyzed here. A discussion of the real-world data can be found in Chapter 7.

### 4.3.1 Simulation

Tables 4.1–4.9 shows a numerical comparison of the simulation estimates for both the pre-existing algorithms, and for the modified algorithms introduced. As explained in Chapter 3, the results shown are averaged over 21 variations of the impulse responses, and

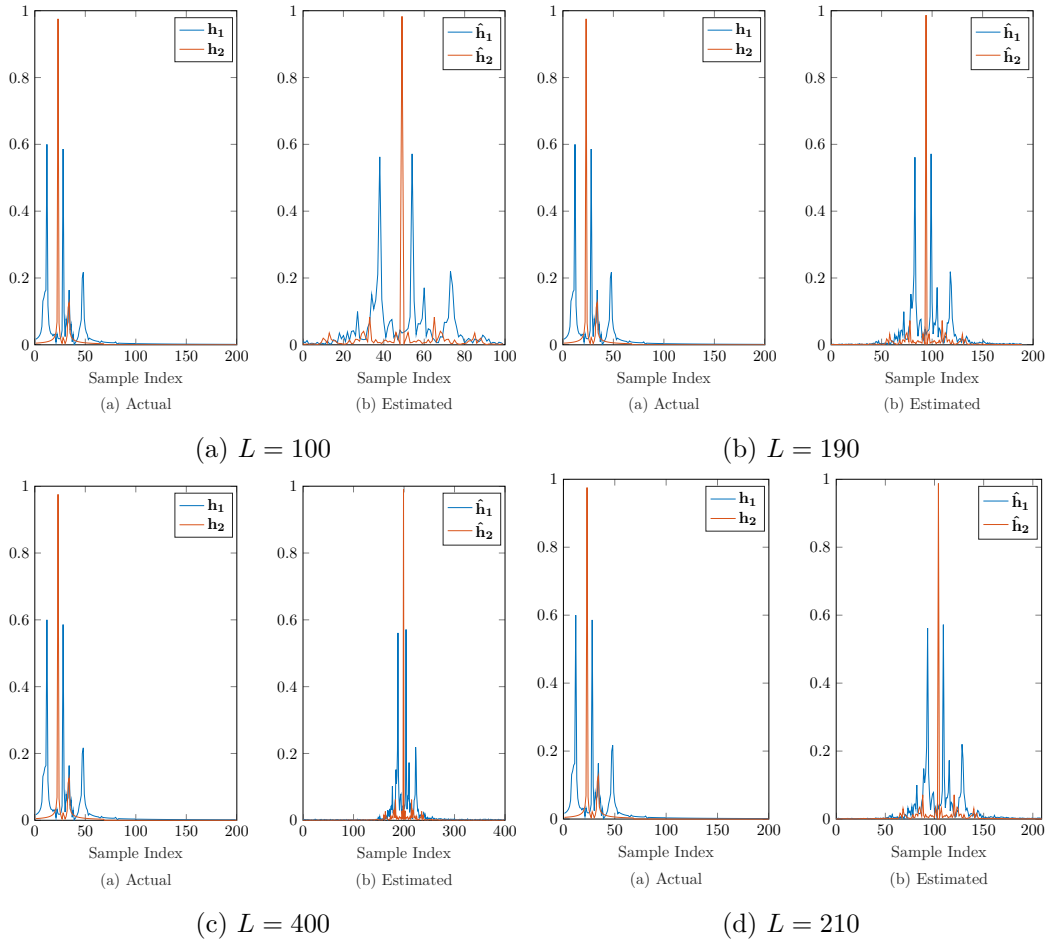


Fig. 4.21: Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect estimates of transfer function length,  $L$ , (a)  $L/2$ , (b)  $L - 10$ , (c)  $L * 2$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

only results for sampling rates  $F_s = 20, 100, 150$ , and 200 MHz are shown in the table for channels 30 and 3000.

Table 4.1 shows the averaged results for all four algorithms in the “Ideal” situation for both impulse response sets. In set IR-2 at 100 MHz for the AED, ModAED, and ModAEDS algorithms show an averaged estimate extremely close to the actual distance of 21.772 meters. The SVD algorithm does much better at the 200 MHz frequency. But other than that, the SVD algorithm typically does poorer at delay estimation than the AED, ModAED, and ModAEDS algorithms. Set IR-1 has less accuracy, and typically has the

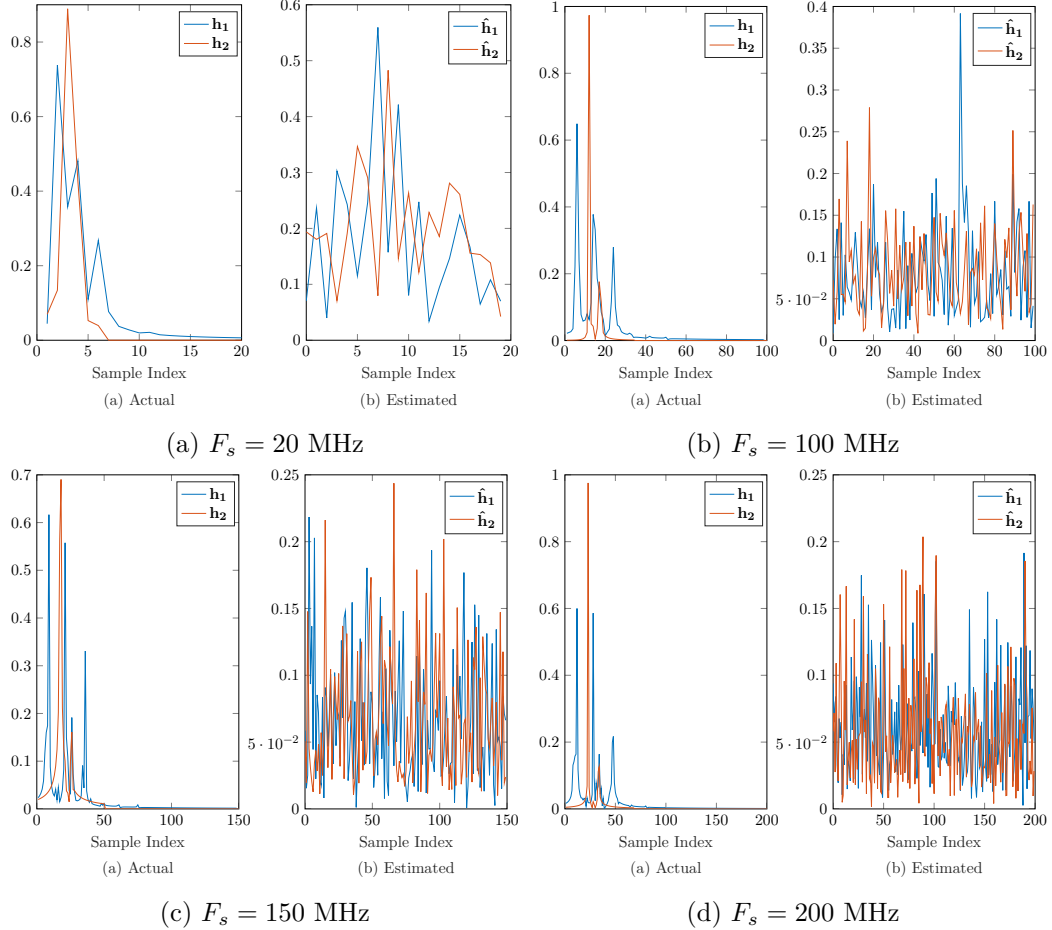


Fig. 4.22: Results of the ModAED algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0, and normalized random initial estimates after 100 iterations.

opposite occurrence, where the SVD typically performs better than the AED, ModAED and ModAEDS algorithms. It is interesting to note that that as the sampling rate increases, there appears to be a correlation in accuracy. Another interesting observance is that the ModAED and ModAEDS algorithms almost consistently give the same estimates, implying that the sparsity constraint may be unnecessary.

Table 4.2 show the averaged results for all four algorithms in the narrowband signal bandwidth situation for both impulse response sets. Set IR-2 typically performs better than set IR-1 once again, however as observed in Figures 4.5, 4.11, and 4.18, there is not much resolution that occurs between channels, and so it might just be random chance. The

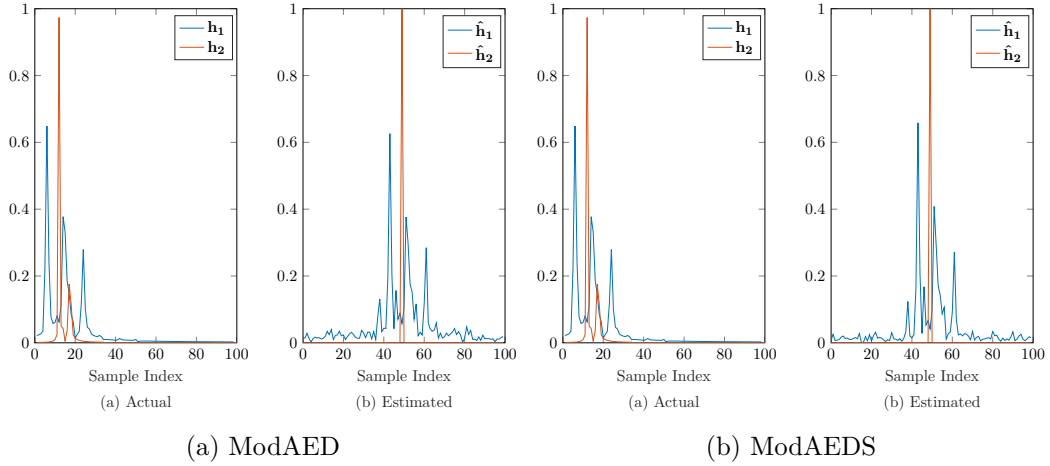


Fig. 4.23: Results of the ModAED and ModAEDS algorithms with 2,500 data samples, with SNR = 60 dB, for wideband simulation at 100 MHz sampling rate for impulse response set IR-1, seed 0.

SVD algorithm does appear to estimate the channel delay fairly accurately for the lower frequency sampling rates (especially for set IR-1), however, for set IR-2 there are much more wild estimates, and even some nonexistent estimates (such as NaN).

In Table 4.3, results for the noisy situations can be seen for both impulse responses. The interesting thing about this situation is that all of the algorithms, except for the SVD show results that are exactly the same as the ideal situation shown in Table 4.1. The SVD algorithm does not have as large of estimates as for the narrowband situation, starts to have larger estimates, and even NaN estimates. Table 4.4 shows results for the smaller data size situation for both impulse responses. It also shows a similar trend to what is shown in Table 4.3. Although the values algorithm estimates aren't exactly the same estimates as the ideal case, they are close, except for the SVD approach.

Tables 4.5–4.8 show the effects that model order has on the algorithms. Underestimating the model order does not appear to affect the adaptive algorithms much. It has a much more drastic effect on the SVD algorithm. It should be noted that for the 20 MHz cases, the NaNs appear for the  $L/2$  case and the  $L - 10$  case for set IR-1 potentially because there just isn't enough room for the peaks to show (as they are the same model order estimate). Overestimating the model order also does not have much of an effect on the

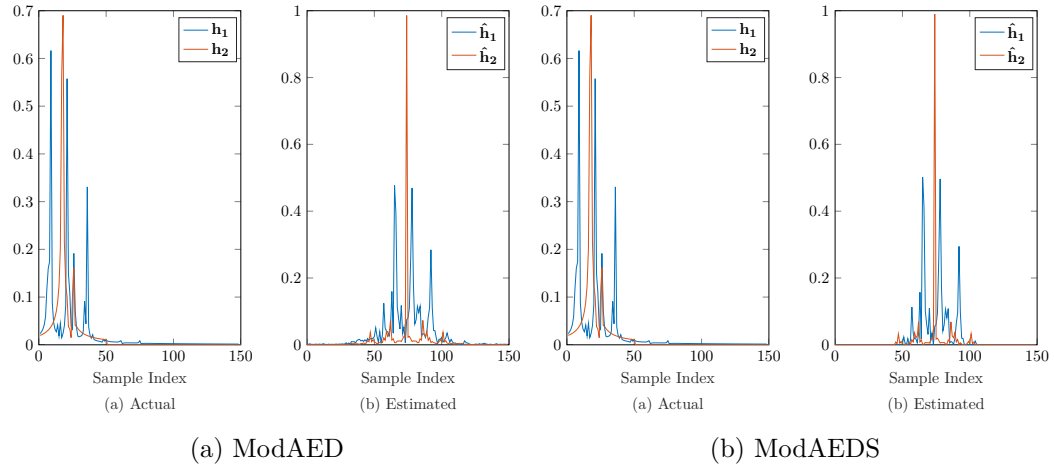


Fig. 4.24: Results of the ModAED and ModAEDS algorithms with 250,000 data samples, with SNR = 20 dB, for wideband simulation at 150 MHz sampling rate for impulse response set IR-1, seed 0.

adaptive algorithms, but does on the SVD algorithm.

The SVD algorithm does not utilize an initial estimate for the impulse response. Thus, it is not included in the last table, Table 4.9. And as can be seen a normalized random start does not provide good estimates.

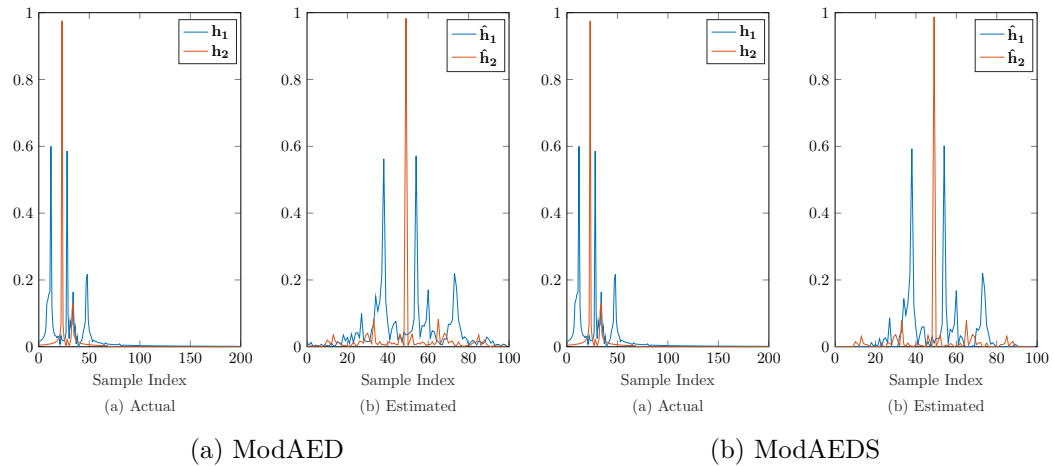


Fig. 4.25: Results of the ModAED and ModAEDS algorithms with 250,000 data samples, with SNR = 60 dB, for wideband simulation at 100 MHz sampling rate for incorrect estimates of model order,  $L = L/2$ , for impulse response set IR-1, seed 0.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	20 MHz	SVD	21.772	15	15	15	15	6.772
		AED		15	0	15	30	6.772
		ModAED		14.286	0	15	30	7.486
		ModAEDS		14.286	0	15	30	7.486
	100 MHz	SVD	21.772	15.286	3	18	24	6.486
		AED		15.429	6	18	36	6.344
		ModAED		16.857	6	18	36	4.915
		ModAEDS		16.857	6	18	36	4.915
	150 MHz	SVD	21.772	17.905	2	18	44	3.867
		AED		17.143	8	18	36	4.629
		ModAED		16.667	8	18	36	5.106
		ModAEDS		16.667	8	18	36	5.106
200 MHz	SVD	21.772	16.786	4.5	18	37.5	4.986	
	AED		12.929	7.5	15	19.5	8.844	
	ModAED		13.5	7.5	15	19.5	8.272	
	ModAEDS		14.071	7.5	16.5	19.5	7.701	
IR-2: Ideal	20 MHz	SVD	21.772	17.857	0	15	30	3.915
		AED		18.571	0	15	30	3.201
		ModAED		18.571	0	15	36	3.201
		ModAEDS		18.571	0	15	36	3.201
	100 MHz	SVD	21.772	17.286	3	18	30	4.486
		AED		21.714	6	24	33	0.058
		ModAED		21.857	6	24	33	0.085
		ModAEDS		21.857	6	24	33	0.085
	150 MHz	SVD	21.772	16.381	0	18	24	5.391
		AED		18.191	4	18	34	3.582
		ModAED		19.810	4	20	34	1.963
		ModAEDS		19.810	4	20	34	1.963
200 MHz	SVD	21.772	21.357	4.5	19.5	64.5	0.415	
	AED		19.786	4.5	22.5	34.5	1.986	
	ModAED		19.357	4.5	19.5	34.5	2.415	
	ModAEDS		19.357	4.5	19.5	34.5	2.415	

Table 4.1: Numerical results for pre-existing and modified algorithms in the “Ideal” case.



Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Narrowband	20 MHz	SVD	21.772	14.286	0	0	60	7.486
		AED		2.857	0	0	60	18.915
	100 MHz	ModAED		2.857	0	0	60	18.915
		ModAEDS		2.857	0	0	60	18.915
IR-1: Narrowband	100 MHz	SVD	21.772	25.714	0	0	120	3.942
		AED		5.714	0	0	60	16.058
	150 MHz	ModAED		5.714	0	0	60	16.0579
		ModAEDS		5.714	0	0	60	16.0579
IR-1: Narrowband	150 MHz	SVD	21.772	22.857	0	0	180	1.085
		AED		2.857	0	0	60	18.915
	200 MHz	ModAED		2.857	0	0	60	18.915
		ModAEDS		2.857	0	0	60	18.915
IR-2: Narrowband	20 MHz	SVD	21.772	17.143	0	0	180	4.629
		AED		2.857	0	0	60	18.915
	100 MHz	ModAED		2.857	0	0	60	18.915
		ModAEDS		5.714	0	0	60	16.058
IR-2: Narrowband	20 MHz	SVD	21.772	22.857	0	0	60	1.085
		AED		17.143	0	0	60	4.629
	100 MHz	ModAED		17.143	0	0	60	4.629
		ModAEDS		17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	SVD	21.772	245.714	0	0	2880	223.942
		AED		17.143	0	0	60	4.629
	150 MHz	ModAED		17.143	0	0	60	4.629
		ModAEDS		17.143	0	0	60	4.629
IR-2: Narrowband	150 MHz	SVD	21.772	NaN	0	NaN	120	NaN
		AED		14.286	0	0	60	7.486
	200 MHz	ModAED		14.286	0	0	60	7.486
		ModAEDS		14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	SVD	21.772	NaN	0	NaN	1620	NaN
		AED		14.286	0	0	60	7.486
	200 MHz	ModAED		14.286	0	0	60	7.486
		ModAEDS		14.286	0	0	60	7.486

Table 4.2: Numerical results for pre-existing and modified algorithms in the narrowband case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{\mu_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\epsilon_{\mu_{1,2}}$
IR-1: Noisy	20 MHz	SVD	21.772	17.143	0	15	45	4.629
		AED		15	0	15	30	6.772
		ModAED		14.286	0	15	30	7.486
		ModAEDS		14.286	0	15	30	7.486
	100 MHz	SVD	21.772	22.143	6	18	78	0.371
		AED		15.429	6	18	36	6.344
		ModAED		16.857	6	18	36	4.915
		ModAEDS		16.857	6	18	36	4.915
	150 MHz	SVD	21.772	61.333	4	40	170	39.561
		AED		17.143	8	18	36	4.629
		ModAED		16.667	8	18	36	5.106
		ModAEDS		16.667	8	18	36	5.106
200 MHz	SVD	21.772	NaN	6	NaN	160.5	NaN	
	AED		12.929	7.5	15	19.5	8.844	
	ModAED		13.5	7.5	15	19.5	8.272	
	ModAEDS		14.071	7.5	16.5	19.5	7.701	
IR-2: Noisy	20 MHz	SVD	21.772	27.143	0	30	75	5.371
		AED		18.571	0	15	30	3.201
		ModAED		18.571	0	15	30	3.201
		ModAEDS		18.571	0	15	30	3.201
	100 MHz	SVD	21.772	36.571	0	27	99	14.799
		AED		21.714	6	24	33	0.058
		ModAED		21.857	6	24	33	0.085
		ModAEDS		21.857	6	24	33	0.085
	150 MHz	SVD	21.772	35.429	4	20	136	13.656
		AED		18.191	4	18	34	3.582
		ModAED		19.810	4	20	34	1.611
		ModAEDS		19.810	4	20	34	1.963
200 MHz	SVD	21.772	NaN	3	NaN	160.5	NaN	
	AED		19.786	4.5	22.5	34.5	1.986	
	ModAED		19.357	4.5	19.5	34.5	2.415	
	ModAEDS		19.357	4.5	19.5	34.5	2.415	

Table 4.3: Numerical results for pre-existing and modified algorithms in the noisy case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	20 MHz	SVD	21.772	15	15	15	15	6.772
		AED		14.286	0	15	30	7.486
		ModAED		14.286	0	15	30	7.486
		ModAEDS		14.286	0	15	30	7.486
	100 MHz	SVD	21.772	16.143	3	18	36	5.629
		AED		16.857	6	18	36	4.915
		ModAED		17.429	6	18	36	4.344
		ModAEDS		17.429	6	18	36	4.344
	150 MHz	SVD	21.772	41.143	2	12	196	19.371
		AED		15.429	8	18	20	6.344
		ModAED		16.191	8	18	36	5.582
		ModAEDS		15.429	8	18	20	6.344
200 MHz	SVD	21.772	172.357	7.5	190.5	250.5	150.585	
	AED		14.429	7.5	16.5	19.5	7.346	
	ModAED		14.426	7.5	16.5	19.5	7.346	
	ModAEDS		14.429	7.5	16.5	19.5	7.344	
IR-2: Fewer Samples	20 MHz	SVD	21.772	17.857	0	15	30	3.915
		AED		19.286	0	15	30	2.486
		ModAED		19.286	0	15	30	2.486
		ModAEDS		19.286	0	15	30	2.486
	100 MHz	SVD	21.772	20	3	18	54	1.772
		AED		20.857	3	24	33	0.915
		ModAED		20.857	3	24	33	0.915
		ModAEDS		20.857	3	24	33	0.915
	150 MHz	SVD	21.772	77.429	0	50	242	55.656
		AED		18.952	4	18	34	2.820
		ModAED		18.952	4	18	34	2.820
		ModAEDS		18.952	4	18	34	2.820
200 MHz	SVD	21.772	172.357	7.5	190.5	250.5	150.585	
	AED		19.143	4.5	16.5	34.5	2.629	
	ModAED		19.143	4.5	16.5	34.5	2.629	
	ModAEDS		19.143	4.5	16.5	34.5	2.629	

Table 4.4: Numerical results for pre-existing and modified algorithms in the smaller data size case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	20 MHz	SVD	21.772	NaN	0	NaN	30	NaN
		AED		NaN	0	NaN	30	NaN
		ModAED		NaN	0	NaN	30	NaN
		ModAEDS		NaN	0	NaN	30	NaN
	100 MHz	SVD	21.772	12.714	3	9	33	9.058
		AED		15.429	6	18	36	6.344
		ModAED		16.857	6	18	36	4.915
		ModAEDS		16.857	6	18	36	4.915
	150 MHz	SVD	21.772	12.095	2	8	38	9.677
		AED		17.142	8	18	36	4.629
		ModAED		16.667	8	18	36	5.106
		ModAEDS		16.667	8	18	36	5.106
200 MHz	SVD	21.772	12.214	4.5	9	28.5	9.558	
	AED		12.929	7.5	15	19.5	8.844	
	ModAED		13.5	7.5	15	19.5	8.272	
	ModAEDS		14.071	7.5	16.5	19.5	7.701	
IR-2: L/2	20 MHz	SVD	21.772	23.571	0	15	45	1.799
		AED		18.571	0	15	30	3.201
		ModAED		18.571	0	15	30	3.201
		ModAEDS		18.571	0	15	30	3.201
	100 MHz	SVD	21.772	15.714	0	15	36	6.058
		AED		21.714	6	24	33	0.058
		ModAED		22.286	6	24	33	0.514
		ModAEDS		22.286	6	24	33	0.514
	150 MHz	SVD	21.772	16.667	4	16	32	5.105
		AED		18.191	4	18	34	3.582
		ModAED		19.810	4	20	34	1.963
		ModAEDS		19.810	4	20	34	1.963
200 MHz	SVD	21.772	14.643	0	16.5	27	7.129	
	AED		20	4.5	22.5	34.5	1.772	
	ModAED		19.643	4.5	19.5	34.5	2.129	
	ModAEDS		19.357	4.5	19.5	34.5	2.415	

Table 4.5: Numerical results for pre-existing and modified algorithms in the wrong model order,  $L/2$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	20 MHz	SVD	21.772	NaN	0	NaN	30	NaN
		AED		NaN	0	NaN	30	NaN
		ModAED		NaN	0	NaN	30	NaN
		ModAEDS		NaN	0	NaN	30	NaN
	100 MHz	SVD	21.772	16.143	6	18	36	5.629
		AED		15.429	6	18	36	6.344
		ModAED		16.857	6	18	36	4.915
		ModAEDS		16.857	6	18	36	4.915
	150 MHz	SVD	21.772	15.524	6	18	36	6.248
		AED		17.143	8	18	36	4.629
		ModAED		16.667	8	18	36	5.106
		ModAEDS		16.667	8	18	36	5.106
200 MHz	SVD	21.772	14.214	6	16.5	36	7.558	
	AED		12.929	7.5	15	19.5	8.844	
	ModAED		14.071	7.5	16.5	19.5	7.701	
	ModAEDS		14.071	7.5	16.5	19.5	7.701	
TF-2: L-10	20 MHz	SVD	21.772	23.571	0	15	45	1.799
		AED		18.571	0	15	30	3.201
		ModAED		18.571	0	15	30	3.201
		ModAEDS		18.571	0	15	30	3.201
	100 MHz	SVD	21.772	16.143	6	18	36	5.629
		AED		21.714	6	24	33	0.058
		ModAED		21.857	6	24	33	0.085
		ModAEDS		21.857	6	24	33	0.085
	150 MHz	SVD	21.772	15.905	4	16	34	5.867
		AED		18.191	4	18	34	3.582
		ModAED		19.810	4	20	34	1.963
		ModAEDS		19.810	4	20	34	1.963
200 MHz	SVD	21.772	12.643	0	15	28.5	9.129	
	AED		19.786	4.5	22.5	34.5	1.986	
	ModAED		19.357	4.5	19.5	34.5	2.415	
	ModAEDS		19.357	4.5	19.5	34.5	2.415	

Table 4.6: Numerical results for pre-existing and modified algorithms in the wrong model order,  $L = 10$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: 2L	20 MHz	SVD	21.772	90.714	0	90	255	68.942
		AED		15	0	15	30	6.772
		ModAED		14.286	0	15	30	7.486
		ModAEDS		14.285	0	15	30	7.486
	100 MHz	SVD	21.772	84	3	63	279	62.228
		AED		15.429	6	18	36	6.344
		ModAED		16.857	6	18	36	4.915
		ModAEDS		16.857	6	18	36	4.915
	150 MHz	SVD	21.772	NaN	2	NaN	256	NaN
		AED		17.143	8	18	36	4.629
		ModAED		16.191	8	18	36	5.582
		ModAEDS		16.191	8	18	36	5.582
200 MHz	SVD	21.772	NaN	12	NaN	117	NaN	
	AED		12.929	7.5	15	19.5	8.844	
	ModAED		14.071	7.5	16.5	19.5	7.701	
	ModAEDS		14.071	7.5	16.5	19.5	7.701	
IR-2: 2L	20 MHz	SVD	21.772	124.286	0	60	285	102.514
		AED		18.571	0	15	30	3.201
		ModAED		18.571	0	15	30	3.201
		ModAEDS		18.571	0	15	30	3.201
	100 MHz	SVD	21.772	109.429	3	114	264	87.656
		AED		21.714	6	24	33	0.058
		ModAED		21.857	6	24	33	0.085
		ModAEDS		21.857	6	24	33	0.085
	150 MHz	SVD	21.772	NaN	16	NaN	294	NaN
		AED		18.191	4	18	34	3.582
		ModAED		19.810	4	20	34	1.963
		ModAEDS		19.810	4	20	34	1.963
200 MHz	SVD	21.772	NaN	37.5	NaN	279	NaN	
	AED		19.786	4.5	22.5	34.5	1.986	
	ModAED		19.357	4.5	19.5	34.5	2.415	
	ModAEDS		19.357	4.5	19.5	34.5	2.415	

Table 4.7: Numerical results for pre-existing and modified algorithms in the wrong model order,  $2L$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L+10	20 MHz	SVD	21.772	31.429	0	0	15	120
		AED		15	0	0	15	30
	100 MHz	ModAED		14.286		0	15	30
		ModAEDS		14.286		0	15	30
	150 MHz	SVD	21.772	13	0	0	9	42
		AED		15.429	6	6	18	36
ModAED			16.857	6	6	18	36	
ModAEDS			16.857	6	6	18	36	
200 MHz	SVD	21.772	16.571	0	0	18	36	
	AED		17.143	8	8	18	36	
	ModAED		16.667	8	8	18	36	
	ModAEDS		16.667	8	8	18	36	
IR-2: L+10	20 MHz	SVD	21.772	16.286	1.5	1.5	16.5	33
		AED		12.929	7.5	7.5	15	19.5
	100 MHz	ModAED		13.5	7.5	7.5	15	19.5
		ModAEDS		14.071	7.5	7.5	16.5	19.5
	150 MHz	SVD	21.772	78.571	0	0	60	165
		AED		18.571	0	0	15	30
ModAED			18.571	0	0	15	30	
ModAEDS			18.571	0	0	15	30	
200 MHz	SVD	21.772	20.714	0	0	18	45	
	AED		21.714	6	6	24	33	
	ModAED		21.857	6	6	24	33	
	ModAEDS		21.857	6	6	24	33	
IR-2: L+10	20 MHz	SVD	21.772	20.476	2	2	20	38
		AED		18.191	4	4	18	34
	100 MHz	ModAED		19.810	4	4	20	34
		ModAEDS		19.810	4	4	20	34
	150 MHz	SVD	21.772	22.643	3	3	22.5	51
		AED		19.786	4.5	4.5	22.5	34.5
200 MHz	ModAED		19.357	4.5	4.5	19.5	34.5	
	ModAEDS		19.357	4.5	4.5	19.5	34.5	

Table 4.8: Numerical results for pre-existing and modified algorithms in the wrong model order,  $L + 10$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{mini_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	20 MHz	AED	21.772	125.714	0	135	300	103.942
		ModAED		189.286	15	195	405	167.514
		ModAEDS		189.286	15	195	405	167.514
100 MHz	AED	AED	21.772	64.286	6	30	258	42.514
		ModAED		138.714	3	135	321	116.942
		ModAEDS		149.571	3	147	321	127.799
150 MHz	AED	AED	21.772	50.667	0	18	202	28.895
		ModAED		122.952	6	126	258	101.180
		ModAEDS		119.333	6	112	258	97.561
200 MHz	AED	AED	21.772	71.143	9	66	201	49.371
		ModAED		85.714	0	69	295.5	63.942
		ModAEDS		85.714	0	69	295.5	63.942
IR-2: Random Start	20 MHz	AED	21.772	90.714	0	75	285	68.942
		ModAED		172.143	15	180	435	150.371
		ModAEDS		172.143	15	180	435	150.371
100 MHz	AED	AED	21.772	72.143	6	42	186	50.371
		ModAED		121	12	108	246	99.228
		ModAEDS		118.857	12	108	246	97.085
150 MHz	AED	AED	21.772	91.333	10	94	226	69.561
		ModAED		109.714	10	92	228	87.942
		ModAEDS		120.476	10	92	282	98.704
200 MHz	AED	AED	21.772	105.857	6	115.5	234	84.085
		ModAED		117.571	9	84	280.5	95.799
		ModAEDS		114.143	9	84	280.5	92.371

Table 4.9: Numerical results for pre-existing and modified algorithms in the normalized random start case.



CHAPTER 5  
THE ADAPTIVE CROSS-CHANNEL WITH SPARSE SHIFT-SUPPRESSION (AXIS)  
METHOD

To combat the possibility of spurious peaks that can result in time-shifted versions of the channel impulse response, an anchor constraint was added to the optimization equation

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \\ & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = 1. \end{aligned} \quad (5.1)$$

The term  $\mathbf{e}_d$  is a unit vector of the form seen in Definition (4.13), where the  $d^{\text{th}}$  element is equal to 1. The update equation is derived similarly to the previous Modified Adaptive Eigenvalue Decomposition (ModAED) algorithm and the ModAED with Sparsity (ModAEDS) algorithm as explained in the previous chapter.

### 5.1 AXIS Derivation

The update is derived from the Lagrangian of the objective equations, (5.1)

$$\begin{aligned} \mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} \\ + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - 1). \end{aligned} \quad (5.2)$$

The derivative is taken with respect to the complex conjugate of the new channel estimate, using the same assumptions explained in Section 4.2,

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d. \quad (5.3)$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d. \quad (5.4)$$

Setting the results of (5.3) to 0 and solving for the transfer function estimate,  $\mathbf{h}^{(n+1)}$  forms (5.4). Replacing  $\mathbf{h}^{(n+1)}$  in the constraints listed in (5.1) with (5.4) allows a system of equations to be set up and the Lagrange multipliers to be solved for simultaneously.

$$\begin{aligned} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r &= \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - 1 &= \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_d \end{aligned} \quad (5.5)$$

$$\begin{bmatrix} \mathbf{x}_r^{\text{H}} \mathbf{x}_r & \mathbf{e}_d^{\text{H}} \mathbf{x}_r \\ \mathbf{x}_r^{\text{H}} \mathbf{e}_d & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - 1 \end{bmatrix} \quad (5.6)$$

$$\begin{aligned} \tilde{\lambda} &= \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r - \mathbf{e}_d^{\text{H}} \mathbf{x}_r)}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \\ \tilde{\mu} &= \frac{\mathbf{x}_r^{\text{H}} \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - \mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \end{aligned} \quad (5.7)$$

Note that  $\tilde{\lambda} = \lambda^{\text{H}}$  and  $\tilde{\mu} = \mu^{\text{H}}$ . The variable  $\epsilon$  has been added to guarantee that the fractional values never divide by zero by adding a small variation to the denominator. Conjugating (5.7), and replacing  $\lambda$  and  $\mu$  respectively in (5.4) supplies the complete update

$$\begin{aligned} \mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\ &\eta \left( \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r - \mathbf{e}_d^{\text{H}} \mathbf{x}_r)}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r - \right. \\ &\quad \left. \left( \frac{\mathbf{x}_r^{\text{H}} \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - \mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \right). \end{aligned} \quad (5.8)$$

This update is referred to as the Adaptive Cross-Channel with Sparse Shift-Suppression (AXIS) algorithm.

The complete derivation of the AXIS algorithm is found in Appendix A.2. The code listing for this algorithm can be found in Appendix B.6.

### 5.1.1 Algorithm Advantages, Implementations, and Modifications

As the algorithm is adaptive, the estimate is made as it sweeps through the data. Note

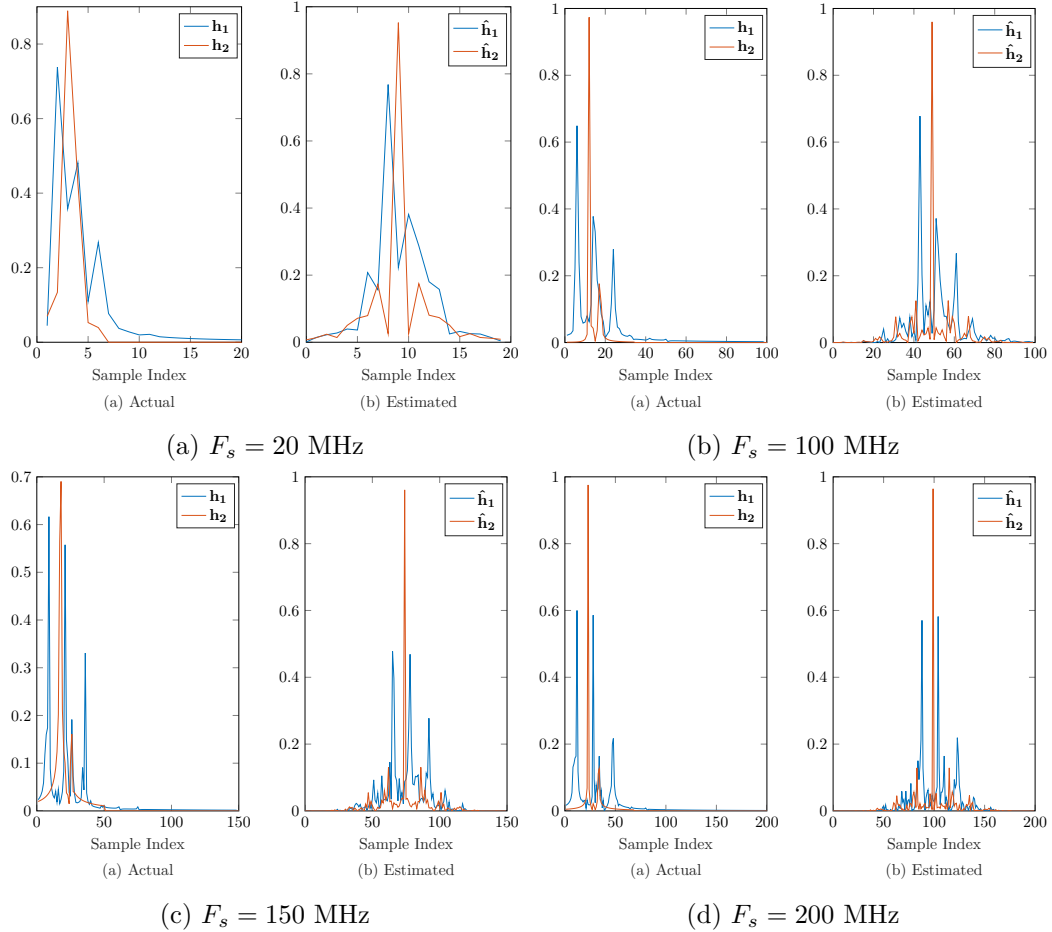


Fig. 5.1: Results of the AXIS algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

that due to the constraint forcing a nonzero value in the transfer function, it cannot be driven to zero, so normalizing the transfer function estimate at each step is no longer necessary. The  $L_1$  regularization reduces spurious peaks and inaccuracies, resulting in a better transfer function estimate. This in turn, lends itself to improved time delay estimation.

The AXIS algorithm is similar to what was done in [28, 43]. Sparseness is implemented in the same way as the AXIS algorithm, but the shift-suppression constraint has not been previously implemented. However, the other methods discuss similar constraints to the shift suppression. This algorithm differs in that it is built around minimizing estimate change, and does not make the non-negativity assumption often used in the acoustic domain.

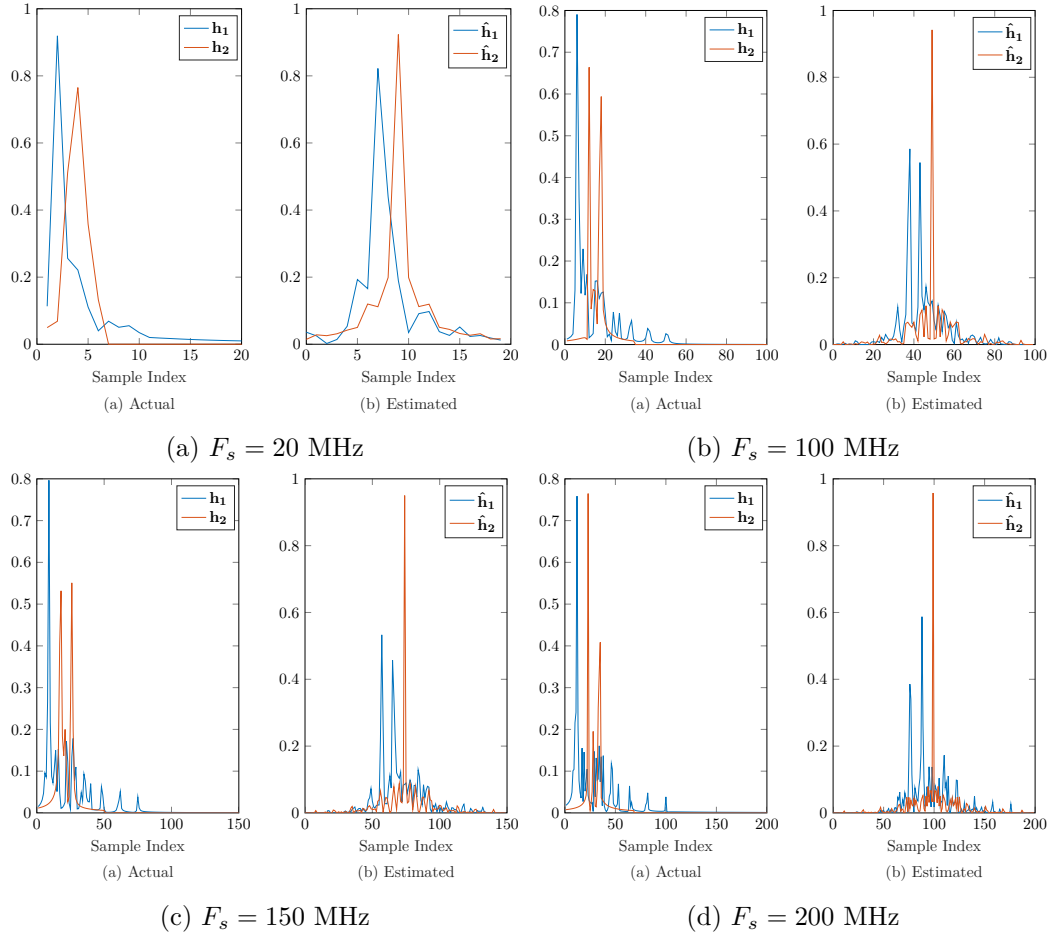


Fig. 5.2: Results of the AXIS algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

The results in Figures 5.1 and 5.2, produce similar plots to the AED algorithm shown in Figure 4.8, although the derivation is closest to the ModAEDS algorithm. The AXIS plots for the ideal situation are sparser than the AED algorithm plots of the same situation. Results of AXIS with less data points are more similar to the ModAED and ModAEDS algorithms results exhibited in Figure 5.4, showing less definition of peaks.

An advantage of the proposed AXIS algorithm is its robustness to over-estimation of the channel order. Testing via simulations (see section 5.3.1) found that over-estimating the channel order even by a factor of 2 still yielded good channel estimates as seen in Figure 5.6. For substantial ( $\text{SNR} \geq 20\text{dB}$ ) added white noise, the AXIS algorithm still provides

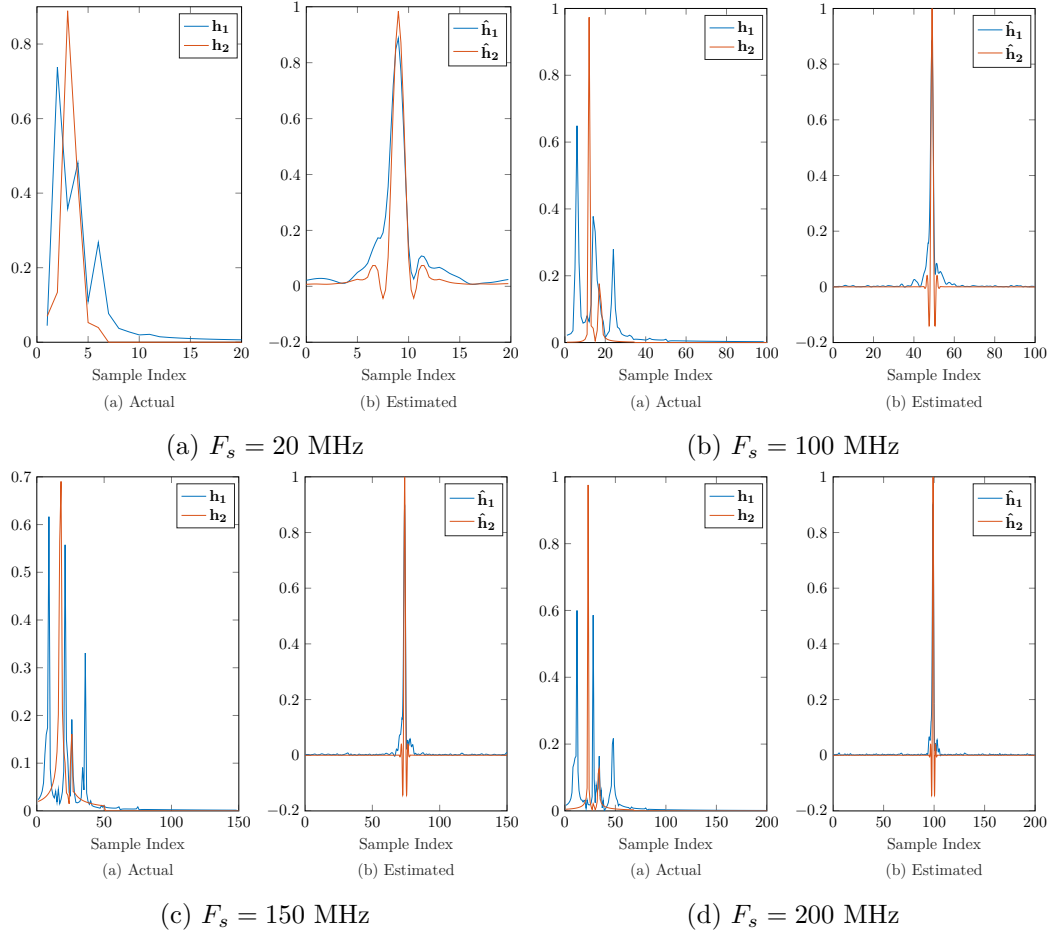


Fig. 5.3: Results of the AXIS algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

an accurate channel estimate shown in Figure 5.5. For the narrowband simulation (Figure 5.3) estimates did not improve from the results of the modified algorithms expounded on in Chapter 4.

It is worth noting that the initial value for  $\mathbf{h}^{(n)}$  will affect its convergence. Starting with  $\mathbf{h}^{(0)} = \mathbf{e}_d$  seems to perform acceptably, though the performance may be improved by starting with a vector which is closer to the actual channel impulse response. With a normalized random start, the results are like those shown in Chapter 4.

The AXIS algorithm requires  $(N - L)(30M(L + 1) + 2)$  additions and multiplies, and must compute a complex exponential  $(N - L)$  times.

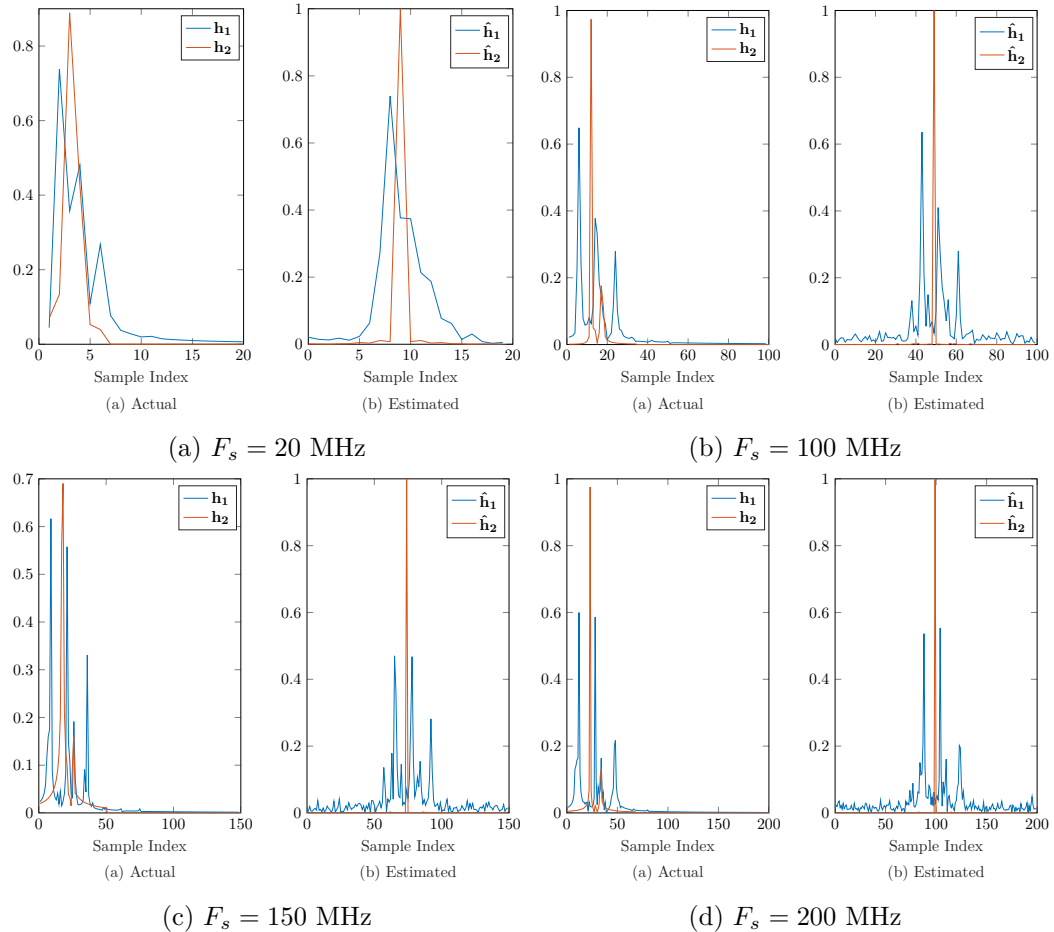


Fig. 5.4: Results of the AXIS algorithm with 2,500 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Numerical results for the AXIS algorithm can be seen in Tables 5.1–5.18, and are discussed in Section 5.3.

## 5.2 Variations

To see if any improvements could be made to the original AXIS algorithm to improve accuracy, some variations of the algorithm were also derived and tested. The numerical results and an analysis on the results from these variations can be seen below in Section 5.3.

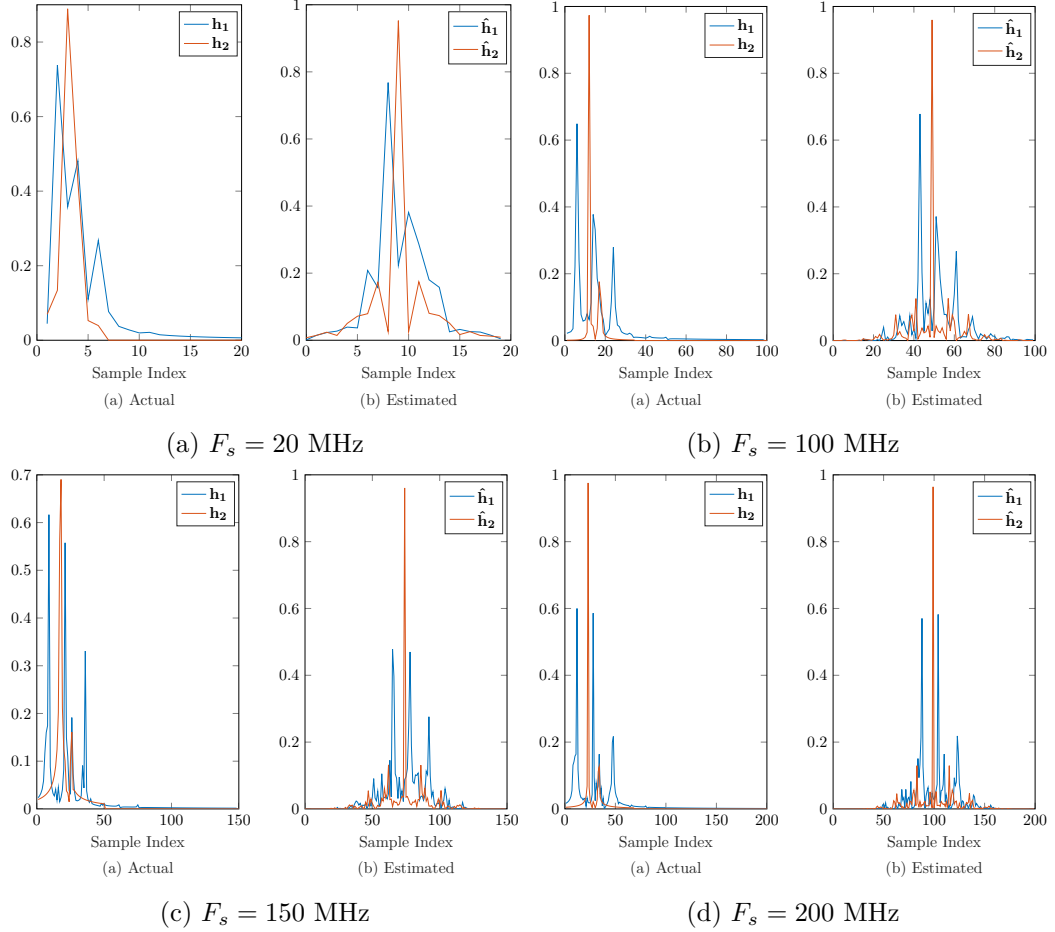


Fig. 5.5: Results of the AXIS algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

### 5.2.1 Varied AXIS

This variation of the AXIS algorithm is slight; the second constraint is more flexible

$$\begin{aligned}
 & \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
 & \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \\
 & \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \zeta.
 \end{aligned} \tag{5.9}$$

The variable  $\zeta$  in the objective function is initialized as the scalar value 1, but after the first iteration,  $\zeta = \mathbf{h}_d^{(n+1)}$ . This modification only changes the derivation of this variant of the

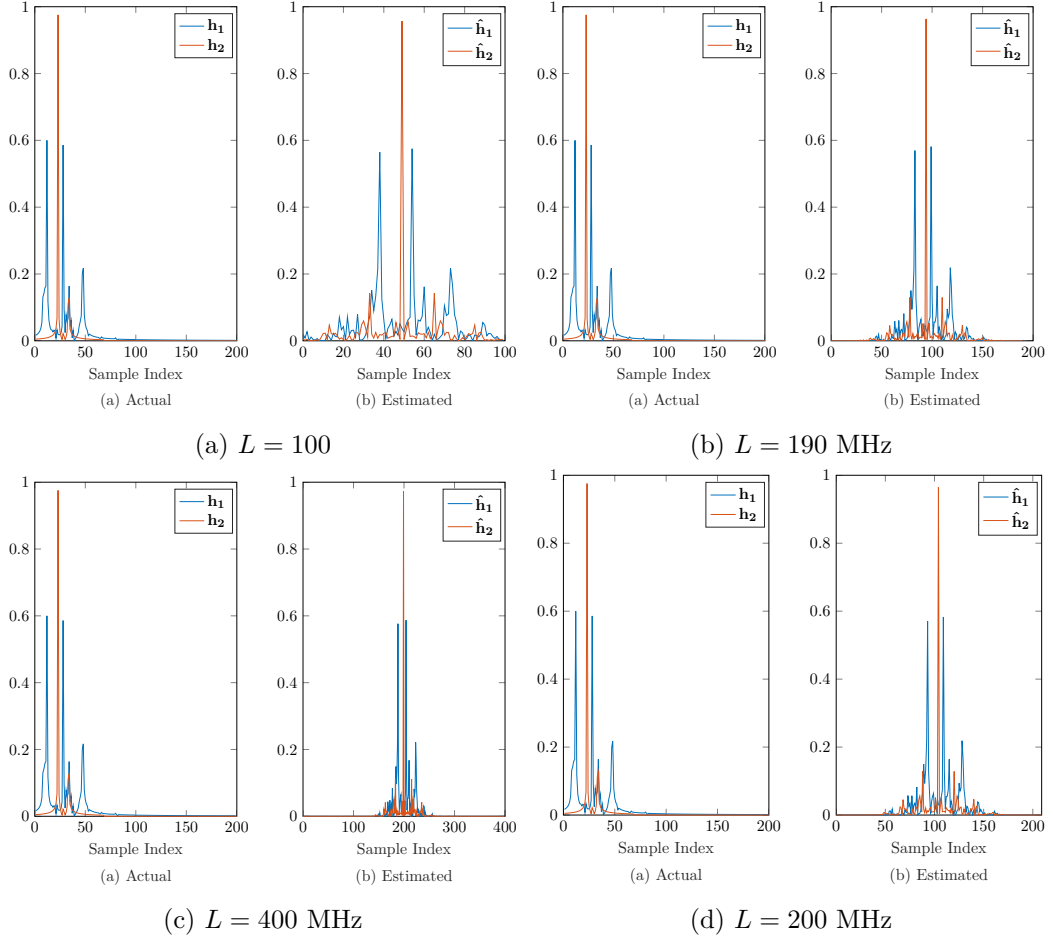


Fig. 5.6: Results of the AXIS algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a)  $L/2$ , (b)  $L - 10$ , (c)  $2L$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

AXIS algorithm update very little. Due to this slight variation, this algorithm is referred to as the Varied AXIS (VAXIS) algorithm. The first few steps result in the same equations as the AXIS algorithm. The first significant change occurs is in setting up the system of equations,

$$\begin{aligned}
 (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r &= \tilde{\lambda} \mathbf{x}_r^H \mathbf{x}_r + \tilde{\mu} e_d^H \mathbf{x}_r \\
 (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \delta &= \tilde{\lambda} \mathbf{x}_r^H \mathbf{e}_d + \tilde{\mu} e_d^H \mathbf{e}_d
 \end{aligned} \tag{5.10}$$

$$\begin{bmatrix} \mathbf{x}_r^H \mathbf{x}_r & e_d^H \mathbf{x}_r \\ \mathbf{x}_r^H \mathbf{e}_d & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \delta \end{bmatrix} \tag{5.11}$$



$$\begin{aligned}
\tilde{\lambda} &= \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \tilde{\delta} \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \\
\tilde{\mu} &= \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon}.
\end{aligned} \tag{5.12}$$

Which, substituting the values of  $\lambda$  and  $\mu$  with the conjugates of  $\tilde{\lambda}$  and  $\tilde{\mu}$  respectively from (5.12), produce a variation on (5.8),

$$\begin{aligned}
\mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\
&\eta \left( \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \tilde{\delta} \mathbf{e}_d^H \mathbf{x}_r)^*}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r - \right. \\
&\left. \left( \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \right). \tag{5.13}
\end{aligned}$$

This modification does not appear to modify the channel estimation greatly in comparison to the original AXIS algorithm as shown in Figure 5.7. The other simulation situations are not shown due to their likeness to the AXIS results.

The full derivation for this algorithm, and the corresponding code are found in Appendices A.2, and B.7 respectively.

It should be noted that this algorithm is just slightly more complex than the AXIS algorithm requiring  $(N - L)(30M(L + 1) + 4)$  computations,  $2(N - L)$  more than the AXIS (which in comparing magnitude matters little). Like the AXIS algorithm, a complex exponential must be computed  $(N - L)$  times.

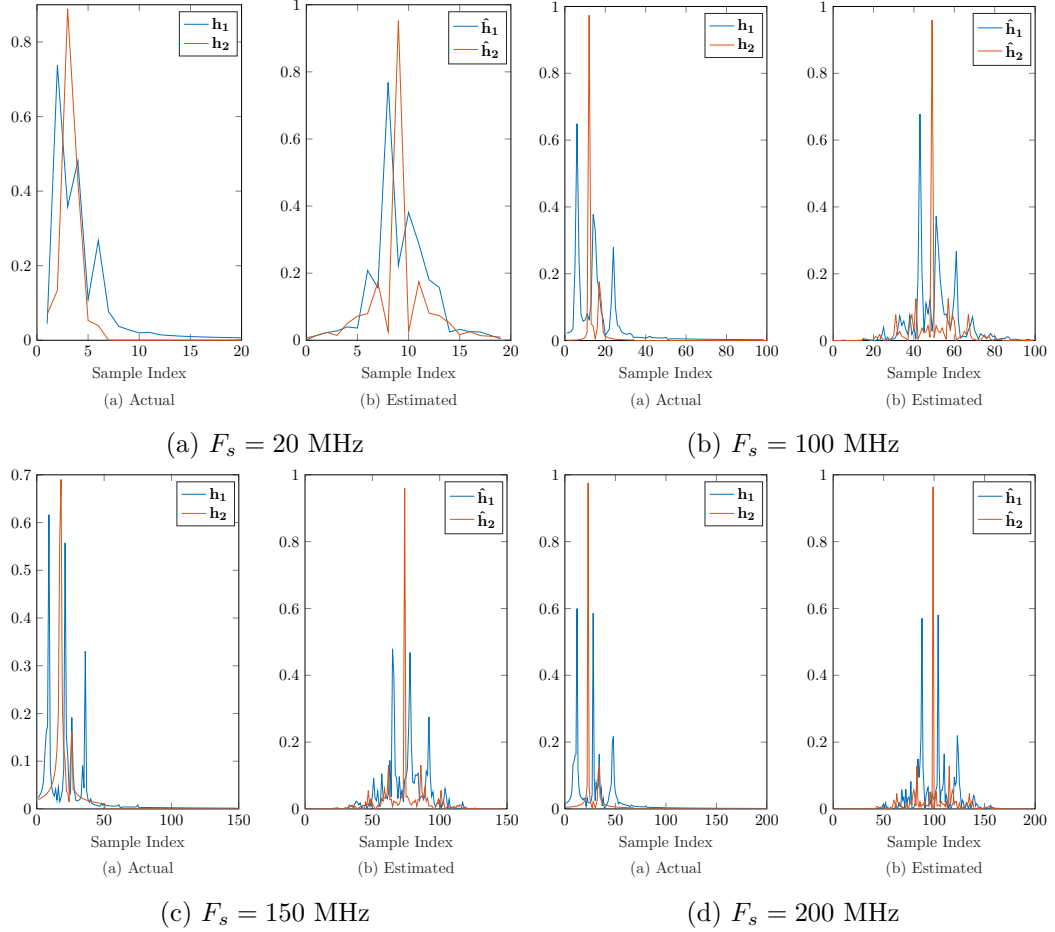


Fig. 5.7: Results of the VAXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

### 5.2.2 Dual Fixed-Peak AXIS

A different variant of the AXIS algorithm involves a third constraint,

$$\begin{aligned}
 & \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
 & \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \\
 & \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \zeta \\
 & \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c = y.
 \end{aligned} \tag{5.14}$$

The third constraint is like the second constraint, however, it focuses on a different index and

can incorporate a different value. In the previous algorithms, fixing only one of the impulse response estimates results in an asymmetric treatment of the impulse responses, with one very constrained, and the other not. As a result, the constrained impulse response typically becomes very focused on the maximum peak value. The other impulse response tends to adopt characteristics of the constrained impulse response profile to keep mathematical relationship balanced. Introducing the third constraint should remedy this, by allowing for more flexibility between channels. This variant of the AXIS algorithm is thus referred to as the Dual Fixed-Peak AXIS (DFPAXIS) algorithm.

The derivation follows previous derivations. First, the Lagrangian is set up,

$$\begin{aligned} \mathcal{L} = & \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} \\ & + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \delta) + \beta (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y). \end{aligned} \quad (5.15)$$

Taking the gradient with respect to  $\mathbf{h}^{(n+1)*}$  simplifies the expression to

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d + \beta \mathbf{e}_c. \quad (5.16)$$

Setting (5.16) to 0, and solving for  $\mathbf{h}^{(n+1)}$  yields

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d - \beta \mathbf{e}_c. \quad (5.17)$$

Equation (5.17) can then be used to solve for the Lagrange multipliers,  $\lambda$ ,  $\mu$ , and  $\beta$ .

$$\begin{aligned} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r &= \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \tilde{\beta} \mathbf{e}_c^{\text{H}} \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - \delta &= \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_c^{\text{H}} \mathbf{e}_d \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_c - y &= \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_c + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_c + \tilde{\mu} \mathbf{e}_c^{\text{H}} \mathbf{e}_c \end{aligned} \quad (5.18)$$

$$\begin{bmatrix} \mathbf{x}_r^H \mathbf{x}_r & \mathbf{e}_d^H \mathbf{x}_r & \mathbf{e}_c^H \mathbf{x}_r \\ \mathbf{x}_r^H \mathbf{e}_d & 1 & 0 \\ \mathbf{x}_r^H \mathbf{e}_c & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \\ \tilde{\beta} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y \end{bmatrix} \quad (5.19)$$

By solving the system of equation, definitions for  $\tilde{\lambda}$ ,  $\tilde{\mu}$ , and  $\tilde{\beta}$  can be found,

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}) \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon}$$

$$\begin{aligned} \tilde{\mu} = & \frac{-\mathbf{x}_r^H \mathbf{e}_d ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \\ & \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}) (\mathbf{x}_r \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \end{aligned} \quad (5.20)$$

$$\begin{aligned} \tilde{\beta} = & \frac{-\mathbf{x}_r^H \mathbf{e}_c ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}))}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \\ & \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) (\mathbf{x}_r \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \end{aligned}$$

Replacing the Lagrange multipliers with the conjugates of (5.20) results in the final update for the DFPAXIS algorithm.

$$\begin{aligned} \mathbf{h}^{(n+1)} = & \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\ & \eta \left( \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} - \right. \right. \\ & \left. \left. \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}) \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r \right. \\ & + \left( \frac{-\mathbf{x}_r^H \mathbf{e}_d ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \right. \\ & \left. \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}) (\mathbf{x}_r \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \\ & + \left( \frac{-\mathbf{x}_r^H \mathbf{e}_c ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta}))}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \right. \\ & \left. \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) (\mathbf{x}_r \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_c \Big). \end{aligned} \quad (5.21)$$

The delay between the two impulse response must already be established prior to utilizing this algorithm to maintain the fundamental existing relationship. As the focus of the research is blind channel identification, this may not seem to be possible. One approach is to use the AXIS algorithm for a block of data, and then implement the DFPAXIS algorithm as a secondary algorithm.

This is done as follows. Assuming the  $M = 2$ , if for the second constraint,  $\mathbf{h}^{(n+1)H} \mathbf{e}_d = z$ ,  $d$  is an index within the first impulse response estimate. After a few iterations, the major peaks should expose themselves, with at least one maximum peak for each impulse response. The first impulse response should have a peak at the  $d^{\text{th}}$  index. The index of the largest peak for the second impulse response should equal  $c$  for the third constraint,  $\mathbf{h}^{(n+1)H} \mathbf{e}_c = y$ . The values for  $y$  and  $z$  should then be values of the impulse responses at their respective indices.

The impulse response estimates can be seen in Figure 5.8. It looks similar to the channel estimate results shown by the AXIS and VAXIS algorithms, without much improvement. The other simulation situations replicate the results of the AXIS algorithm, and so are not shown here.

The complete derivation is shown in Appendix A.2, and the code is shown Appendix B.8.

Due to the third constraint, more computations are required than either the VAXIS or the AXIS algorithms. The DFPAXIS algorithm requires  $(N - L)(52M(L + 1) + 11)$  computations and  $(N - L)$  complex exponentials. In comparison to magnitude, this does not significant.

### 5.2.3 Non-Sparse AXIS

This alternative algorithm keeps the shift-suppression constraint but eliminates the sparsity constraint. The objective function is practically the same as (5.1), and so the math remains the same. However, when observing the update equation, (5.8), letting  $\alpha = 0$ ,

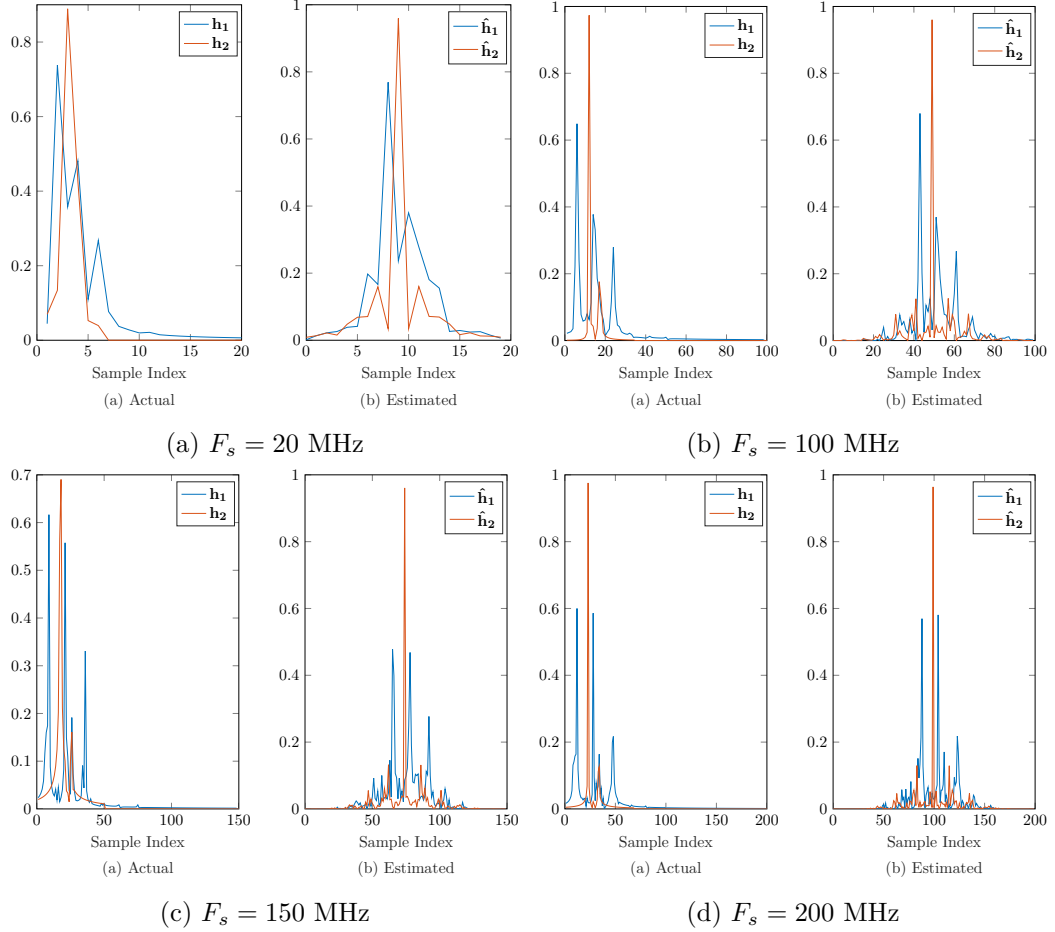


Fig. 5.8: Results of the DFPAXIS algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

produces a Non-Sparse AXIS (NSAXIS) algorithm,

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \eta \left( \left( \frac{\mathbf{h}^{(n)\text{H}} \mathbf{x}_r - (\mathbf{h}^{(n)\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r - \mathbf{e}_d^{\text{H}} \mathbf{x}_r)}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r - \left( \frac{\mathbf{x}_r^{\text{H}} \mathbf{x}_r \mathbf{h}^{(n)\text{H}} \mathbf{e}_d - \mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{h}^{(n)\text{H}} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r - \mathbf{x}_r^{\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \right). \quad (5.22)$$

This modification does not appear to affect the results greatly, if only being slightly noisier as seen in Figure 5.9. Thus, more plots are not shown, like with the previous variants. The derivation and code remain the same as the AXIS algorithm, but letting  $\alpha = 0$ . The derivation can be seen in Appendix A.2, and the code in Appendix B.6.

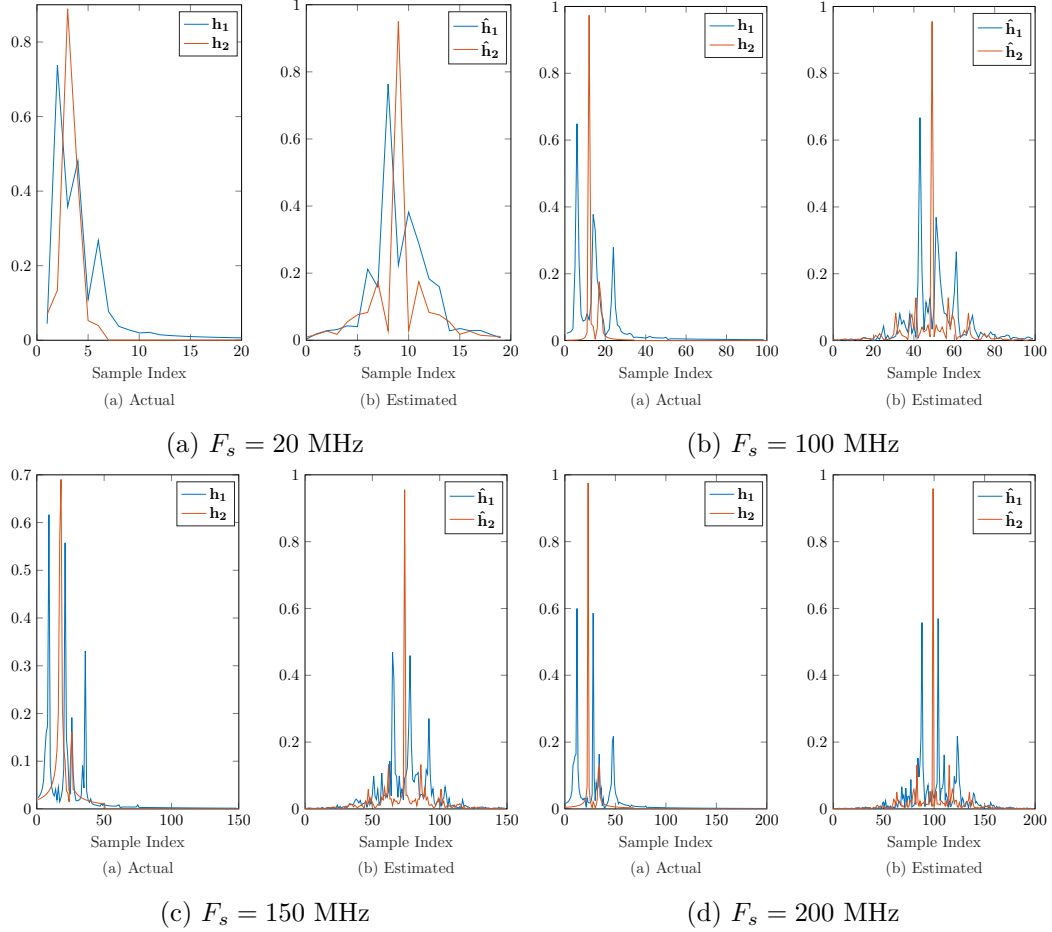


Fig. 5.9: Results of the NSAXIS algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response IR-1 seed 0.

As the sparsity constraint does not need to be computed in the NSAXIS algorithm, fewer computations than the AXIS algorithm are needed, only  $(N - L)(28M(L + 1) + 2)$ , and no complex exponential is required during computation.

#### 5.2.4 Single Constraint AXIS (SC)

The last approach explored condenses the AXIS algorithm into a single constraint objective function. As a result, a weighted enforcement of the constraints condensed into

the objective function yields

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - 1\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0. \end{aligned} \quad (5.23)$$

As there is only one constraint, this alternative algorithm is referred to as the Single Constraint (SC) algorithm.

The derivation is similar as those listed above, set up the Lagrangian, take the derivative with respect to  $\mathbf{h}^{(n+1)*}$ , set it equal to 0, and solve for  $\mathbf{h}^{(n+1)}$ :

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma ((\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)}) - \mathbf{e}_d) + \lambda \mathbf{x}_r, \quad (5.24)$$

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d - \lambda \mathbf{x}_r), \quad (5.25)$$

where  $\mathbf{I}$  is the identity matrix of size  $M(L+1) \times M(L+1)$ .

As there are not multiple constraint, a system of equations does not need to be arranged. Equation (5.25) merely needs to be plugged into the constraint listed in (5.23), and solved for  $\lambda$  to produce

$$\lambda = \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \right)^*. \quad (5.26)$$

Exchanging  $\lambda$  in (5.25), with (5.26) constructs the update for this adaptive algorithm,

$$\begin{aligned} \mathbf{h}^{(n+1)} &= (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \\ & (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \right)^* \mathbf{x}_r). \end{aligned} \quad (5.27)$$

This algorithm unfortunately contains a matrix inverse, and matrix multiplication. However, this matrix inverse is computationally simple to compute as the matrix is a diagonal matrix. Computational complexity is  $(N-L)(9M^2 * (L+1)^2 + 8M(L+1))$  additions and multiplications, which is still much greater than the AXIS complexity.



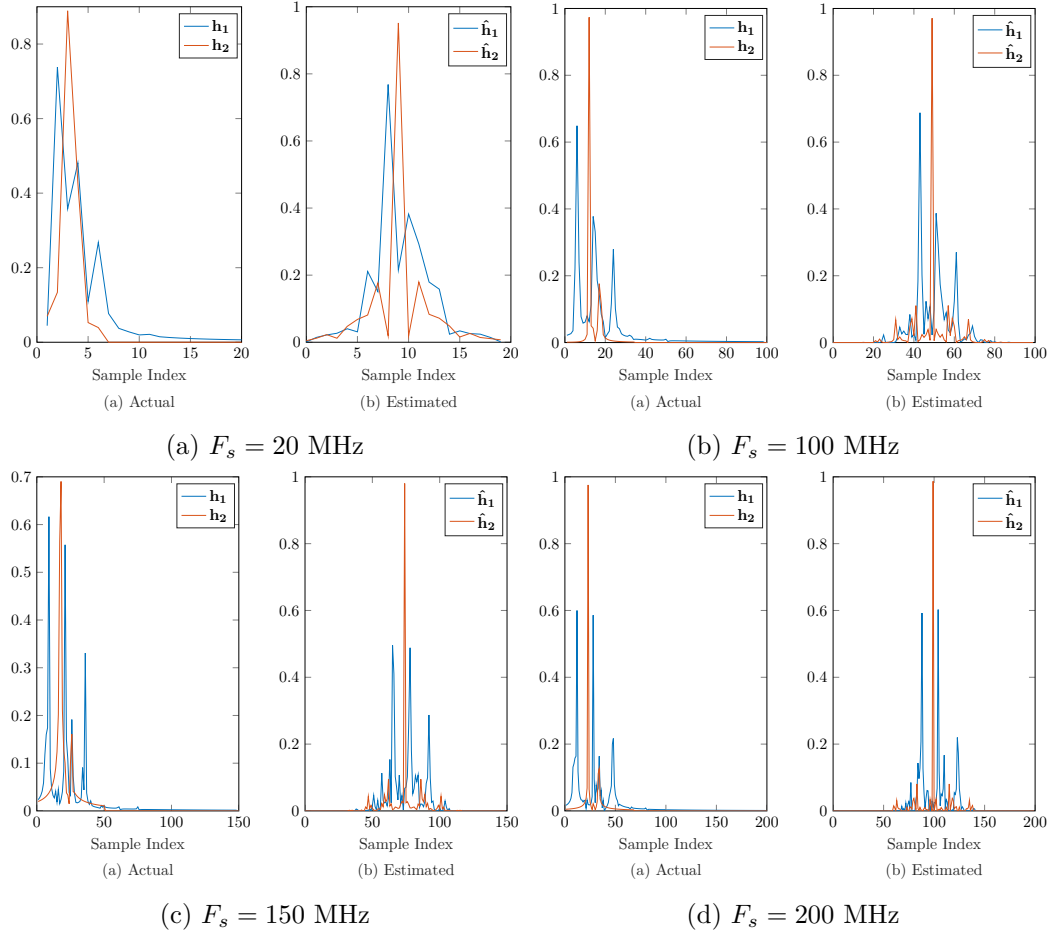


Fig. 5.10: Results of the SC algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

The channel estimates can be seen in Figure 5.10 for the “ideal” case. Like the other variants there does not appear to be much improvement from the AXIS algorithm, and so no other figures are shown. The complete derivation of this algorithm is detailed in Appendix A.2. The code is found in Appendix B.9.

### Variations of the SC Algorithm

As the SC algorithm is a variation of the AXIS algorithm, the other variations can also be applied, including the Varied SC, Dual Fixed-Peak SC, and the Non-Sparse SC algorithms.

**Varied SC (VSC)** The Varied Single Constraint (VSC), is similar to the VAXIS, by switching the 1 in the objective function, (5.23), to a  $\zeta$ ,

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \zeta\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0. \end{aligned} \quad (5.28)$$

Using the same derivation methods described in the other variations, the update for this variant of the SC algorithm results in

$$\begin{aligned} \mathbf{h}^{(n+1)} &= (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \\ & (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \zeta^* \mathbf{e}_d - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \zeta^* \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \right)^* \mathbf{x}_r). \end{aligned} \quad (5.29)$$

As explained above,  $\zeta$  is initialized as 1, and then upon the following iterations,  $\zeta = \mathbf{h}_d^{(n+1)}$ .

The channel estimates are shown in Figure 5.11, once again, there is not much improvement if any on the estimates gleaned for the AXIS algorithm. Only  $(N - L)$  more computations than the SC algorithm are needed for the VSC algorithm. The code for this subvariation is shown in Appendix B.10, and the derivation in Appendix A.2.

**Dual Fixed-Peak SC (DFPSC)** The Dual Fixed-Peak SC is also similar to the DF-PAXIS algorithm derived above, the channel estimates can be seen in Figure 5.12. As  $\gamma \|\mathbf{h}^{(n+1)} \mathbf{e}_d - \zeta\|$  in the objective functions described in (5.28) only affects one of impulse responses, meaning the impulse responses are treated asymmetrically. To combat this, a similar offset for the other impulse response,  $\phi \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y\|$  is added to the objective function,

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \zeta\|_2^2 + \phi \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0. \end{aligned} \quad (5.30)$$

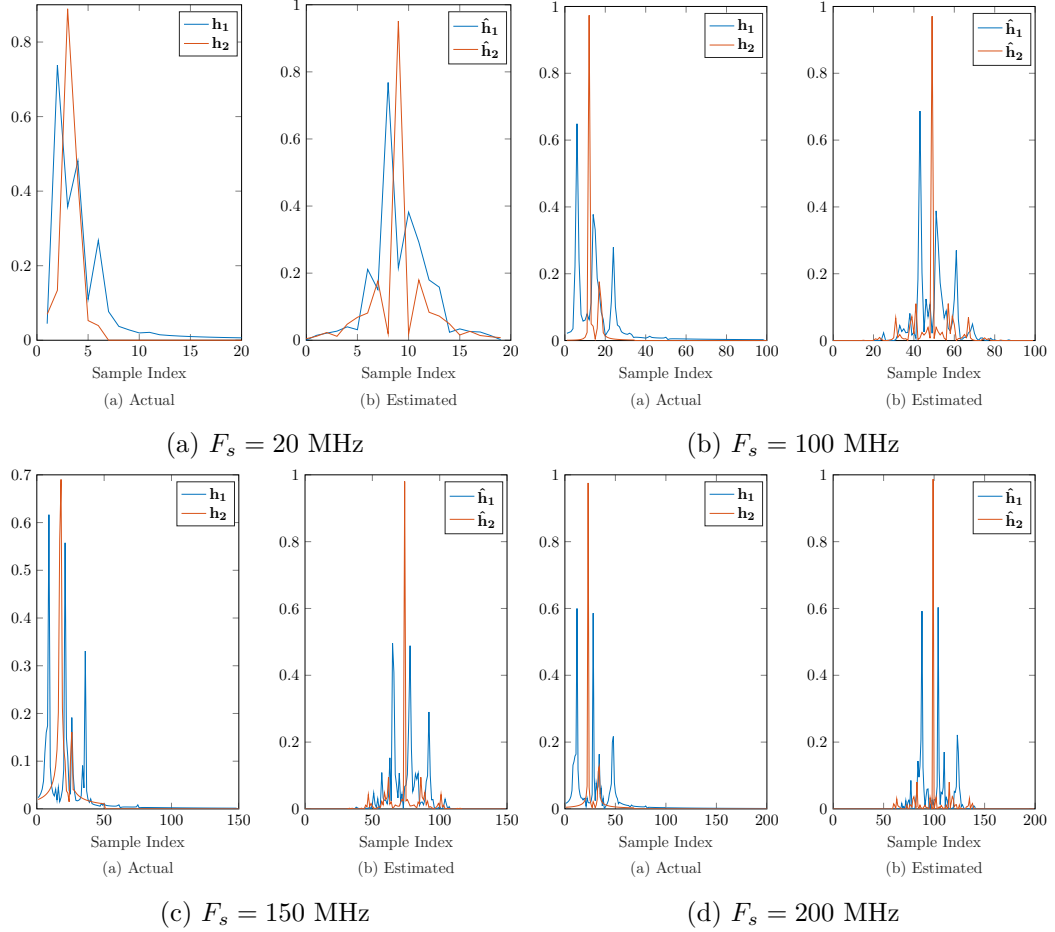


Fig. 5.11: Results of the VSC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

The update resulting from the minimization techniques explored above is a bit shorter than (5.8),

$$\begin{aligned}
 \mathbf{h}^{(n+1)} &= (\mathbf{I} + \gamma e_d e_d^H + \phi e_c e_c^H)^{-1} (\mathbf{h}^{(n)} - \alpha e^{j\angle \mathbf{h}^{(n)}} + \gamma (\hat{\mathbf{z}}^* e_d) + \phi (y^* e_c) \\
 &\quad - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha e^{j\angle \mathbf{h}^{(n)}} + \gamma (\hat{\mathbf{z}}^* e_d) + \phi (y^* e_c))^H (\mathbf{I} + \gamma e_d e_d^H + \phi e_c e_c^H)^{-1} \mathbf{x}_r}{\mathbf{x}_r^H (\mathbf{I} + \gamma e_d e_d^H + \phi e_c e_c^H)^{-1} \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r). \quad (5.31)
 \end{aligned}$$

As also explained previously, this algorithm variation is a secondary algorithm, which can only be utilized once the delay between two transfer functions has already been established. As with the DFPAXIS algorithm, the DFPSC algorithm requires more computations

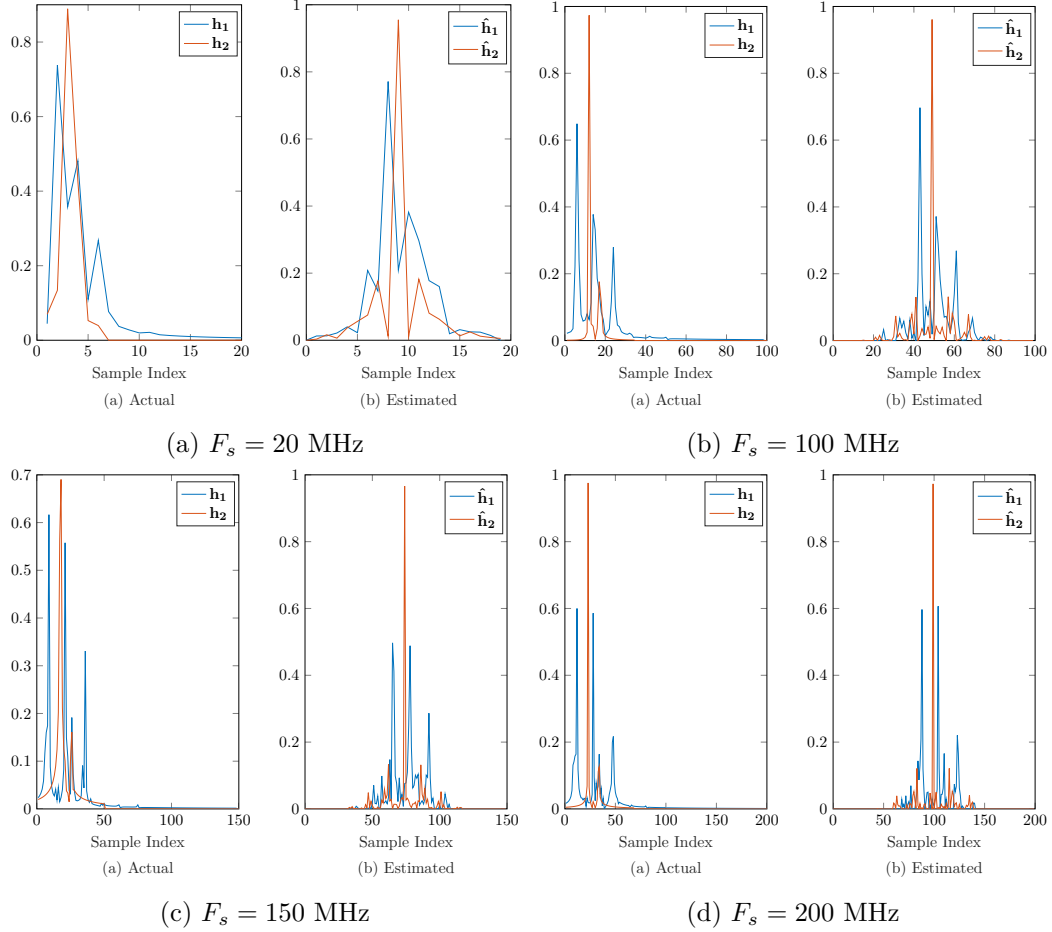


Fig. 5.12: Results of the DFPSC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

than the SC algorithm (and the AXIS algorithm), needing  $(N-L)(12M^2*(L+1)^2+10M(L+1)+1)$  additions and multiplies.

The code for this algorithm and the detailed derivation can be found in Appendix B.11, and Appendix A.2 respectively.

**Non-Sparse SC (NSSC)** Similar to the Non-Sparse AXIS, this subvariation of the algorithm removes the sparsity constraint by merely setting  $\alpha = 0$  in the update Equation

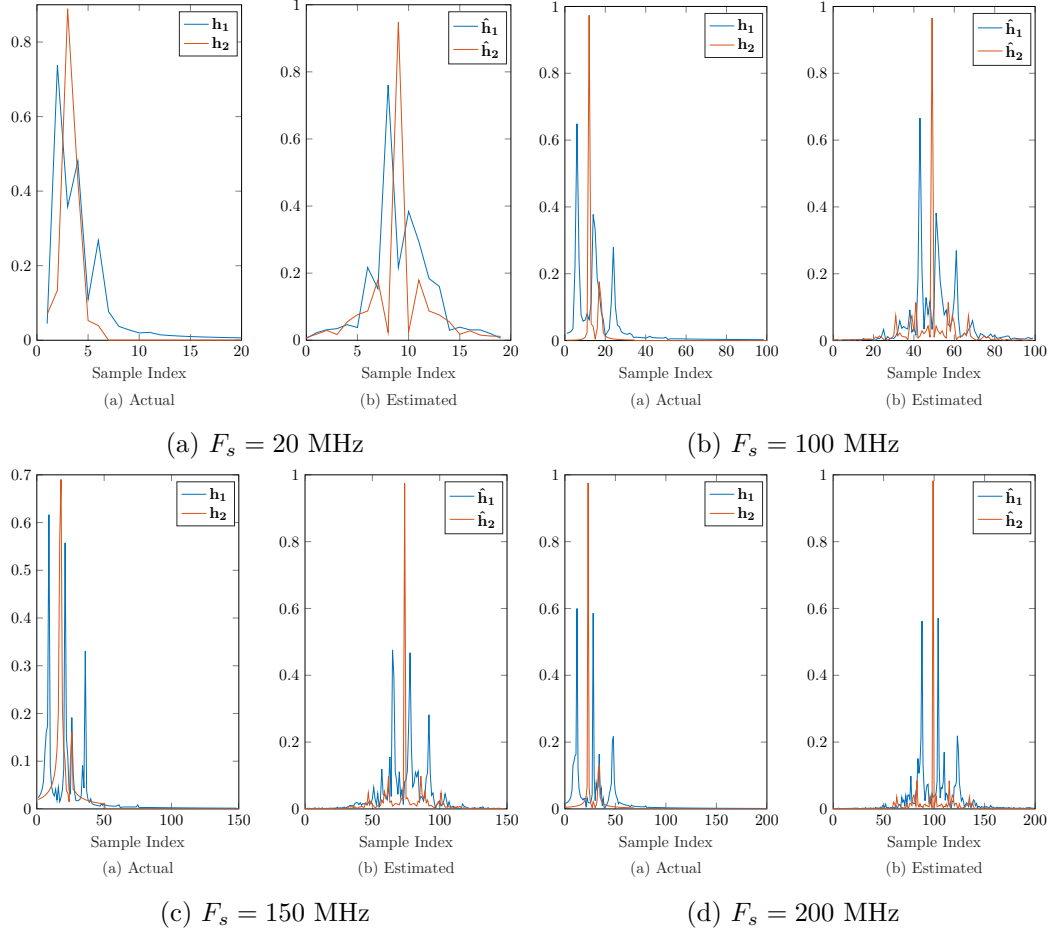


Fig. 5.13: Results of the Non-Sparse SC algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

(5.27). The resulting update equation is

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma e_d e_d^H)^{-1} (\mathbf{h}^{(n)} + \gamma e_d - \eta \left( \frac{(\mathbf{h}^{(n)} + \gamma e_d)^H (\mathbf{I} + \gamma e_d e_d^H)^{-1} \mathbf{x}_r}{\mathbf{x}_r^H (\mathbf{I} + \gamma e_d e_d^H)^{-1} \mathbf{x}_r} \right)^* \mathbf{x}_r). \quad (5.32)$$

With this update, there was also not a significant change compared to the AXIS and SC algorithm results. As there is not much change (if any), only the ideal case is shown in Figure 5.13. And like the NSAXIS algorithm, a few less computations than the SC algorithm are required,  $(N - L)(9M^2 * (L + 1)^2 + 6M(L + 1))$ , which is still more than the AXIS algorithm.

The derivation for this algorithm remains the same as the SC algorithm, however

defining  $\alpha = 0$ . The SC algorithm derivation can be seen in Appendix A.2. The code is also shown in Appendix B.9.

### 5.3 Data Verification

These algorithms were all tested using both simulation and real-world data. The simulation results are discussed in detail here, whereas the real-world data discussion can be found in Chapter 7.

#### 5.3.1 Simulation

The AXIS algorithm and all its variants and subvariants were tested using the same variables at the same values, and on the same computer. The variables, and their respective values can be seen in Table 7.2.

Each of the algorithms was also tested with a variation of key parameters to see their effect on the channel estimation. These parameters include signal bandwidth, signal to noise ratio (SNR), sample rate, and sample size. Tables 5.1–5.18 show the numerical delay estimation results of the simulations for all of the algorithms discussed in Chapter 5.

Each simulation situation is analyzed in more detail below, drawing a direct comparison between the algorithms, and the different impulse response sets.

#### “Ideal” Simulation

The “Ideal” simulation results can be seen in Table 5.1 and Table 5.2 for impulse response set IR-1 and IR-2 respectively. For both impulse response sets, the DFPAXIS algorithm typically has the best results, with the smallest averaged error. However, as was observed previously impulse response set IR-2 has better performance than set IR-1.

#### Signal Bandwidth

Tables 5.3 and 5.4 show the simulation results for the impulse responses using narrowband data. As can be observed, the estimates do not resolve, and so these algorithms produce a much larger error than their wideband equivalent. It is interesting to note that

all of the algorithms in all simulation situations except for the impulse response set IR-2 at 20 MHz, produce the exact same estimates and errors. For set IR-1, the DFPAXIS algorithm produces a smaller estimate with a larger error. It should also be noted that set IR-2, typically has better results, and more realistic channel estimates, however, observing the channel estimates (as shown in Figure 5.3) the channels do not resolve, and so are likely random chance that they are good estimates.

### Noise

Noisy (20 dB SNR) simulation results are shown in Tables 5.5 and 5.6. For both impulse response sets, the results are almost the same as the less noisy counterpart (60 dB SNR). At 150 MHz, for set IR-1, along with both DFP algorithms, the AXIS algorithms also perform better, showing the first significant difference. For impulse response set IR-2, instead of performing better at 20 MHz, the DFPAXIS algorithm performs the same as the other algorithms.

### Sample Size

Comparison of different data sizes was also tested, and the results for the smaller data amount is shown in Table 5.7, and Table 5.8. With less data, no one algorithm appears to stand out as superior than the others. There are some slightly varied results, but those could just be due to random noise differences. The results for the smaller data size are typically not as accurate as with the larger data size, but fairly close to the estimates shown in Tables 5.1, and 5.2.

### Model Order

Different model order for the impulse responses is shown in Tables 5.9–5.16.

Underestimating the model order does not appear to affect the delay estimates for impulse response set IR-2. Even underestimate the channel order by half does not appear to have great effects on the delay estimates. There is slight variation (if any) between the exact model order estimate response and the underestimated model order response.

Overestimating also does not appear to affect set IR-2 greatly. Overestimating has shown delay estimates that overestimate the distance rather than underestimate. Doubling the model order estimate appears to provide more consistent estimates between algorithms.

Set IR-1 reveals that underestimating the channel order has a greater impact. At 20 MHz, underestimating the model order,  $L$  by half ( $\approx 10$ ), and by just 10 less are equivalent, and possibly due to the smallness of a 10-tap filter, no peaks could be realized in this limited impulse response length (this is why there are NaNs in place of numbers). The impulse response sets do much better for the other sampling rates, and typically have better results for the model order that is closest to the true model order. This also shows that underestimating the model order does not have a significant impact. It is interesting to note, however, that for 150 MHz, underestimating by half gives a better estimate overall than the  $L - 10$  estimate. Overestimating the channel model order, like set IR-2 does not have a great effect on the impulse response delay estimates.

### **Initial Estimates**

As seen in Tables 5.17 and Table 5.18, using a normal random initial channel estimate typically yields large, inaccurate results. It should be noted that both the AXIS and NSAXIS algorithms consistently provide reasonable estimates even with the random start, for both impulse response sets. For those two algorithms, the range of estimates was also not extreme.



Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Ideal	20 MHz	AXIS	21.772	15	0	15	30	6.772
		VAXIS		15	0	15	30	6.772
		DFPAXIS		16.429	0	15	45	5.344
		NSAXIS		15	0	15	30	6.772
		SC		15	0	15	30	6.772
		VSC		15	0	15	30	6.772
	100 MHz	DFPSC		15	0	15	30	6.772
		NSSC		15	0	15	30	6.772
		AXIS	21.772	16.714	6	18	36	5.058
		VAXIS		16.714	6	18	36	5.058
		DFPAXIS		17.143	6	18	36	4.629
		NSAXIS		16.714	6	18	36	5.058
150 MHz	100 MHz	SC		17.143	6	18	36	4.629
		VSC		17.143	6	18	36	4.629
		DFPSC		16.714	6	18	36	5.058
		NSSC		17.143	6	18	36	4.629
		AXIS	21.772	16.667	8	18	36	5.106
		VAXIS		16.667	8	18	36	5.106
	200 MHz	DFPAXIS		17.143	8	18	36	4.629
		NSAXIS		16.667	8	18	36	5.106
		SC		16.667	8	18	36	5.106
		VSC		16.667	8	18	36	5.106
		DFPSC		17.143	8	18	36	4.629
		NSSC		16.667	8	18	36	5.106
200 MHz	150 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844
		SC		12.929	7.5	15	19.5	8.844
		VSC		12.929	7.5	15	19.5	8.844
	200 MHz	DFPSC		12.929	7.5	15	19.5	8.844
		NSSC		12.929	7.5	15	19.5	8.844
		AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844

Table 5.1: Numerical results for the AXIS algorithms in the “ideal” case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: Ideal	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		20	0	15	30	1.772
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
		NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
		SC		21.714	6	24	33	0.058
		VSC		21.714	6	24	33	0.058
DFPSC		21.714	6	24	33	0.058		
NonSSC		21.714	6	24	33	0.058		
150 MHz		AXIS	21.772	18.476	4	18	34	3.296
		VAXIS		18.476	4	18	34	3.296
		DFPAXIS		18.476	4	18	34	3.296
		NSAXIS		18.476	4	18	34	3.296
		SC		19.810	4	20	34	1.963
		VSC		19.810	4	20	34	1.963
		DFPSC		18.476	4	18	34	3.296
		NSSC		19.810	4	20	34	1.963
		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		20.143	4.5	19.5	34.5	1.629
		NSAXIS		19.857	4.5	19.5	34.5	1.915
		SC		19.643	4.5	19.5	34.5	2.129
		VSC		19.643	4.5	19.5	34.5	2.129
DFPSC		19.643	4.5	19.5	34.5	2.129		
NonSSC		19.643	4.5	19.5	34.5	2.129		
200 MHz		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		20.143	4.5	19.5	34.5	1.629
		NSAXIS		19.857	4.5	19.5	34.5	1.915
		SC		19.643	4.5	19.5	34.5	2.129
		VSC		19.643	4.5	19.5	34.5	2.129
		DFPSC		19.643	4.5	19.5	34.5	2.129
		NSSC		19.643	4.5	19.5	34.5	2.129
		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		20.143	4.5	19.5	34.5	1.629
		NSAXIS		19.857	4.5	19.5	34.5	1.915
		SC		19.643	4.5	19.5	34.5	2.129
		VSC		19.643	4.5	19.5	34.5	2.129
DFPSC		19.643	4.5	19.5	34.5	2.129		
NonSSC		19.643	4.5	19.5	34.5	2.129		

Table 5.2: Numerical results for the AXIS algorithms in the “ideal” case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$	
IR-1: Narrowband	20 MHz	AXIS	21.772	5.714	0	0	60	16.058	
		VAXIS		5.714	0	0	60	16.058	
		DFPAXIS		2.86	0	0	60	18.915	
		NSAXIS		5.714	0	0	60	16.058	
		SC		8.571	0	0	60	13.201	
		VSC		8.571	0	0	60	13.201	
		DFPSC		8.571	0	0	60	13.201	
		NSSC		8.571	0	0	60	13.201	
		AXIS		5.714	0	0	0	60	16.058
		VAXIS		5.714	0	0	0	60	16.058
		DFPAXIS		5.714	0	0	0	60	16.058
		NSAXIS		5.714	0	0	0	60	16.058
SC		5.714	0	0	0	60	16.058		
VSC		5.714	0	0	0	60	16.058		
DFPSC		5.714	0	0	0	60	16.058		
NSSC		5.714	0	0	0	60	16.058		
150 MHz		AXIS	21.772	2.857	0	0	60	18.915	
		VAXIS		2.857	0	0	60	18.915	
		DFPAXIS		2.857	0	0	60	18.915	
		NSAXIS		2.857	0	0	60	18.915	
		SC		2.857	0	0	60	18.915	
		VSC		2.857	0	0	60	18.915	
		DFPSC		2.857	0	0	60	18.915	
		NSSC		2.857	0	0	60	18.915	
		AXIS		21.772	2.857	0	0	60	18.915
		VAXIS		21.772	2.857	0	0	60	18.915
		DFPAXIS		21.772	2.857	0	0	60	18.915
		NSAXIS		21.772	2.857	0	0	60	18.915
SC		21.772	2.857	0	0	60	18.915		
VSC		21.772	2.857	0	0	60	18.915		
DFPSC		21.772	2.857	0	0	60	18.915		
NSSC		21.772	2.857	0	0	60	18.915		
200 MHz		AXIS	21.772	2.857	0	0	60	18.915	
		VAXIS		2.857	0	0	60	18.915	
		DFPAXIS		2.857	0	0	60	18.915	
		NSAXIS		2.857	0	0	60	18.915	
		SC		2.857	0	0	60	18.915	
		VSC		2.857	0	0	60	18.915	
		DFPSC		2.857	0	0	60	18.915	
		NSSC		2.857	0	0	60	18.915	
		AXIS		21.772	2.857	0	0	60	18.915
		VAXIS		21.772	2.857	0	0	60	18.915
		DFPAXIS		21.772	2.857	0	0	60	18.915
		NSAXIS		21.772	2.857	0	0	60	18.915
SC		21.772	2.857	0	0	60	18.915		
VSC		21.772	2.857	0	0	60	18.915		
DFPSC		21.772	2.857	0	0	60	18.915		
NSSC		21.772	2.857	0	0	60	18.915		

Table 5.3: Numerical results for the AXIS algorithms in the narrowband case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$	
IR-2: Narrowband	20 MHz	AXIS	21.772	17.143	0	0	60	4.629	
		VAXIS		17.143	0	0	60	4.629	
		DFPAXIS		17.143	0	0	60	4.629	
		NSAXIS		17.143	0	0	60	4.629	
		SC		17.143	0	0	60	4.629	
		VSC		17.143	0	0	60	4.629	
	100 MHz	DFPSC		17.143	0	0	60	4.629	
		NSSC		17.143	0	0	60	4.629	
		AXIS	21.772	17.143	0	0	60	4.629	
		VAXIS		17.143	0	0	60	4.629	
		DFPAXIS		17.143	0	0	60	4.629	
		NSAXIS		17.143	0	0	60	4.629	
150 MHz	20 MHz	SC		17.143	0	0	60	4.629	
		VSC		17.143	0	0	60	4.629	
		DFPSC		17.143	0	0	60	4.629	
		NSSC		17.143	0	0	60	4.629	
		AXIS	21.772	14.286	0	0	60	7.486	
		VAXIS		14.286	0	0	60	7.486	
	150 MHz	200 MHz	DFPAXIS		14.286	0	0	60	7.486
			NSAXIS		14.286	0	0	60	7.486
			SC		14.286	0	0	60	7.486
			VSC		14.286	0	0	60	7.486
			DFPSC		14.286	0	0	60	7.486
			NSSC		14.286	0	0	60	7.486
200 MHz	200 MHz	AXIS	21.772	14.286	0	0	60	7.486	
		VAXIS		14.286	0	0	60	7.486	
		DFPAXIS		14.286	0	0	60	7.486	
		NSAXIS		14.286	0	0	60	7.486	
		SC		14.286	0	0	60	7.486	
		VSC		14.286	0	0	60	7.486	
	200 MHz	200 MHz	DFPSC		14.286	0	0	60	7.486
			NSSC		14.286	0	0	60	7.486
			AXIS	21.772	14.286	0	0	60	7.486
			VAXIS		14.286	0	0	60	7.486
			DFPAXIS		14.286	0	0	60	7.486
			NSAXIS		14.286	0	0	60	7.486

Table 5.4: Numerical results for the AXIS algorithms in the narrowband case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Noisy	20 MHz	AXIS	21.772	15	0	15	30	6.772
		VAXIS		15	0	15	30	6.772
		DFPAXIS		16.429	0	15	45	5.344
		NSAXIS		15	0	15	30	6.772
		SC		15	0	15	30	6.772
		VSC		15	0	15	30	6.772
	100 MHz	DFPSC		15	0	15	30	6.772
		NSSC		15	0	15	30	6.772
		AXIS	21.772	16.714	6	18	36	5.058
		VAXIS		16.714	6	18	36	5.058
		DFPAXIS		17.143	6	18	36	4.629
		NSAXIS		16.714	6	18	36	5.058
150 MHz	150 MHz	SC		17.143	6	18	36	4.629
		VSC		17.143	6	18	36	4.629
		DFPSC		16.714	6	18	36	5.058
		NSSC		17.143	6	18	36	4.629
		AXIS	21.772	17.143	8	18	36	4.629
		VAXIS		17.143	8	18	36	4.629
	200 MHz	DFPAXIS		17.143	8	18	36	4.629
		NSAXIS		16.667	8	18	36	5.106
		SC		16.667	8	18	36	5.106
		VSC		16.667	8	18	36	5.106
		DFPSC		17.143	8	18	36	4.629
		NSSC		16.667	8	18	36	5.106
200 MHz	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844
		SC		12.929	7.5	15	19.5	8.844
		VSC		12.929	7.5	15	19.5	8.844
	200 MHz	DFPSC		12.929	7.5	15	19.5	8.844
		NSSC		12.929	7.5	15	19.5	8.844
		AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844

Table 5.5: Numerical results for the AXIS algorithms in the noisy case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: Noisy	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		18.571	0	15	30	3.201
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
	100 MHz	NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
		SC		21.714	6	24	33	0.058
		VSC		21.714	6	24	33	0.058
150 MHz	DFPSC		21.714	6	24	33	0.058	
	NSSC		21.714	6	24	33	0.058	
	AXIS	21.772	18.476	4	18	34	3.296	
	VAXIS		18.476	4	18	34	3.296	
	DFPAXIS		18.476	4	18	34	3.296	
	NSAXIS		18.476	4	18	34	3.296	
	SC		19.810	4	20	34	1.963	
200 MHz	VSC		19.810	4	20	34	1.963	
	DFPSC		18.476	4	18	34	3.296	
	NSSC		19.810	4	20	34	1.963	
	AXIS	21.772	19.643	4.5	19.5	34.5	2.129	
	VAXIS		19.643	4.5	19.5	34.5	2.129	
	DFPAXIS		20.143	4.5	19.5	34.5	1.629	
	NSAXIS		19.643	4.5	19.5	34.5	2.129	
200 MHz	SC		19.357	4.5	19.5	34.5	2.415	
	VSC		19.357	4.5	19.5	34.5	2.415	
	DFPSC		19.643	4.5	19.5	34.5	2.129	
	NSSC		19.643	4.5	19.5	34.5	2.129	

Table 5.6: Numerical results for the AXIS algorithms in the noisy case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Fewer Samples	20 MHz	AXIS	21.772	14.286	0	15	30	7.486
		VAXIS		14.286	0	15	30	7.486
		DFPAXIS		14.286	0	15	30	7.486
		NSAXIS		14.286	0	15	30	7.486
		SC		14.286	0	15	30	7.486
		VSC		14.286	0	15	30	7.486
		DFPSC		14.286	0	15	30	7.486
		NSSC		14.286	0	15	30	7.486
		AXIS	21.772	16.857	6	18	36	4.915
		VAXIS		16.857	6	18	36	4.915
		DFPAXIS		17.429	6	18	36	4.344
		NSAXIS		16.857	6	18	36	4.915
SC		16.857	6	18	36	4.915		
VSC		16.857	6	18	36	4.915		
DFPSC		16.857	6	18	36	4.915		
NSSC		16.857	6	18	36	4.915		
150 MHz	150 MHz	AXIS	21.772	15.429	8	18	20	6.344
		VAXIS		15.429	8	18	20	6.344
		DFPAXIS		15.429	8	18	20	6.344
		NSAXIS		16.191	8	18	36	5.582
		SC		15.429	8	18	20	6.344
		VSC		15.429	8	18	20	6.344
		DFPSC		15.429	8	18	20	6.344
		NSSC		16.191	8	18	36	5.582
		AXIS	21.772	14.429	7.5	16.5	19.5	7.344
		VAXIS		14.429	7.5	16.5	19.5	7.344
		DFPAXIS		14.429	7.5	16.5	19.5	7.344
		NSAXIS		14.429	7.5	16.5	19.5	7.344
SC		14.429	7.5	16.5	19.5	7.344		
VSC		14.429	7.5	16.5	19.5	7.344		
DFPSC		14.214	7.5	15	19.5	7.558		
NSSC		14.429	7.5	16.5	19.5	7.344		
200 MHz	200 MHz	AXIS	21.772	14.429	7.5	16.5	19.5	7.344
		VAXIS		14.429	7.5	16.5	19.5	7.344
		DFPAXIS		14.429	7.5	16.5	19.5	7.344
		NSAXIS		14.429	7.5	16.5	19.5	7.344
		SC		14.429	7.5	16.5	19.5	7.344
		VSC		14.429	7.5	16.5	19.5	7.344
		DFPSC		14.214	7.5	15	19.5	7.558
		NSSC		14.429	7.5	16.5	19.5	7.344

Table 5.7: Numerical results for the AXIS algorithms in the smaller data size case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: Fewer Samples	20 MHz	AXIS	21.772	19.286	0	15	30	2.486
		VAXIS		19.286	0	15	30	2.486
		DFPAXIS		19.286	0	15	30	2.486
		NSAXIS		19.286	0	15	30	2.486
	SC		19.286	19.286	0	15	30	2.486
	VSC		19.286	19.286	0	15	30	2.486
	DFPSC		19.286	19.286	0	15	30	2.486
	NSSC		19.286	19.286	0	15	30	2.486
	100 MHz	AXIS	21.772	20.857	3	24	33	0.915
	VAXIS		20.857	20.857	3	24	33	0.915
	DFPAXIS		20.857	20.857	3	24	33	0.915
	NSAXIS		20.857	20.857	3	24	33	0.915
SC		20.857	20.857	3	24	33	0.915	
VSC		20.857	20.857	3	24	33	0.915	
DFPSC		20.857	20.857	3	24	33	0.915	
NSSC		20.857	20.857	3	24	33	0.915	
150 MHz	150 MHz	AXIS	21.772	18.952	4	18	34	2.820
		VAXIS		18.952	4	18	34	2.820
		DFPAXIS		18.952	4	18	34	2.820
		NSAXIS		18.952	4	18	34	2.820
	SC		18.952	18.952	4	18	34	2.820
	VSC		18.952	18.952	4	18	34	2.820
	DFPSC		19.429	19.429	4	20	34	2.344
	NSSC		18.952	18.952	4	18	34	2.820
	200 MHz	AXIS	21.772	19.143	4.5	16.5	34.5	2.629
	VAXIS		19.143	19.143	4.5	16.5	34.5	2.629
	DFPAXIS		19.143	19.143	4.5	16.5	34.5	2.629
	NSAXIS		19.143	19.143	4.5	16.5	34.5	2.629
SC		19.143	19.143	4.5	16.5	34.5	2.629	
VSC		19.143	19.143	4.5	16.5	34.5	2.629	
DFPSC		19.143	19.143	4.5	16.5	34.5	2.629	
NSSC		19.143	19.143	4.5	16.5	34.5	2.629	

Table 5.8: Numerical results for the AXIS algorithms in the smaller data size case for set IR-2.



Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L/2	20 MHz	AXIS	21.77	NaN	0	NaN	30	NaN
		VAXIS		NaN	0	NaN	30	NaN
		DFPAXIS		NaN	0	NaN	30	NaN
		NSAXIS		NaN	0	NaN	30	NaN
		SC		NaN	0	NaN	30	NaN
		VSC		NaN	0	NaN	30	NaN
		DFPSC		NaN	0	NaN	30	NaN
	100 MHz	NSSC		NaN	0	NaN	30	NaN
		AXIS	21.772	16.714	6	18	36	5.058
		VAXIS		16.714	6	18	36	5.058
		DFPAXIS		16.714	6	18	36	5.058
		NSAXIS		16	6	18	36	5.772
		SC		16.714	6	18	36	5.058
		VSC		16.714	6	18	36	5.058
150 MHz	DFPSC		16	6	18	36	5.772	
	NSSC		16.714	6	18	36	5.058	
	AXIS	21.772	17.143	8	18	36	4.629	
	VAXIS		17.143	8	18	36	4.629	
	DFPAXIS		17.143	8	18	36	4.629	
	NSAXIS		17.143	8	18	36	4.629	
	SC		17.143	8	18	36	4.629	
200 MHz	VSC		17.143	8	18	36	4.629	
	DFPSC		17.143	8	18	36	4.629	
	NSSC		16.667	8	18	36	5.106	
	AXIS	21.772	12.929	7.5	15	19.5	8.844	
	VAXIS		12.929	7.5	15	19.5	8.844	
	DFPAXIS		12.929	7.5	15	19.5	8.844	
	NSAXIS		12.929	7.5	15	19.5	8.844	
SC		12.929	7.5	15	19.5	8.844		
VSC		12.929	7.5	15	19.5	8.844		
DFPSC		12.929	7.5	15	19.5	8.844		
NSSC		12.929	7.5	15	19.5	8.844		

Table 5.9: Numerical results of the AXIS algorithms in the wrong model order,  $L/2$ , case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: L/2	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		20	0	15	30	1.772
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
	100 MHz	NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
		SC		21.714	6	24	33	0.058
		VSC		21.714	6	24	33	0.058
150 MHz	DFPSC		21.714	6	24	33	0.058	
	NSSC		21.714	6	24	33	0.058	
	AXIS	21.772	18.191	4	18	34	3.582	
	VAXIS		18.191	4	18	34	3.582	
	DFPAXIS		18.191	4	18	34	3.582	
	NSAXIS		18.191	4	18	34	3.582	
	SC		18.476	4	18	34	3.296	
200 MHz	VSC		18.476	4	18	34	3.296	
	DFPSC		18.191	4	18	34	3.582	
	NSSC		18.476	4	18	34	3.296	
	AXIS	21.772	19.643	4.5	19.5	34.5	2.129	
	VAXIS		19.643	4.5	19.5	34.5	2.129	
	DFPAXIS		19.643	4.5	19.5	34.5	2.129	
	NSAXIS		19.857	4.5	19.5	34.5	1.915	
SC		19.643	4.5	19.5	34.5	2.129		
VSC		19.643	4.5	19.5	34.5	2.129		
DFPSC		20.143	4.5	19.5	34.5	1.629		
NSSC		19.643	4.5	19.5	34.5	2.129		

Table 5.10: Numerical results of the AXIS algorithms in the wrong model order,  $L/2$ , case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L-10	20 MHz	AXIS	21.772	NaN	0	NaN	30	NaN
		VAXIS		NaN	0	NaN	30	NaN
		DFPAXIS		NaN	0	NaN	30	NaN
		NSAXIS		NaN	0	NaN	30	NaN
		SC		NaN	0	NaN	30	NaN
		VSC		NaN	0	NaN	30	NaN
		DFPSC		NaN	0	NaN	30	NaN
	100 MHz	NSSC		NaN	0	NaN	30	NaN
		AXIS	21.772	16.714	6	18	36	5.058
		VAXIS		16.714	6	18	36	5.058
		DFPAXIS		17.143	6	18	36	4.629
		NSAXIS		16.714	6	18	36	5.058
		SC		17.143	6	18	36	4.629
		VSC		17.143	6	18	36	4.629
150 MHz	DFPSC		16.714	6	18	36	5.058	
	NSSC		17.143	6	18	36	4.629	
	AXIS	21.772	17.143	8	18	36	4.629	
	VAXIS		17.143	8	18	36	4.629	
	DFPAXIS		16.667	8	18	36	5.106	
	NSAXIS		16.667	8	18	36	5.106	
	SC		16.667	8	18	36	5.106	
200 MHz	VSC		16.667	8	18	36	5.106	
	DFPSC		17.143	8	18	36	4.629	
	NSSC		16.667	8	18	36	5.106	
	AXIS	21.772	12.929	7.5	15	19.5	8.844	
	VAXIS		12.929	7.5	15	19.5	8.844	
	DFPAXIS		12.929	7.5	15	19.5	8.844	
	NSAXIS		12.929	7.5	15	19.5	8.844	
200 MHz	SC		13.5	7.5	15	19.5	8.272	
	VSC		13.5	7.5	15	19.5	8.272	
	DFPSC		12.929	7.5	15	19.5	8.844	
	NSSC		13.5	7.5	15	19.5	8.272	

Table 5.11: Numerical results of the AXIS algorithms in the wrong model order,  $L = 10$ , case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: L-10	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		20	0	15	30	1.772
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
		NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
SC		21.714	6	24	33	0.058		
VSC		21.714	6	24	33	0.058		
DFPSC		21.714	6	24	33	0.058		
NSSC		21.714	6	24	33	0.058		
150 MHz		AXIS	21.772	18.476	4	18	34	3.296
		VAXIS		18.476	4	18	34	3.296
		DFPAXIS		18.476	4	18	34	3.296
		NSAXIS		19.047	4	18	34	2.725
		SC		19.810	4	20	34	1.962
		VSC		19.810	4	20	34	1.963
		DFPSC		18.476	4	18	34	3.296
		NSSC		19.810	4	20	34	1.963
		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		19.643	4.5	19.5	34.5	2.129
		NSAXIS		19.643	4.5	19.5	34.5	2.129
SC		19.643	4.5	19.5	34.5	2.129		
VSC		19.643	4.5	19.5	34.5	2.129		
DFPSC		19.643	4.5	19.5	34.5	2.129		
NSSC		19.643	4.5	19.5	34.5	2.129		

Table 5.12: Numerical results of the AXIS algorithms in the wrong model order,  $L = 10$ , case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: 2L	20 MHz	AXIS	21.772	15	0	15	30	6.772
		VAXIS		15	0	15	30	6.772
		DFPAXIS		16.429	0	15	45	5.344
		NSAXIS		15	0	15	30	6.772
		SC		15	0	15	30	6.772
		VSC		15	0	15	30	6.772
	100 MHz	DFPSC		15	0	15	30	6.772
		NSSC		15	0	15	30	6.772
		AXIS	21.772	17.143	6	18	36	4.629
		VAXIS		17.143	6	18	36	4.629
		DFPAXIS		17.143	6	18	36	4.629
		NSAXIS		17.143	6	18	36	4.629
150 MHz		SC		16.857	6	18	36	4.915
		VSC		16.857	6	18	36	4.915
		DFPSC		17.143	6	18	36	4.629
		NSSC		16.857	6	18	36	4.915
		AXIS	21.772	16.667	8	18	36	5.106
		VAXIS		16.667	8	18	36	5.106
		DFPAXIS		16.667	8	18	36	5.106
		NSAXIS		16.667	8	18	36	5.106
		SC		16.191	8	18	36	5.582
		VSC		16.191	8	18	36	5.582
		DFPSC		16.667	8	18	36	5.106
		NSSC		16.191	8	18	36	5.582
200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844	
	VAXIS		12.929	7.5	15	19.5	8.844	
	DFPAXIS		12.929	7.5	15	19.5	8.844	
	NSAXIS		12.929	7.5	15	19.5	8.844	
	SC		14.071	7.5	16.5	19.5	7.701	
	VSC		14.071	7.5	16.5	19.5	7.701	
DFPSC			13.5	7.5	15	19.5	8.272	
	NSSC		14.071	7.5	16.5	19.5	7.701	

Table 5.13: Numerical results of the AXIS algorithms in the wrong model order,  $2L$ , case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: 2L	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		18.571	0	15	30	3.201
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
		NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
		SC		22.286	6	24	33	0.514
		VSC		22.286	6	24	33	0.514
DFPSC		21.714	6	24	33	0.058		
NSSC		22.286	6	24	33	0.514		
150 MHz		AXIS	21.772	19.524	4	18	34	2.248
		VAXIS		19.524	4	18	34	2.248
		DFPAXIS		19.048	4	18	34	2.725
		NSAXIS		19.524	4	18	34	2.248
		SC		19.810	4	20	34	1.963
		VSC		19.810	4	20	34	1.963
		DFPSC		19.810	4	20	34	1.963
		NSSC		19.810	4	20	34	1.963
		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		20.143	4.5	19.5	34.5	1.629
		NSAXIS		19.643	4.5	19.5	34.5	2.129
		SC		19.357	4.5	19.5	34.5	2.415
		VSC		19.357	4.5	19.5	34.5	2.415
DFPSC		19.357	4.5	19.5	34.5	2.415		
NSSC		19.357	4.5	19.5	34.5	2.415		
200 MHz		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		20.143	4.5	19.5	34.5	1.629
		NSAXIS		19.643	4.5	19.5	34.5	2.129
		SC		19.357	4.5	19.5	34.5	2.415
		VSC		19.357	4.5	19.5	34.5	2.415
		DFPSC		19.357	4.5	19.5	34.5	2.415
		NSSC		19.357	4.5	19.5	34.5	2.415

Table 5.14: Numerical results of the AXIS algorithms in the wrong model order,  $2L$ , case for set IR-2.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L+10	20 MHz	AXIS	21.772	15	0	15	30	6.772
		VAXIS		15	0	15	30	6.772
		DFPAXIS		16.429	0	15	45	5.344
		NSAXIS		15	0	15	30	6.772
		SC		15	0	15	30	6.772
		VSC		15	0	15	30	6.772
		DFPSC		15	0	15	30	6.772
		NSSC		15	0	15	30	6.772
		AXIS	21.772	16.714	6	18	36	5.058
		VAXIS		16.714	6	18	36	5.058
		DFPAXIS		17.143	6	18	36	4.629
		NSAXIS		16.714	6	18	36	5.058
SC		17.143	6	18	36	4.629		
VSC		17.143	6	18	36	4.629		
DFPSC		16.714	6	18	36	5.058		
NSSC		17.143	6	18	36	4.629		
150 MHz		AXIS	21.772	16.667	8	18	36	5.106
		VAXIS		16.667	8	18	36	5.106
		DFPAXIS		16.667	8	18	36	5.106
		NSAXIS		16.667	8	18	36	5.106
		SC		16.667	8	18	36	5.106
		VSC		16.667	8	18	36	5.106
		DFPSC		16.667	8	18	36	5.106
		NSSC		16.667	8	18	36	5.106
		AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844
SC		13.5	7.5	15	19.5	8.272		
VSC		13.5	7.5	15	19.5	8.272		
DFPSC		12.929	7.5	15	19.5	8.844		
NSSC		12.929	7.5	15	19.5	8.844		
200 MHz		AXIS	21.772	12.929	7.5	15	19.5	8.844
		VAXIS		12.929	7.5	15	19.5	8.844
		DFPAXIS		12.929	7.5	15	19.5	8.844
		NSAXIS		12.929	7.5	15	19.5	8.844
		SC		13.5	7.5	15	19.5	8.272
		VSC		13.5	7.5	15	19.5	8.272
		DFPSC		12.929	7.5	15	19.5	8.844
		NSSC		12.929	7.5	15	19.5	8.844

Table 5.15: Numerical results of the AXIS algorithms in the wrong model order,  $L + 10$ , case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-2: L+10	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		18.571	0	15	30	3.201
		DFPAXIS		18.571	0	15	30	3.201
		NSAXIS		18.571	0	15	30	3.201
		SC		18.571	0	15	30	3.201
		VSC		18.571	0	15	30	3.201
		DFPSC		18.571	0	15	30	3.201
		NSSC		18.571	0	15	30	3.201
		AXIS	21.772	21.714	6	24	33	0.058
		VAXIS		21.714	6	24	33	0.058
		DFPAXIS		21.714	6	24	33	0.058
		NSAXIS		21.714	6	24	33	0.058
SC		22.143	6	24	33	0.371		
VSC		22.143	6	24	33	0.371		
DFPSC		21.714	6	24	33	0.058		
NSSC		21.714	6	24	33	0.058		
150 MHz		AXIS	21.772	18.476	4	18	34	3.296
		VAXIS		18.476	4	18	34	3.296
		DFPAXIS		18.476	4	18	34	3.296
		NSAXIS		18.476	4	18	34	3.296
		SC		19.810	4	20	34	1.963
		VSC		19.810	4	20	34	1.963
		DFPSC		18.476	4	18	34	3.296
		NSSC		19.810	4	20	34	1.963
		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		19.643	4.5	19.5	34.5	2.129
		NSAXIS		19.643	4.5	19.5	34.5	2.129
SC		19.357	4.5	19.5	34.5	2.415		
VSC		19.357	4.5	19.5	34.5	2.415		
DFPSC		19.643	4.5	19.5	34.5	2.129		
NSSC		19.643	4.5	19.5	34.5	2.129		
200 MHz		AXIS	21.772	19.643	4.5	19.5	34.5	2.129
		VAXIS		19.643	4.5	19.5	34.5	2.129
		DFPAXIS		19.643	4.5	19.5	34.5	2.129
		NSAXIS		19.643	4.5	19.5	34.5	2.129
		SC		19.357	4.5	19.5	34.5	2.415
		VSC		19.357	4.5	19.5	34.5	2.415
		DFPSC		19.643	4.5	19.5	34.5	2.129
		NSSC		19.643	4.5	19.5	34.5	2.129

Table 5.16: Numerical results of the AXIS algorithms in the wrong model order,  $L + 10$ , case for set IR-2.



Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Random Start	20 MHz	AXIS	21.772	15	0	15	30	6.772
		VAXIS		75	15	45	195	53.228
		DFPAXIS		59.286	15	15	195	37.514
		NSAXIS		15	0	15	30	6.772
		SC		125.714	0	135	300	103.942
		VSC		125.714	0	135	300	103.942
		DFPSC		107.857	0	60	300	86.085
		NonSSC		125.714	0	135	300	103.942
		AXIS	21.772	16	6	18	36	5.772
		VAXIS		58.571	6	33	159	36.799
		DFPAXIS		58.571	6	33	159	36.799
		NSAXIS		16	6	18	36	5.772
		SC		117.286	3	114	303	95.514
VSC		117.286	3	114	303	95.514		
DFPSC		75.286	6	51	258	53.514		
NSSC		114.143	3	111	303	92.371		
150 MHz		AXIS	21.772	16.476	8	18	36	5.296
		VAXIS		42.286	8	20	124	20.514
		DFPAXIS		44	8	20	124	22.228
		NSAXIS		16.476	8	18	36	5.296
		SC		116.667	6	102	258	94.894
		VSC		116.667	6	102	258	94.894
		DFPSC		112.476	6	116	224	90.704
		NSSC		116.667	6	102	258	94.895
		AXIS	21.772	13.429	7.5	15	19.5	8.344
		VAXIS		43.214	1.5	21	126	21.442
		DFPAXIS		43.214	1.5	21	126	21.442
		NSAXIS		13.429	7.5	15	19.5	8.344
		SC		85.714	0	69	295.5	63.942
VSC		85.714	0	69	295.5	63.942		
DFPSC		95.571	9	72	219	73.799		
NSSC		85.714	0	69	295.5	63.942		
200 MHz		AXIS	21.772	16.476	8	18	36	5.296
		VAXIS		42.286	8	20	124	20.514
		DFPAXIS		44	8	20	124	22.228
		NSAXIS		16.476	8	18	36	5.296
		SC		116.667	6	102	258	94.894
		VSC		116.667	6	102	258	94.894
		DFPSC		112.476	6	116	224	90.704
		NSSC		116.667	6	102	258	94.895
		AXIS	21.772	13.429	7.5	15	19.5	8.344
		VAXIS		43.214	1.5	21	126	21.442
		DFPAXIS		43.214	1.5	21	126	21.442
		NSAXIS		13.429	7.5	15	19.5	8.344
		SC		85.714	0	69	295.5	63.942
VSC		85.714	0	69	295.5	63.942		
DFPSC		95.571	9	72	219	73.799		
NSSC		85.714	0	69	295.5	63.942		

Table 5.17: Numerical results of the AXIS algorithms in the normalized random start case for set IR-1.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
		VAXIS		85	0	30	255	63.228
		DFPAXIS		97.143	0	90	285	75.371
		NSAXIS		18.571	0	15	30	3.201
		SC		97.143	0	75	345	75.371
		VSC		97.143	0	75	345	75.371
	100 MHz	DFPSC		85.714	0	60	240	63.942
		NSSC		97.143	0	75	345	75.371
		AXIS	21.772	19.857	3	21	33	1.915
		VAXIS		61.429	3	54	135	39.656
		DFPAXIS		61.429	3	54	135	39.656
		NSAXIS		19.857	3	21	33	1.915
150 MHz	200 MHz	SC		113.571	9	93	249	91.799
		VSC		113.571	9	93	249	91.799
		DFPSC		83.571	6	63	273	61.799
		NSSC		113.571	9	93	249	91.799
		AXIS	21.772	19.810	6	18	34	1.963
		VAXIS		70.381	2	82	152	48.609
	200 MHz	DFPAXIS		70.381	2	82	152	48.609
		NSAXIS		19.810	6	18	34	1.962
		SC		112.381	8	92	228	90.609
		VSC		112.381	8	92	228	90.609
		DFPSC		118.476	8	108	238	96.704
		NSSC		112.381	8	92	228	90.609
200 MHz	AXIS	21.772	18.357	4.5	16.5	40.5	3.415	
	VAXIS		70.143	4.5	70.5	132	48.371	
	DFPAXIS		70.143	4.5	70.5	132	48.371	
	NSAXIS		18.357	4.5	16.5	40.5	3.415	
	SC		117.643	9	84	280.5	95.871	
	VSC		117.643	9	84	280.5	95.871	
200 MHz	DFPSC		98.857	3	66	270	77.085	
	NSSC		117.643	9	84	280.5	95.871	

Table 5.18: Numerical results of the AXIS algorithms in the normalized random start case for set IR-2.

#### 5.4 Algorithm Comparison

All the algorithms discussed in this chapter are derivations of the first algorithm introduced, the AXIS algorithm. As a result, there is little difference in the visual results and within a meter difference in numerical distance measurements.

All these algorithms did an excellent job in computing the distances via simulation for the second impulse response set, IR-2 with errors less than three meters. It is interesting to note that the DFPAXIS algorithm typically performed the best in almost all scenarios. The AXIS and the NSAXIS algorithms provided reasonable results for the normalized random initial channel estimate, that were almost the same as their unit vector equivalents. The AXIS based algorithms also did well with the presence of more noise, with fewer data samples, and with the incorrect model order.

As these algorithms all provide similar results, the computational complexity might be of interest. The DFPSC algorithm requires the most additions and multiplies, about  $(N - L)(12M^2 * (L + 1)^2 + 10M(L + 1) + 1)$ , with  $N - L$  complex exponentials needing to be computed. The NSAXIS algorithm requires the least amount of operations,  $(N - L)(28M(L + 1) + 2)$ , and no complex exponentials are needed. A complete table of algorithm computational complexity can be found in Table 7.1.

CHAPTER 6  
THE “BRUTE FORCE” METHOD

An alternative approach explored uses the same modifications as the AXIS algorithms but is rearranged such that it is more like the SVD algorithm. Instead of trying to minimize the updates between iterations, it minimizes the CR shown in Equation (4.17)

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{X}\mathbf{h}^{(n+1)}\|_2^2 + \alpha\|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}}\mathbf{e}_d = 1. \end{aligned} \tag{6.1}$$

The way that the objective function is written uses all the data matrix,  $\mathbf{X}$  at once instead of in rows or blocks of data as in the other algorithms derived in the previous chapter, although a more adaptive version has also been explored. This algorithm was not derived to be efficient, but rather to try to force the data to comply and yield accurate channel estimates. The name for this method is called the “Brute Force” (BF) algorithm as a result.

### 6.1 Brute Force Derivation

The derivation of this algorithm, referred to as the “Brute Force” (BF) method, also starts by setting up the Lagrangian,

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}}\mathbf{X}^H\mathbf{X}\mathbf{h}^{(n+1)} + \alpha\|\mathbf{h}^{(n+1)*}\|_1 + \lambda(\mathbf{h}^{(n+1)\text{H}}\mathbf{e}_d - 1). \tag{6.2}$$

The update that minimizes the Lagrangian is obtained by taking the gradient with respect to the conjugate of the updated impulse response

$$\frac{d}{d\mathbf{h}^{(n+1)*}}\mathcal{L} = \mathbf{X}^H\mathbf{X}\mathbf{h}^{(n+1)} + \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n+1)}} + \lambda\mathbf{e}_d. \tag{6.3}$$

With the assumptions made in Chapter 3, upon setting (6.3) to 0, and solving for  $\mathbf{h}^{(n+1)}$  yields

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}^H \mathbf{X})^{-1}(\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{e}_d), \quad (6.4)$$

as an update equation.

Since there is only one constraint, and thus one Lagrange multiplier, a system of equations is not required. Replacing  $\mathbf{h}^{(n+1)}$  in the constraint with the update derived in (6.4) produces this definition for  $\tilde{\lambda}$

$$\tilde{\lambda} = \frac{-1 - [(\mathbf{X}^H \mathbf{X})^{-1}(\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}^H \mathbf{X})^{-1})^H \mathbf{e}_d}. \quad (6.5)$$

Exchanging  $\lambda$  in Equation (6.4) with the conjugate of the definition of Equation (6.5) provides the complete update equation used in this algorithm,

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}^H \mathbf{X})^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \frac{-1 - [(\mathbf{X}^H \mathbf{X})^{-1}(\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}^H \mathbf{X})^{-1})^H \mathbf{e}_d} \mathbf{e}_d \right). \quad (6.6)$$

The full derivation and code are located in Appendices A.2, and B.12 respectively.

### 6.1.1 Algorithm Advantages, Disadvantages and Implementation

The main advantage of this algorithm is only a single step is required to achieve the impulse response estimate, like the SVD algorithm. The major disadvantage to this algorithm is the matrix operations required. Both a matrix inverse, which is computationally expensive,  $\mathcal{O}((M(L+1))^3)$ , and matrix multiplication, which can also be computationally expensive,  $\mathcal{O}((M(L+1))^2(N-L))$ .

To implement this algorithm, the initial value of  $h^{(n)} = \mathbf{e}_d$ , just as for the ModAED and AXIS based algorithm. Although starting with a better initial estimate may provide better results, as there is only one step.

Most of the channel estimates resulting from the non-adaptive BF algorithm, as shown in Figures 6.1 and 6.2 do not appear to be very good estimates. The 20 MHz channel estimates are actually quite good, for both impulse response sets, however the others are

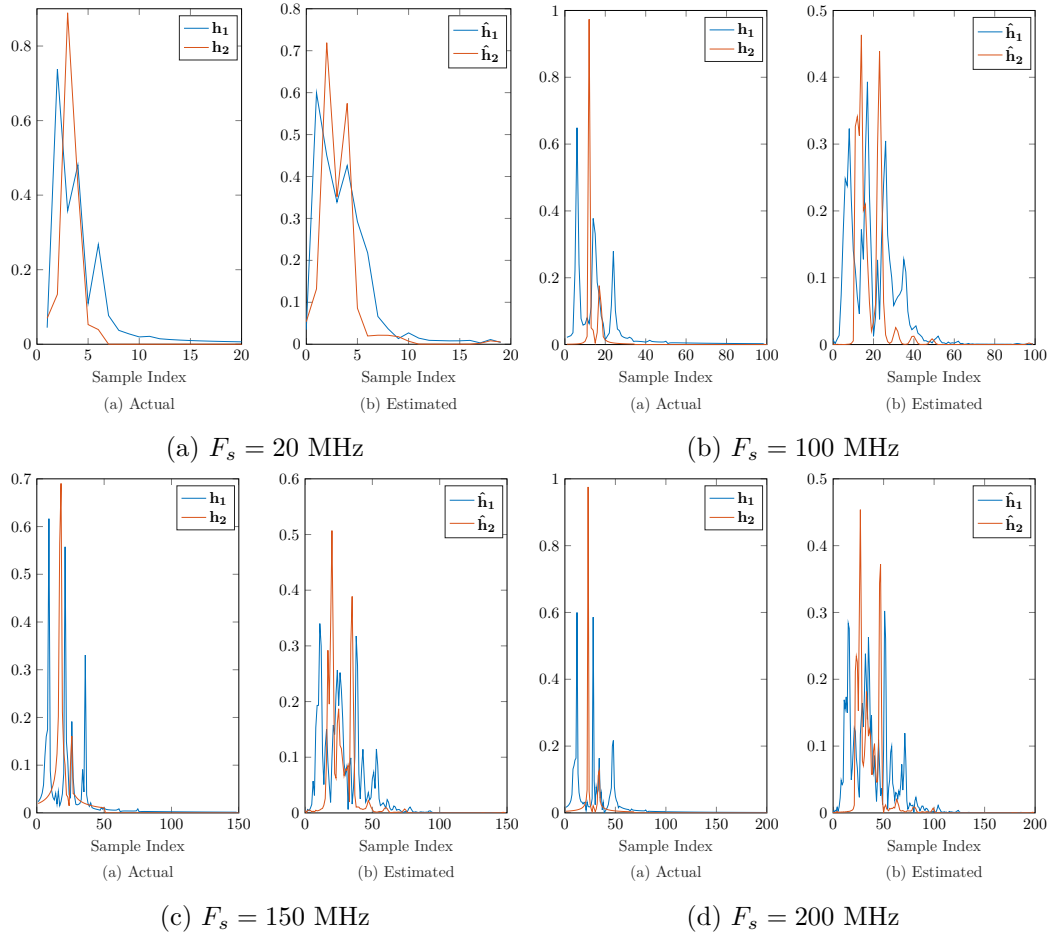


Fig. 6.1: Results of the non-adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

noisy, and seemingly random. As the majority of the results for the “Ideal” situation seemed random, the results for the other simulation situations are not shown, as there is not much to glean from them.

## 6.2 Variations

A couple variations were derived and tested as well to see if any improvement could be made to the BF algorithm. Some of these variations are similar to the variations for the AXIS algorithm.

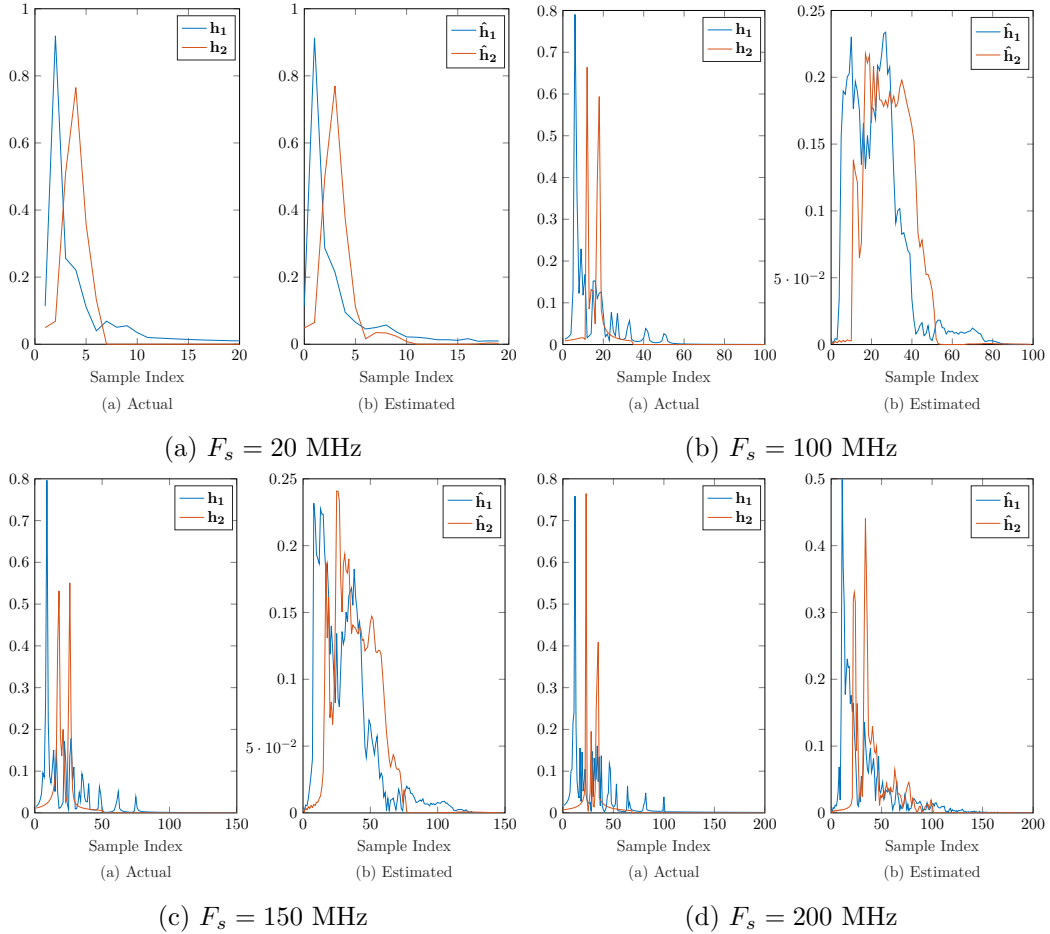


Fig. 6.2: Results of the non-adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

### 6.2.1 Adaptive Brute Force

The BF algorithm may seem an ideal approach due to its single step nature. However, it contains large matrix inverses, which are costly both in time and in computations, and does not show promising results. This algorithm can easily be converted into an adaptive algorithm, dividing the data matrix,  $\mathbf{X}$  into blocks of data can help combat the heavy computations, and theoretically guide the algorithm into providing a better estimate.

Let  $\mathbf{X}_r$  denote a section of rows, or a block of data, then (6.6) would become

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \eta \frac{-1 + [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d} \mathbf{e}_d \right). \quad (6.7)$$

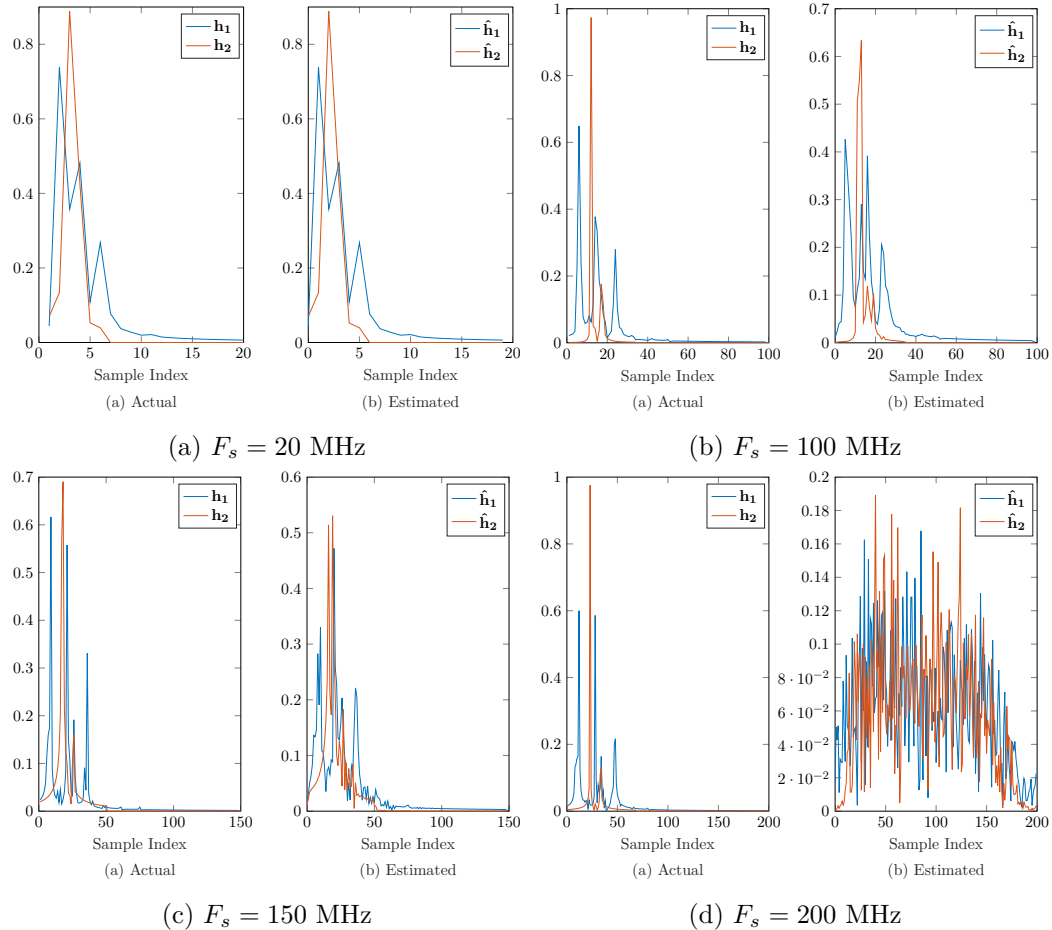


Fig. 6.3: Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Depending on the size of the blocks of data, this can take very few steps. The larger the block of data, the fewer steps are needed, however, the matrix inverse is larger, which would still be a lengthy process.

Channel estimate results for the ideal simulation can be seen in Figures 6.3 and 6.4. These results are not much better than the non-adaptive BF algorithm, but are improved. Impulse response set IR-1 appears to have all but the 200 MHz sampling rate producing good results. The 200 MHz sampling rate is quite random in comparison. Set IR-2 does not have as good of estimates as the first set, but there are some good peaks, if not in a very noisy channel estimate.



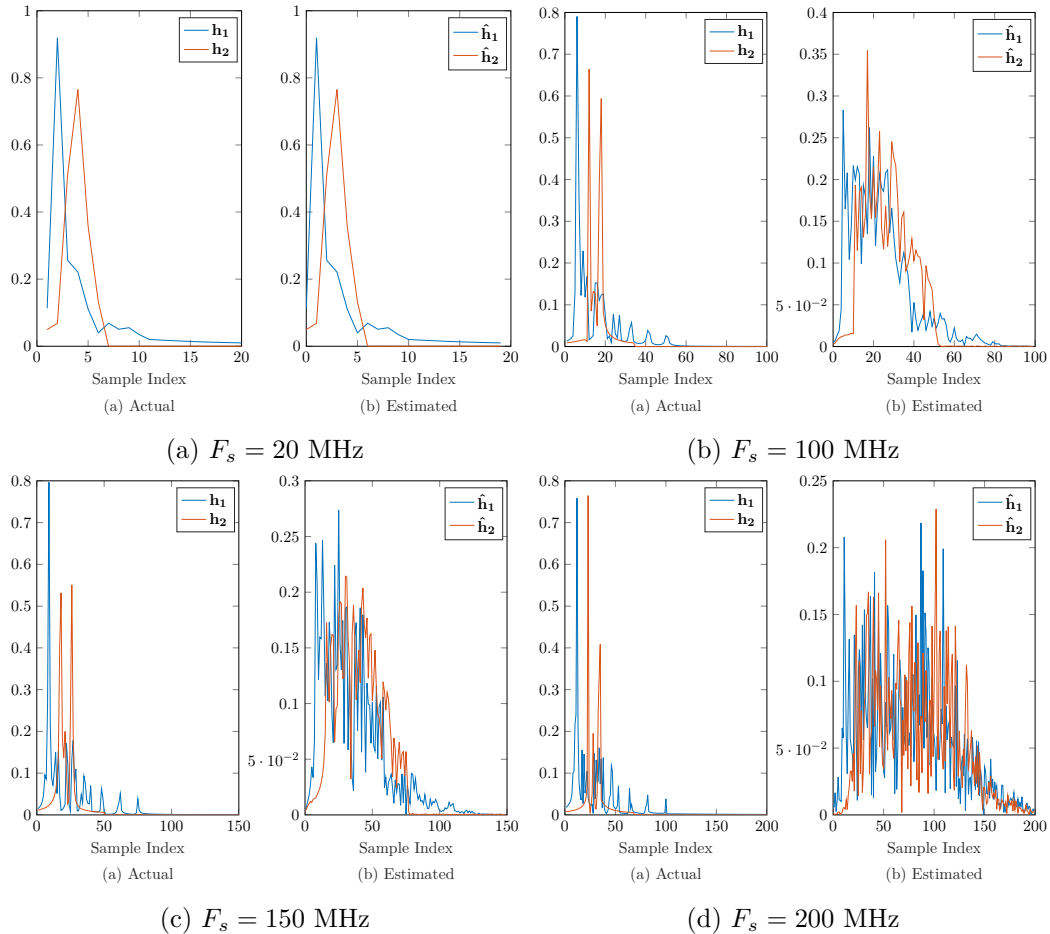


Fig. 6.4: Results of the Adaptive BF algorithm with 250,000 data samples, with  $SNR = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

As the adaptive results were a bit more promising, and not as computationally expensive, some nonideal conditions can be seen in Figures 6.5–6.9. Like the previous algorithms, the narrowband simulation did not produce a good estimate, there is little resolution between channels, as shown in Figure 6.5. Fewer data samples shows similarities with the ideal case. The smaller data size results show that some of the peaks have split, and not yet resolved completely; see Figure 6.6 for a comparison. Interestingly, Figure 6.7 more noise ( $SNR = 20$  dB) actually appears to yield noisy, but good channel estimates with distinct peaks. Not having the correct channel order does appear to affect this algorithm, as shown in Figure 6.8, involving more chaotic signals, that look random than not. What

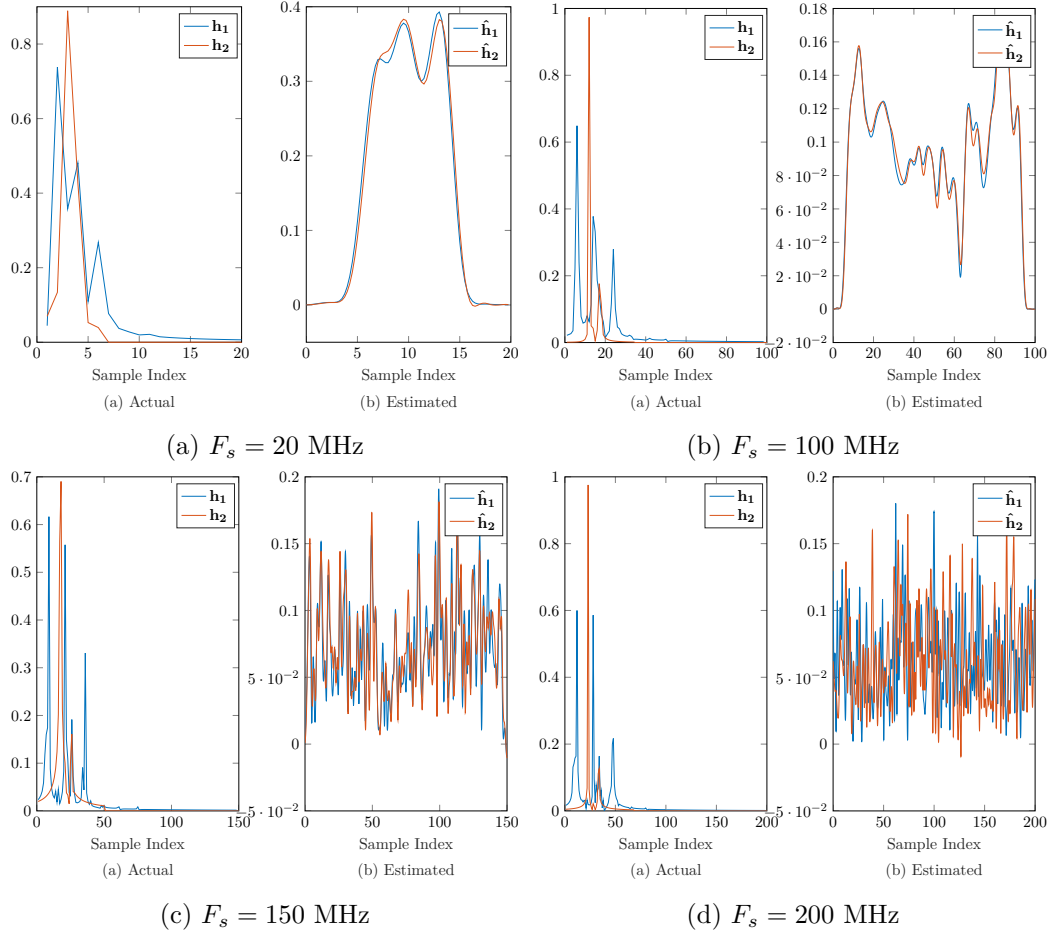


Fig. 6.5: Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

is interesting to note is that the normalized random start, as seen in Figure 6.9 does not appear to have a great impact on the channel estimation, producing similar results to those shown in Figure 6.3.

The complete derivation and code for this algorithm are located in Appendix A.2, and B.12 respectively. It should be noted that this is the exact same algorithm as the non-adaptive BF algorithm, the only difference between the two algorithms is which data matrix is being used: the block data matrix,  $\mathbf{X}_r$  (for the adaptive BF), or the full data matrix,  $\mathbf{X}$  (for the non-adaptive) data matrix. These can be switched interchangeably depending on which algorithm is desired.

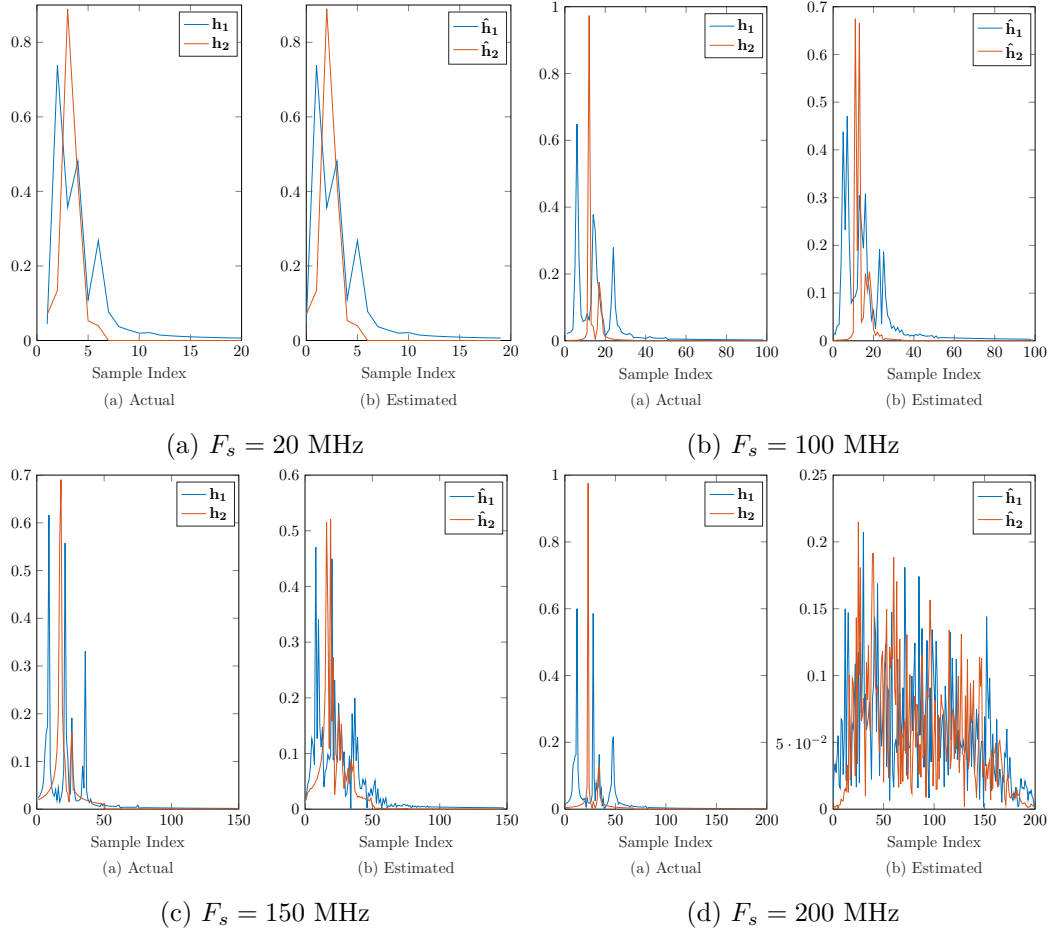


Fig. 6.6: Results of the Adaptive BF algorithm with 2,500 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

## 6.2.2 Varied Adaptive Brute Force

The first variation is minute, by merely changing the constraint to allow for a second variable, like the VAXIS algorithm shown in Chapter 5. As this algorithm involves variation, it is referred to as the Varied “Brute Force” (VBF) algorithm. The corresponding objective function to the VBF algorithm is shown as

$$\begin{aligned}
 & \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{X}_r \mathbf{h}^{(n+1)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
 & \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \zeta.
 \end{aligned} \tag{6.8}$$

As explained in the previous chapter, the variable  $\zeta$  is initialized as 1, but after the first

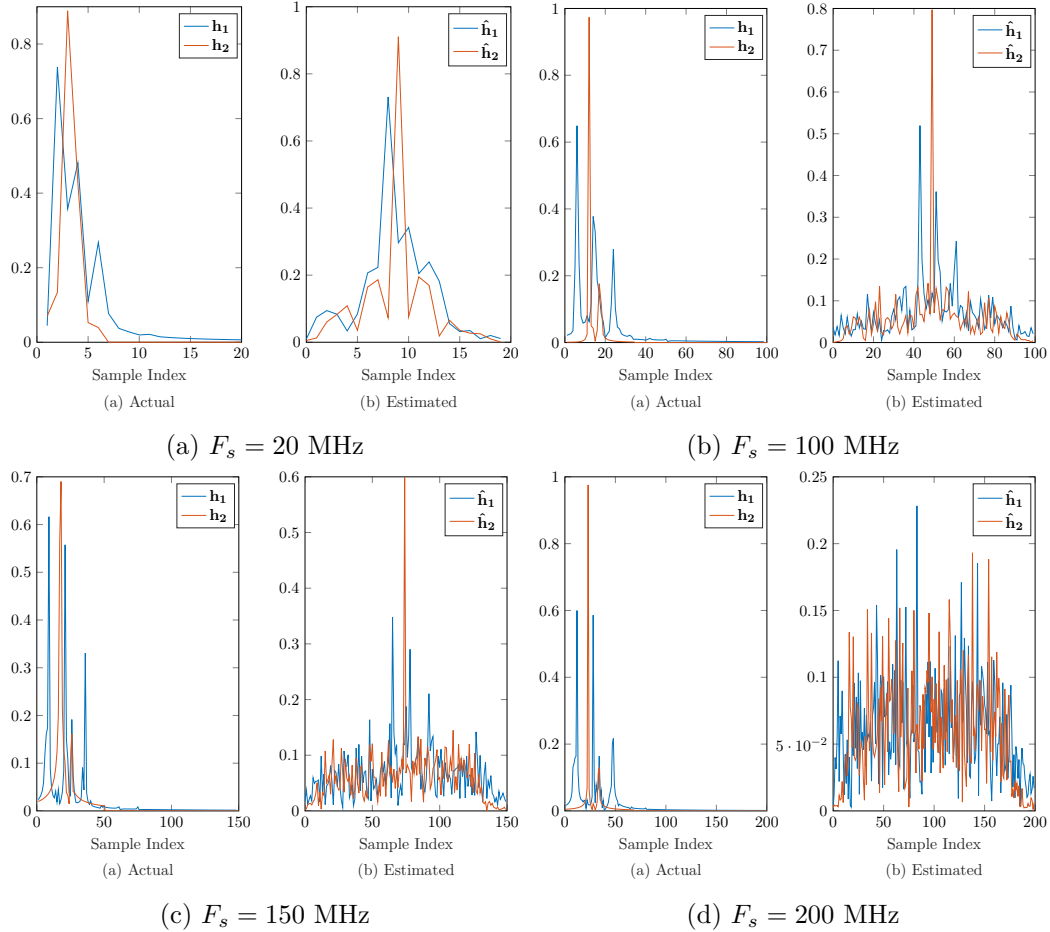


Fig. 6.7: Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 20 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

iteration,  $\zeta = \mathbf{h}_d^{(n+1)}$ . This small alteration only changes the derivation slightly in that  $\tilde{\lambda}$  has a  $\zeta$  instead of a 1. The update equation becomes

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \eta \frac{-\zeta + [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d} \mathbf{e}_d \right). \quad (6.9)$$

The code and complete derivation can be seen in Appendices B.13, and A.2.

This does not vary the results much, as can be seen in Figure 6.10 and Figure 6.11. The channel estimates are very similar to the estimates shown for the adaptive BF algorithm, and so no further VBF channel estimates will be shown for the other simulation situations.

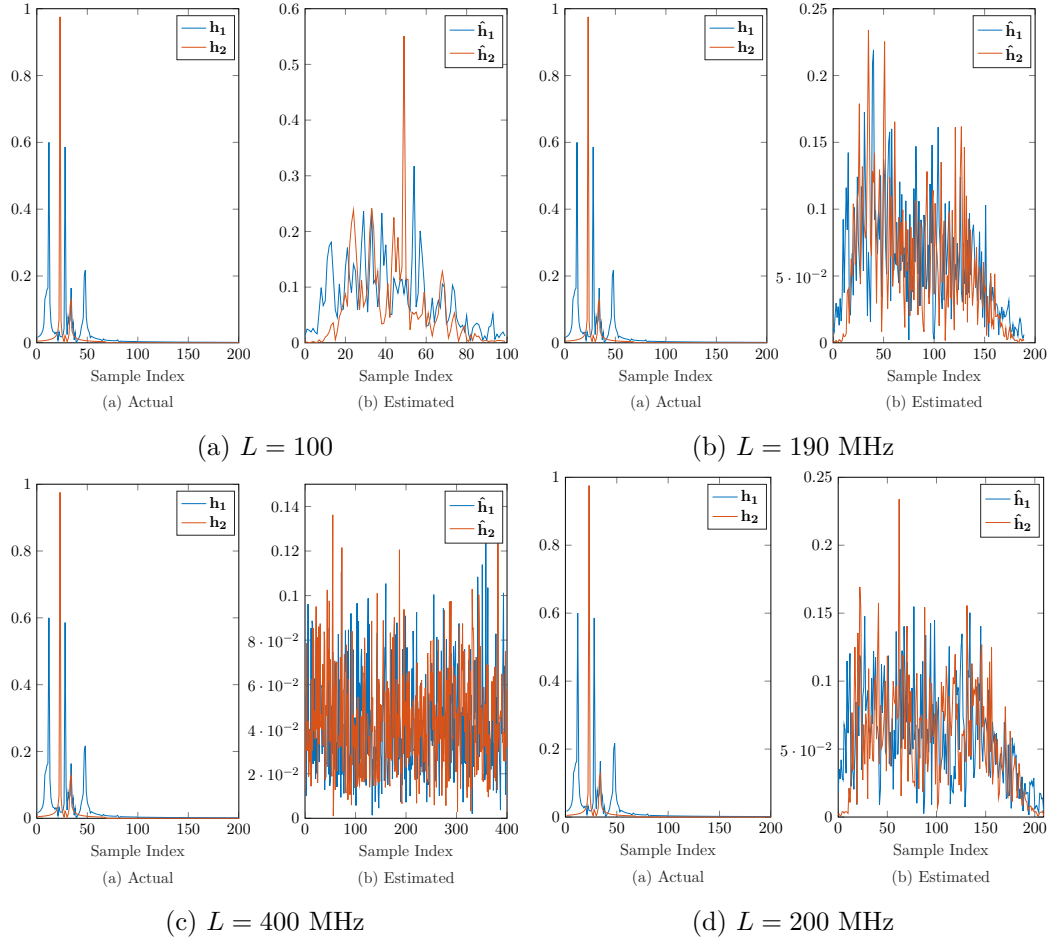


Fig. 6.8: Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a)  $L/2$ , (b)  $L - 10$ , (c)  $2L$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

### 6.2.3 Dual Fixed-Peak Brute Force

Again, like the DFPAXIS algorithm in Chapter 5, a secondary constraint was added to combat the asymmetric treatment of the transfer functions, due to the first constraint. This algorithm in turn is called the Dual Fixed-Peak BF (DFPBF) algorithm.

$$\begin{aligned}
 & \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{X}_r \mathbf{h}^{(n+1)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
 & \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{z} \\
 & \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c = y.
 \end{aligned} \tag{6.10}$$

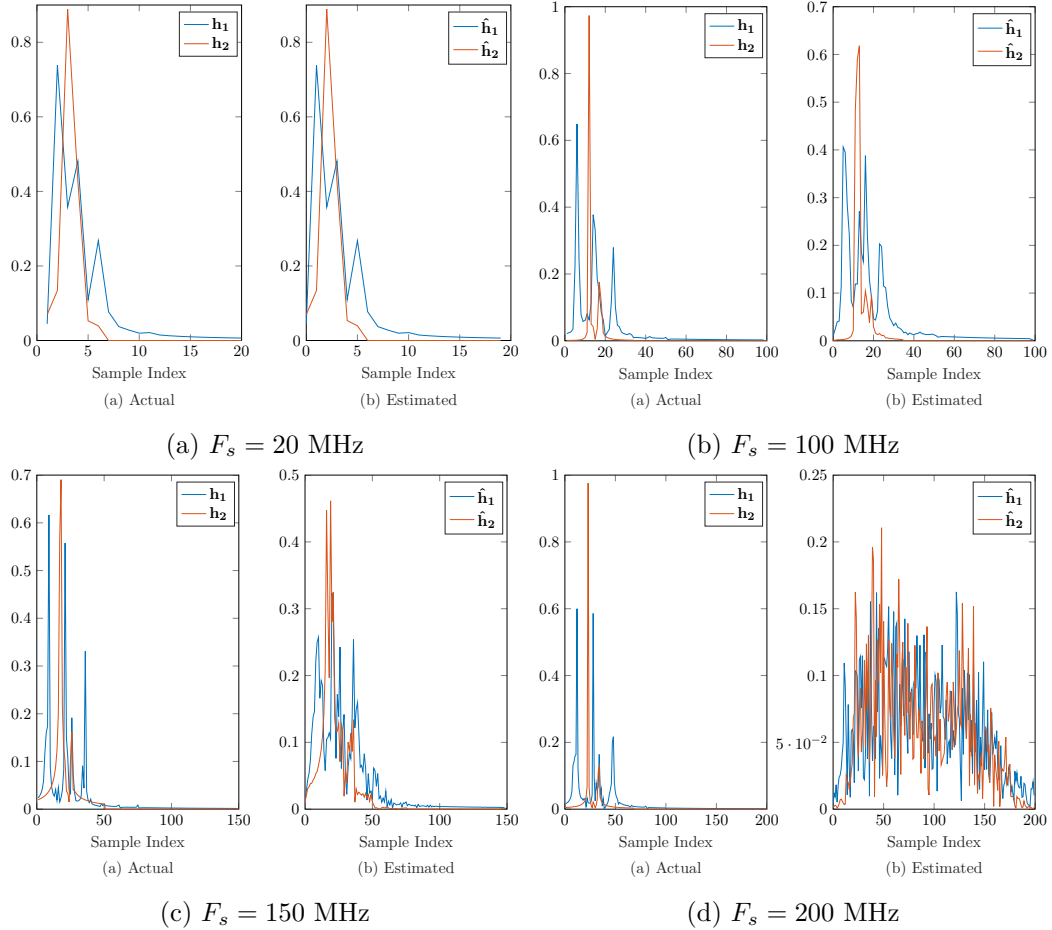


Fig. 6.9: Results of the Adaptive BF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1 with a normalized random start.

As with the DFPAXIS algorithm, this secondary constraint enforces a peak for the second impulse response, alleviate the focus on the first impulse response.

The derivation for this algorithm starts, like the others, by first implementing the Lagrangian,

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{X}_r^{\text{H}} \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \delta) + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y). \quad (6.11)$$

Then, the derivative is taken with respect to the conjugate of the transfer function estimates,

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{X}_r^{\text{H}} \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c. \quad (6.12)$$

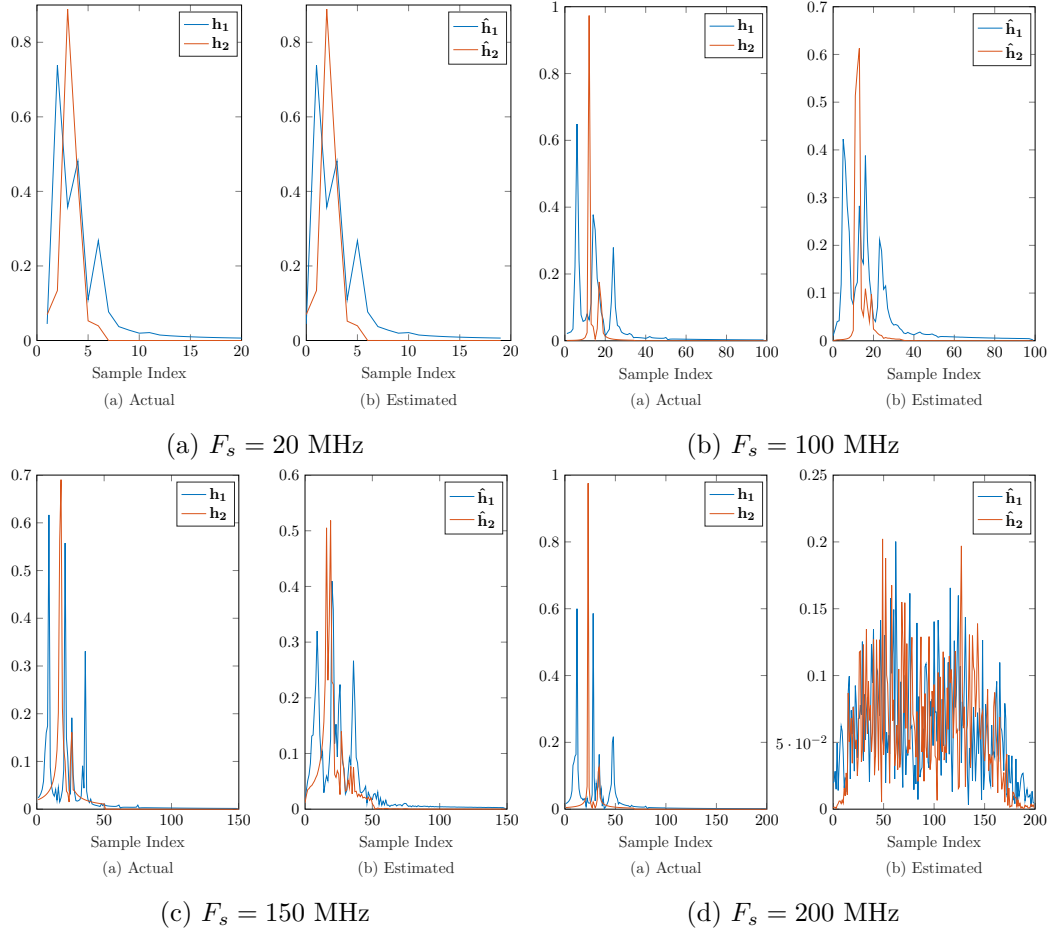


Fig. 6.10: Results of the VBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Setting (6.12) to 0 and solving for  $\mathbf{h}^{(n+1)}$  produces the update,

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c). \quad (6.13)$$

Replacing the update equation, (6.13), in place of  $\mathbf{h}^{(n+1)}$  in the constraints listed in (6.10) and solving produces a system of equations,

$$\begin{aligned} -\tilde{z} - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d &= \tilde{\lambda} \mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d + \tilde{\mu} \mathbf{e}_c^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d \\ -y - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c &= \tilde{\lambda} \mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_c + \tilde{\mu} \mathbf{e}_c^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_c \end{aligned} \quad (6.14)$$

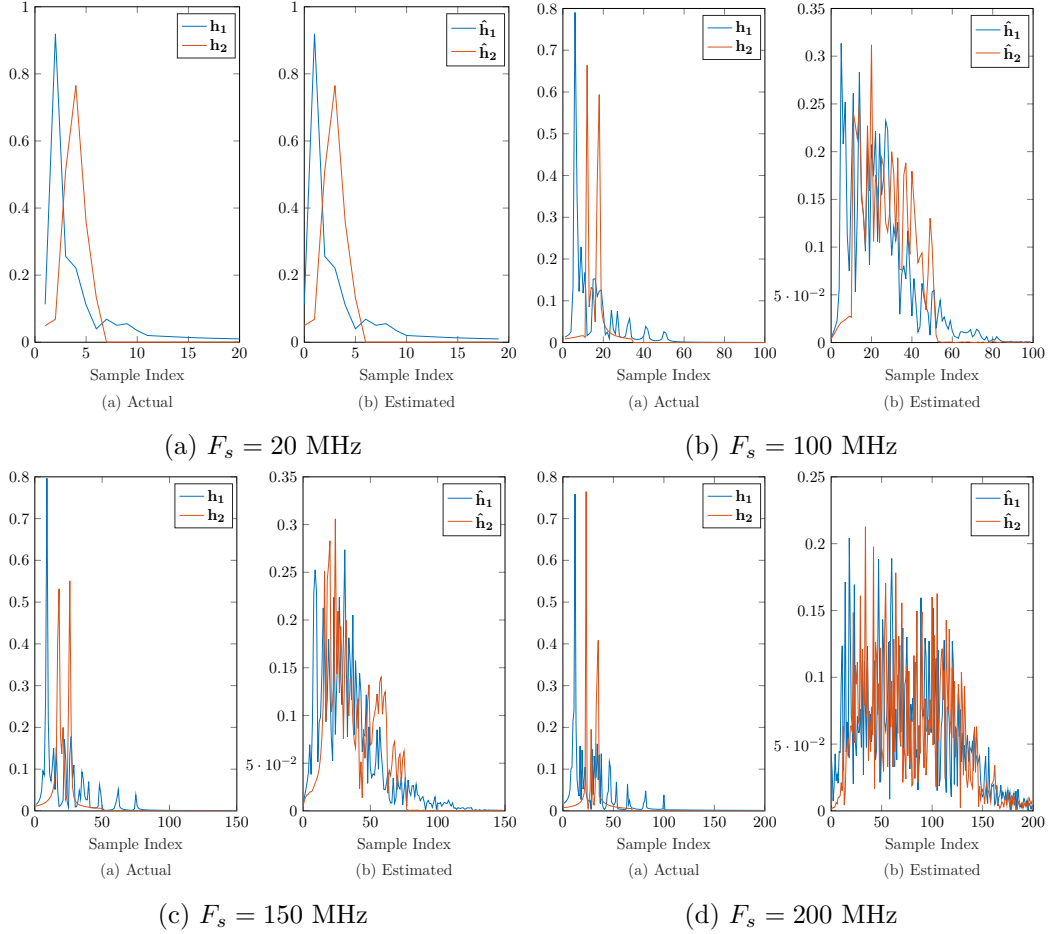


Fig. 6.11: Results of the VBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

$$\begin{bmatrix} \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d & \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \\ \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c & \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} -\tilde{\zeta} - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \\ -y - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c \end{bmatrix}. \quad (6.15)$$

The variables  $\tilde{\lambda}$  and  $\tilde{\mu}$  are defined as

$$\begin{aligned} \tilde{\lambda} &= \frac{(y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d - (\tilde{\zeta} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c}{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d + \epsilon} \\ \tilde{\mu} &= \frac{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c (\tilde{\zeta} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d (y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c}{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d + \epsilon} \end{aligned} \quad (6.16)$$



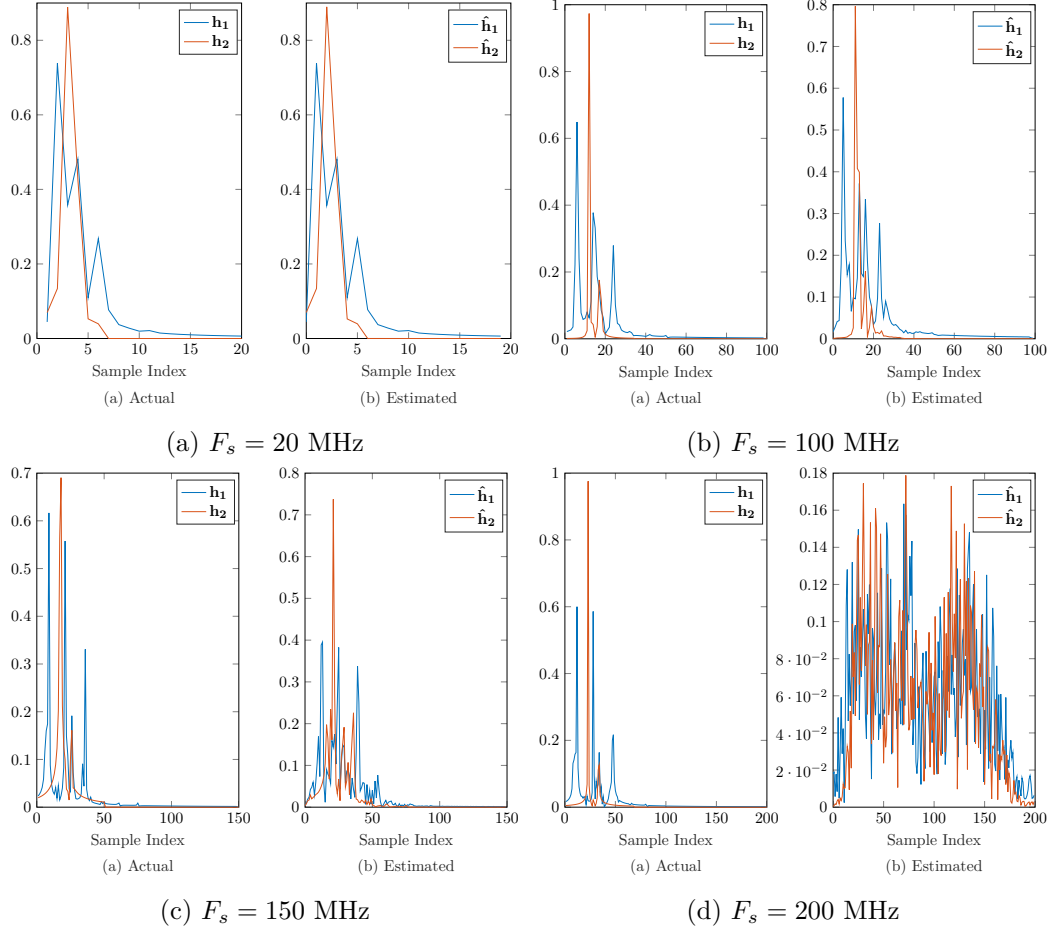


Fig. 6.12: Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

Replacing  $\lambda$  and  $\mu$  with the conjugates of (6.16) in (6.13), becomes

$$\begin{aligned}
 \mathbf{h}^{(n+1)} = & -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \eta \left( \right. \right. \\
 & \left( \frac{(y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)H}} (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c) \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d - (\tilde{y} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)H}} (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d) \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c}{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d + \epsilon} \right)^* \mathbf{e}_d + \\
 & \left. \left. \left( \frac{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c (\tilde{y} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)H}} (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d) - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d (y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)H}} (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c)}{\mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c - \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_c \mathbf{e}_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d + \epsilon} \right)^* \mathbf{e}_c \right) \right).
 \end{aligned} \tag{6.17}$$

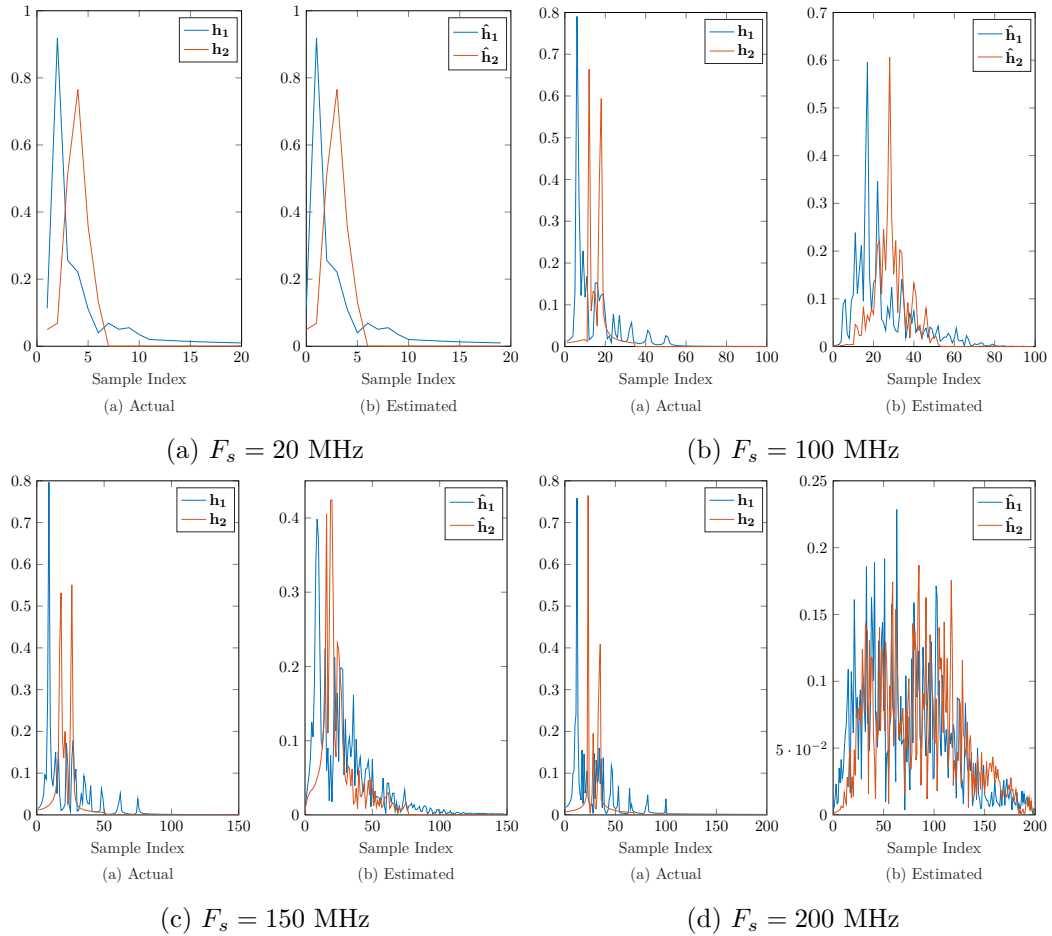


Fig. 6.13: Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-2, seed 0.

As explained in Chapter 5, the DFPBF algorithm is a secondary algorithm to try to eliminate the asymmetrical treatment of the transfer functions. And, as with the other DFP algorithms,  $\zeta$  is initialized to 1. The  $c^{th}$  element,  $y$  cannot get its value until the relationship between the impulse responses has been established. Thus, for the first iteration or two of the algorithm, one of the other algorithms must be used to initialize the channel estimates. Then, once the delay relationship has been established,  $y$  can take its value from the maximum peak of the second impulse response.

The impulse response estimates for the DFPBF algorithm can be seen in Figures 6.12–6.18. There are some similarities with the adaptive BF results, but there are some

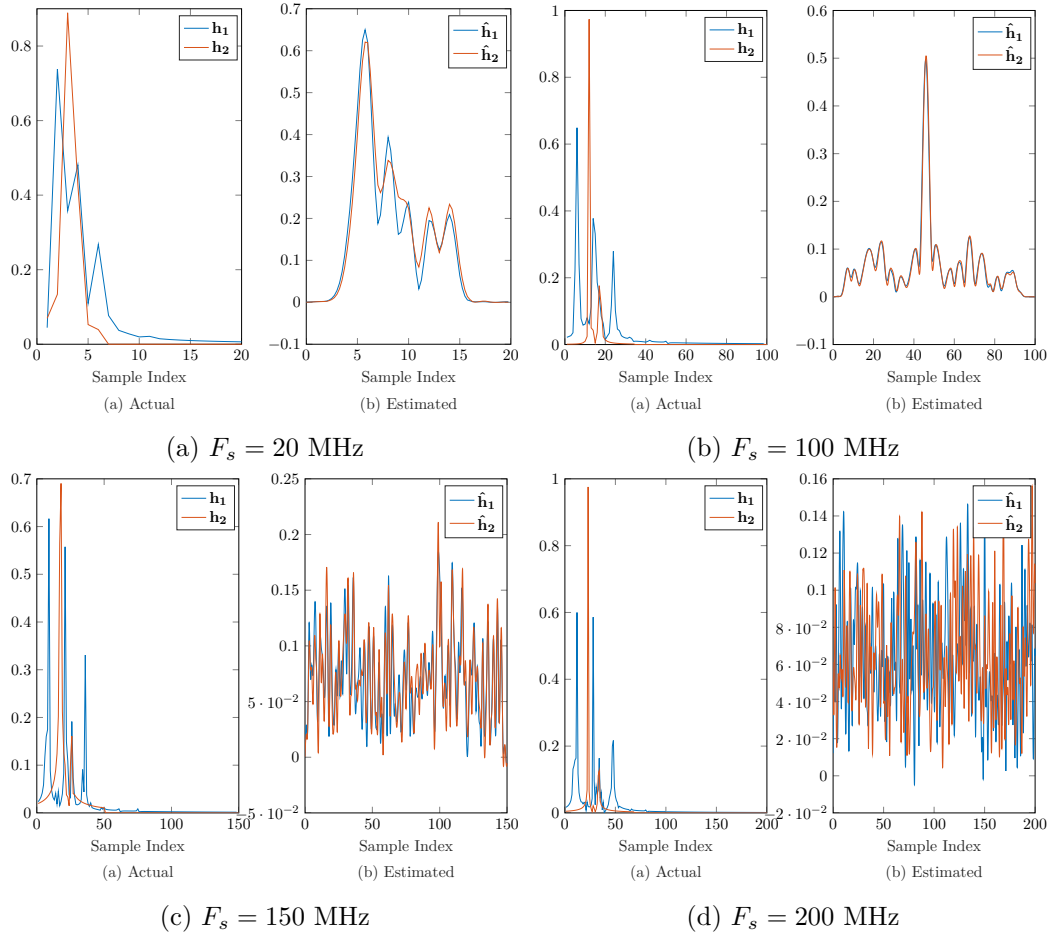


Fig. 6.14: Results of the DFPBFB algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for narrowband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

distinction, and so all of the simulation situation variations are shown here.

Figure 6.12 and Figure 6.13 show the ideal channel estimation results. Like with the adaptive BF results, the 200 MHz sampling rate estimates are not sparse. It is interesting to note that the channel estimates for the 100 MHz and 150 MHz sampling rates are also quite good for the second impulse response set, IR-2, unlike the adaptive BF results.

There is little to no resolution in the narrowband simulation, as is typical for all of these algorithms explored in this thesis, see Figure 6.14. With respect to the size of the data, the fewer samples that were used to estimate the channel estimates in Figure 6.15, actually show some improvement with respect to the BF algorithm. There appears to be

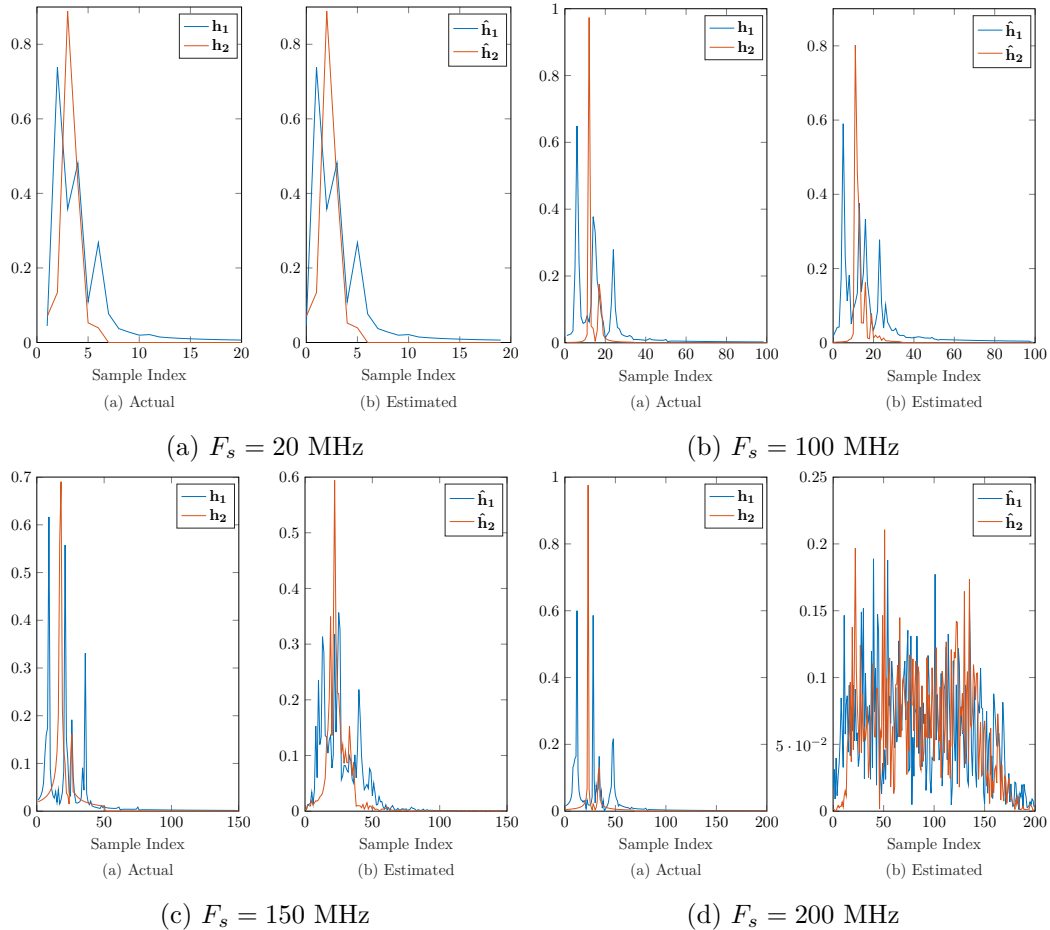


Fig. 6.15: Results of the DFPBF algorithm with 2,500 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

some reduction in the duplicate peaks seen in Figure 6.6. In comparison with the adaptive BF algorithm the noisy case (20 dB SNR) shows that the peaks have shifted from their placement in the adaptive BF algorithm, displayed in Figure 6.16. Figure 6.17 shows the effects of the incorrect model order. Underestimating by one half appears to produce good results, but the other incorrect model order estimates produce similar results to the ideal case.

A complete derivation of the DFPBF algorithm can be found in Appendix A.2. The code listing for the DFPBF is found in Appendix B.14.

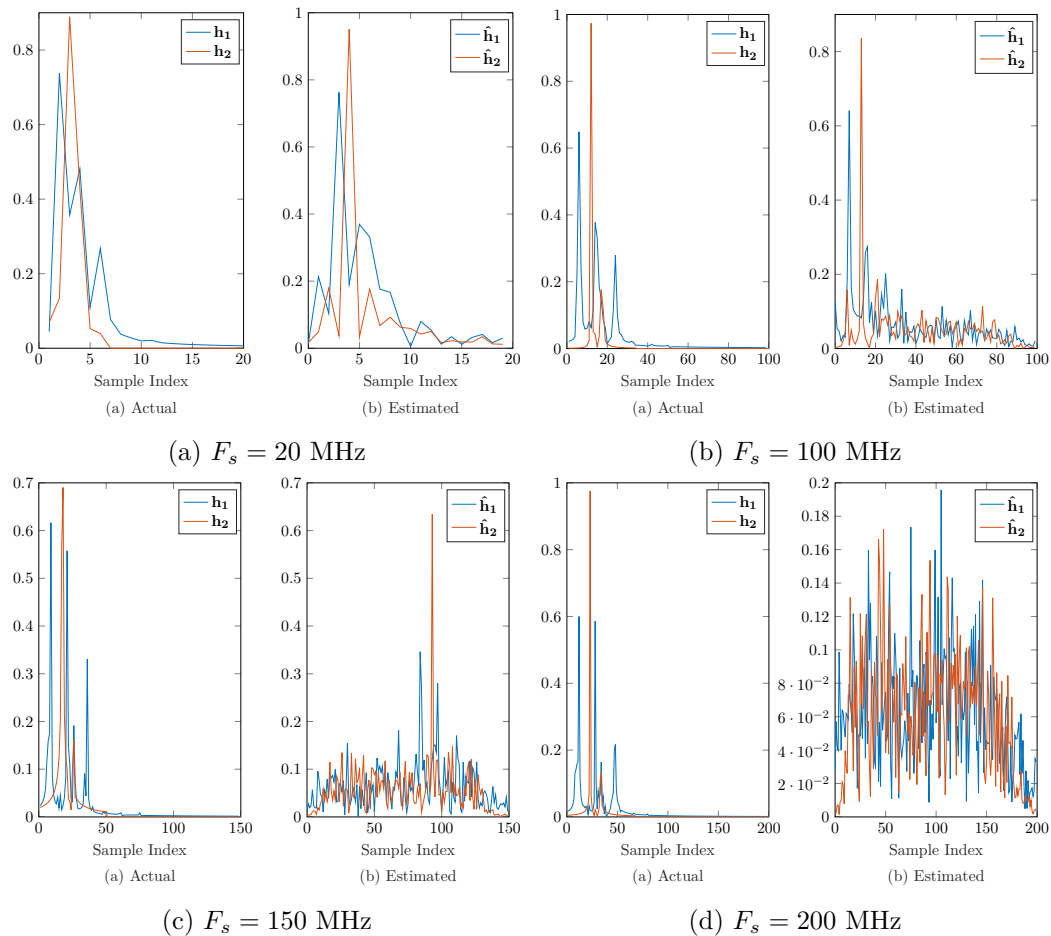


Fig. 6.16: Results of the DFPBFB algorithm with 250,000 data samples, with  $\text{SNR} = 20$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0.

### 6.3 Data Verification

As seen in both the Figures, and the tables below, the BF group of algorithms is not very reliable, and so the algorithms were only tested in simulation.

#### 6.3.1 Simulation

Like with the algorithms discussed in Chapters 4 and 5, these algorithms discussed in this chapter were also tested thoroughly in simulation. The numerical results of these simulations are discussed below.

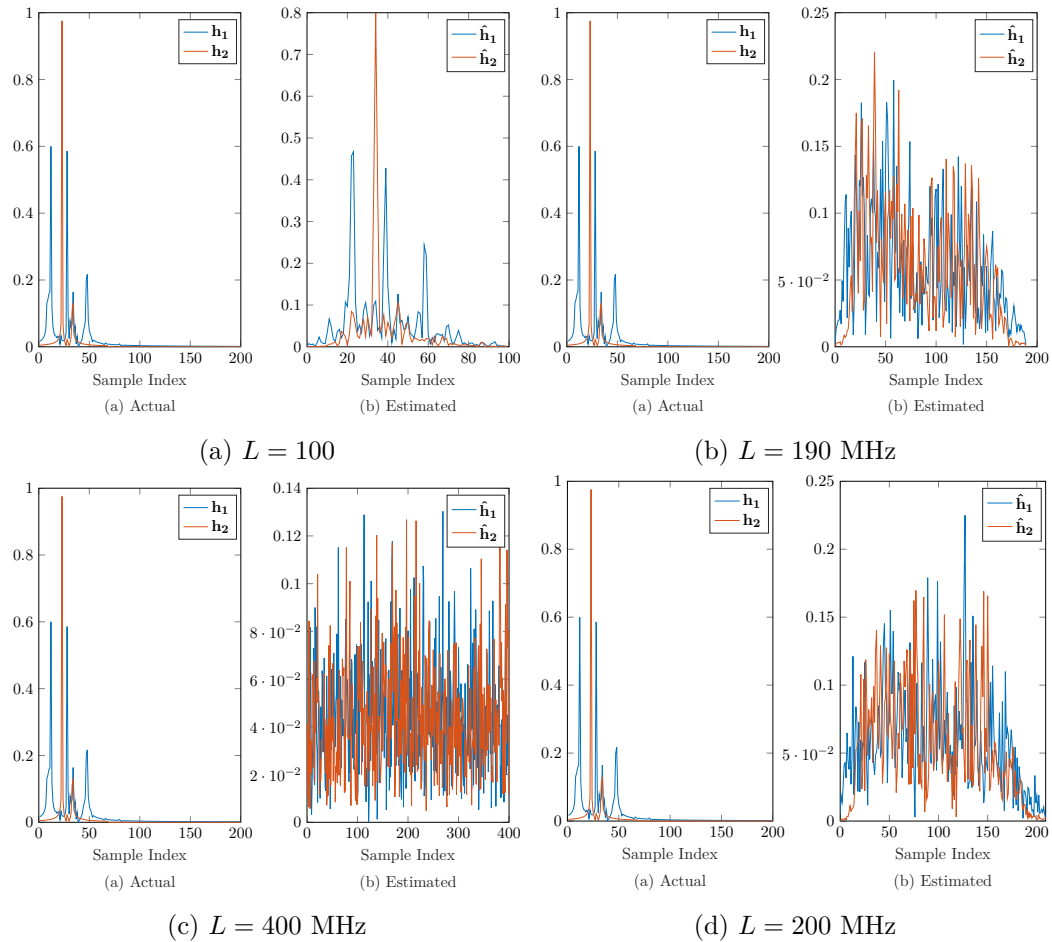


Fig. 6.17: Results of the DFPBF algorithm with 250,000 data samples, with SNR = 60 dB for wideband simulation at 200 MHz sampling rate for incorrect model order estimates (a)  $L/2$ , (b)  $L - 10$ , (c)  $2L$ , and (d)  $L + 10$ , for impulse response set IR-1, seed 0.

### “Ideal” Simulation

The numerical results for the “ideal” simulation can be seen in Table 6.1. For the ideal simulation, the 20 MHz sampling rate simulations are actually very good. This follows from the Figures shown, as the low sampling rate does a very good job estimating the channel estimates. It is interesting to note that for both impulse response sets the NBF algorithm typically has the best averaged estimate, but in comparing the channel estimates, do a poor job of estimating the channel. At 200 MHz sampling rate, the range for the adaptive BF algorithms is quite large, which makes the channel estimates quite large. At that higher frequency sampling rate, the results are very chaotic as shown in the Figures, so this large

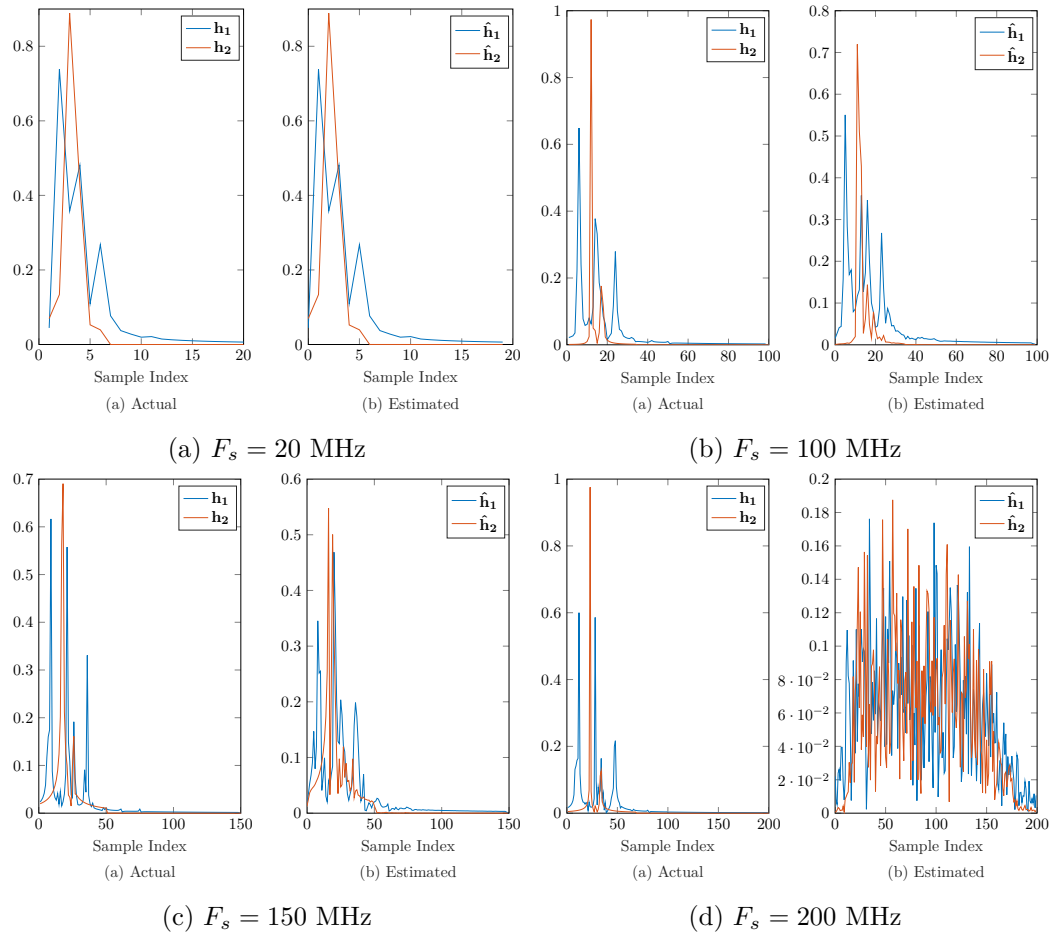


Fig. 6.18: Results of the DFPBF algorithm with 250,000 data samples, with  $\text{SNR} = 60$  dB, at various sampling rates for wideband simulation, (a) 20 MHz, (b) 100 MHz, (c) 150 MHz, and (d) 200 MHz, for impulse response set IR-1, seed 0 with a normalized random start.

range makes sense. And as observed in other chapters, impulse response set IR-2 typically had better results.

### Signal Bandwidth

The narrowband results, shown in Table 6.2, involve a mixture of overly large distance estimates, and NaN estimates. NaN estimates typically occur when there is no definitive peak in one or either channel. This could happen if the channel estimates deteriorate into normalized random noise. Looking at the range of the estimates, some have a maximum distance estimate of greater than 10000 m, and a minimum distance greater than the actual

distance estimate, meaning that there is no possibility of estimating the distance correctly, or even closely.

### Noise

Table 6.3 shows the noisier estimates, with  $SNR = 20$  dB, actually have some decent estimates for impulse response set IR-1, and fairly good estimates for set IR-2. Interestingly, for set IR-2, the estimates are improved from the “ideal” case. But like the “ideal” case, at the 200 MHz sampling rate most of the algorithms performs poorly, with a wide range of estimates, although not as large as found the narrowband results. It should be noted that the NBF algorithm does provide reasonable distance estimates for the 200 MHz.

### Sample Size

With fewer data samples, the algorithms still maintain the performance shown in the ideal case, as seen in Table 6.4. As with the noisy case and the ideal case, NBF yields better more consistent distance estimates. The adaptive BF algorithms do have decent results excluding the 200 MHz sampling rate (as expected).

### Model Order

As the model order is crucial to the SVD algorithm, it is also crucial to the BF algorithms. This is logical as the BF algorithms involve a similar setup to the SVD algorithm. Different model order estimation results can be found in Tables 6.5–6.8.

The BF algorithm is not as affected by underestimating the model order as it is by overestimating. Results for underestimating the model order can be seen in Table 6.5 and Table 6.6. In these results, there are a few NaNs, especially for the 20 MHz sampling rate. This could be due to there not being enough room to develop peaks, or because the estimate degenerated into a normalized random signal before iterations were finished. Underestimating the channel order does affect the range of the distance estimates. Although there appear to be some good estimates, those few estimates have a large range of distance



estimate. Once again, the NBF algorithm has the best estimate most of the sampling rates, and a consistent estimate for the others.

Overestimating the model order produces many more NaNs, and very large distance estimates, as can be seen in Tables 6.7 and 6.8. Doubling the model order yields only NaNs and inaccurate overly large estimates. By merely increasing the overestimation by 10 samples, only the 20 MHz sampling rate is affected greatly, producing very large estimates with large range differences.

### **Initial Estimates**

Initializing the BF algorithms with a normalized random start can be seen in Table 6.9. The results for using a normalized random are more chaotic, showing a mixture of large estimates with some accurate estimates depending on the sampling rate. It should be noted that the NBF algorithm provides the most consistent estimates, although there are some wide range estimates associated with them.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Ideal	20 MHz	NBF	21.772	18.571	0	15	30	3.201
		ABF		15	15	15	15	6.772
		VBF		15	15	15	15	6.772
		DFPBF		15	15	15	15	6.772
	100 MHz	NBF	21.772	14.143	3	12	45	7.629
		ABF		14.571	0	15	30	7.201
		VBF		14.571	0	15	30	7.201
		DFPBF		14	3	15	36	7.772
	150 MHz	NBF		18.381	6	18	42	3.391
		ABF		14.381	2	14	36	7.391
		VBF		14.381	2	14	36	7.391
		DFPBF		15.905	8	18	36	5.867
200 MHz	NBF	21.772	17.643	0	15	36	4.129	
	ABF		31.429	1.5	19.5	124.5	9.656	
	VBF		31.429	1.5	19.5	124.5	9.656	
	DFPBF		40	3	21	112.5	18.228	
IR-2: Ideal	20 MHz	NBF	21.772	16.429	0	15	30	5.344
		ABF		17.857	0	15	30	3.915
		VBF		17.857	0	15	30	3.915
		DFPBF		17.857	0	15	30	3.915
	100 MHz	NBF	21.772	20.714	3	21	33	1.078
		ABF		20.714	3	15	72	1.058
		VBF		20.714	3	15	72	1.058
		DFPBF		15.571	3	15	33	6.201
	150 MHz	NBF	21.772	20.095	2	22	36	1.677
		ABF		22.571	6	22	42	0.799
		VBF		22.571	6	22	42	0.799
		DFPBF		18.571	4	22	34	3.201
200 MHz	NBF	21.772	23.357	0	24	57	1.585	
	ABF		53.071	0	52.5	133.5	31.299	
	VBF		53.071	0	52.5	133.5	31.299	
	DFPBF		52.357	0	48	145.5	30.585	

Table 6.1: Numerical results for the BF algorithms in the “ideal” case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Narrowband	20 MHz	NBF	21.772	71.429	0	0	240	49.656
		ABF		62.857	0	60	240	41.085
		VBF		62.857	0	60	240	41.085
		DFPBF		NaN	0	NaN	120	NaN
100 MHz	100 MHz	NBF	21.772	260	0	0	2700	238.228
		ABF		434.286	0	0	4320	412.514
		VBF		434.286	0	0	4320	412.514
		DFPBF		2.857	0	0	60	18.915
150 MHz	150 MHz	NBF	21.772	NaN	60	NaN	60	NaN
		ABF		2494.286	0	2580	6120	2472.514
		VBF		2494.286	0	2580	6120	2472.514
		DFPBF		1134.286	0	60	4800	1112.514
200 MHz	200 MHz	NBF	21.772	NaN	0	NaN	0	NaN
		ABF		5242.857	180	5520	10680	5221.085
		VBF		5242.857	180	5520	10680	5221.085
		DFPBF		3742.857	0	2820	8580	3721.085
IR-2: Narrowband	20 MHz	NBF	21.772	122.857	0	60	300	101.085
		ABF		68.571	0	0	420	46.799
		VBF		68.571	0	0	420	46.799
		DFPBF		NaN	0	NaN	60	NaN
100 MHz	100 MHz	NBF	21.772	597.143	0	0	4860	575.371
		ABF		200	0	60	3060	178.228
		VBF		200	0	60	3060	178.228
		DFPBF		NaN	0	NaN	60	NaN
150 MHz	150 MHz	NBF	21.772	NaN	0	NaN	60	NaN
		ABF		2302.857	0	2100	5940	2281.085
		VBF		2302.857	0	2100	5940	2281.085
		DFPBF		1980	0	2100	6840	1958.228
200 MHz	200 MHz	NBF	21.772	NaN	NaN	NaN	NaN	NaN
		ABF		3697.143	60	3060	11280	3675.371
		VBF		3697.143	60	3060	11280	3675.371
		DFPBF		NaN	0	NaN	10440	NaN

Table 6.2: Numerical results for the BF algorithms in the narrowband case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Noisy	20 MHz	NBF	21.772	14.286	0	15	30	7.486
		ABF		14.286	0	15	30	7.486
		VBF		14.286	0	15	30	7.486
		DFPBF		13.571	0	15	15	8.201
	100 MHz	NBF	21.772	15.857	6	18	36	5.915
		ABF		15.857	6	18	36	5.915
		VBF		15.857	6	18	36	5.915
		DFPBF		23.714	6	18	195	1.942
	150 MHz	NBF	21.772	16.095	8	18	36	5.677
		ABF		15.619	8	18	22	6.153
		VBF		15.619	8	18	22	6.153
		DFPBF		15.143	6	16	36	6.629
200 MHz	NBF	21.772	13.643	7.5	16.5	19.5	8.129	
	ABF		48.857	7.5	37.5	159	27.085	
	VBF		48.857	7.5	37.5	159	27.085	
	DFPBF		54.357	9	39	153	32.585	
IR-2: Noisy	20 MHz	NBF	21.772	20.714	0	30	45	1.058
		ABF		20	0	15	45	1.772
		VBF		20	0	15	45	1.772
		DFPBF		20	0	15	60	1.772
	100 MHz	NBF	21.772	20.857	3	24	33	0.9145
		ABF		19.857	3	21	33	1.915
		VBF		19.857	3	21	33	1.915
		DFPBF		21.286	3	24	39	0.486
	150 MHz	NBF	21.772	19.143	4	22	34	2.629
		ABF		18.476	4	18	34	3.296
		VBF		18.476	4	18	34	3.296
		DFPBF		20.857	0	18	74	0.915
200 MHz	NBF	21.772	19.929	4.5	22.5	34.5	1.844	
	ABF		48	1.5	39	148.5	26.228	
	VBF		48	1.5	39	148.5	26.228	
	DFPBF		63.571	1.5	51	159	41.799	

Table 6.3: Numerical results for the BF algorithms in the noisy case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Fewer Samples	20 MHz	NBF	21.772	15	15	15	15	6.772
		ABF		15	15	15	15	6.772
		VBF		15	15	15	15	6.772
		DFPBF		15	15	15	15	6.772
	100 MHz	NBF	21.772	16.429	6	18	39	5.344
		ABF		16.143	6	15	30	5.629
		VBF		16.143	6	15	30	5.629
		DFPBF		14	6	12	33	7.772
	150 MHz	NBF	21.772	17.143	2	16	44	4.629
		ABF		14.667	2	16	36	7.106
		VBF		14.667	2	16	36	7.106
		DFPBF		16	8	18	36	5.772
200 MHz	NBF	21.772	16.714	6	16.5	36	5.058	
	ABF		42.929	0	31.5	153	21.156	
	VBF		42.929	0	31.5	153	21.156	
	DFPBF		53.286	3	55.5	136.5	31.514	
IR-2: Fewer Samples	20 MHz	NBF	21.772	17.857	0	15	30	3.915
		ABF		17.857	0	15	30	3.915
		VBF		17.857	0	15	30	3.915
		DFPBF		17.857	0	15	30	3.915
	100 MHz	NBF	21.772	20.286	6	21	33	1.486
		ABF		19	3	18	45	2.772
		VBF		19	3	18	45	2.772
		DFPBF		16.286	3	15	33	5.486
	150 MHz	NBF	21.772	19.524	0	22	34	2.248
		ABF		23.524	4	24	58	1.752
		VBF		23.524	4	24	58	1.752
		DFPBF		17.905	2	20	34	3.867
200 MHz	NBF	21.772	23.286	0	24	46.5	1.514	
	ABF		33.357	1.5	30	84	11.585	
	VBF		33.357	1.5	30	84	11.585	
	DFPBF		52.714	4.5	37.5	126	30.942	

Table 6.4: Numerical results for the BF algorithms in the smaller data size case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$	
IR-1: L/2	20 MHz	NBF	21.772	NaN	0	NaN	NaN	NaN	
		ABF		NaN	0	NaN	NaN	NaN	
		VBF		NaN	0	NaN	NaN	NaN	
		DFPBF		NaN	0	NaN	NaN	NaN	
	100 MHz	NBF	21.772	19.429	6	18	18	45	2.344
		ABF		17.857	6	18	18	45	3.915
		VBF		17.857	6	18	18	45	3.915
		DFPBF		13.571	6	15	15	24	8.201
	150 MHz	NBF	21.772	14.952	6	18	18	22	6.820
		ABF		16.857	6	16	16	36	4.915
		VBF		16.857	6	16	16	36	4.915
		DFPBF		16.952	8	18	18	38	4.820
200 MHz	NBF	21.772	18.714	7.5	16.5	16.5	46.5	3.058	
	ABF		13.787	1.5	13.5	13.5	42	7.986	
	VBF		13.786	1.5	13.5	13.5	42	7.986	
	DFPBF		13.929	6	15	15	28.5	7.844	
IR-2: L/2	20 MHz	NBF	21.772	23.571	0	30	30	45	1.799
		ABF		23.571	0	30	30	45	1.799
		VBF		23.571	0	30	30	45	1.799
		DFPBF		NaN	0	NaN	NaN	30	NaN
	100 MHz	NBF	21.772	25.857	3	24	24	51	4.085
		ABF		26.429	0	27	27	51	4.656
		VBF		26.429	0	27	27	51	4.656
		DFPBF		17.571	3	18	18	33	4.201
	150 MHz	NBF	21.772	22.857	0	22	22	52	1.085
		ABF		21.238	0	16	16	50	0.534
		VBF		21.238	0	16	16	50	0.534
		DFPBF		13.810	2	14	14	32	7.963
200 MHz	NBF	21.772	20.857	1.5	18	18	54	0.915	
	ABF		19.143	1.5	16.5	16.5	49.5	2.629	
	VBF		19.143	1.5	16.5	16.5	49.5	2.629	
	DFPBF		13.571	3	15	15	27	8.201	

Table 6.5: Numerical results for the BF algorithms in the wrong model order,  $L/2$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L-10	20 MHz	NBF	21.772	NaN	0	NaN	30	NaN
		ABF		NaN	0	NaN	30	NaN
		VBF		NaN	0	NaN	30	NaN
		DFPBF		NaN	0	NaN	30	NaN
	100 MHz	NBF	21.772	20.143	6	18	117	1.629
		ABF		15.857	6	18	36	5.915
		VBF		15.857	6	18	36	5.915
		DFPBF		16.429	6	18	36	5.344
	150 MHz	NBF	21.772	15.905	4	18	38	5.867
		ABF		15.429	4	18	36	6.344
		VBF		15.429	4	18	36	6.344
		DFPBF		15.905	6	18	36	5.867
200 MHz	NBF	21.772	14.929	7.5	15	34.5	6.844	
	ABF		14.357	1.5	15	27	7.415	
	VBF		14.357	1.5	15	27	7.415	
	DFPBF		22.643	0	19.5	57	0.871	
IR-2: L-10	20 MHz	NBF	21.772	23.571	0	30	45	1.799
		ABF		22.857	0	30	45	1.085
		VBF		22.857	0	30	45	1.085
		DFPBF		NaN	0	NaN	30	NaN
	100 MHz	NBF	21.772	21	3	15	57	0.772
		ABF		14.143	0	15	27	7.629
		VBF		14.143	0	15	27	7.629
		DFPBF		17.143	3	18	33	4.629
	150 MHz	NBF	21.772	19.905	2	20	54	1.867
		ABF		17.905	2	18	50	3.867
		VBF		17.905	2	18	50	3.867
		DFPBF		16.191	4	16	42	5.582
200 MHz	NBF	21.772	22.357	1.5	22.5	63	0.585	
	ABF		33.786	4.5	27	120	12.014	
	VBF		33.786	4.5	27	120	12.014	
	DFPBF		38.214	3	31.5	109.5	16.442	

Table 6.6: Numerical results for the BF algorithms in the wrong model order,  $L = 10$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2L	20 MHz	NBF	21.772	162.143	15	225	285	140.371
		ABF		152.143	0	90	345	130.371
		VBF		152.143	0	90	345	130.371
		DFFBF		13.571	0	15	15	8.201
	100 MHz	NBF	21.772	133.143	6	99	285	111.371
		ABF		61.714	0	30	234	39.942
		VBF		61.714	0	30	234	39.942
		DFFBF		92.286	6	60	270	70.514
	150 MHz	NBF	21.772	NaN	4	NaN	314	NaN
		ABF		220.476	20	226	452	198.704
		VBF		220.476	20	226	452	198.704
		DFFBF		189.143	22	200	420	167.371
200 MHz	NBF	21.772	NaN	6	NaN	168	NaN	
	ABF		NaN	93	NaN	340.5	NaN	
	VBF		NaN	93	NaN	340.5	NaN	
	DFFBF		NaN	15	NaN	286.5	NaN	
IR-2: 2L	20 MHz	NBF	21.772	150	0	180	285	128.228
		ABF		146.429	0	120	315	124.656
		VBF		146.429	0	120	315	124.656
		DFFBF		19.2857	0	15	30	2.486
	100 MHz	NBF	21.772	132.857	3	111	303	111.085
		ABF		96.429	0	93	267	74.656
		VBF		96.429	0	93	267	74.656
		DFFBF		NaN	3	NaN	300	NaN
	150 MHz	NBF	21.772	NaN	4	NaN	302	NaN
		ABF		NaN	4	NaN	446	NaN
		VBF		NaN	4	NaN	446	NaN
		DFFBF		189.714	34	92	512	167.942
200 MHz	NBF	21.772	NaN	NaN	NaN	NaN	NaN	
	ABF		NaN	109.5	NaN	507	NaN	
	VBF		NaN	109.5	NaN	507	NaN	
	DFFBF		NaN	7.5	NaN	534	NaN	

Table 6.7: Numerical results for the BF algorithms in the wrong model order, 2L, case.



Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: L+10	20 MHz	NBF	21.772	65.714	0	60	135	43.942
		ABF		100.714	0	135	195	78.942
		VBF		100.714	0	135	195	78.942
		DFPBF		NaN	0	NaN	15	NaN
	100 MHz	NBF	21.772	21.429	3	18	66	0.344
		ABF		18.857	0	18	45	2.915
		VBF		18.857	0	18	45	2.915
		DFPBF		15.143	0	18	36	6.629
	150 MHz	NBF	21.772	20.191	2	14	62	1.582
		ABF		15.905	0	14	38	5.867
		VBF		15.905	0	14	38	5.867
		DFPBF		13.238	0	16	22	8.534
200 MHz <sup>?</sup>	NBF	21.772	20.286	1.5	13.5	63	1.486	
	ABF		76.5	6	66	136.5	54.728	
	VBF		76.5	6	66	136.5	54.728	
	DFPBF		43.5	1.5	33	145.5	21.728	
IR-2: L+10	20 MHz	NBF	21.772	70	0	75	150	48.228
		ABF		56.429	0	30	165	34.656
		VBF		56.429	0	30	165	34.656
		DFPBF		20	0	15	30	1.772
	100 MHz	NBF	21.772	26.571	3	24	57	4.799
		ABF		26.143	0	18	66	4.371
		VBF		26.143	0	18	66	4.371
		DFPBF		19.857	6	15	39	1.915
	150 MHz	NBF	21.772	24.952	0	28	56	3.180
		ABF		18.667	4	16	44	3.106
		VBF		18.667	4	16	44	3.106
		DFPBF		14.857	2	14	52	6.915
200 MHz	NBF	21.772	26.214	1.5	27	55.5	4.442	
	ABF		53.714	6	49.5	120	31.942	
	VBF		53.714	6	49.5	120	31.942	
	DFPBF		58.429	3	55.5	124.5	36.656	

Table 6.8: Numerical results for the BF algorithms in the wrong model order,  $L + 10$ , case.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	20 MHz	NBF	21.772	33.571	0	30	90	11.799
		ABF	21.772	112.143	0	135	195	90.372
		VBF	21.772	112.143	0	135	195	90.372
		DFFBF	21.772	15	0	15	30	6.772
	100 MHz	NBF	21.772	21.714	6	21	36	0.058
		ABF	21.772	20.571	0	21	48	1.201
		VBF	21.772	20.571	0	21	48	1.201
		DFFBF	21.772	15.571	6	18	36	6.201
	150 MHz	NBF	21.772	23.810	2	20	68	2.037
		ABF	21.772	20.667	2	18	44	1.106
		VBF	21.772	20.667	2	18	44	1.106
		DFFBF	21.772	17.905	4	18	60	3.867
200 MHz	NBF	21.772	17.286	7.5	15	61.5	4.486	
	ABF	21.772	48.286	4.5	27	150	26.514	
	VBF	21.772	48.286	4.5	27	150	26.514	
	DFFBF	21.772	64.286	7.5	69	172.5	42.514	
IR-2: Random Start	20 MHz	NBF	21.772	35	0	30	105	13.228
		ABF	21.772	84.286	15	105	180	62.514
		VBF	21.772	84.286	15	105	180	62.514
		DFFBF	21.772	17.857	0	15	30	3.915
	100 MHz	NBF	21.772	21.286	0	18	48	0.486
		ABF	21.772	30.857	6	27	81	9.085
		VBF	21.772	30.857	6	27	81	9.085
		DFFBF	21.772	19	3	21	33	2.772
	150 MHz	NBF	21.772	22.191	2	22	50	0.418
		ABF	21.772	24	4	22	44	2.228
		VBF	21.772	24	4	22	44	2.228
		DFFBF	21.772	21.714	0	22	58	0.058
200 MHz	NBF	21.772	22.286	4.5	21	84	0.513	
	ABF	21.772	50.143	13.5	42	108	28.371	
	VBF	21.772	50.143	13.5	42	108	28.371	
	DFFBF	21.772	57.783	1.5	51	145.5	36.013	

Table 6.9: Numerical results for the BF algorithms in the normalized random start case.

#### 6.4 Algorithm Comparison

The purpose of the algorithms presented in this chapter was to try to force the algorithms to yield an accurate channel estimate, hence the nomenclature for this group of algorithms. Unfortunately these algorithms do not consistently provide good channel estimates, except at a very low sampling rate. Interestingly, although the BF algorithms do not provide good channel estimates, the NBF algorithm especially provides very good averaged distance estimates consistently. Although the averaged distance estimates were usually quite good, the ranges of estimates were very large, typically exceeding the size of the simulation building. The large range makes the distance estimates unreliable.

The BF algorithms all stem from the NBF algorithm. In order to attempt to improve channel and distance estimation results, and computational complexity, variants of the NBF algorithm were developed. These adaptive variants did improve channel estimation and computational complexity slightly, but did not improve distance estimation.

Due to the unreliability, and computational complexity, these algorithms are not considered further in this thesis.

## CHAPTER 7

### RESULTS

The algorithms presented, and the algorithms introduced by this thesis were tested thoroughly using simulation. The algorithms were also tested with data collected from a real environment, but the equipment available for testing, was incomplete. As a result, only limited real-world results are discussed here.

#### 7.1 Comparison of Algorithms

All the algorithms proposed in this thesis will be compared here. Algorithm complexity, simulation results and observed results all contribute to the viability of the algorithms usage in a real-world environment.

##### 7.1.1 Algorithm Complexity

Computational complexity is a rough estimate of how efficient an algorithm is. It cannot describe exactly how much more work goes into processing the data, as it only represents the additions and multiplies, as explained in [52]. However, it is still a common way to express efficiency of an algorithm, so computational complexity is shown below for some of the algorithms discussed in this thesis<sup>1</sup>. Table 7.1 shows the computational complexity of the algorithms using both the adaptive line-by-line approach, and the modified block approach explained in more detail in Appendix B.1.

As can be seen in the table, the SVD algorithm is by far the most computationally expensive algorithm and cannot be made any simpler. Looking at only the line-by-line complexity in the second column, the DFPSC has high complexity compared with the other algorithms, but it is of the same magnitude as the other SC algorithms. For the non-single constraint and SVD algorithms, the DFPAXIS has the most computational complexity. The

---

<sup>1</sup>The BF group of algorithms is not discussed further in this thesis as the simulation results were not good.

Algorithm	Complexity (Line)	Complexity (Block)
SVD	$4((N-L)^2M(L+1) + 8((N-L)(M(L+1))^2) + 9(M(L+1))^3)$	N/A
AED	$(N-L)(9M(L+1) - 1)$	$\frac{(N-L)}{b}(M(L+1)(4b+7) - (b+2) + 1)$
ModAED	$(N-L)(6M(L+1))$	$\frac{(N-L)}{b}(M(L+1)(6b+2) - 2b)$
ModAEDS	$(N-L)(8M(L+1))$	$\frac{(N-L)}{b}(M(L+1)(6b+4) - 2b)$
AXIS	$(N-L)(30M(L+1) + 2)$	$\frac{(N-L)}{b}(M(L+1)(24b+9) - 2)$
VAXIS	$(N-L)(30M(L+1) + 4)$	$\frac{(N-L)}{b}(M(L+1)(9+24b) + 2(b-1))$
DFPAXIS	$(N-L)(52M(L+1) + 11)$	$\frac{(N-L)}{b}(M(L+1)(10+46b) + 5b)$
NSAXIS	$(N-L)(28M(L+1) + 2)$	$\frac{(N-L)}{b}(M(L+1)(24b+7) - 2)$
SC	$(N-L)(9M^2(L+1)^2 + 8M(L+1))$	$\frac{(N-L)}{b}(M(L+1)(9+7b) - 2b+1)$
VSC	$(N-L)(9M^2(L+1)^2 + 8M(L+1) + 1)$	$\frac{(N-L)}{b}(M(L+1)(9+7b) - 2b+2)$
DFPSC	$(N-L)(12M^2(L+1)^2 + 10M(L+1) + 1)$	$\frac{(N-L)}{b}(M(L+1)(11+7b) - 2b+4)$
NSSC	$(N-L)(9M^2(L+1)^2 + 6M(L+1))$	$\frac{(N-L)}{b}(7M(L+1)(b+1) - 2b+1)$

Table 7.1: Comparison of the computational complexity of the algorithms.

Variable	Value
$\epsilon$	1e-10
$\alpha$	1e-5
$\gamma$	1e-5
$\phi$	1e-5

Table 7.2: Table of various variables used in simulation.

ModAED algorithm has the smallest complexity, only needing  $6M(N - L)(L + 1)$  additions and multiplies.

The third column of the table shows the computational complexity for the modified block approach developed to be more computationally efficient. It should be noted that the modified block implementation of the code was not implemented to be more computationally efficient, but to yield more accurate results, however it appears that in some cases, the algorithms might actually be more efficient, such as the SC variants of the AXIS algorithm. The DFPAXIS algorithm requires the most computations, and the AED algorithm the least.

### 7.1.2 Simulation Results

All the algorithms were tested using simulation first. In simulation, all but the SVD algorithm derived by Xu et al. used the same simulation conditions.

#### MATLAB Conditions

During simulation, two different data sizes, two different noise levels, two different bandwidths, two different initial estimates, and five different model orders at 11 different sample rates<sup>2</sup> were tested. However, with all the variation that occurred during testing, there was some consistency between parameter variations. MATLAB was used for all simulation testing on the same computer. Table 7.2 show the constant variables used during simulation. These values were arbitrarily decided, prior to processing.

<sup>2</sup>See Section 3.3 for an explanation of these variations.

## Results

Numerical results for the algorithms discussed here can be found in Chapters 4 and 5, although the full results can be found in Appendix D.

In the ideal case, all of the adaptive algorithms had comparable results as seen in Tables 4.1, 5.1 and 5.2. Although the DFPAXIS algorithm typically has the best channel estimate, all the distance estimates are typically within 1 meter from each other. As observed previously, impulse response set IR-2 typically has better results than set IR-1. For set IR-1, the 20 MHz SVD has arguably the best estimate due to the range of the estimate being zero. However, with that argument, the SVD also has the worst estimate for set IR-2 at 200 MHz as it has a very large range, although the error is quite small. It is interesting to note that although mathematically, the AXIS algorithms are more like the ModAED and ModAEDS algorithms, they produce results more similar to the AED algorithm. Another interesting observation is that when the SVD suffers, the adaptive algorithms tend to do much better.

***Effects of Various Parameters:*** There were various parameters that were adjusted during simulation to test the robustness of the algorithms. This included testing the robustness to signal bandwidth, noise, sample rate, data size, channel order, and initial channel estimates. The effects of these parameters will be discussed below.

Signal Bandwidth and Sample Rate: As was seen in all the figures in Chapters 4 and 5, the narrowband simulations for all the algorithms resulted in very bad channel estimates that showed that the channels did not appear to resolve, but rather were duplicates of each other. Table 4.2 and Tables 5.3 and 5.4 display large estimation ranges. Impulse response set IR-1, show on average, very small distance estimates, while set IR-2 show larger estimates that are closer to the estimates typically used.

The wideband simulations have shown to do much better. In these simulations, bandwidth corresponds with sample rate directly. It is assumed that the higher the sample rate, the higher the bandwidth. It appears that higher bandwidth could correlate to better accuracy.

Noise: The results for more noise can be seen in Table 4.3, Table 5.5 and Table 5.6. The noisy results, excluding the SVD algorithm, are not that different from the less noisy case.

The SVD algorithm suffers greatly with the added noise. The results in the table shows a wide range of estimates with high average values, that degenerate at the higher sampling rate. The plots in Chapter 4 show that the estimates degenerate overall. For the highest sample rate (200 MHz), the SVD algorithm is not able to resolve the channels at all. NaN is the average distance estimate for both impulse response sets at 200 MHz.

Like the less noisy case, the DFPAXIS algorithm tends to do better than the other algorithms, although it is a slight margin over the other adaptive algorithms. There are even some instances where the noisy distance estimates are better than the ideal case. It should be noted that typically, however, the algorithms either do the same or slightly poorer.

Sample Size: Tables 4.4, 5.7 and 5.8 show the numerical results for the simulations which used fewer samples. There is very little difference from the more samples case excluding the SVD approach. Some of the algorithm results show that they even perform better than with more samples, and some show the opposite.

The adaptive algorithms with fewer samples on average produce the same results. There is not one algorithm whose performance stands out significantly compared with the others. The SVD approach does fine for the smaller sample rates, but degenerates at the high sample rates. This could possibly be due to the model order being too close to the sample amount. It should be noted that the SVD algorithm wasn't able to use as many samples as the other algorithms due to memory issues, but the ratio between the different data sizes is the same.

Channel Order: The wrong model order results can be seen in Tables 4.5–4.8, and Tables 5.9–5.16. The model order results show that most of the algorithms are robust to model order error, and some are not.



Underestimating the channel order for both the SVD and the adaptive algorithms does quite terribly for the 20 MHz sampling rate in the first impulse response set, IR-1, yielding only NaNs. These NaNs possibly occurred due to there is not enough room for the peaks to establish themselves within 10 samples. There does not appear to be much effect on the adaptive algorithms. The distance estimates are similar (within  $\approx 1$  or 2 m difference) to those shown in the correct model order cases. The SVD appears to suffer more with the underestimation of the channel order than the adaptive algorithms. The SVD has a much greater disparity between the correct model order and the underestimated model order.

As can be seen in Tables 4.7 and 4.8 overestimating the channel order is much more detrimental to the SVD algorithm. Although estimated distance are not NaNs, the channel estimates shown in Chapter 4 show that even overestimating by 10 samples degenerates the estimate, although not as doubling the model order. Interestingly, the adaptive algorithms were not as affected by overestimating the model order. The estimates were only slightly different from the estimates with the actual order.

Initial Channel Estimates: In the ideal case, and for all the cases described above, the initial channel estimate for the adaptive algorithms was a unit vector, shown in (4.13). A different initial channel estimate was also tested. As can be seen in Table 4.9, Table 5.17, and Table 5.18, using a normalized random start does not produce good estimates. The distance estimates for all but the AXIS and NSAXIS are extremely large, with very large ranges of estimates. The SVD algorithm does not use an initial estimate, so it is not included in the tables or this section.

### 7.1.3 Results with Real Observed Data

As explained in Chapter 3, the testing with real data was limited to powerful very narrowband signals, or a very weak wideband signals. As a result, the algorithms were not thoroughly tested with the observed data. The few tests and results for the real data will be discussed here.

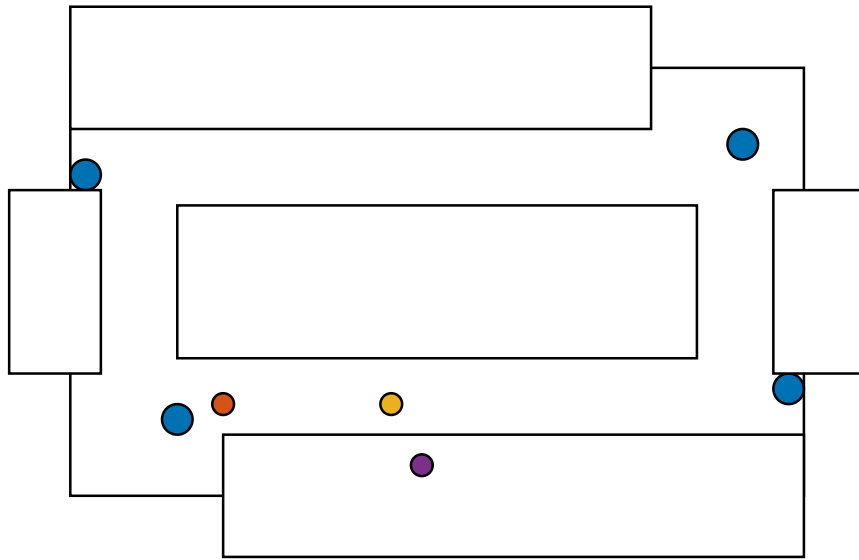


Fig. 7.1: The approximate locations of the antennas used during real-world data testing. The large blue circles correspond to the location of the receivers, and the smaller multi-colored circles the location of the transmitters. Note that this figure is not to scale.

### Testing Conditions

Two data sets for three different transmission locations were tested. Figure 7.1 shows the relative locations of the four receivers and the transmission locations with which the data sets were collected for the second floor of the SANT building. Table 7.3 shows the relative horizontal distance between antennas in meters, elevation was not considered.

The HT data sets could produce signals that could be seen at each receiver at about -20 decibel-milliwatts (dBm), with a -80 dBm noise floor. In contrast, the Gaussian data sets could only be seen when the transmitter was within one to two meters of the individual receivers. Even in close proximity, the signal only had a maximum power of about -60 dBm with a -80 dBm noise floor. A working amplifier at the frequency range used for testing was not available.

The GNURadio program with which the data was collected limited the sampling rate to 16 MHz, when the SDRs used for collection could perform at least double that. For time synchronization between radios, the GNURadio program had to be used. A sample difference at 16 MHz is a difference of 18.75 meters, which is quite large for a building that is just over twice that in size.

Antenna Positions (m)	NW	NE	SW	SE	Corner	Hall	Lab
NW	0	34.135	9.845	37.704	9.967	15.362	21.275
NE	34.135	0	31.670	9.449	31.455	22.921	21.6713
SW	9.845	31.670	0	32.918	0.975	9.357	12.375
SE	37.704	9.449	32.918	0	32.675	23.592	21.031
Corner	9.967	31.455	0.975	32.675	0	9.016	12.527
Hall	15.362	22.921	9.357	23.592	9.016	0	4.633
Lab	21.275	21.672	12.375	21.031	12.527	4.633	0

Table 7.3: The distance in meters between antennas (receiver or transmitters).

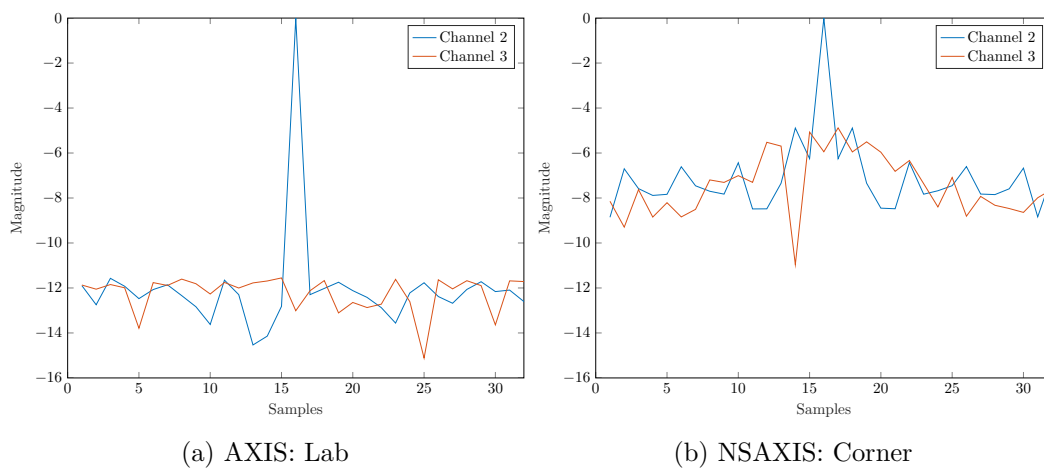


Fig. 7.2: Channel estimates for the Gaussian Data sets with the most “reasonable” results, using a model order of  $L = 32$ . These plots show  $\ln(\mathbf{h}_i)$ , for  $i = 2, 3$ , as the values received by the receiver are very small.

The combination of limited transmitters, and low distance resolution do not proffer ideal conditions for real-world testing, but a few of the real-world results are included for completeness. See Chapter 3 for more explanation of the testing conditions.

## Results

As testing conditions were poor, only a few cases of the results are shown to show how poor the testing conditions really were. As the wideband signal could barely be seen at the receivers, only channels 2, and 3 were used for testing, as they are closest to the transmitter locations, and there is some LOS.

The estimates shown in Figure 7.2 are the better estimates for the Gaussian Data Sets

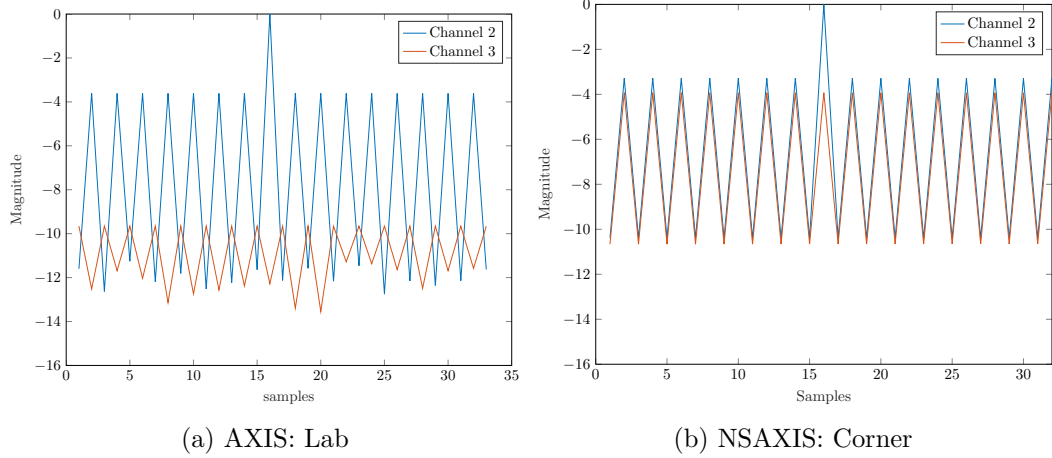


Fig. 7.3: Channel estimates for the HT Data sets, using a model order of  $L = 32$ . These plots show  $\ln(\mathbf{h}_i)$ , for  $i = 2, 3$ , to match the results shown above in Figure 7.2.

with channels 2 and 3 (the SW and NW receiver locations respectively). As the received power was so small for the receivers especially at larger distances than 1 meter, the natural logarithm was used on the plots of the estimates in order to see some response on the secondary channel. Both algorithms show a delay estimate of 1 sample,  $\approx 18.75m$ . It should be noted that the received signals were normalized prior to processing through the algorithms to try to prevent power discrepancies between the channels.

Figure 7.3 shows the case for the same pairing of data sets, channels and algorithms as Figure 7.2, but for the HT transmitted signal. The HT Data sets are very narrowband, and like with the simulation results, have difficulty separating the channels, thus yielding to terrible channel estimates.

## 7.2 Final Observations

The result show that the Adaptive Algorithms on average yield the best distance estimates in most of the circumstances tested in simulation. This is interesting, because mathematically the SVD method would be the assumed best. A possible reason for this could be that with such large data matrices, the SVD gets lost in the rounding errors.

Of the adaptive algorithms, the DFPAXIS algorithm typically had the best distance estimates, but it was more computationally complex than most of the others (excluding the

SC algorithms). As the other algorithms are typically within 1 meter of error, the ModAED would be an acceptable choice as it is the least computationally complex. The AXIS algorithm includes the shift-suppression, and sparsity constraints, and so would probably be the better option, as it produces a cleaner, less spurious channel estimate.

The results of the algorithms with real-world testing were quite poor. Real-world data testing was not expected to work as well as the simulation, but the hardware had a few issues that caused the testing to fail. During real-world testing, the equipment available was not powerful enough to transmit a strong wideband signal by itself, and a working amplifier was not readily available. Thus, adequate testing could not be done, and no conclusions about the algorithms could be made.

## CHAPTER 8

### CONCLUSION

As wireless devices have gained popularity, there is a drawing need for accurate source localization inside buildings, especially when the signal is unknown. This is a difficult task as a direct “line-of-sight” (LOS) is not readily available in buildings due to walls, and furniture.

The most common methods of source localization for an unknown signal without guaranteed LOS include received signal strength (RSS), and time difference of arrival (TDOA). RSS is typically not as accurate as TDOA, so TDOA is the preferred choice. In order to compute TDOA, time delay estimation (TDE) techniques are used.

The purpose of this research was to determine TDE algorithms for accurate blind, multipath TDOA source localization. Of the TDE techniques, the blind channel identification technique of utilizing the cross-relation (CR) was explored in depth in this thesis. Two pre-existing CR algorithms were looked at, a Singular Value Decomposition (SVD) approach [38], and the Adaptive Eigenvalue Decomposition (AED) approach [25]. The research discussed in this thesis included CR approaches that used an adaptive approach but minimized the difference between iterations rather than minimize the CR itself.

#### 8.1 Research

Thirteen algorithms and variations for blind channel identification in multipath environments were introduced and explored in this research. Of those algorithms, 10 are viable algorithms. At least four algorithms consistently appeared to have better than or matching results to the existing CR algorithms in simulation.

This research introduces the use of a shift-suppression constraint in tandem with a sparsity constraint to prevent spurious peaks from occurring and causing noisy channel estimates. In simulation, the adaptive algorithms show that they are more robust to noise,

and channel model order than the SVD algorithm, providing accurate results even in the presence of these non-ideal cases.

Real-world data were also tested, but due to the limited availability of transmitters, power restrictions, and small sample rate, the algorithm results were disappointing.

## 8.2 Future Work

Improvement of these algorithms are always possible, and there is much more to explore that could not be covered in this thesis.

To improve algorithm accuracy, there are a few possibilities that could be researched in more detail. One of those possibilities would be to use NLOS identification and mitigation techniques. As explained in Chapter 3, NLOS conditions can cause more peaks to occur, which if cleared up would allow for easier first peak identification. Also, with some preliminary testing (which was discovered after the results had been compiled), it has also been proposed that perhaps the MUSIC algorithm [50, 55] could be used for peak identification, rather than taking the maximum peaks. Another avenue would be to use a more probabilistic model, as the estimation could be more optimal as explain in Chapter 2.

To improve speed, it also stands to reason that if the algorithms start with a better initial estimates of the impulse response  $\mathbf{h}^{(n)}$ , then the algorithms would converge quicker for large quantities of data. Additionally, a Newton's Method approach for step size adaptation theoretically would remove the need for such large data sizes [56].

As TDOA is not a perfect source localization technique, perhaps the estimates could be augmented by combining the results with another source localization technique using Dilution of Precision [57–59]. It should be noted that this research has assumed a stationary transmitter, a possible future extension of this project is for moving transmitters. Perhaps this could be achieved with a Kalman filter [50] to predict movement, or by implementing an FDOA system to work in tandem with the TDOA algorithms as explained in Chapter 2. This research also assumed a SIMO system, which is ideal. There are likely to be more than one transmitters in indoor environments. Another extension of this research would include adapting the algorithms for multiple input multiple output (MIMO) systems.

As thorough testing with real-world data was not achievable during this research, using a wider variety of transmitters would help remediate this problem. As the transmitter that was used had a low transmitting power, adding an amplifier could help improve this issue. All the algorithms showed that they do not perform with narrowband data. Narrowband data is easier to achieve than wideband data. There is the possibility to use multiple SDRs tuned to different center frequencies to capture a more wideband signal, as proposed in [60].



## REFERENCES

- [1] R. Zekavat and M. Buehrer, *Handbook of Position Localization: Theory, Practice and Advances*. Wiley-IEEE Press, 2012, ch. 1, 2,16, pp. 3–66,523–555.
- [2] M. B. Montminy, “Passive geolocation of low-power emitters in urban environments using TDOA,” Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 2007.
- [3] K. Lee, H. Kwon, and K. You, “TDOA based geolocation using IRLS algorithm,” in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Nov 2017, pp. 92–95.
- [4] K. Pahlaven, X. Li, M. Ylianttila, R. Chana, and M. Latva-aho, “An overview of wireless indoor geolocation techniques and systems,” in *Proc. Mobile and Wireless Communications Networks: IFIP-TC6/European Commission NETWORKING 2000 International Workshop*, May 2000, pp. 1–13.
- [5] X. Guo, B. Li, L. Chu, and D. Wang, “Near-field source localization in complex indoor environment using uniform circular array,” in *2014 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)*, July 2014, pp. 412–415.
- [6] D. E. Gustafson, J. M. Elwell, and J. A. Soltz, “Innovative indoor geolocation using RF multipath diversity,” in *2006 IEEE/ION Position, Location, And Navigation Symposium*, April 2006, pp. 904–912.
- [7] A. Abbasi and M. H. Kahaei, “Improving source localization in LOS and NLOS multipath environments for uwb signals,” in *2009 14th International CSI Computer Conference*, Oct 2009, pp. 310–316.
- [8] Apple Inc<sup>®</sup>. (2019) Find my friends. [Online]. Available: <https://apps.apple.com/us/app/find-my-friends/id466122094>
- [9] Kin Seong Leong, Mun Leng Ng, and P. H. Cole, “Investigation on the deployment of HF and UHF RFID tag in livestock identification,” in *2007 IEEE Antennas and Propagation Society International Symposium*, June 2007, pp. 2773–2776.
- [10] M. Gharibi, R. Boutaba, and S. L. Waslander, “Internet of drones,” *IEEE Access*, vol. 4, pp. 1148–1162, 2016.
- [11] S. Whiting, “Radio-frequency transmitter geolocation using non-ideal received signal strength indicators,” Master’s thesis, Utah State University, Old Main Hill, Logan, UT, 2018.
- [12] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing (Signals and Communication Technology)*. Berlin, Heidelberg: Springer-Verlag, 2006.

- [13] D. Salvati and S. Canazza, "Adaptive time delay estimation using filter length constraints for source localization in reverberant acoustic environments," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 507–510, May 2013.
- [14] J. Benesty, Y. Huang, and J. Chen, "Time delay estimation via minimum entropy," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 157–160, March 2007.
- [15] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *The Journal of the Acoustic Society of America*, vol. 107, pp. 384–391, 2000.
- [16] T. P. Bhardwaj and R. Natha, "Maximum likelihood estimation of time delays in multipath acoustic channel - performance versus number of multipath and noise," in *2010 4th International Conference on Signal Processing and Communication Systems*, Dec 2010, pp. 1–5.
- [17] D. Musicki and W. Koch, "Geolocation using TDOA and FDOA measurements," in *2008 11th International Conference on Information Fusion*, June 2008, pp. 1–8.
- [18] X. Qu, L. Xie, and W. Tan, "Iterative source localization based on TDOA and FDOA measurements," in *2016 35th Chinese Control Conference (CCC)*, July 2016, pp. 436–440.
- [19] K. C. Ho and Wenwei Xu, "Localization of a moving source using tdoa and fdoa measurements," in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, vol. 4, May 2003, pp. IV–IV.
- [20] D. Wang, P. Zhang, Z. Yang, F. Wei, and C. Wang, "A novel estimator for TDOA and FDOA positioning of multiple disjoint sources in the presence of calibration emitters," *IEEE Access*, vol. 8, pp. 1613–1643, 2020.
- [21] X. Qu, L. Xie, and W. Tan, "Iterative constrained weighted least squares source localization using TDOA and FDOA measurements," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3990–4003, Aug 2017.
- [22] P. C. Chestnut, "Emitter location accuracy using TDOA and differential doppler," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 2, pp. 214–218, March 1982.
- [23] M. L. Fowler and X. Hu, "Signal models for TDOA/FDOA estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1543–1550, Oct 2008.
- [24] P. M. Schultheiss and E. Weinstein, "Estimation of differential doppler shifts," *The Journal of the Acoustic Society of America*, vol. 66, pp. 1412–1419, 1979.
- [25] Y. Huang, J. Benesty, and G. W. Elko, "Adaptive eigenvalue decomposition algorithm for real time acoustic source localization system," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 2, March 1999, pp. 937–940 vol.2.

- [26] B. Lee and T. Kalker, "Maximum a posteriori estimation of time delay," in *Proc. 2007 2nd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2007, pp. 285–288.
- [27] Weidong Liu, E. Ding, and Xiaojing Meng, "The research of time delay estimation algorithm," in *2011 International Conference on Computer Science and Service System (CSSS)*, June 2011, pp. 878–881.
- [28] Y. Lin, J. Chen, Y. Kim, and D. D. Lee, "Blind sparse-nonnegative (BSN) channel identification for acoustic time-difference-of-arrival estimation," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 2007, pp. 106–109.
- [29] Zhen-Ya He and Li-Hua Li, "High-resolution multipath time delay estimation using a neural network," in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, April 1991, pp. 1469–1472 vol.2.
- [30] M. R. Bai, S. Lan, and J. Huang, "Time difference of arrival (TDOA)-based acoustic source localization and signal extraction for intelligent audio classification," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, July 2018, pp. 632–636.
- [31] J. Cho, D. Hwang, and K. Kim, "Improving TDoA based positioning accuracy using machine learning in a LoRaWan environment," in *2019 International Conference on Information Networking (ICOIN)*, Jan 2019, pp. 469–472.
- [32] C. Wu, H. Hou, W. Wang, Q. Huang, and X. Gao, "TDOA based indoor positioning with NLOS identification by machine learning," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct 2018, pp. 1–6.
- [33] D. A. Bibb, Z. Yun, and M. F. Iskander, "Machine learning for source localization in urban environments," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Nov 2016, pp. 401–405.
- [34] Lang Tong and S. Perreau, "Multichannel blind identification: from subspace to maximum likelihood methods," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1951–1968, Oct 1998.
- [35] K. Abed-Meraim, Wanzhi Qiu, and Yingbo Hua, "Blind system identification," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1310–1322, Aug 1997.
- [36] Yingbo Hua, "Fast maximum likelihood for blind identification of multiple FIR channels," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Oct 1994, pp. 415–419 vol.1.
- [37] J. Gunther and A. Swindlehurst, "Algorithms for blind equalization with multiple antennas based on frequency domain subspaces," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 5, May 1996, pp. 2419–2422 vol. 5.

- [38] Guanghan Xu, Hui Liu, Lang Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2982–2993, Dec 1995.
- [39] Hui Liu, Guanghan Xu, and Lang Tong, "A deterministic approach to blind identification of multi-channel FIR systems," in *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. iv, April 1994, pp. IV/581–IV/584 vol.4.
- [40] A. Aïssa-El-Bey, K. Abed-Meraim, and C. Laot, "Adaptive blind estimation of sparse SIMO channels," in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, May 2011, pp. 348–351.
- [41] Y. Li, K. Lee, and Y. Bresler, "Identifiability in blind deconvolution with subspace or sparsity constraints," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 4266–4275, July 2016.
- [42] S. Choudhary and U. Mitra, "Sparse blind deconvolution: What cannot be done," in *2014 IEEE International Symposium on Information Theory*, June 2014, pp. 3002–3006.
- [43] M. Crocco and A. Del Bue, "Estimation of TDOA for room reflections by iterative weighted L1 constraint," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3201–3205.
- [44] S. R. Drake and K. Dogancay, "Geolocation by time difference of arrival using hyperbolic asymptotes," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2004, pp. ii–361.
- [45] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, Aug 1994.
- [46] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, August 1976.
- [47] "Cross-correlation (xcorr) - MATLAB." [Online]. Available: <https://www.mathworks.com/help/matlab/ref/xcorr.html>
- [48] J. J. Shynk, *Probability, Random Variables, and Random Processes: Theory and Signal Processing Applications*. Wiley, 2013.
- [49] C. R. Comsa, "Source localization via time difference of arrival," Ph.D. dissertation, New Jersey Institute of Technology, Newark, NJ, 2011.
- [50] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2000, ch. 3,6,7,11,12,13, pp. 132–133,339,392,500,573,591–616.
- [51] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, Aug 1972.

- [52] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 1989, ch. 1,5, pp. 19–20,239.
- [53] Y. Chen. (2019) Subgradient methods. [Online]. Available: [http://www.princeton.edu/~yc5/ele522\\_optimization/lectures/subgradient\\_methods.pdf](http://www.princeton.edu/~yc5/ele522_optimization/lectures/subgradient_methods.pdf)
- [54] S. Boyd and L. Vandenberghe. (2008) Subgradient methods. [Online]. Available: [https://see.stanford.edu/materials/lsoceee364b/01-subgradients\\_notes.pdf](https://see.stanford.edu/materials/lsoceee364b/01-subgradients_notes.pdf)
- [55] J. G. Proakis, C. M. Rader, F. Ling, C. L. Nikias, M. Moonen, and I. K. Proudler, *Algorithms for Statistical Signal Processing*. Upper Saddle River, NJ: Prentice-Hall, Inc, 2002, ch. 9, pp. 488–489.
- [56] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2004, ch. 9, pp. 484–496.
- [57] J. D. Bard and F. M. Ham, “Time difference of arrival dilution of precision and applications,” *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 521–523, Feb 1999.
- [58] D. J. Torrieri, “Statistical theory of passive location systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 183–198, March 1984.
- [59] C. Chen, K. Chen, J. Huang, and Y. Li, “Using genetic algorithms to approximate weighted geometric dilution of precision,” in *2016 International Symposium on Computer, Consumer and Control (IS3C)*, July 2016, pp. 895–898.
- [60] J. Gunther, C. Lindstrom, and D. Sorensen, “Exponent: Arbitrary bandwidth receiver architecture,” *Proceedings of the GNU Radio Conference*, vol. 4, no. 1, 2019. [Online]. Available: <https://pubs.gnuradio.org/index.php/grcon/article/view/59>

APPENDICES

## APPENDIX A

## Math Derivations and Notation

## A.1 Notation

$n$	The $n^{\text{th}}$ sample, $n = 0, 1, 2, \dots, N$
$N$	The number of samples of data used, arbitrarily decided
$M$	The number of receivers, $m = 1, 2, \dots, M$
$L$	The model order, length of the impulse response
$s(n)$	The $n^{\text{th}}$ sample of the signal transmitted
$h_i^*(n)$	The $n^{\text{th}}$ sample of the complex conjugate of the $i^{\text{th}}$ impulse response
$w_i(n)$	Additive white noise at the $i^{\text{th}}$ receiver
$x_i(n)$	The $n^{\text{th}}$ sample received at the $i^{\text{th}}$ receiver from the transmitted signal: $x_i(n) = s(n) * h_i^*(n) + w_i(n)$
$X_m(L)$	the Toeplitz matrix of samples $n = 0 \rightarrow n = N$ for data received at the $m^{\text{th}}$ receiver, of size $(N - L) \times (L + 1)$ $\begin{bmatrix} x_m(L) & x_m(L - 1) & \cdots & x_m(0) \\ x_m(L + 1) & x_m(L) & \cdots & x_m(1) \\ \vdots & \ddots & \ddots & \vdots \\ x_m(N) & x_m(N - 1) & \cdots & x_m(N - L - 1) \end{bmatrix}$
$\mathbf{h}_m$	The vector of impulse response samples of size $(L + 1) \times 1$ : $\mathbf{h}_m \triangleq [h_m(0), \dots, h_m(L)]^T$
$\mathbf{h}^{(n)}$	The stacked previous impulse response estimates of size $M(L + 1)$ (For example, $j \neq i$ , and $M = 2$ ): $\begin{bmatrix} \mathbf{h}_j \\ \mathbf{h}_i \end{bmatrix}$
$\mathbf{h}^{(n+1)}$	The update impulse response estimates of size $M(L + 1)$ , (same form as $\mathbf{h}^{(n)}$ )

- $\mathbf{X}$  The combined data matrix of multiple receivers of size  $(N - L) \times M(L + 1)$   
 (For example,  $j \neq i$ , and  $M = 2$ ):  $\begin{bmatrix} \mathbf{X}_i(L) & \vdots & -\mathbf{X}_j(L) \end{bmatrix}$
- $x_r$  A row of the data matrix  $\mathbf{X}$  (for the  $k^{\text{th}}$  row) of size  $1 \times M(L + 1)$   
 (For example,  $j \neq i$ , and  $M = 2$ ):  $\begin{bmatrix} x_i(k + L) & \cdots & x_i(k) & \vdots & -x_j(k + L) & \cdots & -x_j(k) \end{bmatrix}$
- $X_r$  A block of the data matrix  $\mathbf{X}$  of size  $b \times M(L + 1)$  (Note,  $j \neq i$ , and  $M = 2$ ):  

$$\begin{bmatrix} x_i(k + L) & \cdots & x_i(k) & \vdots & -x_j(k + L) & \cdots & -x_j(k) \\ x_i(k + L + 1) & \cdots & x_i(k + 1) & \vdots & -x_j(k + L + 1) & \cdots & -x_j(k + 1) \\ \vdots & \cdots & \ddots & \vdots & \ddots & \cdots & \vdots \\ x_m(k + L + b) & \cdots & x_m(k + b) & \vdots & -x_j(k + L + b) & \cdots & -x_j(k + b) \end{bmatrix}$$
- $\lambda$  A Lagrange multiplier
- $\mu$  A Lagrange multiplier
- $\beta$  A Lagrange multiplier
- $\tilde{\{}$  The conjugate of the Lagrange multiplier  $\lambda$ ,  $\mu$  or  $\beta$
- $\alpha$  A weighting value for a component of the objective function
- $\gamma$  A weighting value for a component of the objective function
- $\phi$  A weighting value for a component of the objective function
- $\eta$  A step size for update equation
- $\epsilon$  A small nonzero number added to the denominator such that the denominator will never equal 0 without significantly affecting the answer
- $\mathbf{I}$  The identity matrix of size  $M(L + 1) \times M(L + 1)$ :  

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$
- $F_s$  sampling rate
- $\mathcal{L}$  The variable denoting the Lagrangian (equation)



Hz	hertz, representing the frequency measurement
MHz	Mega-hertz, representing $10^6$ Hz
$\tilde{\zeta}$	A scalar value
$y$	A scalar value
$e_d$	A elementary vector where the $d^{\text{th}}$ element is a 1 of size $M(L+1) \times 1$ : $\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^T$
$e_c$	A elementary vector where the $c^{\text{th}}$ element is a 1 of size $M(L+1) \times 1$ (Note $d \neq c$ ): $\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^T$
$\{\}^*$	The conjugate
$\{\}^H$	The conjugate transpose or rather Hermitian
$\{\}^{-1}$	The inverse
$c$	the speed of light: $= 3 \times 10^8$ m/s
$R_{x_i, x_j}(\tau)$	The cross-correlation (scalar form): $E[x_i(n)x_j(n+\tau)]$
$\mathbf{R}_{x_i, x_j}$	The cross-correlation (vector form): $E[\mathbf{x}_i \mathbf{x}_j^T]$
$\mathbf{R}$	The covariance matrix+: $\begin{bmatrix} \mathbf{R}_{x_1, x_1} & \mathbf{R}_{x_1, x_2} \\ \mathbf{R}_{x_2, x_1} & \mathbf{R}_{x_2, x_2} \end{bmatrix}$

### Multiple Receiver Format

The above notation assumes that the number of channels is small,  $M = 2$ . This section expounds on the formatting and notation for multiple receivers ( $> 2$ ) for a larger set of linear equations. Note that this method is similar to the one described in [38]:

For some arbitrary number of receivers,  $M$ , a matrix of all the pairs between the receivers is made as follows:

$$\mathbf{X}^m(L) = \left[ \begin{array}{cccccc} \mathbf{0} & \cdots & \mathbf{0} & X_{m+1}(L) & -X_m(L) & \mathbf{0} & \mathbf{0} \\ & & \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & X_M(L) & \mathbf{0} & \cdots & -X_m(L) \end{array} \right] \left. \vphantom{\begin{array}{cccccc} \mathbf{0} & \cdots & \mathbf{0} & X_{m+1}(L) & -X_m(L) & \mathbf{0} & \mathbf{0} \\ & & \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & X_M(L) & \mathbf{0} & \cdots & -X_m(L) \end{array}} \right\} M - m \text{ blocks}$$

$\underbrace{\hspace{10em}}_{m-1 \text{ blocks}} \quad \underbrace{\hspace{10em}}_{M-m+1 \text{ blocks}}$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^1(L) \\ \vdots \\ \mathbf{X}^{M-1} \end{bmatrix} \quad \mathbf{h}^{(n)} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_M \end{bmatrix}$$

## A.2 Derivations

Each algorithm derivation is shown step by step as follows:

- The Modified Adaptive Eigenvalue Decomposition (ModAED)
- The Modified Adaptive Eigenvalue Decomposition with Sparsity (ModAEDS)
- The Adaptive Cross-Channel Identification with Sparse Shift-Suppression (AXIS)
- The Variable Adaptive Cross-Channel Identification with Sparse Shift-Suppression (VAXIS)
- The Dual Fixed-Peak Adaptive Cross-Channel Identification with Sparse Shift-Suppression (DFPAXIS)
- The Single Constraint Adaptive Cross-Channel Identification with Sparse Shift-Suppression (SC)
- The Variable Single Constraint Adaptive Cross-Channel Identification with Sparse Shift-Suppression (VSC)
- The Dual Fixed-Peak Single Constraint Adaptive Cross-Channel Identification with Sparse Shift-Suppression (DFPSC)
- The “Brute Force” (BF)
- The Variable “Brute Force” (VBF)
- The Dual Fixed-Peak “Brute Force” (DFPBF)

Note that each vector is assumed to be column formatted for each derivation.

### Modified Adaptive Eigenvalue Decomposition

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \end{aligned} \quad (\text{A.1})$$

Derivation:

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r \quad (\text{A.2})$$

$$\frac{d}{d\mathbf{h}^{(n+1)}} \mathcal{L} = 2\mathbf{h}^{(n+1)} - 2\mathbf{h}^{(n)} + \lambda \mathbf{x}_r \quad (\text{A.3})$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \lambda \mathbf{x}_r \quad (\text{A.4})$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \lambda \mathbf{x}_r \quad (\text{A.5})$$

Constraints:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$(\mathbf{h}^{(n)} - \lambda \mathbf{x}_r)^{\text{H}} \mathbf{x}_r = 0 \quad (\text{A.6})$$

$$\mathbf{h}^{(n)\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r \quad (\text{A.7})$$

$$\tilde{\lambda} = \frac{\mathbf{h}^{(n)\text{H}} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r} \quad (\text{A.8})$$

$$\lambda = \frac{\mathbf{x}_r^{\text{H}} \mathbf{h}^{(n)}}{\|\mathbf{x}_r\|_2^2} \quad (\text{A.9})$$

Complete Update:

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \eta \frac{\mathbf{x}_r^{\text{H}} \mathbf{h}^{(n)}}{\|\mathbf{x}_r\|_2^2} \mathbf{x}_r \quad (\text{A.10})$$

### Modified Adaptive Eigenvalue Decomposition with Sparsity

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)\text{H}}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r} = 0 \end{aligned} \quad (\text{A.11})$$

Derivation:

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}}\mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}}\mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}}\mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}}\mathbf{h}^{(n)} + \alpha \|\mathbf{h}^{(n+1)\text{H}}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r}. \quad (\text{A.12})$$

Assuming,  $\text{sgn}(\mathbf{h}^{(n+1)}) \approx \text{sgn}(\mathbf{h}^{(n)})$ , which for complex numbers,  $\text{sgn}(\mathbf{h}^{(n)}) = \mathbf{e}^{j\angle\mathbf{h}^{(n)}}$ ,

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} + \lambda \mathbf{x}_r \quad (\text{A.13})$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} + \lambda \mathbf{x}_r \quad (\text{A.14})$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \lambda \mathbf{x}_r \quad (\text{A.15})$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r} = 0:$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \lambda \mathbf{x}_r)_{\mathbf{x}_r}^{\text{H}} = 0 \quad (\text{A.16})$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})_{\mathbf{x}_r}^{\text{H}} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r \quad (\text{A.17})$$

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})_{\mathbf{x}_r}^{\text{H}}}{\mathbf{x}_r^{\text{H}} \mathbf{x}_r} \quad (\text{A.18})$$

$$\lambda = \frac{\mathbf{x}_r^{\text{H}} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})}{\|\mathbf{x}_r\|_2^2} \quad (\text{A.19})$$

Complete Update

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \eta \frac{\mathbf{x}_r^{\text{H}} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle\mathbf{h}^{(n)}})}{\|\mathbf{x}_r\|_2^2} \mathbf{x}_r \quad (\text{A.20})$$

## Adaptive Cross-Channel Identification with Sparse Shift-Suppression

Objective Function:

$$\begin{aligned}
 \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
 \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \\
 & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = 1
 \end{aligned} \tag{A.21}$$

Derivation:

$$\begin{aligned}
 \mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} \\
 + \alpha \|\mathbf{h}^{(n+1)}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - 1)
 \end{aligned} \tag{A.22}$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d \tag{A.23}$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d \tag{A.24}$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d \tag{A.25}$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d)^{\text{H}} \mathbf{x}_r = 0 \tag{A.26}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r \tag{A.27}$$

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = 1:$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d)^{\text{H}} \mathbf{e}_d = 1 \tag{A.28}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - 1 = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r \tag{A.29}$$

System of Equations:

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r \tag{A.30}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - 1 = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_d$$

$$\begin{bmatrix} \mathbf{x}_r^H \mathbf{x}_r & \mathbf{e}_d^H \mathbf{x}_r \\ \mathbf{x}_r^H \mathbf{e}_d & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - 1 \end{bmatrix} \quad (\text{A.31})$$

$$\begin{aligned} \tilde{\lambda} &= \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \\ \tilde{\mu} &= \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \end{aligned} \quad (\text{A.32})$$

$$\begin{aligned} \lambda &= \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r)^*}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \\ \mu &= \left( \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \end{aligned} \quad (\text{A.33})$$

Complete Update:

$$\begin{aligned} \mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\ &\eta \left( \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r)^*}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r - \right. \\ &\quad \left. \left( \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \right) \end{aligned} \quad (\text{A.34})$$

## Varied AXIS

### Objective Function

$$\begin{aligned}
& \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
& \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r} = 0 \\
& \quad \quad \quad \mathbf{h}^{(n+1)\text{H}}_{\mathbf{e}_d} = \tilde{\zeta}
\end{aligned} \tag{A.35}$$

### Derivation:

$$\begin{aligned}
\mathcal{L} = & \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} \\
& + \alpha \|\mathbf{h}^{(n+1)}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r} + \mu (\mathbf{h}^{(n+1)\text{H}}_{\mathbf{e}_d} - \tilde{\zeta})
\end{aligned} \tag{A.36}$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d \tag{A.37}$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d \tag{A.38}$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d \tag{A.39}$$

### Constraints:

$$\mathbf{h}^{(n+1)\text{H}}_{\mathbf{x}_r} = 0:$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d)_{\mathbf{x}_r}^{\text{H}} = 0 \tag{A.40}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})_{\mathbf{x}_r}^{\text{H}} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} + \tilde{\mu} \mathbf{e}_d^{\text{H}} \tag{A.41}$$

$$\mathbf{h}^{(n+1)\text{H}}_{\mathbf{e}_d} = \tilde{\zeta}:$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d)_{\mathbf{e}_d}^{\text{H}} = \tilde{\zeta} \tag{A.42}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})_{\mathbf{e}_d}^{\text{H}} - \tilde{\zeta} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} + \tilde{\mu} \mathbf{e}_d^{\text{H}} \tag{A.43}$$

### System of Equations:

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})_{\mathbf{x}_r}^{\text{H}} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} + \tilde{\mu} \mathbf{e}_d^{\text{H}} \tag{A.44}$$

$$(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})_{\mathbf{e}_d}^{\text{H}} - \tilde{\zeta} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} + \tilde{\mu} \mathbf{e}_d^{\text{H}}$$



$$\begin{bmatrix} \mathbf{x}_r^H \mathbf{x}_r & \mathbf{e}_d^H \mathbf{x}_r \\ \mathbf{x}_r^H \mathbf{e}_d & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \end{bmatrix} \quad (\text{A.45})$$

$$\begin{aligned} \tilde{\lambda} &= \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \tilde{\delta} \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \\ \tilde{\mu} &= \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon}. \end{aligned} \quad (\text{A.46})$$

$$\begin{aligned} \lambda &= \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \tilde{\delta} \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \\ \mu &= \left( \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right)^* \end{aligned} \quad (\text{A.47})$$

Complete Update

$$\begin{aligned} \mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\ &\eta \left( \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \tilde{\delta} \mathbf{e}_d^H \mathbf{x}_r)^*}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right) \mathbf{x}_r - \right. \\ &\quad \left. \left( \frac{\mathbf{x}_r^H \mathbf{x}_r (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \tilde{\delta} \mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r)^*}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r + \epsilon} \right) \mathbf{e}_d \right) \end{aligned} \quad (\text{A.48})$$

## Dual-Fixed Peak AXIS

Objective Function:

$$\begin{aligned}
& \min_{\mathbf{h}^{(n+1)*}} \quad \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\
& \text{subject to} \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \\
& \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta} \\
& \quad \quad \quad \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c = y
\end{aligned} \tag{A.49}$$

Derivation:

$$\begin{aligned}
\mathcal{L} = & \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} \\
& + \alpha \|\mathbf{h}^{(n+1)}\|_1 + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \tilde{\zeta}) + \beta (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y)
\end{aligned} \tag{A.50}$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d + \beta \mathbf{e}_c \tag{A.51}$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \lambda \mathbf{x}_r + \mu \mathbf{e}_d + \beta \mathbf{e}_c \tag{A.52}$$

$$\mathbf{h}^{(n+1)} = \mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d - \beta \mathbf{e}_c \tag{A.53}$$

Constraints:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$(\mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d - \beta \mathbf{e}_c)^{\text{H}} \mathbf{x}_r = 0 \tag{A.54}$$

$$(\mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{x}_r + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{x}_r + \tilde{\beta} \mathbf{e}_c^{\text{H}} \mathbf{x}_r \tag{A.55}$$

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta}:$$

$$(\mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r - \mu \mathbf{e}_d - \beta \mathbf{e}_c)^{\text{H}} \mathbf{e}_d = \tilde{\zeta} \tag{A.56}$$

$$(\mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - \tilde{\zeta} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_d + \tilde{\beta} \mathbf{e}_c^{\text{H}} \mathbf{e}_d \tag{A.57}$$

$$(\mathbf{h}^n - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^{\text{H}} \mathbf{e}_d - \tilde{\zeta} = \tilde{\lambda} \mathbf{x}_r^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_d^{\text{H}} \mathbf{e}_d + \tilde{\beta} \cdot 0 \tag{A.58}$$

$$\mathbf{h}^{(n+1)\text{H}}\mathbf{e}_c = y:$$

$$(\mathbf{h}^n - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}} - \lambda\mathbf{x}_r - \mu\mathbf{e}_d - \beta\mathbf{e}_c)\text{H}\mathbf{e}_c = y \quad (\text{A.59})$$

$$(\mathbf{h}^n - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y = \tilde{\lambda}\mathbf{x}_r\text{H}\mathbf{e}_c + \tilde{\mu}\mathbf{e}_d\text{H}\mathbf{e}_c + \tilde{\beta}\mathbf{e}_c\text{H}\mathbf{e}_c \quad (\text{A.60})$$

$$(\mathbf{h}^n - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y = \tilde{\lambda}\mathbf{x}_r\text{H}\mathbf{e}_c + \tilde{\mu} \cdot \mathbf{0} + \tilde{\beta}\mathbf{e}_c\text{H}\mathbf{e}_c \quad (\text{A.61})$$

System of Equations:

$$\begin{aligned} (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r &= \tilde{\lambda}\mathbf{x}_r\text{H}\mathbf{x}_r + \tilde{\mu}\mathbf{e}_d\text{H}\mathbf{x}_r + \tilde{\beta}\mathbf{e}_c\text{H}\mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta &= \tilde{\lambda}\mathbf{x}_r\text{H}\mathbf{e}_d + \tilde{\mu}\mathbf{e}_d\text{H}\mathbf{e}_d + \tilde{\beta}\mathbf{e}_c\text{H}\mathbf{e}_d \\ (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y &= \tilde{\lambda}\mathbf{x}_r\text{H}\mathbf{e}_c + \tilde{\mu}\mathbf{e}_d\text{H}\mathbf{e}_c + \tilde{\beta}\mathbf{e}_c\text{H}\mathbf{e}_c \end{aligned} \quad (\text{A.62})$$

$$\begin{bmatrix} \mathbf{x}_r\text{H}\mathbf{x}_r & \mathbf{e}_d\text{H}\mathbf{x}_r & \mathbf{e}_c\text{H}\mathbf{x}_r \\ \mathbf{x}_r\text{H}\mathbf{e}_d & 1 & 0 \\ \mathbf{x}_r\text{H}\mathbf{e}_c & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \\ \tilde{\beta} \end{bmatrix} = \begin{bmatrix} (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r \\ (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta \\ (\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y \end{bmatrix} \quad (\text{A.63})$$

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y)\mathbf{e}_c\text{H}\mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta)\mathbf{e}_d\text{H}\mathbf{x}_r}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon}$$

$$\begin{aligned} \tilde{\mu} = \frac{-\mathbf{x}_r\text{H}\mathbf{e}_d((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y)\mathbf{e}_c\text{H}\mathbf{x}_r)}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon} + \\ \frac{((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta)(\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r)}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon} \end{aligned} \quad (\text{A.64})$$

$$\begin{aligned} \tilde{\beta} = \frac{-\mathbf{x}_r\text{H}\mathbf{e}_c((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r - \mathbf{e}_d\text{H}\mathbf{x}_r((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta))}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon} + \\ \frac{((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y)(\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r)}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon} \end{aligned}$$

$$\lambda = \left( \frac{(\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_c - y)\mathbf{e}_c\text{H}\mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha\mathbf{e}^{j\angle\mathbf{h}^{(n)}})\text{H}\mathbf{e}_d - \delta)\mathbf{e}_d\text{H}\mathbf{x}_r}{\mathbf{x}_r\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_d\mathbf{e}_d\text{H}\mathbf{x}_r - \mathbf{x}_r\text{H}\mathbf{e}_c\mathbf{e}_c\text{H}\mathbf{x}_r + \epsilon} \right)^*$$

$$\mu = \left( \frac{-\mathbf{x}_r^H \mathbf{e}_d ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \check{\delta})(x_r x_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \quad (\text{A.65})$$

$$\beta = \left( \frac{-\mathbf{x}_r^H \mathbf{e}_c ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \check{\delta}))}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y)(x_r x_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^*$$

Complete Update:

$$\begin{aligned} \mathbf{h}^{(n+1)} &= \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \\ &\quad \eta \left( \left( \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} - \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \check{\delta}) \mathbf{e}_d^H \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r \right. \\ &+ \left( \frac{-\mathbf{x}_r^H \mathbf{e}_d ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y) \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \check{\delta})(x_r x_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_d \\ &+ \left. \left( \frac{-\mathbf{x}_r^H \mathbf{e}_c ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{x}_r - \mathbf{e}_d^H \mathbf{x}_r ((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_d - \check{\delta}))}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} + \frac{((\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H \mathbf{e}_c - y)(x_r x_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r)}{\mathbf{x}_r^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_d \mathbf{e}_d^H \mathbf{x}_r - \mathbf{x}_r^H \mathbf{e}_c \mathbf{e}_c^H \mathbf{x}_r + \epsilon} \right)^* \mathbf{e}_c \right) \quad (\text{A.66}) \end{aligned}$$

**Single Constraint**Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)\text{H}}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \mathbf{1}\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \end{aligned} \quad (\text{A.67})$$

Derivation:

$$\begin{aligned} \mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ + \gamma (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} + \mathbf{1} - \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d) + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r \end{aligned} \quad (\text{A.68})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma ((\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)}) - \mathbf{e}_d) + \lambda \mathbf{x}_r \quad (\text{A.69})$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma ((\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)}) - \mathbf{e}_d) + \lambda \mathbf{x}_r \quad (\text{A.70})$$

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d - \lambda \mathbf{x}_r) \quad (\text{A.71})$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d - \lambda \mathbf{x}_r))^{\text{H}} \mathbf{x}_r = 0 \quad (\text{A.72})$$

$$((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d))^{\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r \quad (\text{A.73})$$

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \quad (\text{A.74})$$

$$\lambda = \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \right)^* \quad (\text{A.75})$$

Complete Update

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} \left( \mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \mathbf{e}_d)^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} \mathbf{x}_r}{\mathbf{x}_r^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} \mathbf{x}_r} \right)^* \mathbf{x}_r \right) \quad (\text{A.76})$$

### Varied Single Constraint

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)\text{H}}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \tilde{\delta}\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \end{aligned} \quad (\text{A.77})$$

Derivation:

$$\begin{aligned} \mathcal{L} = & \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ & + \gamma (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} + |\tilde{\delta}|^2 - \tilde{\delta} \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \tilde{\delta}^* \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d) + \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r \end{aligned} \quad (\text{A.78})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \tilde{\delta}^* \mathbf{e}_d) + \lambda \mathbf{x}_r \quad (\text{A.79})$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \tilde{\delta}^* \mathbf{e}_d) + \lambda \mathbf{x}_r \quad (\text{A.80})$$

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \tilde{\delta}^* \mathbf{e}_d - \lambda \mathbf{x}_r) \quad (\text{A.81})$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \tilde{\delta}^* \mathbf{e}_d - \lambda \mathbf{x}_r))^{\text{H}} \mathbf{x}_r = 0 \quad (\text{A.82})$$

$$((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \tilde{\delta}^* \mathbf{e}_d))^{\text{H}} \mathbf{x}_r = \tilde{\lambda} \mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r \quad (\text{A.83})$$

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \tilde{\delta}^* \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \quad (\text{A.84})$$

$$\lambda = \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \tilde{\delta}^* \mathbf{e}_d)^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}})^{-1} \mathbf{x}_r} \right)^* \quad (\text{A.85})$$

Complete Update:

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \delta^* \mathbf{e}_d - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma \delta^* \mathbf{e}_d)^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} \mathbf{x}_r}{\mathbf{x}_r^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H)^{-1} \mathbf{x}_r} \right)^* \mathbf{x}_r) \quad (\text{A.86})$$



### Dual Fixed-Peak Single Constraint

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{h}^{(n+1)} - \mathbf{h}^{(n)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)\text{H}}\|_1 + \gamma \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \delta\|_2^2 + \phi \|\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y\|_2^2 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0 \end{aligned} \quad (\text{A.87})$$

Derivation:

$$\begin{aligned} \mathcal{L} = & \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n+1)} - \mathbf{h}^{(n+1)\text{H}} \mathbf{h}^{(n)} - \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n+1)} + \mathbf{h}^{(n)\text{H}} \mathbf{h}^{(n)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \\ & \lambda \mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r + \gamma (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} + |\delta|^2 - \delta \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \delta^* \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d) \\ & + \phi (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c \mathbf{e}_c^{\text{H}} \mathbf{h}^{(n+1)} + |y|^2 - y \mathbf{e}_c^{\text{H}} \mathbf{h}^{(n+1)} - y^* \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c). \end{aligned} \quad (\text{A.88})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{x}_r + \gamma (\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \delta^* \mathbf{e}_d) + \phi (\mathbf{e}_c \mathbf{e}_c^{\text{H}} \mathbf{h}^{(n+1)} - y^* \mathbf{e}_c) \quad (\text{A.89})$$

$$0 = \mathbf{h}^{(n+1)} - \mathbf{h}^{(n)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{x}_r + \gamma (\mathbf{e}_d \mathbf{e}_d^{\text{H}} \mathbf{h}^{(n+1)} - \delta^* \mathbf{e}_d) + \phi (\mathbf{e}_c \mathbf{e}_c^{\text{H}} \mathbf{h}^{(n+1)} - y^* \mathbf{e}_c) \quad (\text{A.90})$$

$$\mathbf{h}^{(n+1)} = (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c)) \quad (\text{A.91})$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{x}_r = 0:$$

$$((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} - \lambda \mathbf{x}_r + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c)))^{\text{H}} \mathbf{x}_r = 0 \quad (\text{A.92})$$

$$\begin{aligned} ((\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c)))^{\text{H}} \mathbf{x}_r = \\ \tilde{\lambda} \mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} \mathbf{x}_r \end{aligned} \quad (\text{A.93})$$

$$\tilde{\lambda} = \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c))^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} \mathbf{x}_r + \epsilon} \quad (\text{A.94})$$

$$\lambda = \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c))^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} \mathbf{x}_r}{\mathbf{x}_r^{\text{H}} (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^{\text{H}} + \phi \mathbf{e}_c \mathbf{e}_c^{\text{H}})^{-1} \mathbf{x}_r + \epsilon} \right)^* \quad (\text{A.95})$$

Complete Update

$$\begin{aligned}
\mathbf{h}^{(n+1)} &= (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H + \phi \mathbf{e}_c \mathbf{e}_c^H)^{-1} (\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c)) \\
&\quad - \eta \left( \frac{(\mathbf{h}^{(n)} - \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \gamma (\delta^* \mathbf{e}_d) + \phi (y^* \mathbf{e}_c))^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H + \phi \mathbf{e}_c \mathbf{e}_c^H)^{-1} \mathbf{x}_r}{\mathbf{x}_r^H (\mathbf{I} + \gamma \mathbf{e}_d \mathbf{e}_d^H + \phi \mathbf{e}_c \mathbf{e}_c^H)^{-1} \mathbf{x}_r + \epsilon} \right)^* \mathbf{x}_r \quad (\text{A.96})
\end{aligned}$$

**Brute Force**Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{X}_r \mathbf{h}^{(n+1)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = 1 \end{aligned} \quad (\text{A.97})$$

Derivation:

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - 1). \quad (\text{A.98})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d \quad (\text{A.99})$$

$$0 = \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d \quad (\text{A.100})$$

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d) \quad (\text{A.101})$$

Constraint:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = 1:$$

$$(-(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d))^H \mathbf{e}_d = 1 \quad (\text{A.102})$$

$$-((\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}}))^H \mathbf{e}_d - 1 = \tilde{\lambda} \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \quad (\text{A.103})$$

$$\tilde{\lambda} = \frac{-1 - [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d} \quad (\text{A.104})$$

$$\lambda = \left( \frac{-1 - [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d} \right)^* \quad (\text{A.105})$$

Complete Update:

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \eta \left( -\frac{1 + [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})]^H \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1})^H \mathbf{e}_d} \right)^* \mathbf{e}_d \right) \quad (\text{A.106})$$

### Varied Brute Force

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{X}_r \mathbf{h}^{(n+1)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta} \end{aligned} \quad (\text{A.107})$$

Derivation:

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \tilde{\zeta}). \quad (\text{A.108})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d \quad (\text{A.109})$$

$$0 = \mathbf{X}_r^H \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d \quad (\text{A.110})$$

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d) \quad (\text{A.111})$$

Constraint:  $\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta}$ :

$$-(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d) \mathbf{e}_d = \tilde{\zeta} \quad (\text{A.112})$$

$$-((\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})) \mathbf{e}_d - \tilde{\zeta} = \tilde{\lambda} \mathbf{e}_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} \mathbf{e}_d \quad (\text{A.113})$$

$$\tilde{\lambda} = \frac{-\tilde{\zeta} - [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})] \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1}) \mathbf{e}_d} \quad (\text{A.114})$$

$$\lambda = \left( \frac{-\tilde{\zeta} - [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})] \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1}) \mathbf{e}_d} \right)^* \quad (\text{A.115})$$

Complete Update:

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \eta \left( \frac{-\tilde{\zeta} + [(\mathbf{X}_r^H \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})] \mathbf{e}_d}{\mathbf{e}_d^H ((\mathbf{X}_r^H \mathbf{X}_r)^{-1}) \mathbf{e}_d} \right)^* \mathbf{e}_d \right) \quad (\text{A.116})$$

### Dual Fixed-Peak Brute Force

Objective Function:

$$\begin{aligned} \min_{\mathbf{h}^{(n+1)*}} \quad & \|\mathbf{X}_r \mathbf{h}^{(n+1)}\|_2^2 + \alpha \|\mathbf{h}^{(n+1)*}\|_1 \\ \text{subject to} \quad & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta} \\ & \mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c = y \end{aligned} \quad (\text{A.117})$$

Derivation:

$$\mathcal{L} = \mathbf{h}^{(n+1)\text{H}} \mathbf{X}_r^{\text{H}} \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \|\mathbf{h}^{(n+1)*}\|_1 + \lambda (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d - \tilde{\zeta}) + \mu (\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c - y) \quad (\text{A.118})$$

$$\frac{d}{d\mathbf{h}^{(n+1)*}} \mathcal{L} = \mathbf{X}_r^{\text{H}} \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c \quad (\text{A.119})$$

$$0 = \mathbf{X}_r^{\text{H}} \mathbf{X}_r \mathbf{h}^{(n+1)} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c \quad (\text{A.120})$$

$$\mathbf{h}^{(n+1)} = -(\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c) \quad (\text{A.121})$$

Constraints:

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_d = \tilde{\zeta}:$$

$$(-(\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c))^{\text{H}} \mathbf{e}_d = \tilde{\zeta} \quad (\text{A.122})$$

$$-\tilde{\zeta} - ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})^{\text{H}} \mathbf{e}_d = \tilde{\lambda} \mathbf{e}_d^{\text{H}} ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1})^{\text{H}} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_c^{\text{H}} ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1})^{\text{H}} \mathbf{e}_d \quad (\text{A.123})$$

$$\mathbf{h}^{(n+1)\text{H}} \mathbf{e}_c = y:$$

$$(-(\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} (\alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}} + \lambda \mathbf{e}_d + \mu \mathbf{e}_c))^{\text{H}} \mathbf{e}_c = y \quad (\text{A.124})$$

$$-y - ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})^{\text{H}} \mathbf{e}_c = \tilde{\lambda} \mathbf{e}_d^{\text{H}} ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1})^{\text{H}} \mathbf{e}_c + \tilde{\mu} \mathbf{e}_c^{\text{H}} ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1})^{\text{H}} \mathbf{e}_c \quad (\text{A.125})$$

Cramer's Rule:

$$\begin{aligned} -\tilde{\zeta} - ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})^{\text{H}} \mathbf{e}_d &= \tilde{\lambda} \mathbf{e}_d^{\text{H}} (\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \mathbf{e}_d + \tilde{\mu} \mathbf{e}_c^{\text{H}} (\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \mathbf{e}_d \\ -y - ((\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n+1)}})^{\text{H}} \mathbf{e}_c &= \tilde{\lambda} \mathbf{e}_d^{\text{H}} (\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \mathbf{e}_c + \tilde{\mu} \mathbf{e}_c^{\text{H}} (\mathbf{X}_r^{\text{H}} \mathbf{X}_r)^{-1} \mathbf{e}_c \end{aligned} \quad (\text{A.126})$$

$$\begin{bmatrix} e_d^H(\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d & e_c^H(\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d \\ e_d^H(\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c & e_c^H(\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} -\tilde{\gamma} - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H e_d \\ -y - ((\mathbf{X}_r^H \mathbf{X}_r)^{-1} \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H e_c \end{bmatrix} \quad (\text{A.127})$$

$$\tilde{\lambda} = \frac{(y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon} \quad (\text{A.128})$$

$$\tilde{\mu} = \frac{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d (y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon}$$

$$\lambda = \left( \frac{(y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon} \right)^* \quad (\text{A.129})$$

$$\mu = \left( \frac{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d (y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon} \right)^*$$

Complete Update:

$$\begin{aligned} \mathbf{h}^{(n+1)} = & -(\mathbf{X}_r^H \mathbf{X}_r)^{-1} \left( \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}} + \eta \left( \right. \right. \\ & \left. \left( \frac{(y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon} \right)^* e_d + \right. \\ & \left. \left. \left( \frac{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c (\tilde{\gamma} + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d (y + \alpha \mathbf{e}^{j\angle \mathbf{h}^{(n)}})^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c}{e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c - e_d^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_c e_c^H (\mathbf{X}_r^H \mathbf{X}_r)^{-1} e_d + \epsilon} \right)^* e_c \right) \right) \end{aligned} \quad (\text{A.130})$$

## APPENDIX B

## Code Listings

**B.1 Modifications**

The code listings here utilize a modification that has shown to provide faster and more accurate results without the need for a myriad of iterations over the same data. The adaptive algorithms use multiple rows of data simultaneously such that each step includes more information and can thus make a better approximation. A mathematical explanation for the ModAED algorithm is shown here for improved understanding, although it should be noted, that this modification applies to the other algorithms the same way.

Using the notation described in Appendix A.1, the data block,  $\mathbf{X}_r$ , is of size  $b \times M(L+1)$ , and the previous update vector,  $\mathbf{h}^{(n)}$ , is of size  $M(L+1) \times 1$ , as shown below:

$$\mathbf{X}_r = \begin{bmatrix} \text{---}x_1\text{---} \\ \text{---}x_2\text{---} \\ \vdots \\ \text{---}x_{b-1}\text{---} \\ \text{---}x_b\text{---} \end{bmatrix} \quad \mathbf{h}^{(n)} = \begin{bmatrix} | \\ \mathbf{h}_j^{(n)} \\ | \\ \dots \\ | \\ \mathbf{h}_i^{(n)} \\ | \end{bmatrix}.$$

where  $M = 2$ . Note that for this explanation, not all the vector are assumed to be in column formatting; rather the vectors will follow the same formatting shown above.

As it is repeatedly mathematically shown in Appendix A, a typical update involves a scaled version of the previous impulse response estimate vector multiplied by a row of the data matrix and then added to the previous impulse response estimate. The update for the ModAED algorithm as shown in Appendix A.2, is

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \eta \frac{\mathbf{x}_r^* \mathbf{h}^{(n)}}{\|\mathbf{x}_r\|_2^2} \mathbf{x}_r^T. \quad (\text{B.1})$$

As the update is written, the matrix multiplication would be

$$\underbrace{\begin{bmatrix} \mathbf{h}_j^{(n+1)} \\ \vdots \\ \mathbf{h}_i^{(n+1)} \end{bmatrix}}_{M(L+1) \times 1} = \underbrace{\begin{bmatrix} \mathbf{h}_j^{(n)} \\ \vdots \\ \mathbf{h}_i^{(n)} \end{bmatrix}}_{M(L+1) \times 1} - \eta \underbrace{\left( \underbrace{\begin{bmatrix} -\mathbf{x}_r^* \\ \vdots \\ \mathbf{h}_j^{(n)} \\ \vdots \\ \mathbf{h}_i^{(n)} \end{bmatrix}}_{1 \times 1} \right) / \left( \underbrace{\begin{bmatrix} -\mathbf{x}_r \\ \mathbf{x}_r^H \end{bmatrix}}_{1 \times 1} \right)}_{1 \times 1} \underbrace{\begin{bmatrix} \mathbf{x}_r^T \\ \vdots \end{bmatrix}}_{M(L+1) \times 1}. \quad (\text{B.2})$$

This method updates each individual coefficient of the previous update vector,  $\mathbf{h}^{(n)}$ , with the same weighted scalar of the data.

The data block modification requires Equation (B.1) to be rearranged, and modified without loss of generality, to

$$\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} - \eta \mathbf{X}_r^T \left( \frac{\mathbf{h}^{(n)H} \mathbf{X}_r^T}{\|\mathbf{X}_r\|_2^2} \right)^H. \quad (\text{B.3})$$

The matrix multiplication would (as follows) be slightly different from Equation (B.2).

$$\underbrace{\begin{bmatrix} \mathbf{h}_j^{(n+1)} \\ \vdots \\ \mathbf{h}_i^{(n+1)} \end{bmatrix}}_{M(L+1) \times 1} = \underbrace{\begin{bmatrix} \mathbf{h}_j^{(n)} \\ \vdots \\ \mathbf{h}_i^{(n)} \end{bmatrix}}_{M(L+1) \times 1} - \eta \underbrace{\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{b-1} \end{bmatrix}}_{M(L+1) \times b}$$

$$\underbrace{\left( \underbrace{\begin{bmatrix} -\mathbf{h}_j^{(n)*} & \vdots & -\mathbf{h}_i^{(n)*} \end{bmatrix}}_{1 \times b} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{b-1} \end{bmatrix} \right) / \left( \underbrace{\begin{bmatrix} \|\mathbf{x}_1\|_2^2 & \|\mathbf{x}_2\|_2^2 & \cdots & \|\mathbf{x}_b\|_2^2 \end{bmatrix}}_{1 \times b} \right)^H}_{b \times 1}. \quad (\text{B.4})$$



Note that for this instance, the  $./$  is an element-wise divide, meaning that each element of the left side is divided by the corresponding element on the right side of the divide (this is the same notation as in MATLAB).

When the impulse response estimate,  $\mathbf{h}^{(n+1)}$ , is then updated, the method differs in that each individual coefficient of the filter is individually updated, providing a better, more accurate estimate. This modification packs more information into each step, thus eliminating the need for more passes through the data, and a smarter more specialized adaptive method.

## B.2 The Least Squares (SVD) Approach According to Xu et al.

Listing B.1: LS Function

```
1 % Least Squares Singular Value Decomposition Algorithm
2 %   - The channel estimation algorithm that utilizes the SVD
3 %   Inputs:
4 %       - x: Toeplitz data matrix size: (N - L)xM(L + 1)
5 %       - L: the length of a transfer function estimate
6 %       - M: the number of channels
7 %   Outputs:
8 %       - EstChannels: vector containing the new updated estimate for
9 %         transfer functions
10 function EstChannels = LS(x, L, M)
11 %% Pre-Processing
12 %% Processing
13     % Call the SVD
14     [~, ~, hnew] = svd(x);
15 %% Output
16     % Format output
17     EstChannels = reshape(hnew(:, end), L + 1, M);
18 end
```

### B.3 The Adaptive Eigenvalue Decomposition

Listing B.2: AED Function

```

1 % Adaptive Eigenvalue Decomposition Algorithm
2 %   - The most basic update algorithm for channel estimation
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - L: the length of a transfer function estimate
10 %      - N: the amount of data in the Toeplitz data matrix
11 %      - M: the number of channels
12 %      - maxBlockReset: the data block size used for processing
13 %      - maxIter: the maximum amount of passes through the data
14 %   Outputs:
15 %       - EstChannels: vector containing new updated estimate for transfer
16 %         functions
17 function EstChannels = AED(hprev, x, eta, epsilon, L, N, M, ...,
18                               maxBlockReset, maxIter)
19 %% Pre-processing and Initialization
20 % Determine iterations through data
21     iters = N - (L + 1);
22 %% Processing
23 % Make Multiple Passes through data (if data chunks are sufficiently
24 % small)
25     for iter = 1:maxIter
26         % Reset block size
27         block = maxBlockReset;
28         % Iterate through data
29         for row = 1:block:iters
30             % Verify that the indices are valid
31             if row > N - block
32                 % Set maxBlock such that the indices are not violated

```

```
33         block = N - row - (L + 1);
34     end
35     % Get "row" of data
36     xr = x(row:row + block - 1, :).';
37     % Compute Lagrange multiplier
38     tilde_lambda = (hprev'*xr)/(norm(hprev) + epsilon);
39     % Compute update
40     hnew = hprev - eta*xr*tilde_lambda';
41     hnew = hnew/norm(hnew);
42     % Save new update for next iteration
43     hprev = hnew;
44 end
45 end
46 %% Output
47 % Format output
48 EstChannels = reshape(hnew, L + 1, M);
49 end
```

## B.4 The Modified Adaptive Eigenvalue Decomposition

Listing B.3: modAED Function

```

1 % Modified Adaptive Eigenvalue Decomposition Algorithm
2 %   - The most basic update algorithm for channel estimation
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - L: the length of a transfer function estimate
10 %      - N: the amount of data in the Toeplitz data matrix
11 %      - M: the number of channels
12 %      - maxBlockReset: the data block size used for processing
13 %      - maxIter: the maximum amount of passes through the data
14 %   Outputs:
15 %       - EstChannels: vector containing new updated estimate for transfer
16 %         functions
17 function EstChannels = modAED(hprev, x, eta, epsilon, L, N, M, ...
18                               maxBlockReset, maxIter)
19 %% Pre-processing and Initializatin
20 % Determine iterations through data
21     iters = N - (L + 1);
22 %% Processing
23 % Make Multiple Passes through data (if data chunks are sufficiently
24 % small)
25     for iter = 1:maxIter
26         % Reset block size
27         block = maxBlockReset;
28         % Iterate through data
29         for row = 1:block:iters
30             % Verify that the indices are valid
31             if row > N - block
32                 % Set maxBlock such that the indices are not violated

```

```
33         block = N - row - (L + 1);
34     end
35     % Get "row" of data
36     xr = x(row:row + block - 1, :).';
37     % Compute Lagrange multiplier
38     tilde_lambda = (hprev'*xr)/(norm(xr) + epsilon);
39     % Compute update
40     hnew = hprev - eta*xr*tilde_lambda';
41     % Save new update for next iteration
42     hprev = hnew;
43 end
44 end
45 %% Output
46 % Format output
47 EstChannels = reshape(hnew, L + 1, M);
48 end
```

## B.5 The Modified Adaptive Eigenvalue Decomposition with Sparsity

Listing B.4: modAEDS Function

```

1 % Modified Adaptive Eigenvalue Decomposition Algorithm with Sparsity
2 %   - The basic update algorithm with a sparsity constraint
3 %     for channel estimation
4 %   Inputs:
5 %     - hprev: column vector containing all the transfer functions
6 %             estimates size: M(L + 1)x1
7 %     - x: Toeplitz data matrix size: (N - L)xM(L + 1)
8 %     - eta: stepsize for update (scalar)
9 %     - alpha: the weight for the sparsity constraint
10 %    - epsilon: the offset for the denominator to be nonzero (small)
11 %    - L: the length of a transfer function estimate
12 %    - N: the amount of data in the Toeplitz data matrix
13 %    - M: the number of channels
14 %    - maxBlockReset: the data block size used for processing
15 %    - maxIter: the maximum amount of passes through the data
16 %   Outputs:
17 %     - EstChannels: vector containing new updated estimate for transfer
18 %       functions
19 function EstChannels = AEDS(hprev, x, eta, alpha, epsilon, L, N, M,...
20                               maxBlockReset, maxIter)
21 %% Pre-processing and Initialization
22   % Determine iterations through data
23   iters = N - (L + 1);
24 %% Processing
25   % Make Multiple Passes through data (if data chunks are sufficiently
26   % small)
27   for iter = 1:maxIter
28       % Reset block size
29       block = maxBlockReset;
30       % Iterate through data
31       for row = 1:block:iters
32           % Verify that the indices are valid

```

```

33     if row > N - block
34         % Set maxBlock such that the indices are not violated
35         block = N - row - (L + 1);
36     end
37     % Get "row" of data
38     xr = x(row:row + block - 1, :).';
39     % Compute Lagrange multiplier
40     tilde_lambda = ((hprev - alpha*exp(1j*phase(hprev)))'*xr)/...
41                   (norm(xr) + epsilon);
42     % Compute update
43     hnew = hprev - alpha*exp(1j*phase(hprev)) - ...
44           eta*xr*tilde_lambda';
45     % Save new update for next iteration
46     hprev = hnew;
47     end
48 end
49 %% Output
50 % Format output
51 EstChannels = reshape(hnew, L + 1, M);
52 end

```



## B.6 The Adaptive Cross-Channel Identification with Sparse Shift-Suppression

Listing B.5: AXIS Function

```

1 % Adaptive Cross-Channel Identification with Sparse Shift-Suppression
2 % Algorithm
3 %   - The modified update algorithm for channel estimation with sparsity
4 %     and shift-suppression
5 %   Inputs:
6 %     - hprev: column vector containing all the transfer functions
7 %               estimates size:  $M(L + 1) \times 1$ 
8 %     - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
9 %     - eta: stepsize for update (scalar)
10 %     - epsilon: the offset for the denominator to be nonzero (small)
11 %     - alpha: the weight for the sparsity constraint
12 %     - L: the length of a transfer function estimate
13 %     - N: the amount of data in the Toeplitz data matrix
14 %     - M: the number of channels
15 %     - maxBlockReset: the data block size used for processing
16 %     - maxIter: the maximum amount of passes through the data
17 %     - d: the index for the elementary vector
18 %   Outputs:
19 %     - EstChannels: vector containing new updated estimate for transfer
20 %       functions
21 function EstChannels = AXIS(hprev, x, eta, alpha, epsilon, L, N, M,...
22                               maxBlockReset, maxIter, d)
23 %% Pre-Processing and Initialization
24 % Determine iterations through data
25     iters = N - (L + 1);
26 % Set elementary vector
27     ed = zeros(M*(L + 1), 1);
28     ed(d) = 1;
29 %% Processing
30 % Make multiple passes through data (if data chunks are sufficiently
31 % small)
32     for iter = 1:maxIter

```

```

33     % Reset the block size
34     block = maxBlockReset;
35     % Iterate through data
36     for row = 1:block:iters
37         % Verify that the indices are valid
38         if row > N - block
39             % Set maxBlock such that the indices are not violated
40             block = N - row - (L + 1);
41         end
42         % Get "row" of data
43         xr = x(row:row + block - 1, :).';
44         % set ed block matrix for ease of computation
45         edMat = zeros(size(xr));
46         edMat(d,:) = ones(1, block);
47         % Compute place holder variable for repeated variables
48         hest = (hprev - alpha*exp(1j*phase(hprev)));
49         % Compute Lagrange Multipliers
50         den = (vecnorm(xr).' - (xr'*ed).*(ed'*xr).' + epsilon).';
51         tilde_lambda = (hest'*xr - (hest'*ed)*(ed'*xr) + ed'*xr)./den;
52         tilde_mu = (vecnorm(xr)*(hest'*ed) - vecnorm(xr) - ...
53                     (xr'*ed).'.*(hest'*xr))./den;
54         % Compute Update
55         hnew = hest - eta*(xr*tilde_lambda' + edMat*tilde_mu');
56         % Save new update for next iteration
57         hprev = hnew;
58     end
59 end
60 %% Output
61 % Format output
62 EstChannels = reshape(hnew, L + 1, M);
63 end

```

## B.7 The Varied Adaptive Cross-Channel Identification with Sparse Shift-Suppression

Listing B.6: VAXIS Function

```

1 % Varied Adaptive Cross-Channel Identification with Sparse Shift-
2 % Suppression Algorithm
3 %   - The modified update algorithm for channel estimation with sparsity,
4 %     shift-suppression and flexibility
5 %   Inputs:
6 %     - hprev: column vector containing all the transfer functions
7 %             estimates size:  $M(L + 1) \times 1$ 
8 %     - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
9 %     - eta: stepsize for update (scalar)
10 %    - epsilon: the offset for the denominator to be nonzero (small)
11 %    - alpha: the weight for the sparsity constraint
12 %    - L: the length of a transfer function estimate
13 %    - N: the amount of data in the Toeplitz data matrix
14 %    - M: the number of channels
15 %    - maxBlockReset: the data block size used for processing
16 %    - maxIter: the maximum amount of passes through the data
17 %    - d: the index for the elementary vector
18 %   Outputs:
19 %     - EstChannels: vector containing new updated estimate for transfer
20 %       functions
21 function EstChannels = vAXIS(hprev, x, eta, alpha, epsilon, L, N, M,...
22                               maxBlockReset, maxIter, d)
23 %% Pre-Processing and Initialization
24 % Determine iterations through data
25 iters = N - (L + 1);
26 % Set elementary vector
27 ed = zeros(M*(L + 1), 1);
28 ed(d) = 1;
29 % initialize flexible variable
30 z = 1;
31 %% Processing
32 % Make multiple passes through data (if data chunks are sufficiently

```

```

33 % small)
34 for iter = 1:maxIter
35     % Reset the block size
36     block = maxBlockReset;
37     % Iterate through data
38     for row = 1:block:iters
39         % Verify that the indices are valid
40         if row > N - block
41             % Set maxBlock such that the indices are not violated
42             block = N - row - (L + 1);
43         end
44         % Get "row" of data
45         xr = x(row:row + block - 1, :).';
46         % set ed block matrix for ease of computation
47         edMat = zeros(size(xr));
48         edMat(d,:) = ones(1, block);
49         % Compute place holder variable for repeated variables
50         hest = (hprev - alpha*exp(1j*phase(hprev)));
51         % Compute Lagrange Multipliers
52         den = (vecnorm(xr).' - (xr'*ed).*(ed'*xr).' + epsilon).';
53         tilde_lambda = (hest'*xr - (hest'*ed)*(ed'*xr) + ...
54                                 z*ed'*xr)./den;
55         tilde_mu = (vecnorm(xr)*(hest'*ed) - z*vecnorm(xr) - ...
56                                 (xr'*ed).*(hest'*xr)).'/den;
57         % Compute Update
58         hnew = hest - eta*(xr*tilde_lambda' + edMat*tilde_mu');
59         % Save new update for next iteration
60         hprev = hnew;
61         % Determine new value for z
62         z = hprev(d);
63     end
64 end
65 %% Output
66 % Format output
67 EstChannels = reshape(hnew, L + 1, M);

```



## B.8 The Dual Fixed-Peak Adaptive Cross-Channel Identification with Sparse Shift-Suppression

Listing B.7: DFPAXIS Function

```

1 % Dual Fixed-Peak Adaptive Cross-Channel Identification with Sparse Shift-
2 % Suppression Algorithm
3 %   - The modified update algorithm to combat asymmetric treatment of the
4 %     channels for channel estimation with sparsity,
5 %     shift-suppression and flexibility
6 %   Inputs:
7 %     - hprev: column vector containing all the transfer functions
8 %             estimates size: M(L + 1)x1
9 %     - x: Toeplitz data matrix size: (N - L)xM(L + 1)
10 %    - eta: stepsize for update (scalar)
11 %    - epsilon: the offset for the denominator to be nonzero (small)
12 %    - alpha: the weight for the sparsity constraint
13 %    - L: the length of a transfer function estimate
14 %    - N: the amount of data in the Toeplitz data matrix
15 %    - M: the number of channels
16 %    - maxBlockReset: the data block size used for processing
17 %    - maxIter: the maximum amount of passes through the data
18 %    - d: the index for the elementary vector
19 %   Outputs:
20 %     - EstChannels: vector containing new updated estimate for transfer
21 %       functions
22 function EstChannels = DFPAXIS(hprev, x, eta, alpha, epsilon, L, N, M,...
23                               maxBlockReset, maxIter, d)
24 %% Pre-Processing and Initialization
25 % Determine iterations through data
26     iters = N - (L + 1);
27 % Set elementary vector
28     ed = zeros(M*(L + 1), 1);
29     ed(d) = 1;
30 % initialize flexible variable
31     z = 1;

```

```

32 %% Processing
33 % Make multiple passes through data (if data chunks are sufficiently
34 % small)
35 for iter = 1:maxIter
36     % Reset the block size
37     block = maxBlockReset;
38     % Iterate through data
39     for row = 1:block:iters
40         % Verify that the indices are valid
41         if row > N - block
42             % Set maxBlock such that the indices are not violated
43             block = N - row - (L + 1);
44         end
45         % Get "row" of data
46         xr = x(row:row + block - 1, :).';
47         % set ed block matrix for ease of computation
48         edMat = zeros(size(xr));
49         edMat(d,:) = ones(1, block);
50         % Check if first iteration
51         if row == 1
52             % Call normal AXIS algorithm to start algorithm
53             EstChannels = AXIS(hprev, xr.', eta, alpha, epsilon, L,...
54                               (L + 1) + block, M, block, 1, d);
55             % Reformat output
56             hprev = reshape(EstChannels, M*(L + 1), 1);
57             % Determine which indice for the max peak
58             [~, c] = max(hprev(L + 1:end));
59             % Make corresponding elementary vector
60             ec = zeros(M*(L + 1), 1);
61             ec(c) = 1;
62             % If not the first iteration
63         else
64             % set ec block matrix for ease of computation
65             ecMat = zeros(size(xr));
66             ecMat(c,:) = ones(1, block);

```

```

67     % Compute place holder variable for repeated variables
68     hest = (hprev - alpha*exp(1j*phase(hprev)));
69     % Compute Lagrange Multipliers
70     den = (vecnorm(xr).' - (xr'*ed).*(ed'*xr).' - ...
71           (xr'*ec).*(ec'*xr).' + epsilon).';
72     tilde_lambda = (hest'*xr - (hest'*ec - y)*ec'*xr ...
73                   - (hest'*ed - z)*ed'*xr)./den;
74     tilde_mu = ((hest'*ed - z)*(vecnorm(xr).' - ...
75               (xr'*ec).*(ec'*xr).' - ...
76               (xr'*ed).*(hest'*xr - ...
77                 (hest'*ec - y)*ec'*xr).')'.)/den;
78     tilde_beta = ((hest'*ec - y)*(vecnorm(xr).' - ...
79                 (xr'*ed).*(ed'*xr).' - ...
80                 (xr'*ec).*(hest'*xr - ...
81                   (ed'*xr).*(hest'*ed - z)).')'.)/den;
82     % Compute Update
83     hnew = hest - eta*(xr*tilde_lambda' + ...
84               edMat*tilde_mu' + ecMat*tilde_beta');
85     % Save new update for next iteration
86     hprev = hnew;
87     end
88     % Determine new value for z and y
89     z = hprev(d);
90     y = hprev(c);
91     end
92     end
93 %% Output
94 % Format output
95 EstChannels = reshape(hnew, L + 1, M);
96 end

```



## B.9 The Single Constraint Method

Listing B.8: SC Function

```

1 % Single Constraint Algorithm
2 %   - The algorithm with only one extra constraint
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - alpha: the weight for the sparsity constraint
10 %      - gamma: the weight for the shift-suppression constraint
11 %      - L: the length of a transfer function estimate
12 %      - N: the amount of data in the Toeplitz data matrix
13 %      - M: the number of channels
14 %      - maxBlockReset: the data block size used for processing
15 %      - maxIter: the maximum amount of passes through the data
16 %      - d: the index for the elementary vector
17 %   Outputs:
18 %       - EstChannels: vector containing new updated estimate for transfer
19 %         functions
20 function EstChannels = SC(hprev, x, eta, alpha, gamma, epsilon, L, N, M,...
21                           maxBlockReset, maxIter, d)
22 %% Pre-Processing and Initialization
23 % Determine iterations through data
24 iters = N - L;
25 % Set elementary vector
26 ed = zeros(M*(L + 1), 1);
27 ed(d) = 1;
28 % Set up identity matrix
29 I = eye(M*(L + 1));
30 %% Processing
31 % Make multiple passes through data (if data chunks are sufficiently
32 % small)

```

```

33     for iter = 1:maxIter
34         % Reset the block size
35         block = maxBlockReset;
36         % Iterate through data
37         for row = 1:block:iters
38             % Verify that the indices are valid
39             if row > N - block
40                 % Set maxBlock such that the indices are not violated
41                 block = N - row - (L + 1);
42             end
43             % Get "row" of data
44             xr = x(row:row + block - 1, :).';
45             % Compute place holder variable for repeated variables
46             inv = (I + gamma*ed*ed')\I;
47             hest = (hprev - alpha*exp(1j*phase(hprev)) + gamma*ed);
48             den = zeros(block, 1); xrt = xr';
49             for cols = 1:block
50                 temp = inv*xr(:, cols);
51                 den(cols) = xrt(cols, :)*temp;
52             end
53             % Compute Lagrange Multiplier
54             tilde_lambda = ((hest'*inv)*xr)./(den + epsilon).';
55             % Compute Update
56             hnew = inv*(hest - eta*(xr*tilde_lambda'));
57             % Save new update for next iteration
58             hprev = hnew;
59         end
60     end
61 %% Output
62 % Format output
63 EstChannels = reshape(hnew, L + 1, M);
64 end

```

## B.10 The Varied Single Constraint Method

Listing B.9: VAXIS Function

```

1 % Varied Single Constraint Algorithm
2 %   - The algorithm with only one extra constraint with flexibility
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - alpha: the weight for the sparsity constraint
10 %      - gamma: the weight for the shift-suppression constraint
11 %      - L: the length of a transfer function estimate
12 %      - N: the amount of data in the Toeplitz data matrix
13 %      - M: the number of channels
14 %      - maxBlockReset: the data block size used for processing
15 %      - maxIter: the maximum amount of passes through the data
16 %      - d: the index for the elementary vector
17 %   Outputs:
18 %       - EstChannels: vector containing new updated estimate for transfer
19 %         functions
20 function EstChannels = vSC(hprev, x, eta, alpha, gamma, epsilon, L, N, M,...
21                             maxBlockReset, maxIter, d)
22 %% Pre-Processing and Initialization
23     % Determine iterations through data
24     iters = N - L;
25     % Set elementary vector
26     ed = zeros(M*(L + 1), 1);
27     ed(d) = 1;
28     % Set up identity matrix
29     I = eye(M*(L + 1));
30     % initialize flexible variable
31     z = 1;
32 %% Processing

```

```

33 % Make multiple passes through data (if data chunks are sufficiently
34 % small)
35 for iter = 1:maxIter
36     % Reset the block size
37     block = maxBlockReset;
38     % Iterate through data
39     for row = 1:block:iters
40         % Verify that the indices are valid
41         if row > N - block
42             % Set maxBlock such that the indices are not violated
43             block = N - row - (L + 1);
44         end
45         % Get "row" of data
46         xr = x(row:row + block - 1, :).';
47         % Compute place holder variable for repeated variables
48         inv = (I + gamma*ed*ed')\I;
49         hest = (hprev - alpha*exp(1j*phase(hprev)) + gamma*ed*conj(z));
50         den = zeros(block, 1); xrt = xr';
51         for cols = 1:block
52             temp = inv*xr(:, cols);
53             den(cols) = xrt(cols, :)*temp;
54         end
55         % Compute Lagrange Multiplier
56         tilde_lambda = ((hest'*inv)*xr)./(den + epsilon).';
57         % Compute Update
58         hnew = inv*(hest - eta*(xr*tilde_lambda'));
59         % Save new update for next iteration
60         hprev = hnew;
61         % Determine new value for z
62         z = hprev(d);
63     end
64 end
65 %% Output
66 % Format output
67 EstChannels = reshape(hnew, L + 1, M);

```



## B.11 The Dual Fixed-Peak Single Constraint Method

Listing B.10: DFPSC Function

```

1 % Dual Fixed-Point Single Constraint Algorithm
2 %   - The algorithm with only one extra constraint
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - alpha: the weight for the sparsity constraint
10 %      - gamma: the weight for the shift-suppression constraint
11 %      - L: the length of a transfer function estimate
12 %      - N: the amount of data in the Toeplitz data matrix
13 %      - M: the number of channels
14 %      - maxBlockReset: the data block size used for processing
15 %      - maxIter: the maximum amount of passes through the data
16 %      - d: the index for the elementary vector
17 %   Outputs:
18 %       - EstChannels: vector containing new updated estimate for transfer
19 %         functions
20 function EstChannels = DFPSC(hprev, x, eta, alpha, gamma, phi, epsilon, ...
21                             L, N, M, maxBlockReset, maxIter, d)
22 %% Pre-Processing and Initialization
23 % Determine iterations through data
24 iters = N - L;
25 % Set elementary vector
26 ed = zeros(M*(L + 1), 1);
27 ed(d) = 1;
28 % Set up identity matrix
29 I = eye(M*(L + 1));
30 % initialize flexible variable
31 z = 1;
32 %% Processing

```

```

33 % Make multiple passes through data (if data chunks are sufficiently
34 % small)
35 for iter = 1:maxIter
36     % Reset the block size
37     block = maxBlockReset;
38     % Iterate through data
39     for row = 1:block:iters
40         % Verify that the indices are valid
41         if row > N - block
42             % Set maxBlock such that the indices are not violated
43             block = N - row - (L + 1);
44         end
45         % Get "row" of data
46         xr = x(row:row + block - 1, :).';
47         % Check if first iteration
48         if row == 1
49             % Call normal SC algorithm to start algorithm
50             EstChannels = SC(hprev, x, eta, alpha, gamma, epsilon, ...
51                             L, N, M, maxBlockReset, maxIter, d);
52             % Reformat output
53             hprev = reshape(EstChannels, M*(L + 1), 1);
54             % Determine which indice for the max peak
55             [~, c] = max(hprev(L + 1:end));
56             % Make corresponding elementary vector
57             ec = zeros(M*(L + 1), 1);
58             ec(c) = 1;
59             % If not the first iteration
60         else
61             % Compute place holder variable for repeated variables
62             inv = (I + gamma*ed*ed' + phi*ec*ec')\I;
63             hest = (hprev - alpha*exp(1j*phase(hprev)) + ...
64                     phi*ec*conj(y) + gamma*ed*conj(z));
65             den = zeros(block, 1); xrt = xr';
66             for cols = 1:block
67                 temp = inv*xr(:, cols);

```

```
68         den(cols) = xrt(cols, :)*temp;
69     end
70     % Compute Lagrange Multiplier
71     tilde_lambda = (hest'*inv*xr)./(den + epsilon).';
72     % Compute Update
73     hnew = inv*(hest - eta*(xr*tilde_lambda'));
74     % Save new update for next iteration
75     hprev = hnew;
76 end
77 % Determine new value for z
78 z = hprev(d);
79 y = hprev(c);
80 end
81 end
82 %% Output
83 % Format output
84 EstChannels = reshape(hnew, L + 1, M);
85 end
```



## B.12 The Brute Force Method

Listing B.11: BF Function

```

1 % "Brute" Force Algorithm
2 %   - The update algorithm to that "forces" the results
3 %   Inputs:
4 %       - hprev: column vector containing all the transfer functions
5 %             estimates size:  $M(L + 1) \times 1$ 
6 %       - x: Toeplitz data matrix size:  $(N - L) \times M(L + 1)$ 
7 %       - eta: stepsize for update (scalar)
8 %       - epsilon: the offset for the denominator to be nonzero (small)
9 %       - alpha: the weight for the sparsity constraint
10 %      - L: the length of a transfer function estimate
11 %      - N: the amount of data in the Toeplitz data matrix
12 %      - M: the number of channels
13 %      - maxBlockReset: the data block size used for processing
14 %      - maxIter: the maximum amount of passes through the data
15 %      - d: the index for the elementary vector
16 %   Outputs:
17 %       - EstChannels: vector containing new updated estimate for transfer
18 %         functions
19 function EstChannels = BF(hprev, x, eta, epsilon, alpha, L, N, M, ...
20                               maxBlockReset, maxIter, d)
21 %% Pre-Processing and Initialization
22 % Determine iterations through data
23     iters = N - (L + 1);
24 % Set elementary vector
25     ed = zeros(M*(L + 1), 1);
26     ed(d) = 1;
27 % Set up identity matrix
28     I = eye(M*(L + 1));
29 %% Processing
30 % Make multiple passes through data (if data chunks are sufficiently
31 % small)
32     for iter = 1:maxIter

```

```

33     % Reset the block size
34     block = maxBlockReset;
35     % Iterate through data
36     for row = 1:block:iters
37         % Verify that the indices are valid
38         if row > N - block
39             % Set maxBlock such that the indices are not violated
40             block = N - row - (L + 1);
41         end
42         % Get "row" of data
43         xr = x(row:row + block - 1, :);
44         % Compute place holder variable for repeating value
45         Xinv = (xr'*xr)\I;
46         % Compute Lagrange Multiplier
47         tilde_lambda = (-1 - (Xinv*alpha*exp(1j*phase(hprev)))'*ed)...
48                             /(ed'*Xinv*ed + epsilon);
49         % Compute Update
50         hnew = -Xinv*(alpha*exp(1j*phase(hprev)) + ...
51                             eta*conj(tilde_lambda)*ed);
52         % Save new update for next iteration
53         hprev = hnew;
54     end
55 end
56 %% Output
57 % Format output
58 EstChannels = reshape(hnew, L + 1, M);
59 end

```

## B.13 The Varied Brute Force Method

Listing B.12: VBF Function

```

1 % Varied "Brute" Force Algorithm
2 %   - The update algorithm to that "forces" the results, with some
3 %   flexibility
4 %   Inputs:
5 %       - hprev: column vector containing all the transfer functions
6 %             estimates size: M(L + 1)x1
7 %       - x: Toeplitz data matrix size: (N - L)xM(L + 1)
8 %       - eta: stepsize for update (scalar)
9 %       - epsilon: the offset for the denominator to be nonzero (small)
10 %      - alpha: the weight for the sparsity constraint
11 %      - L: the length of a transfer function estimate
12 %      - N: the amount of data in the Toeplitz data matrix
13 %      - M: the number of channels
14 %      - maxBlockReset: the data block size used for processing
15 %      - maxIter: the maximum amount of passes through the data
16 %      - d: the index for the elementary vector
17 %   Outputs:
18 %       - EstChannels: vector containing new updated estimate for transfer
19 %         functions
20 function EstChannels = vBF(hprev, x, eta, epsilon, alpha, L, N, M, ...
21                             maxBlockReset, maxIter, d)
22 %% Pre-Processing and Initialization
23     % Determine iterations through data
24     iters = N - (L + 1);
25     % Set elementary vector
26     ed = zeros(M*(L + 1), 1);
27     ed(d) = 1;
28     % Set up identity matrix
29     I = eye(M*(L + 1));
30     % Initialize flexible variable
31     z = 1;
32 %% Processing

```

```

33 % Make multiple passes through data (if data chunks are sufficiently
34 % small)
35 for iter = 1:maxIter
36     % Reset the block size
37     block = maxBlockReset;
38     % Iterate through data
39     for row = 1:block:iters
40         % Verify that the indices are valid
41         if row > N - block
42             % Set maxBlock such that the indices are not violated
43             block = N - row - (L + 1);
44         end
45         % Get "row" of data
46         xr = x(row:row + block - 1, :);
47         % Compute place holder variable for repeating value
48         Xinv = (xr'*xr)\I;
49         % Compute Lagrange Multiplier
50         tilde_lambda = (-z - (Xinv*alpha*exp(1j*phase(hprev)))'*ed)...
51                         /(ed'*Xinv*ed + epsilon);
52         % Compute Update
53         hnew = -Xinv*(alpha*exp(1j*phase(hprev)) + ...
54                 eta*conj(tilde_lambda)*ed);
55         % Save new update for next iteration
56         hprev = hnew;
57     end
58 end
59 %% Output
60 % Format output
61 EstChannels = reshape(hnew, L + 1, M);
62 end

```

## B.14 The Dual Fixed-Peak Brute Force Method

Listing B.13: DFPBF Function

```

1 % Dual Fixed "Brute" Force Algorithm
2 %   - The update algorithm to that "forces" the results, treating the two
3 %     transfer functions symmetrically
4 %   Inputs:
5 %     - hprev: column vector containing all the transfer functions
6 %             estimates size: M(L + 1)x1
7 %     - x: Toeplitz data matrix size: (N - L)xM(L + 1)
8 %     - eta: stepsize for update (scalar)
9 %     - epsilon: the offset for the denominator to be nonzero (small)
10 %    - alpha: the weight for the sparsity constraint
11 %    - L: the length of a transfer function estimate
12 %    - N: the amount of data in the Toeplitz data matrix
13 %    - M: the number of channels
14 %    - maxBlockReset: the data block size used for processing
15 %    - maxIter: the maximum amount of passes through the data
16 %    - d: the index for the elementary vector
17 %   Outputs:
18 %     - EstChannels: vector containing new updated estimate for transfer
19 %       functions
20 function EstChannels = DFPBF(hprev, x, eta, epsilon, alpha, L, N, M, ...
21                               maxBlockReset, maxIter, d)
22 %% Pre-Processing and Initialization
23 % Determine iterations through data
24 iters = N - (L + 1);
25 % Set elementary vector
26 ed = zeros(M*(L + 1), 1);
27 ed(d) = 1;
28 % Set up identity matrix
29 I = eye(M*(L + 1));
30 % Initialize flexible variable
31 z = 1;
32 %% Processing

```

```

33 % Make multiple passes through data (if data chunks are sufficiently
34 % small)
35 for iter = 1:maxIter
36     % Reset the block size
37     block = maxBlockReset;
38     % Iterate through data
39     for row = 1:block:iters
40         % Verify that the indices are valid
41         if row > N - block
42             % Set maxBlock such that the indices are not violated
43             block = N - row - (L + 1);
44         end
45         % Get "row" of data
46         xr = x(row:row + block - 1, :);
47         % Check if first iteration
48         if row == 1
49             % Call normal BF algorithm to start algorithm
50             EstChannels = BF(hprev, xr, eta, alpha, epsilon, L,...
51                             block, M, block, 1, d);
52             % Reformat output
53             hprev = reshape(EstChannels, M*(L + 1), 1);
54             % Determine which indice for the max peak
55             [~, c] = max(hprev(L + 1:end));
56             % Make corresponding elementary vector
57             ec = zeros(M*(L + 1), 1);
58             ec(c) = 1;
59             % If not the first iteration
60             else
61                 % Compute place holder variables for repeating value
62                 Xinv = (xr'*xr)\I;
63                 hed = (-z - (Xinv*alpha*exp(1j*phase(hprev)))*ec);
64                 hec = (-y - (Xinv*alpha*exp(1j*phase(hprev)))*ec);
65                 den = epsilon + ed'*Xinv*ed*ec'*Xinv*ec - ...
66                             ed'*Xinv*ec*ec'*Xinv*ed;
67                 % Compute Lagrange Multipliers

```

```
68         tilde_lambda = (hed*ec'*Xinv*ec - hec*ec'*Xinv*ed)/den;
69         tilde_mu = (ed'*Xinv*ed*hec - ed'*Xinv*ec*hed)/den;
70         % Compute Update
71         hnew = -Xinv*(alpha*exp(1j*phase(hprev)) + eta*(...
72             conj(tilde_lambda)*ed) + conj(tilde_mu)*ec);
73         % Save new update for next iteration
74         hprev = hnew;
75     end
76     % Determine new value for z and y
77     z = hprev(d);
78     y = hprev(c);
79     end
80 end
81 %% Output
82 % Format output
83 EstChannels = reshape(hnew, L + 1, M);
84 end
```

## B.15 The Cross Correlation Function

Listing B.14: CC Function

```
1 % Cross Correlation Delay Estimation Algorithm
2 %   - The channel estimation algorithm that utilizes the SVD
3 %   Inputs:
4 %       - x: data vector size: MxN
5 %   Outputs:
6 %       - correlation: vector containing cross correlation values
7 function correlation = CC(x)
8 %% Pre-processing
9     % Separate matrices
10    x1 = x(1, :);
11    x2 = x(2, :);
12 %% Processing/Output
13    % Get correlation
14    correlation = xcorr(x1, x2);
15 end
```



## B.16 The Overhead File to Call the Algorithms

Listing B.15: Overhead File

```

1 %% Overhead - Real
2 % This version incorporates multiple functions for ease of processing for
3 % real observed data collected using the SDRs
4 clear all;
5 close all;
6 clc;
7 % Add all files needed to working directory
8 dirs{1} = '/home/madi/Research_2018-2019/tdoa-bcid/Algorithms_MATLAB_Take2/Real/
           algorithms/';
9 dirs{2} = '/home/madi/Research_2018-2019/tdoa-bcid/Algorithms_MATLAB_Take2/Real/
           data/';
10 % Cycle through and add paths
11 for fold = 1:length(dirs)
12     % Add folders
13     addpath(dirs{fold});
14 end
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 % Initialization and Pre-Processing
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 % Initializations and pre-processing for algorithm comparison
19     % Sampling Rate (FIX)
20     Fs = 16e6;
21     % Convert sampling rate to MHz
22     Fs_MHz = Fs/1e6;
23     % Label sampling rate for output
24     sampleRate = [num2str(Fs_MHz), ' MHz'];
25     % Compute L
26     L = Fs_MHz;
27     % Speed of light
28     c = 3e8;
29     % Distance conversion
30     dist = (c/Fs).';

```

```

31 % Determine which data to read from and label data set for output
32 dateDat = '002_2020-02-26_14-51'; dataSet = 'WalkieTalkie_Inside';
33 %   dateDat = '002_2020-02-26_14-52'; dataSet = 'WalkieTalkie_Hallway';
34 %   dateDat = '002_2020-02-26_14-54'; dataSet = 'WalkieTalkie_Under';
35 % Initialize file to output results to
36 fileName = ['Method_Comparisons_Real_' date '_v1.csv'];
37 % Check if file already exists to prevent overwrite
38 while exist(fileName, 'file')
39     % Determine version
40     if fileName(end - 5) == 'v'
41         % If single digits
42         ver = str2double(fileName(end-4));
43     elseif fileName(end - 6) == 'v'
44         % If double digits
45         ver = str2double(fileName(end-5:end-4));
46     elseif fileName(end - 7) == 'v'
47         % If triple digits
48         str2double(fileName(end-6:end-4));
49     else
50         % If more
51         msg11 = 'HELP! More than triple digits method comparison\n';
52         msg12 = 'files, add more to if statement';
53         msg = [msg11 msg12];
54         error(msg);
55     end
56     % Update version
57     new_ver = ver + 1;
58     % Determine new filename
59     fileName = ['Method_Comparisons_Real_' date '_v' ...
60                                     num2str(new_ver) '.csv'];
61 end
62 % Set first row of data table array
63 header = {'Data_Set', '&', 'Channels', '&', 'Algorithm', '&', ...
64           'Sample_Rate', '&', 'Estimate_m', '\\\'};
65 % Initialize number of columns for output table;

```

```

66 dataTabCol = 5;
67 % Set algorithms (1:19 selects which algorithm to run, NOTE multiple
68 % algorithms can be selected)
69 alg_list = {'CC', 'SVD', ... % [1 2]
70            'AED', 'ModAED', 'ModAEDS', ... %[3 4 5]
71            'AXIS', 'VAXIS', 'DFPAXIS', 'NSAXIS',... %[6 7 8 9]
72            'NBF', 'ABF', 'VBF', 'DFPBF', 'NSBF',... %[10 11 12 13 14]
73            'SC', 'VSC', 'DFPSC', 'NSSC'}; %[15 16 17 18]
74 % Select the adaptive algorithms and the CC algorithm for processing
75 % real data
76 algs = alg_list([1, 3, 4, 5, 6, 7, 8, 9 15, 16, 17, 18]);
77 % Determine number of algorithms
78 numAlgs = length(algs);
79 % Create dataTable
80 dataTable = cell(numAlgs, 2*dataTabCol);
81 % Read in data
82 data = readDataFile(dirs{2}, dateDat);
83 % Modify to processable size
84 % N = 2500;
85 N = length(data);
86 data = data(:, 1:N);
87 % Determine which channels to run through processing (1 2 3 4)
88 channels = [1 3];
89 % Process all data through algorithms
90 for a = 1:numAlgs
91     % Process algorithms
92     delayEst = algSelect(algs{a}, data, L, channels);
93     % Compute distance estimate
94     distEst = delayEst*dist;
95     % Add to table
96     dataTable(a, :) = {dataSet, '&', [num2str(channels(1)) '\&' ...
97                               num2str(channels(2))], '&', algs{a}, '&', ...
98                               sampleRate, '&', distEst, '\\'};
99 end
100 % Output to file

```

```
101 T = cell2table(vertcat(header, dataTable));  
102 writetable(T, fileName, 'Delimiter', '');
```

Listing B.16: Select Algorithm

```

1 %% Algorithm Selection Function
2 % This function determines which algorithm is needed to process the data
3 %   Input:
4 %       - alg: denotes the algorithm to use
5 %       - data: the data to process the algorithm with
6 %       - L: the model order
7 %       - channels: which two of the channels are desired for processing
8 %   Output:
9 %       - delayEst: time Distance Estimate
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 function delayEst = algSelect(alg, data, L, channels)
12     % Set amount for interpolation
13     interpolationAmt = 1;
14     % Number of channels
15     M = 2;
16     % determine data size
17     N = length(data);
18     % determine peak index for unitary vector
19     peak_index = floor((L + 1)/2);
20     % Start unitary vector initial estimate
21     hest = zeros(M*(L + 1), 1);
22     hest(peak_index) = 1;
23     % Set nonzero offset for denominator
24     epsilon = 0.1e-9;
25     % Set step size for adaptive algorithms
26     eta = 0.5e-3;
27     % Set weighting variables for objective constraints
28     alpha = 0.1e-4;
29     gamma = 0.1e-4;
30     phi = 0.1e-4;
31     % Set default blocksize for faster algorithm processing
32     maxBlockReset = 1500;
33     % Set number of iterations for algorithm
34     maxIter = 1;

```

```

35 % Separate into two channels and normalize
36 channel1 = data(channels(1), :).'; channel1 = channel1/norm(channel1);
37 channel2 = data(channels(2), :).'; channel2 = channel2/norm(channel2);
38 % Create the data matrices
39 cols = channel1((L + 1):end);
40 rows = flipud(channel1(1:(L + 1)));
41 X1 = toeplitz(cols, rows);
42 cols = channel2((L + 1):end);
43 rows = flipud(channel2(1:(L + 1)));
44 X2 = toeplitz(cols, rows);
45 % Create toeplitz data matrix
46 X = [X1 -X2];
47 % Determine which algorithm to use
48 switch alg
49     % If Cross-Correlation algorithm
50     case 'CC'
51         EstChannel = CC(data(channels, :));
52     % If Least Squares according to Xu et. al (SVD) algorithm
53     case 'SVD'
54         EstChannel = LS(X, L, M);
55     % If Adaptive Eigenvalue Decomposition algorithm
56     case 'AED'
57         EstChannel = AED(hest, X, eta, epsilon, L, ...
58                         N, M, maxBlockReset, ...
59                         maxIter);
60     % If Modified Adaptive Eigenvalue Decomposition algorithm
61     case 'ModAED'
62         EstChannel = modAED(hest, X, eta, epsilon, ...
63                             L, N, M, ...
64                             maxBlockReset, maxIter);
65     % If Modified Sparse Adaptive Eigenvalue Decomposition
66     % algorithm
67     case 'ModAEDS'
68         EstChannel = AEDS(hest, X, eta, alpha, ...
69                             epsilon, L, N, M, ...

```

```

70         maxBlockReset, maxIter);
71     % If Non Sparse Adaptive Cross-Channel Identification with
72     % Shift-Suppression algorithm
73     case 'NSAXIS'
74         EstChannel = NS_AXIS(hest, X, eta, alpha, ...
75                             epsilon, L, N, M, ...
76                             maxBlockReset, maxIter, ...
77                             peak_index);
78     % If Adaptive Cross-Channel Identification with Sparse
79     % Shift-Suppression algorithm
80     case 'AXIS'
81         EstChannel = AXIS(hest, X, eta, alpha, ...
82                           epsilon, L, N, M, ...
83                           maxBlockReset, maxIter, ...
84                           peak_index);
85     % If Varied Adaptive Cross-Channel Identification with Sparse
86     % Shift-Suppression
87     case 'VAXIS'
88         EstChannel = vAXIS(hest, X, eta, alpha, ...
89                            epsilon, L, N, M, ...
90                            maxBlockReset, maxIter, ...
91                            peak_index);
92     % If Dual Fixed-Peak Adaptive Cross-Channel Identification with
93     % Spares Shift-Suppression algorithm
94     case 'DFPAXIS'
95         EstChannel = DFPAXIS(hest, X, eta, alpha, ...
96                              epsilon, L, N, M, ...
97                              maxBlockReset, maxIter, ...
98                              peak_index);
99     % If Non Sparse "Brute Force" algorithm
100    case 'NSBF'
101        EstChannel = BF_NS(hest, X, eta, alpha, ...
102                            epsilon, L, N, M, ...
103                            N - L, maxIter, ...
104                            peak_index);

```

```

105 % If NonAdaptive "Brute Force" algorithm
106 case 'NBF'
107     EstChannel = BF(hest, X, eta, alpha, ...
108                   epsilon, L, N, M, ...
109                   N - L, maxIter, ...
110                   peak_index);
111 % If Adaptive "Brute Force" algorithm
112 case 'ABF'
113     EstChannel = BF(hest, X, eta, alpha, ...
114                   epsilon, L, N, M, ...
115                   maxBlockReset, maxIter, ...
116                   peak_index);
117 % If Varied "Brute Force" algorithm
118 case 'VBF'
119     EstChannel = vBF(hest, X, eta, alpha, ...
120                    epsilon, L, N, M, ...
121                    maxBlockReset, maxIter, ...
122                    peak_index);
123 % If Dual Fixed-Peak "Brute Force" algorithm
124 case 'DFPBF'
125     EstChannel = DFPBF(hest, X, eta, alpha, ...
126                      epsilon, L, N, M, ...
127                      maxBlockReset, maxIter, ...
128                      peak_index);
129 % If Non-Sparse Single Constraint algorithm
130 case 'NSSC'
131     EstChannel = NS_SC(hest, X, eta, alpha, gamma, ...
132                      epsilon, L, N, M, ...
133                      maxBlockReset, maxIter, ...
134                      peak_index);
135 % If Single Constraint algorithm
136 case 'SC'
137     EstChannel = SC(hest, X, eta, alpha, gamma, ...
138                   epsilon, L, N, M, ...
139                   maxBlockReset, maxIter, ...

```



```

140                                     peak_index);
141 % If Varied Single Constraint algorithm
142 case 'VSC'
143     EstChannel = vSC(hest, X, eta, alpha, gamma,...
144                     epsilon, L, N, M, ...
145                     maxBlockReset, maxIter, ...
146                     peak_index);
147 % If Dual Fixed-Peak Single Constraint algorithm
148 case 'DFPSC'
149     EstChannel = DFPSC(hest, X, eta, alpha, ...
150                       gamma, phi, epsilon,L, ...
151                       N, M, maxBlockReset, ...
152                       maxIter, peak_index);
153 end
154 % Compute delay
155 if strcmp(alg, 'CC')
156     % Normalize
157     EstChannel = EstChannel/max(EstChannel);
158     % Estimate Delay
159     [~, pe] = max(interp(abs(EstChannel), ...
160                         interpolationAmt));
161     indexes = ...
162             interp(-(length(data) - 1):(length(data) - 1),...
163                   interpolationAmt);
164     delayEst = abs(indexes(pe));
165 else
166     % Determine peak location
167     [~, ind1] = max(abs(EstChannel(:, 1)));
168     [~, ind2] = max(abs(EstChannel(:, 2)));
169     % Compute difference
170     delayEst = abs(ind1 - ind2);
171 end
172 end

```

## Listing B.17: Read in Data

```

1 %% Read In GNU Radio Data
2 % GNU Radio data is collected in subfolder dat files, this code cycles
3 % through the subfolders and reads in the dat files
4 %   Input:
5 %       - directory: This is the location of the data files
6 %       - dateDat: This denotes which folders to cycle through via the date
7 %   Output:
8 %       - data: This holds all of the channels data
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 function data = readDataFile(directory, dateDat)
11     % Determine length of date
12     lenD = length(dateDat);
13     % Get files in directory
14     files = dir(directory);
15     directoryNames = {files([files.isdir]).name};
16     % Cycle through directory to add data subfolders
17     for fold = 1:length(directoryNames)
18         % Check if folder matches date
19         if length(directoryNames{fold}) > lenD
20             if strcmp(directoryNames{fold}(1:lenD), dateDat)
21                 % Save directory
22                 subDir = directoryNames{fold};
23                 % break out of loop
24                 break
25             end
26         elseif fold == length(directoryNames)
27             % Add Warning and Break
28             disp('Directory does not exist in specified folder, pausing.')
29             pause();
30         end
31     end
32     % Add subDir to path
33     subDirPath = [directory subDir];
34     addpath(subDirPath);

```

```
35 % Grab dat files in folder
36 dataFiles = dir([subDirPath '/*.dat']);
37 dataFilesName = strings(length(dataFiles), 1);
38 for datFile = 1:length(dataFiles)
39     dataFilesName(datFile, 1) = dataFiles(datFile, 1).name;
40 end
41 % data = zeros(length(dataFiles), 1);
42 % Load in data files
43 for d = 1:length(dataFilesName)
44     % Load in data files
45     fid = fopen(dataFilesName(d));
46     dat = fread(fid, 'int16');
47     fclose(fid);
48     % Format
49     dat = reshape(dat, [length(dat)/2 2]);
50     dat = dat(:, 1) + 1j*dat(:, 2);
51     % Put in data array
52     data(d,:) = dat;
53 end
54 end
```

APPENDIX C  
Oversized Figures

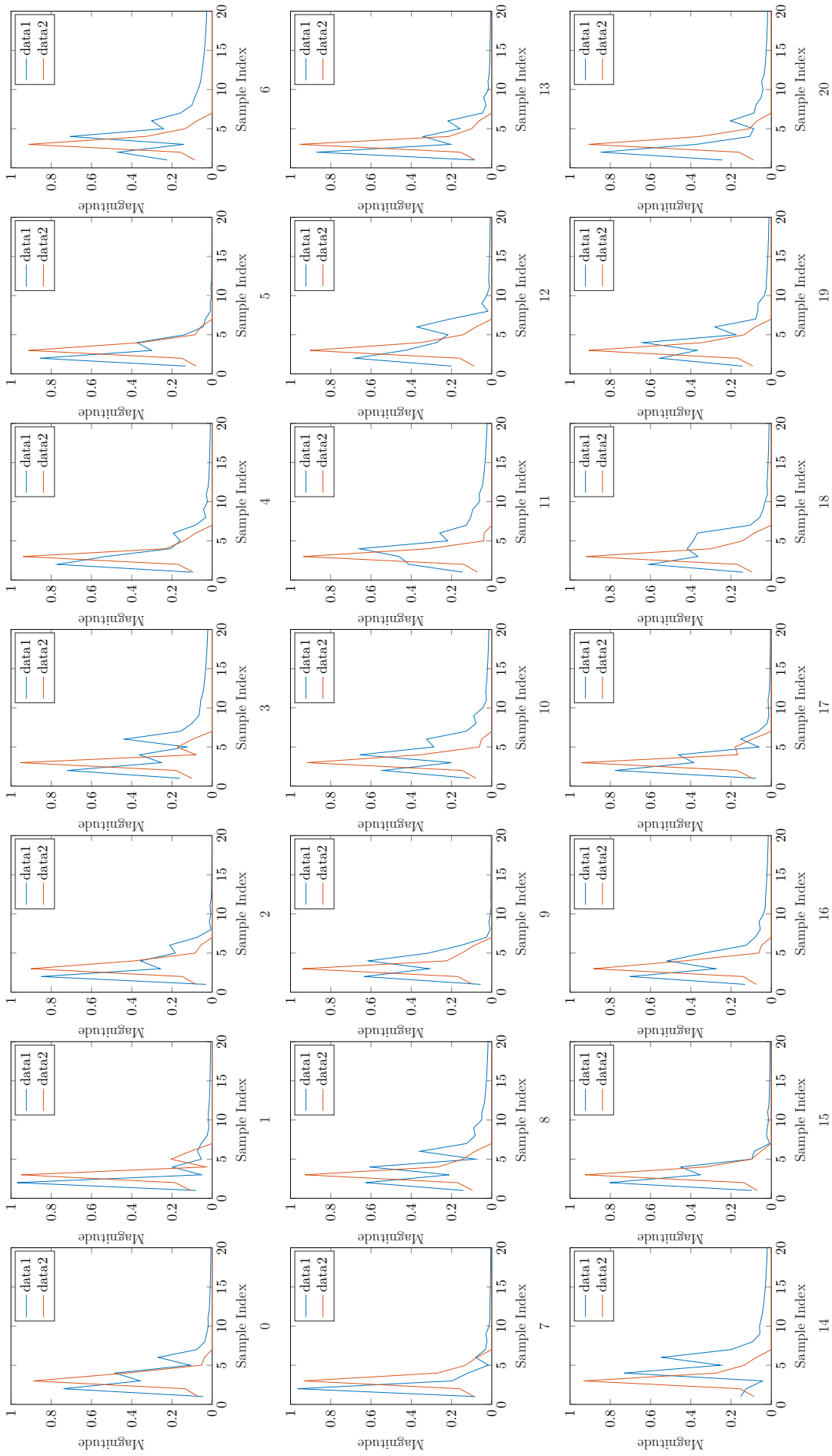


Fig. C.1: Comparison of the 20 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

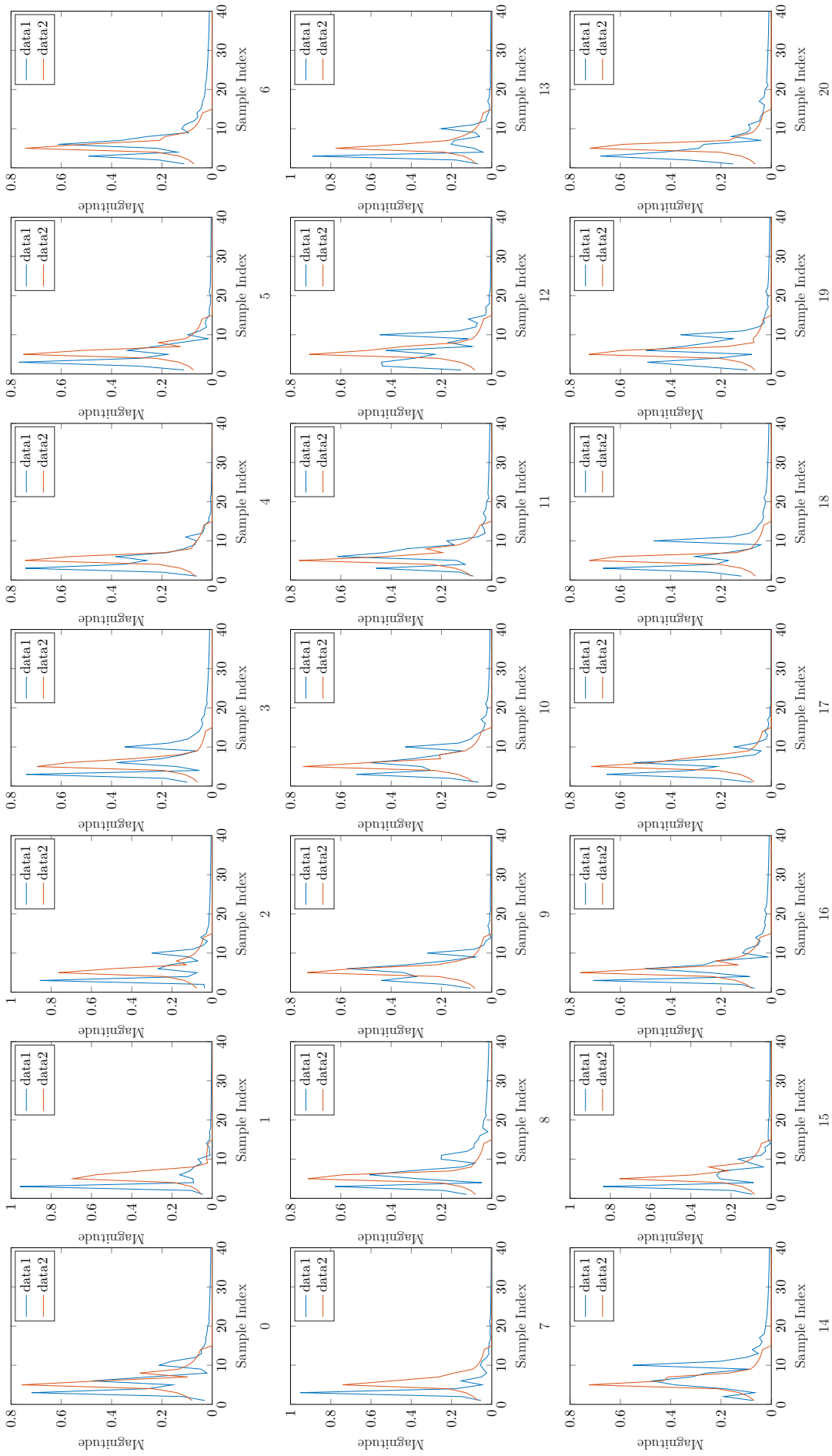


Fig. C.2: Comparison of the 40 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

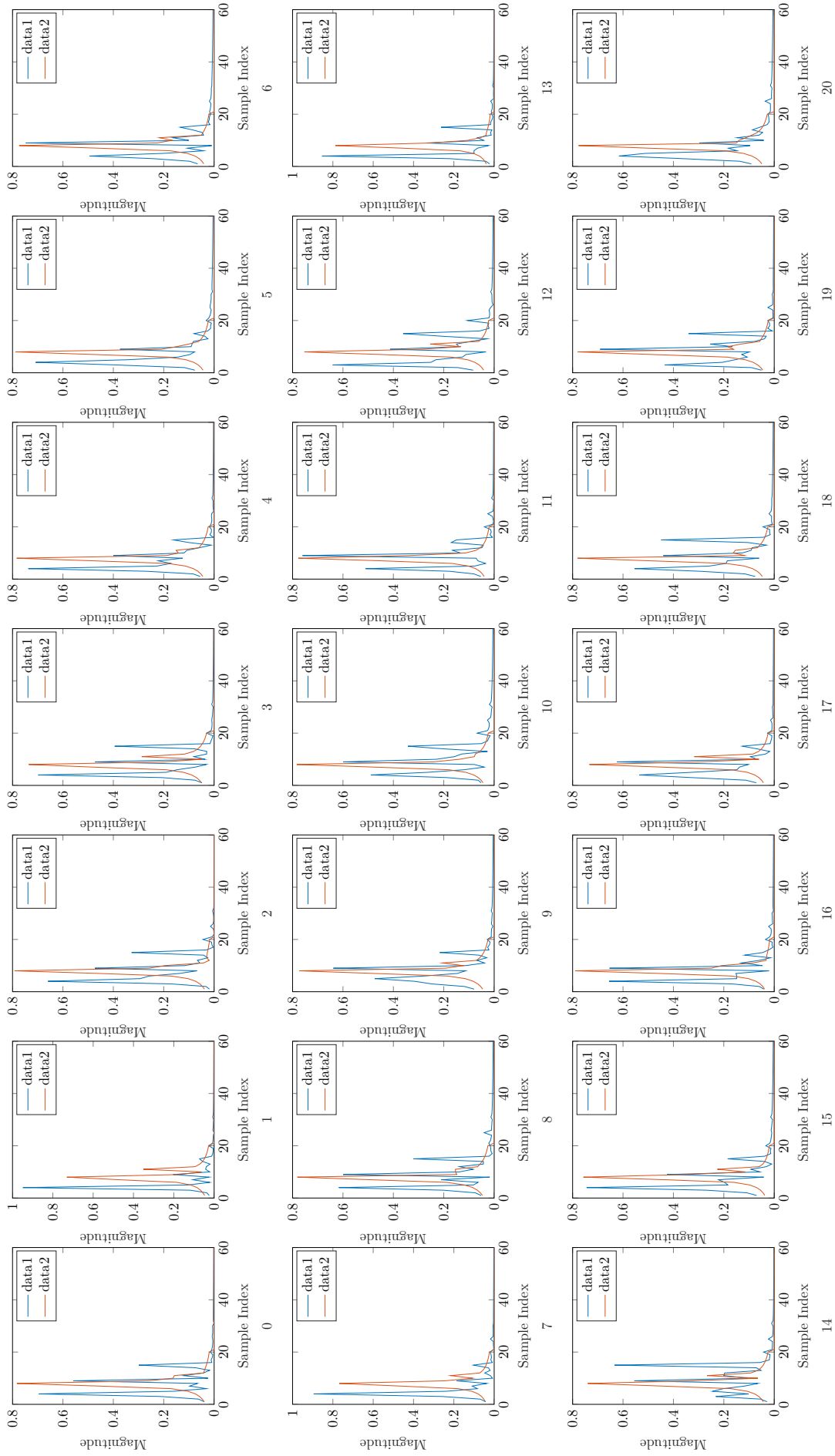


Fig. C.3: Comparison of the 60 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

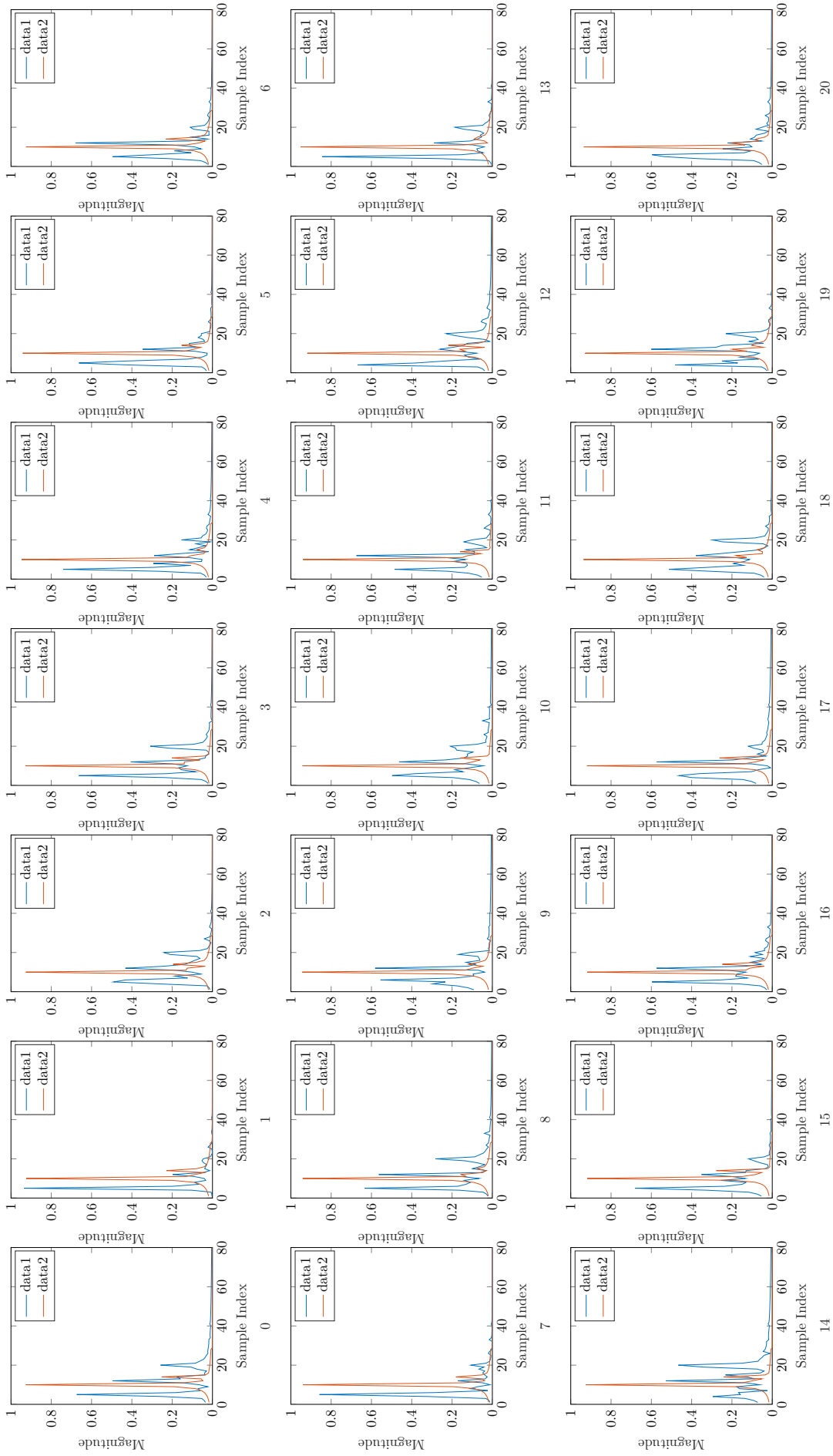


Fig. C.4: Comparison of the 80 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000



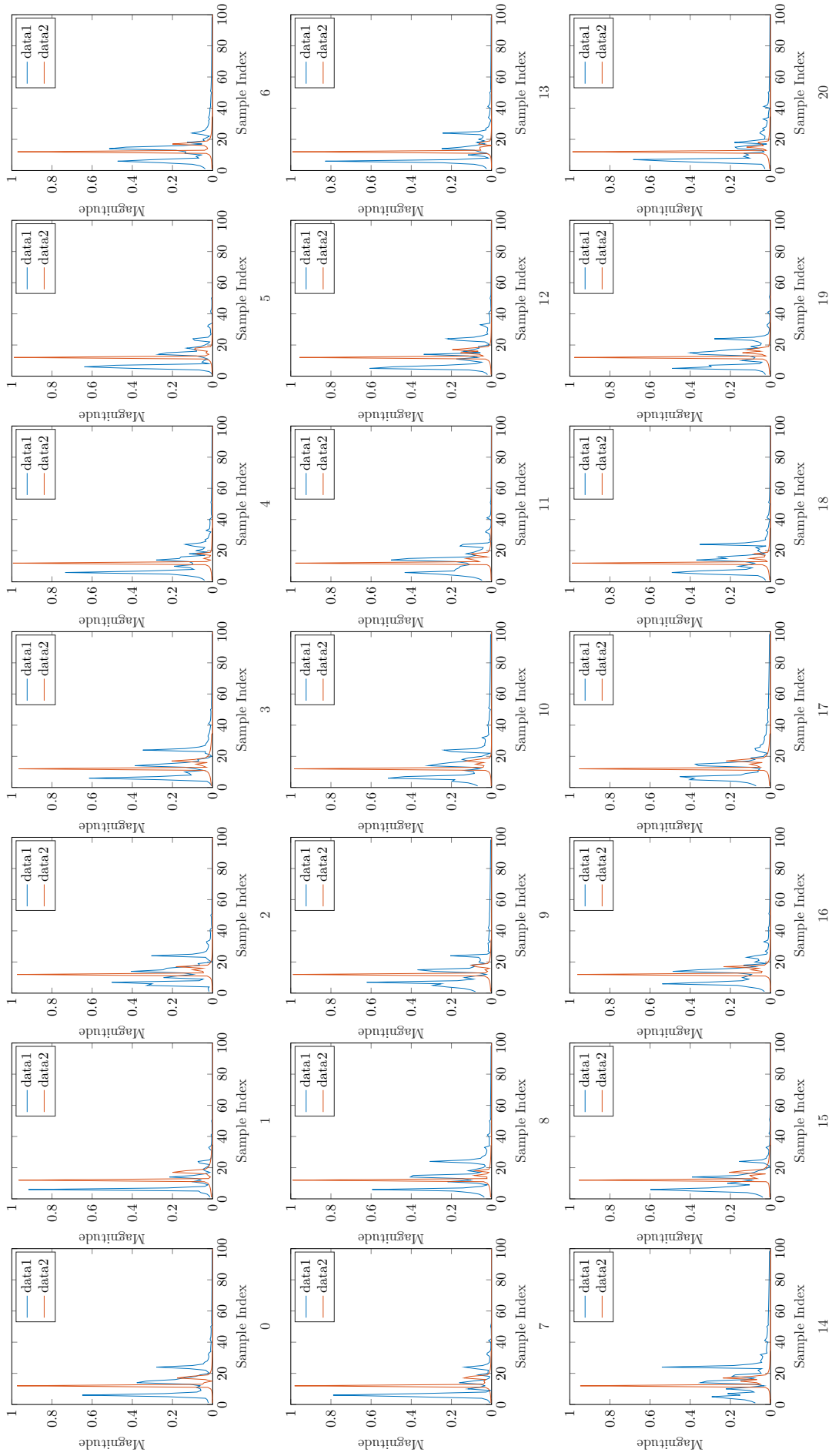


Fig. C.5: Comparison of the 100 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

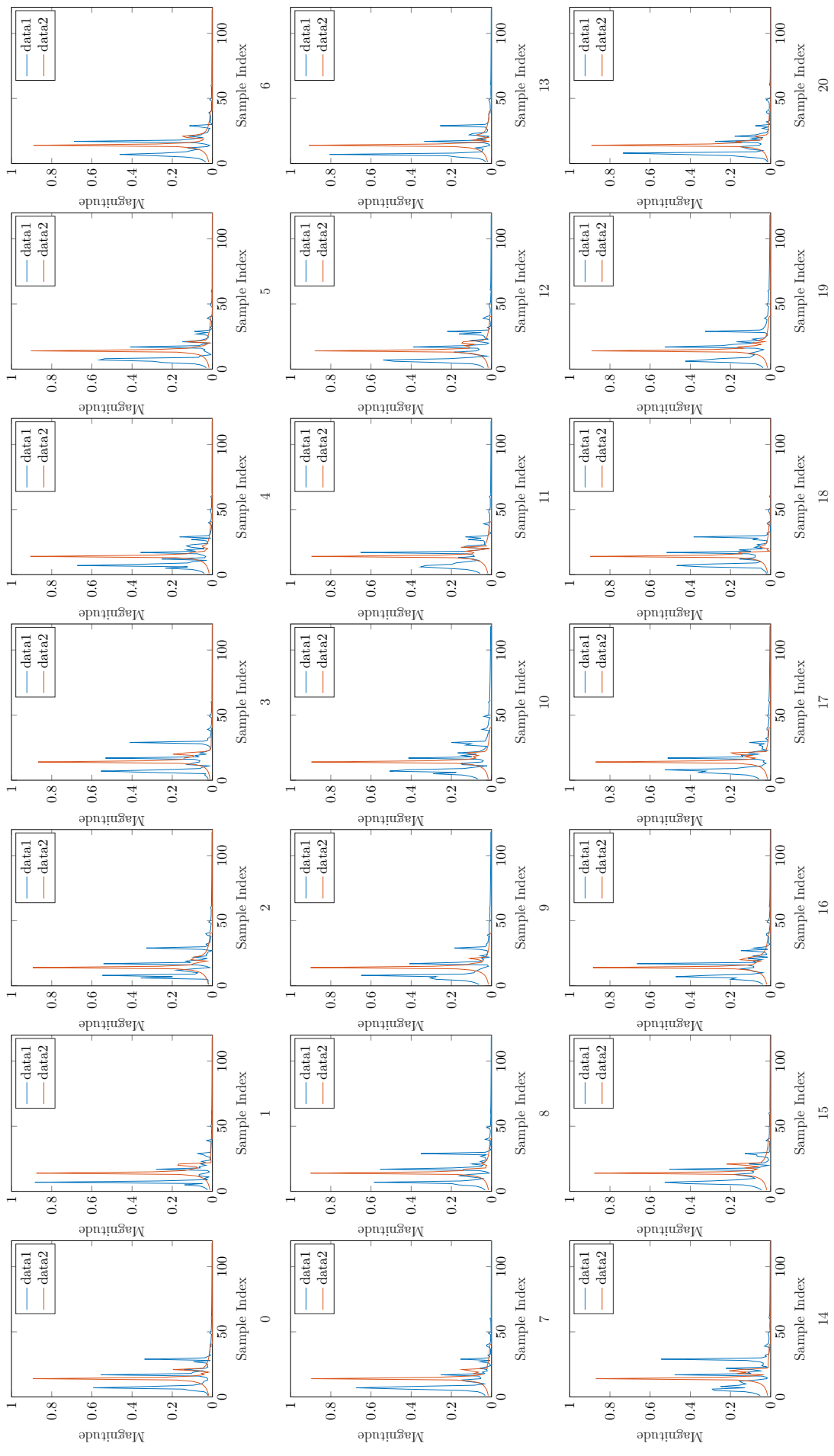


Fig. C.6: Comparison of the 120 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

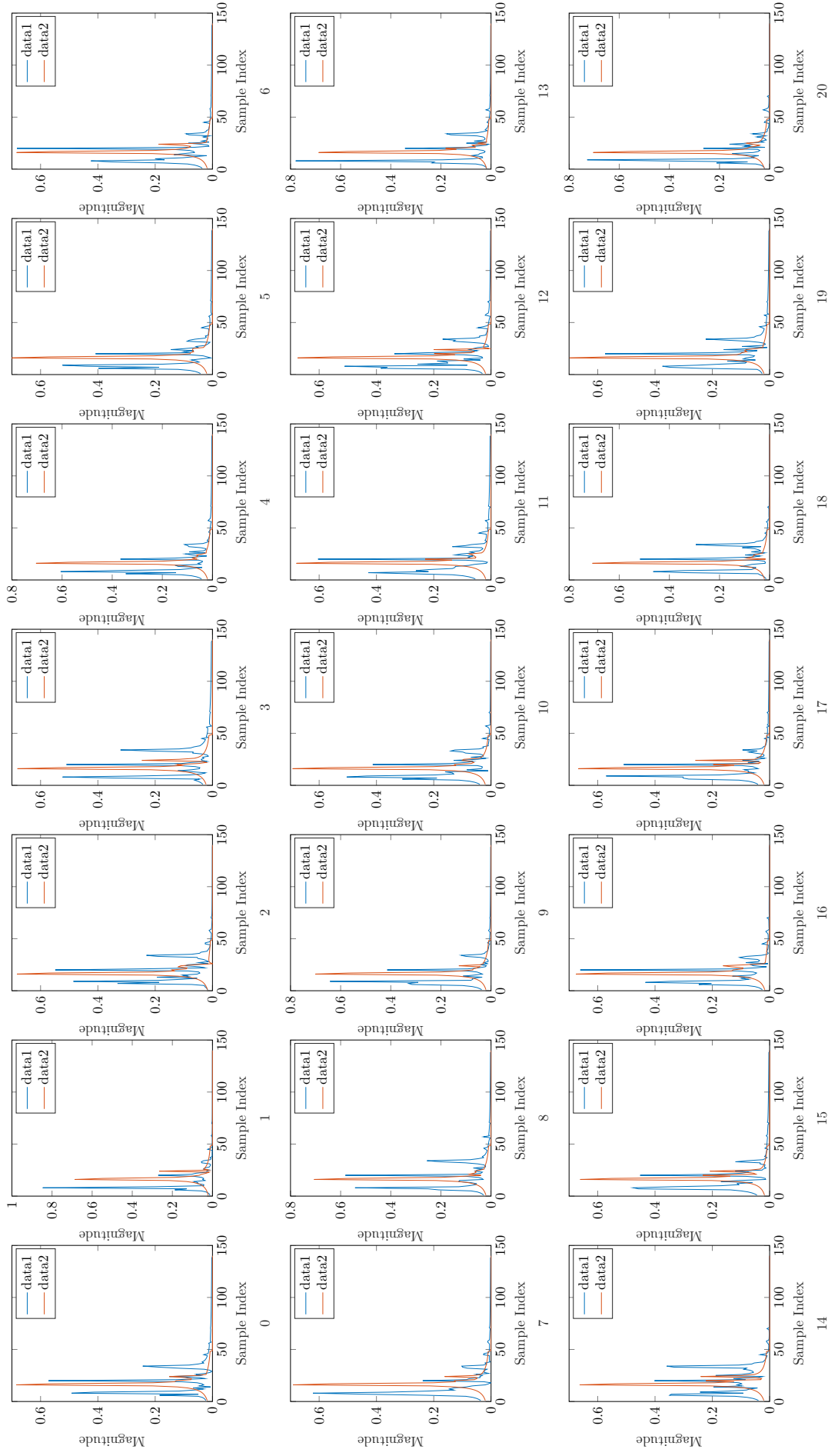


Fig. C.7: Comparison of the 140 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

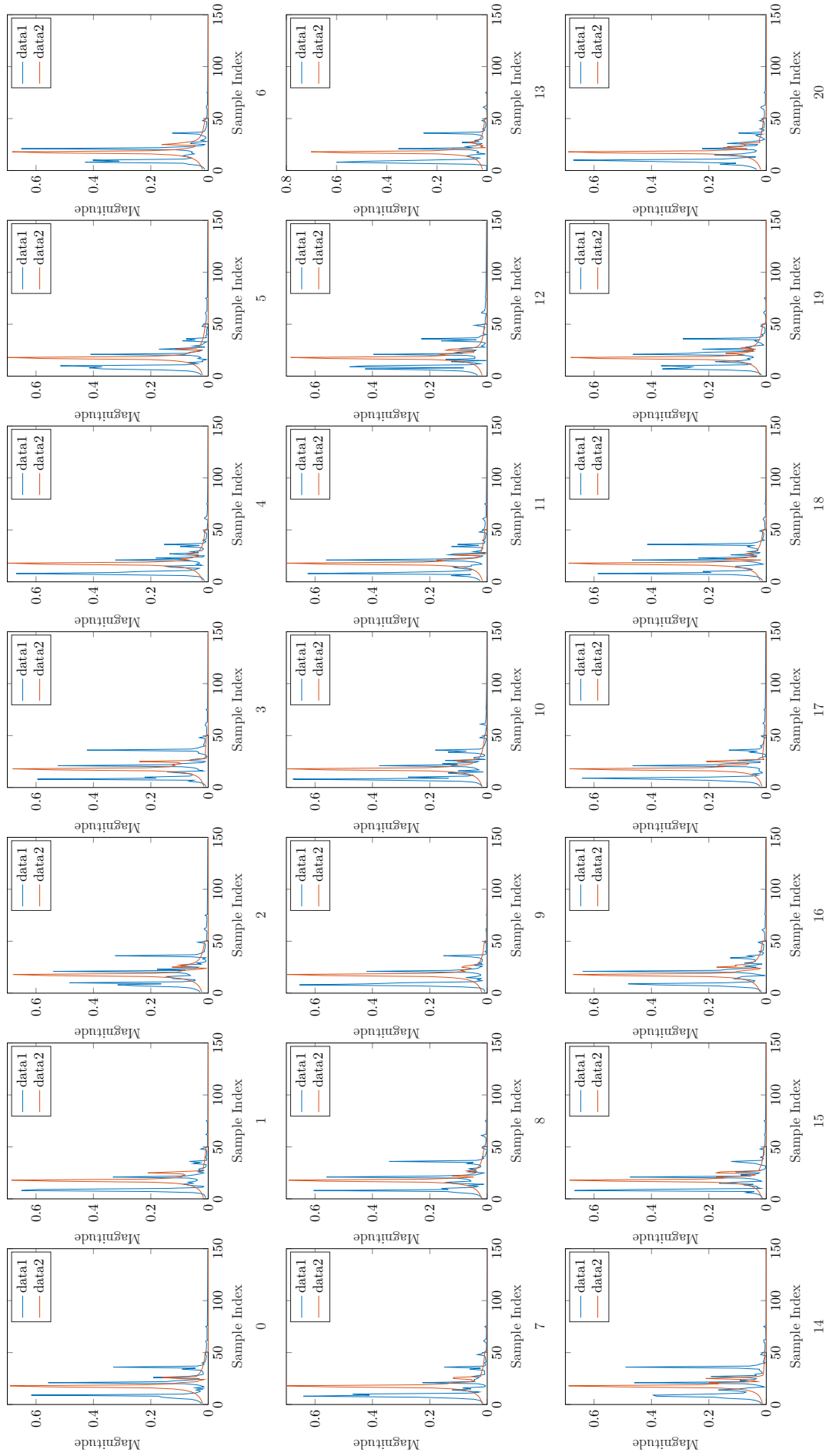


Fig. C.8: Comparison of the 150 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

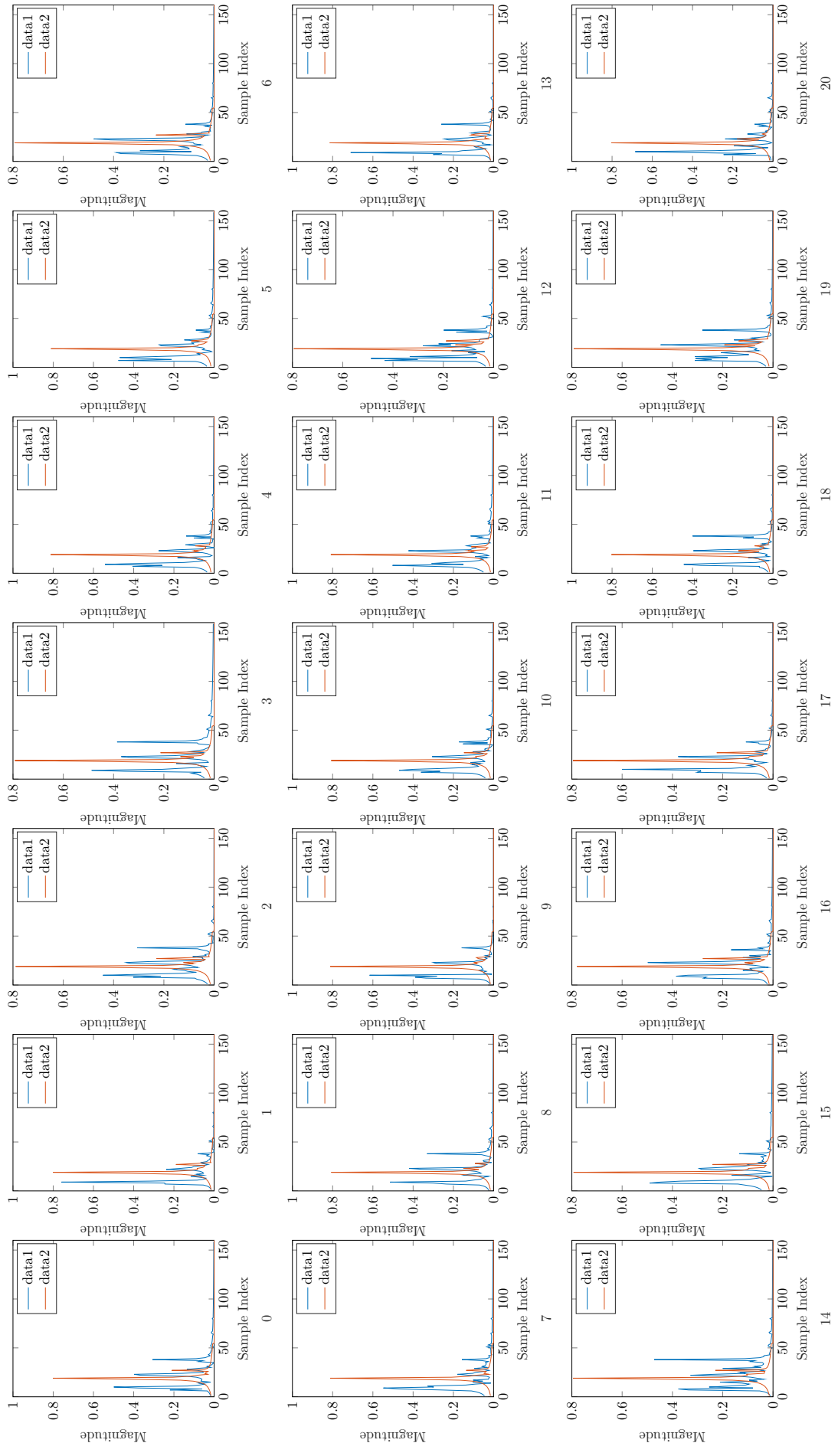


Fig. C.9: Comparison of the 160 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

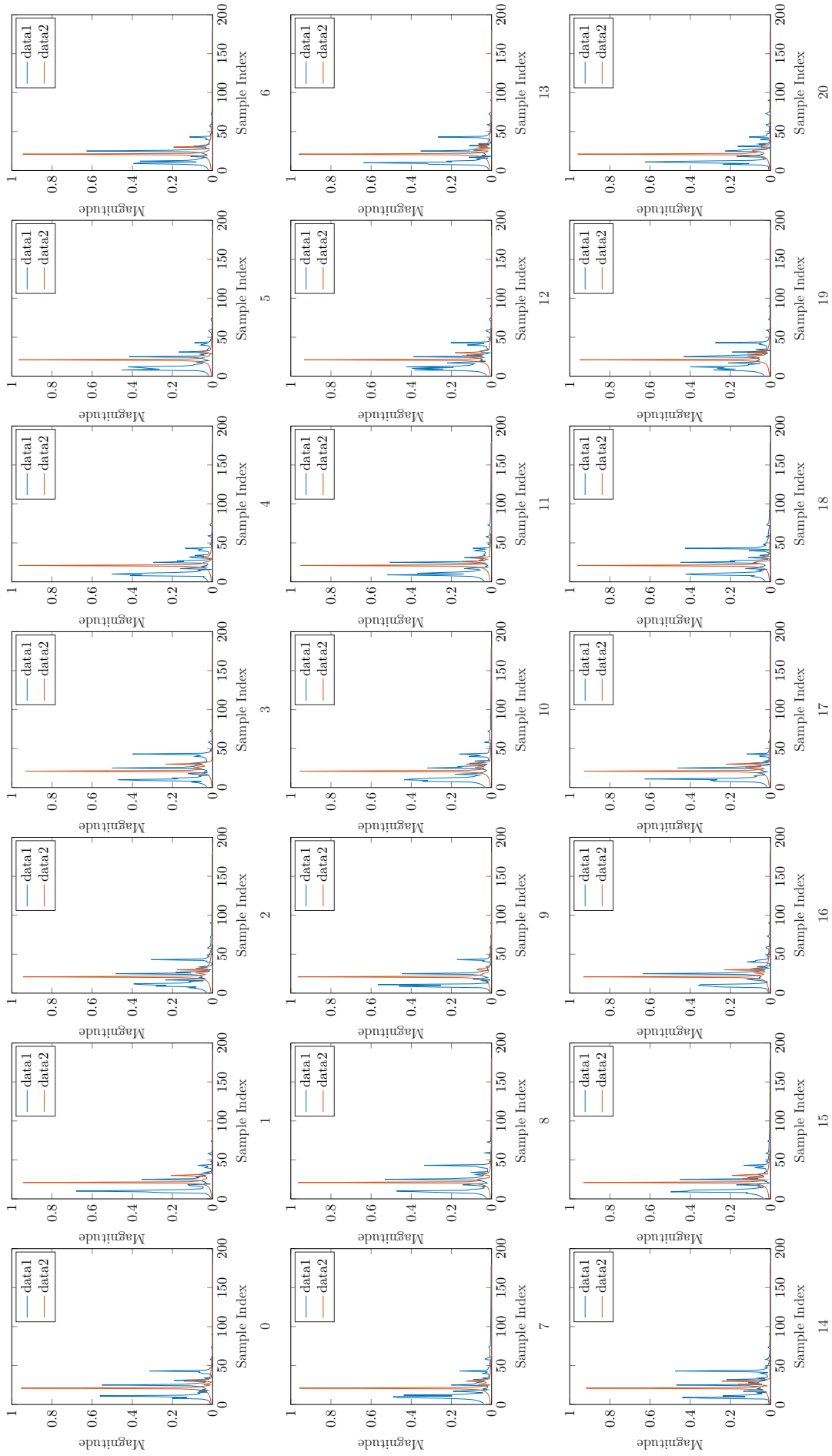


Fig. C.10: Comparison of the 180 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

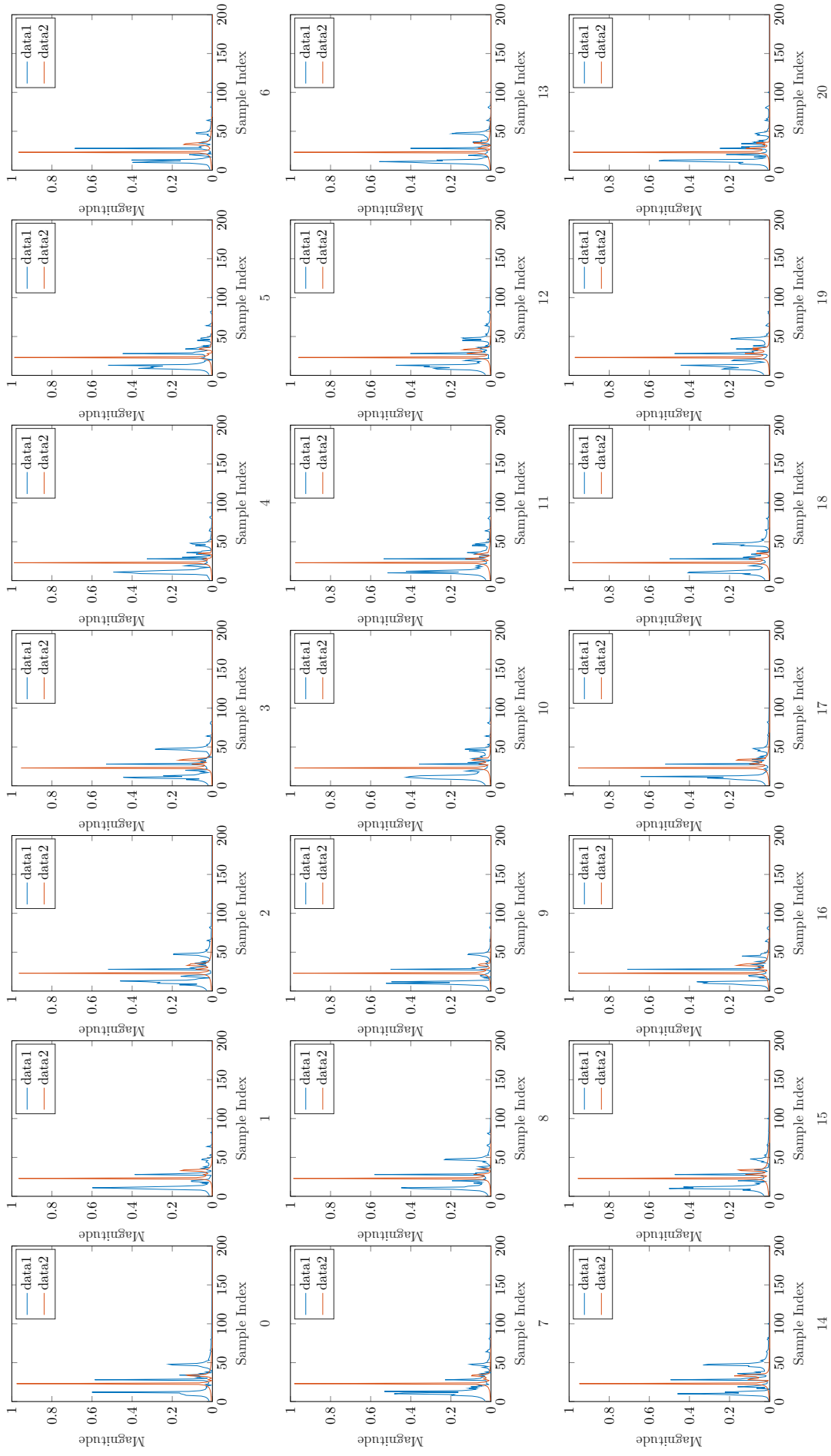


Fig. C.11: Comparison of the 200 MHz transfer functions for impulse response set IR-1 for each seed (0-20) for channels 30 vs 3000

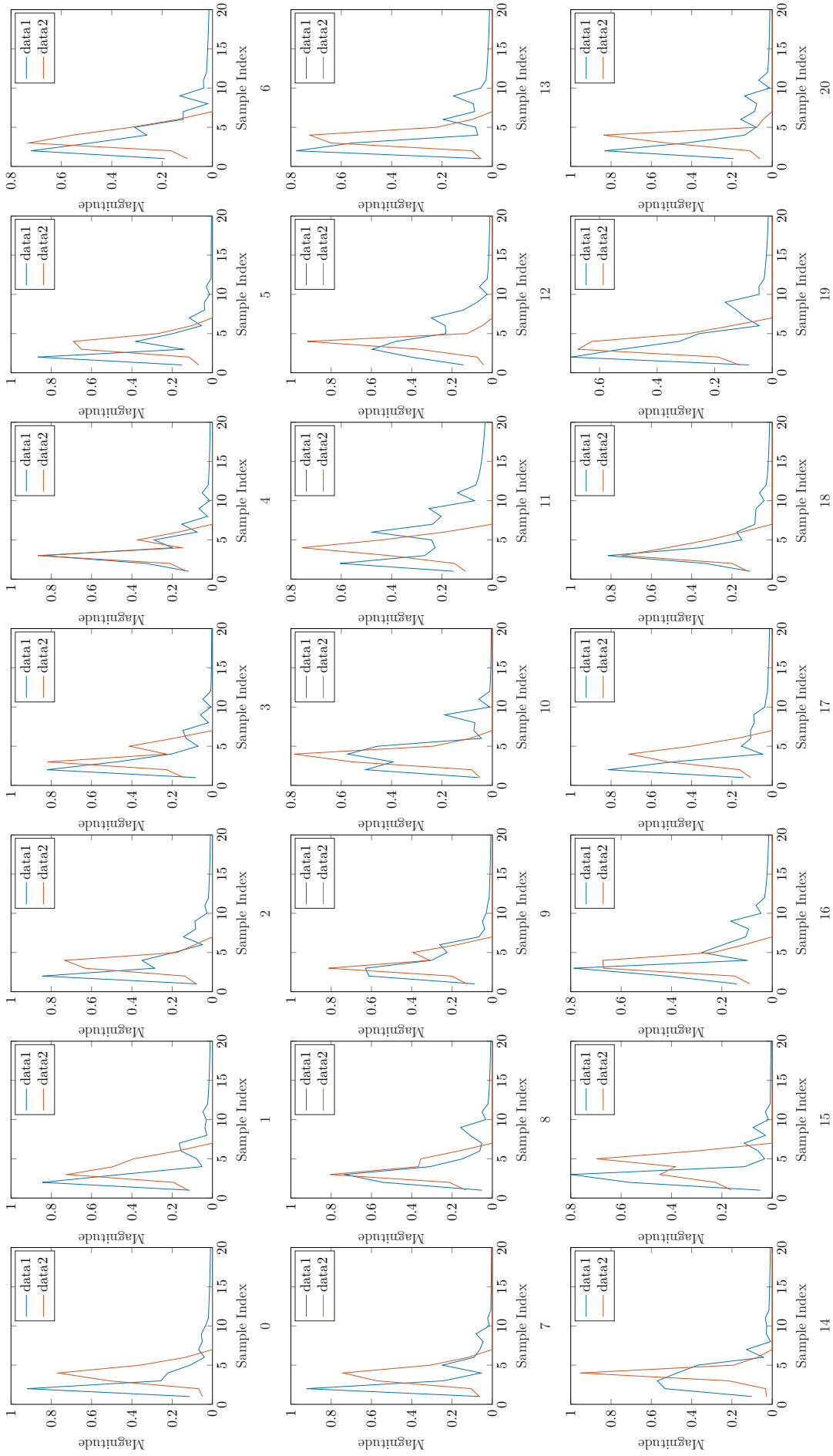


Fig. C.12: Comparison of the 20 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000



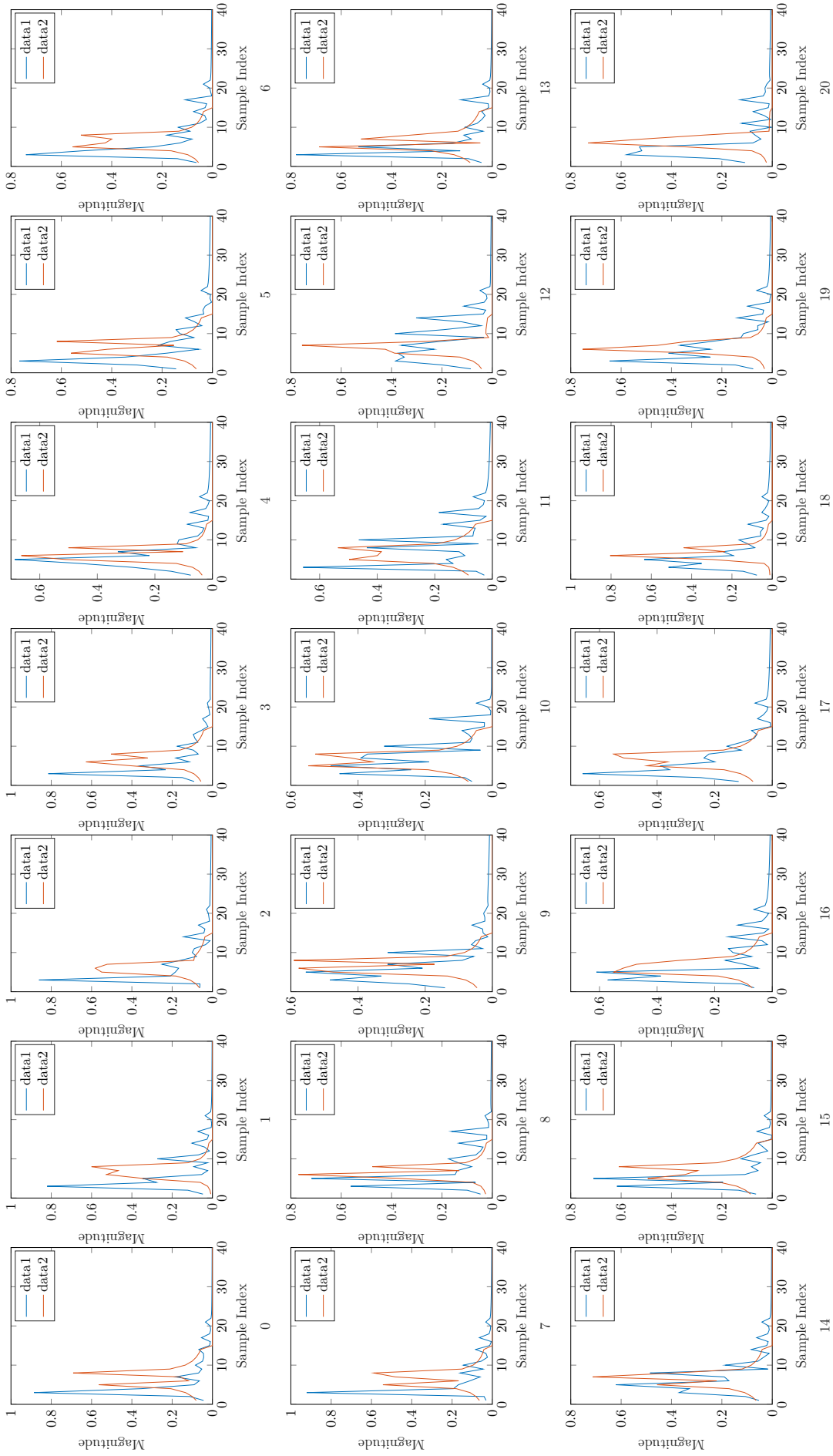


Fig. C.13: Comparison of the 40 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

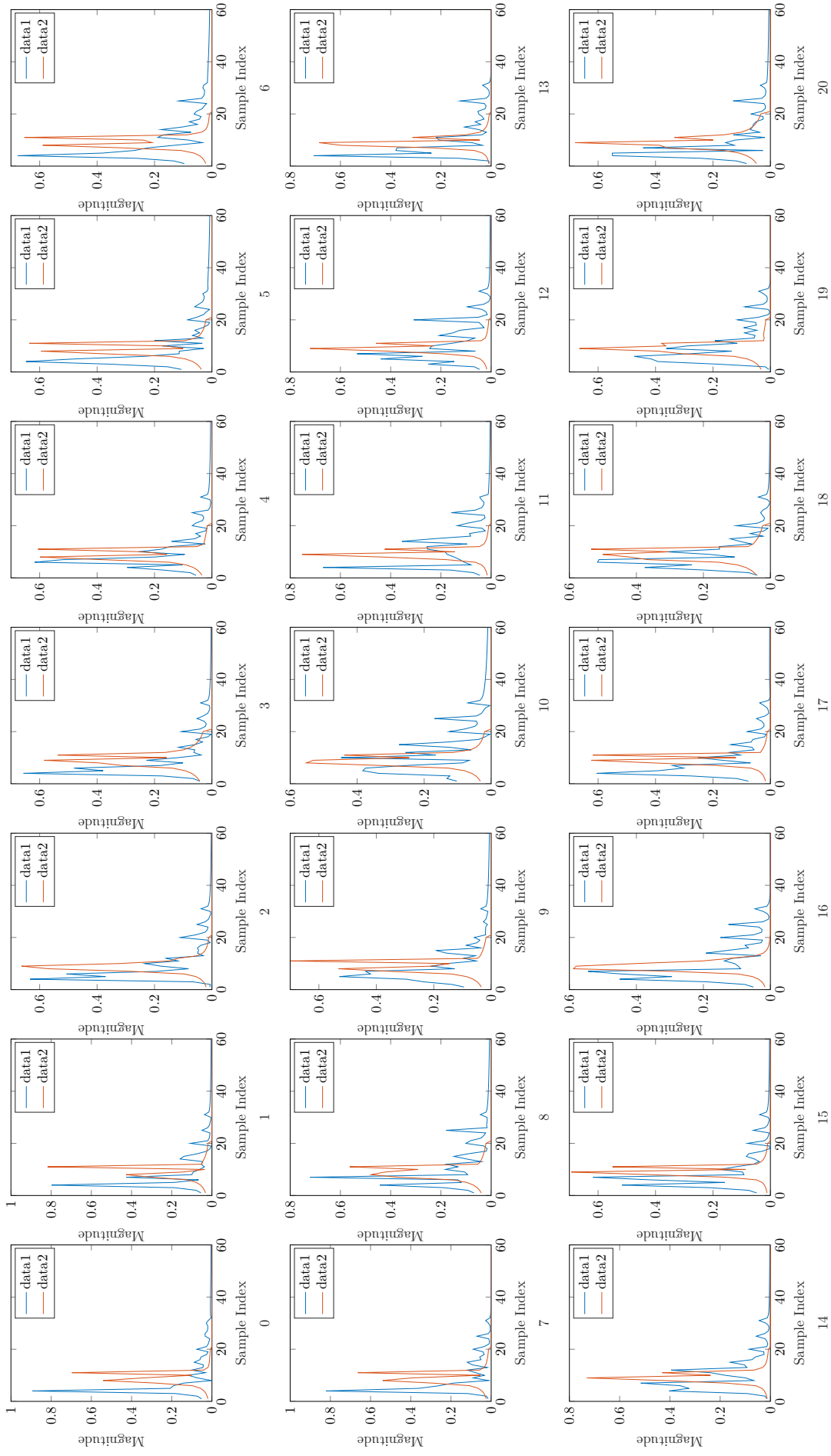


Fig. C.14: Comparison of the 60 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

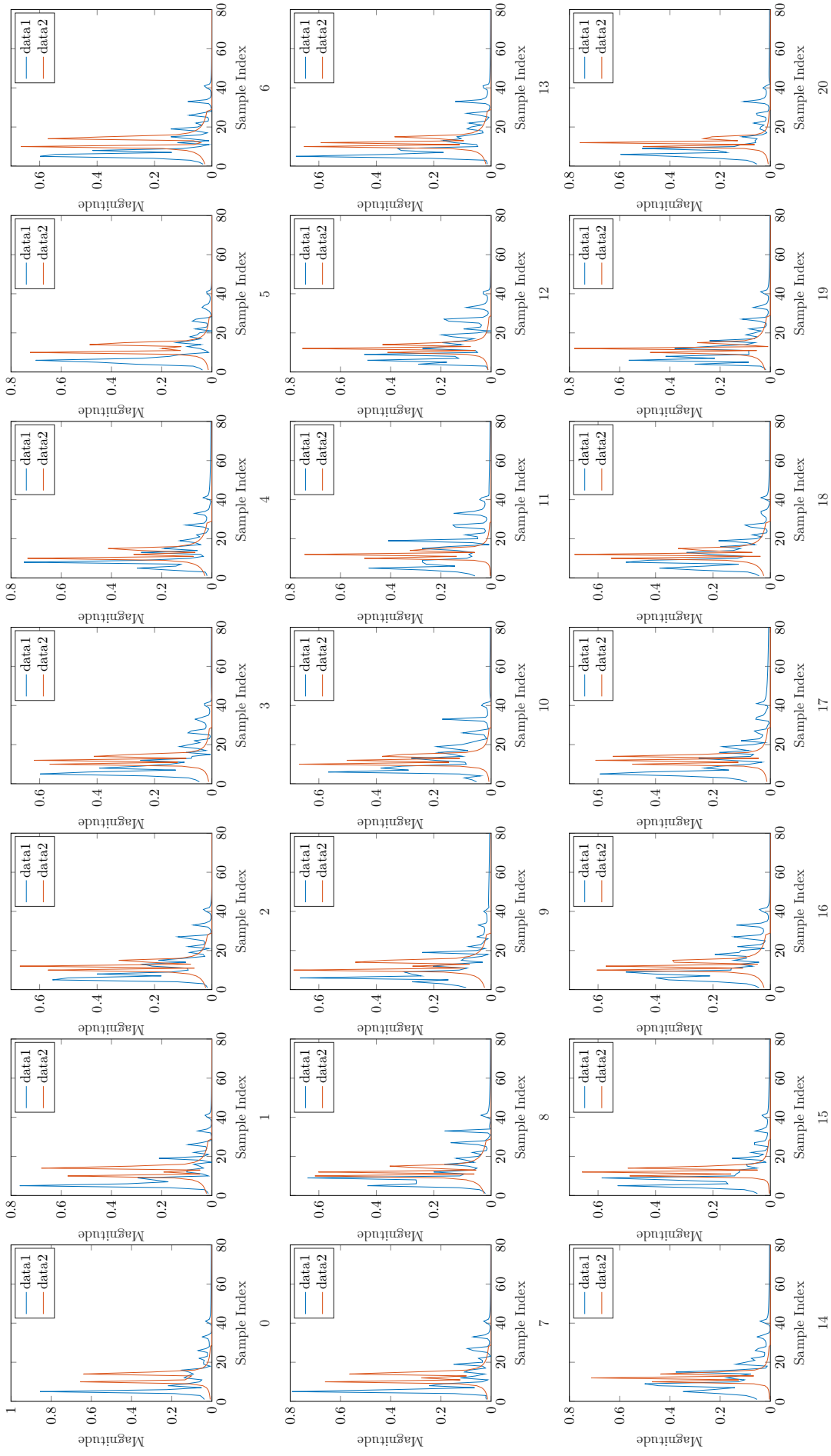


Fig. C.15: Comparison of the 80 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

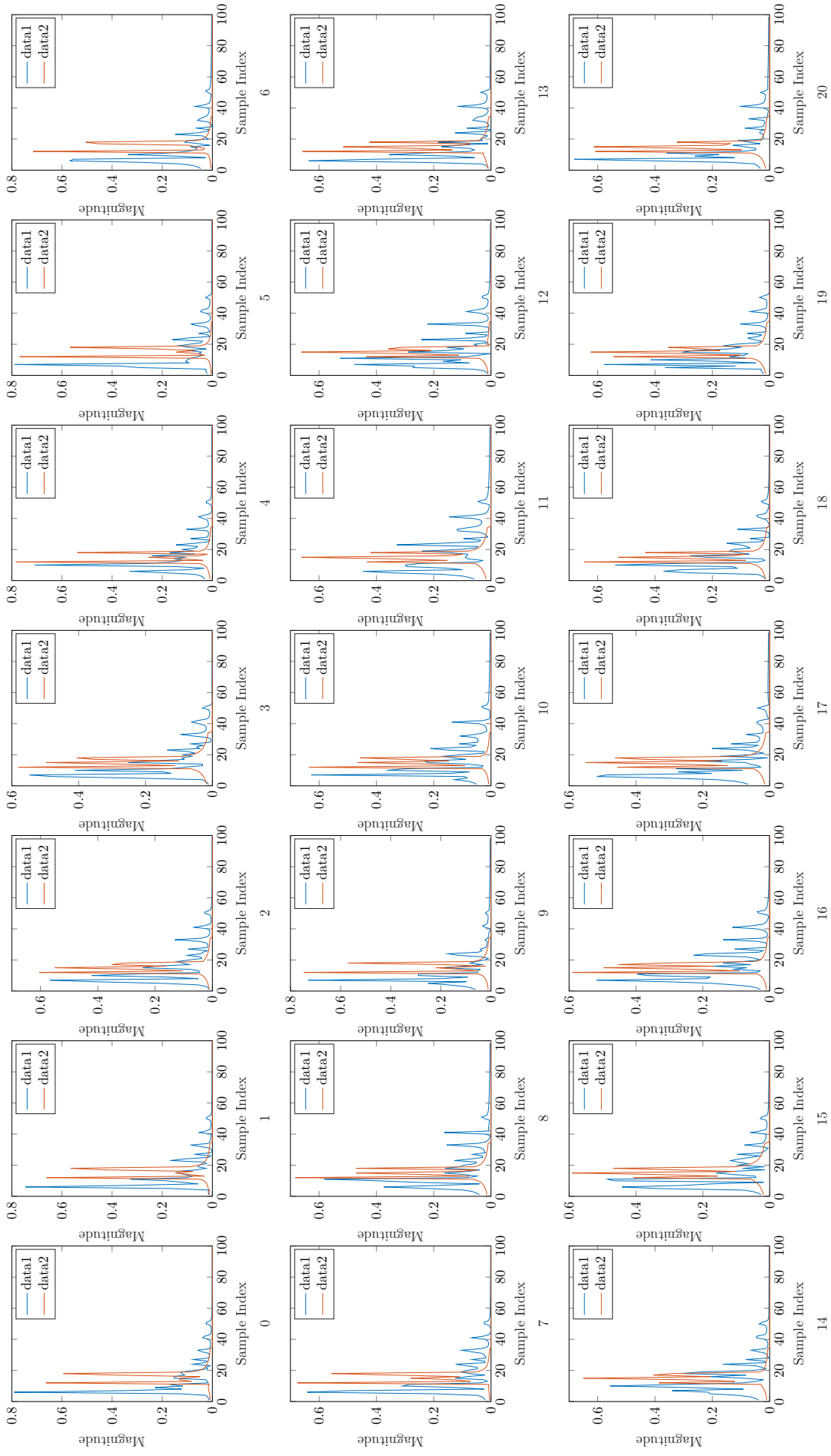


Fig. C.16: Comparison of the 100 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

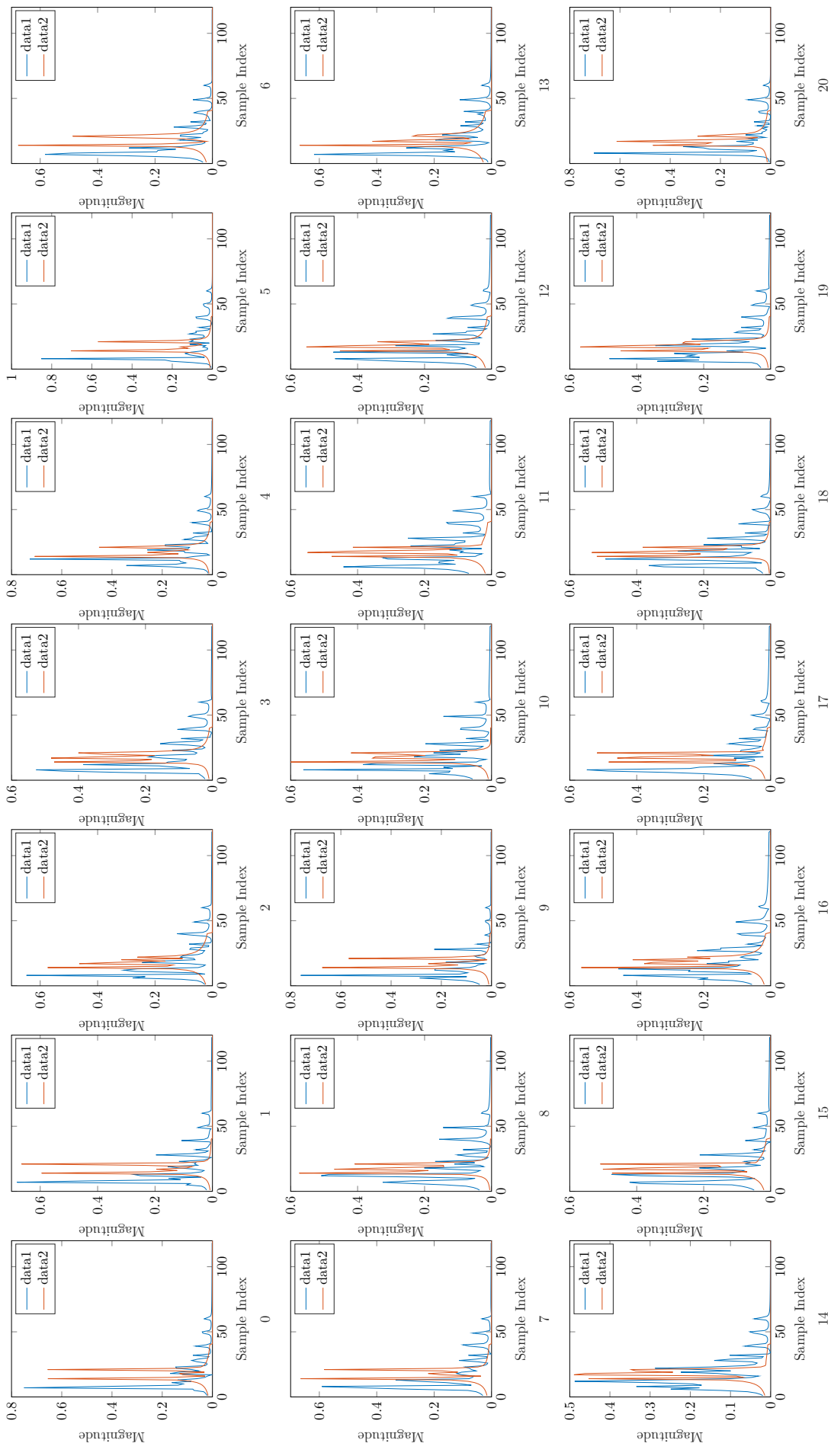


Fig. C.17: Comparison of the 120 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

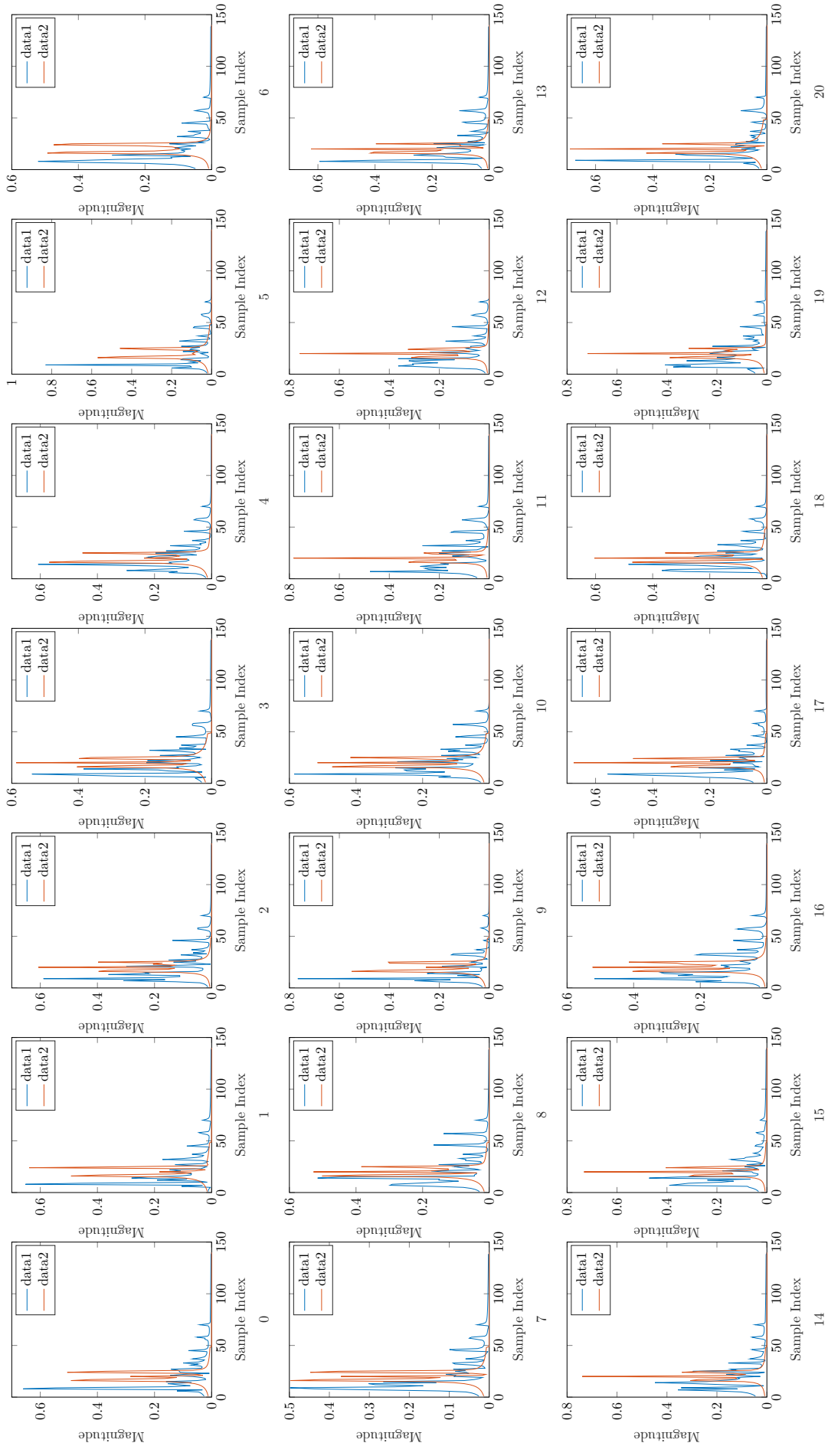


Fig. C.18: Comparison of the 140 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

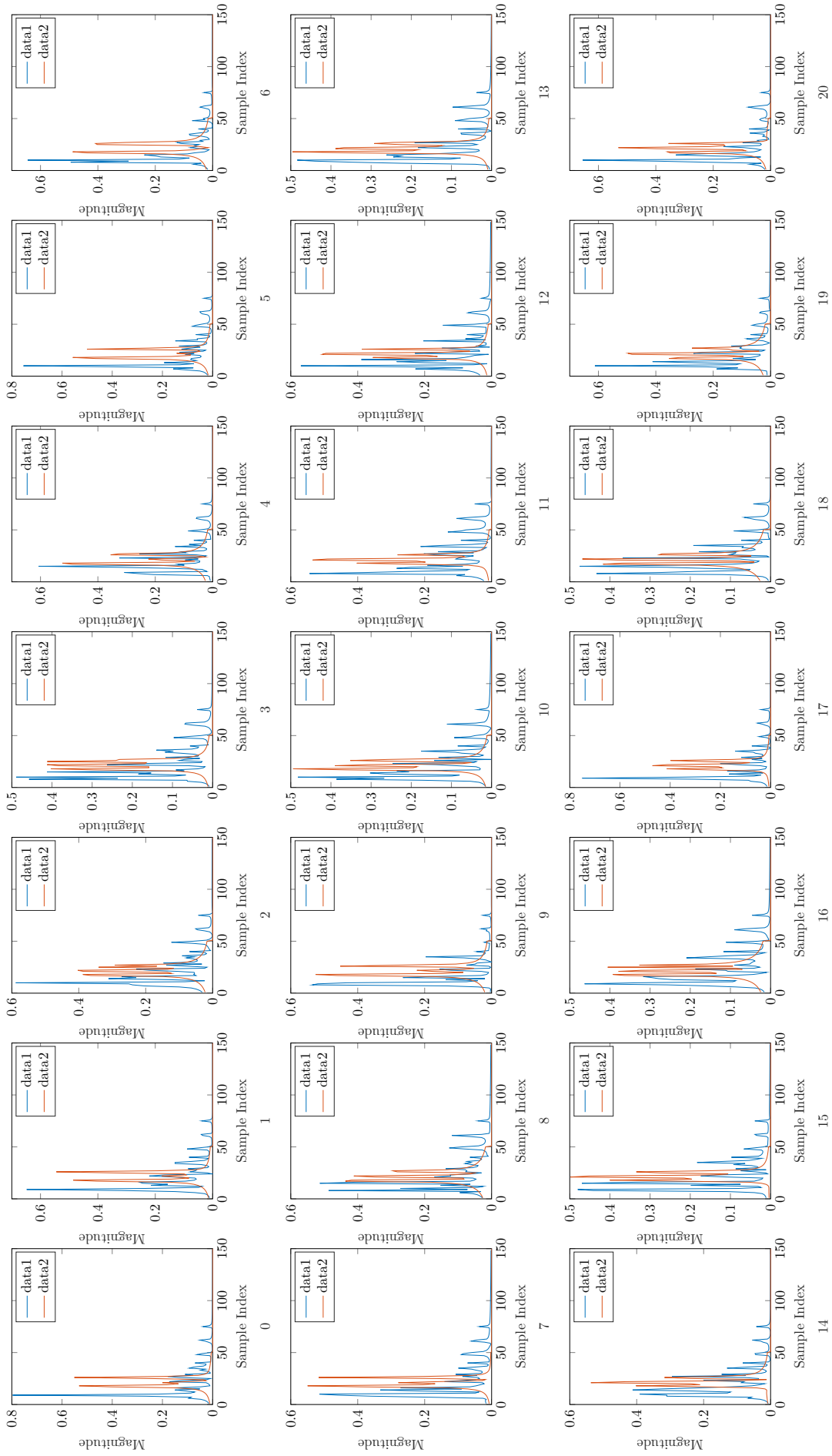


Fig. C.19: Comparison of the 150 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

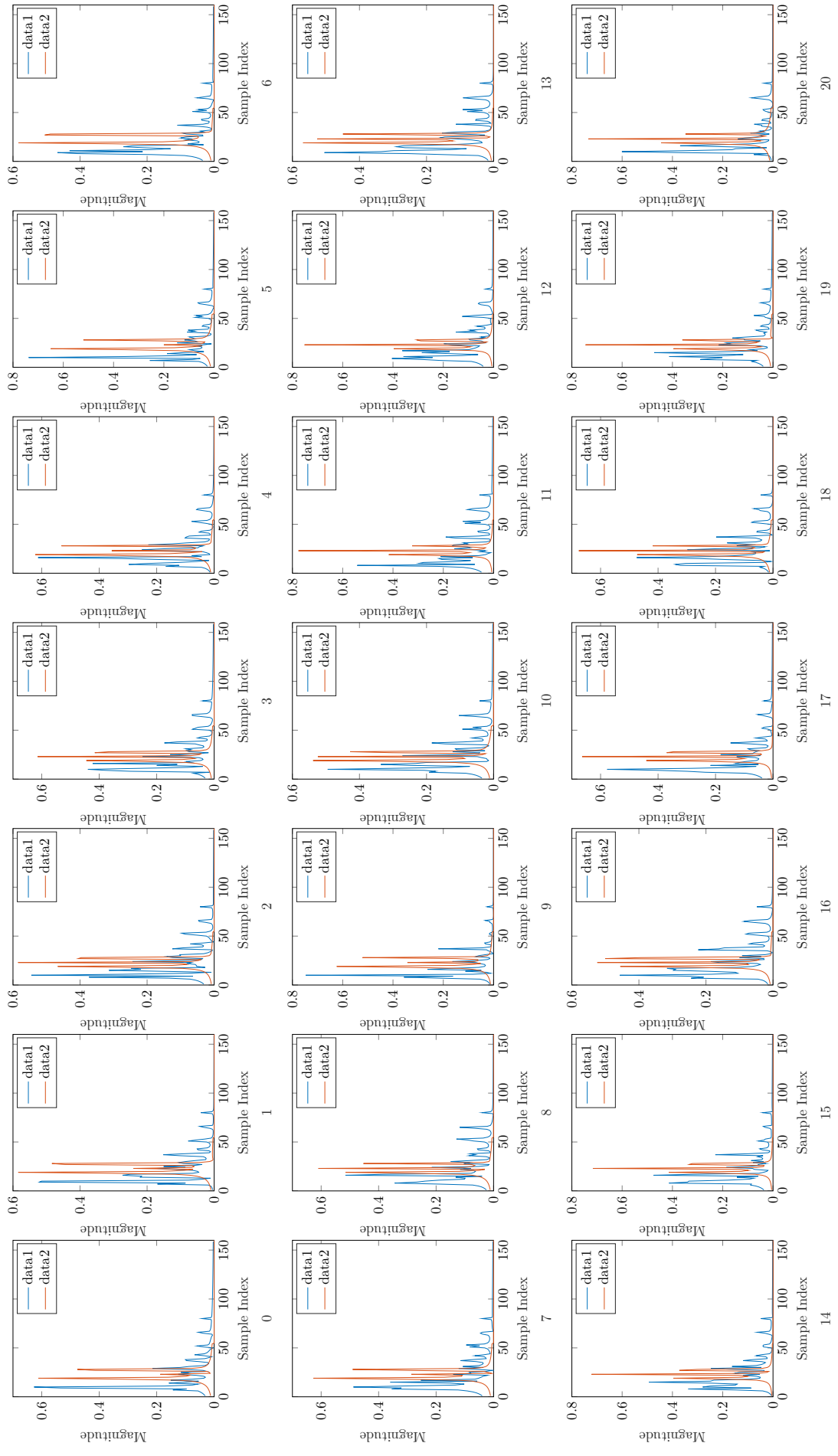


Fig. C.20: Comparison of the 160 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000



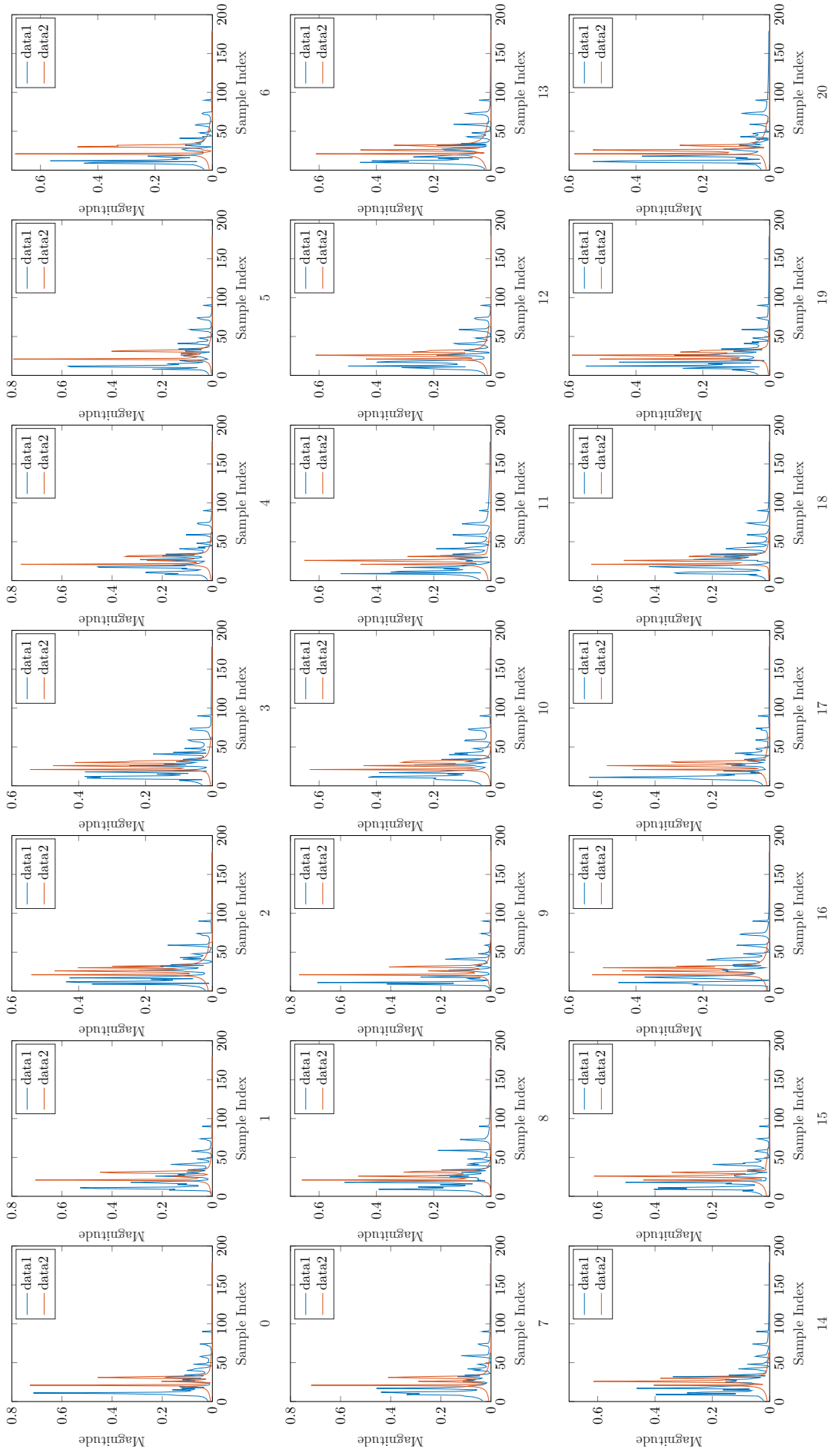


Fig. C.21: Comparison of the 180 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

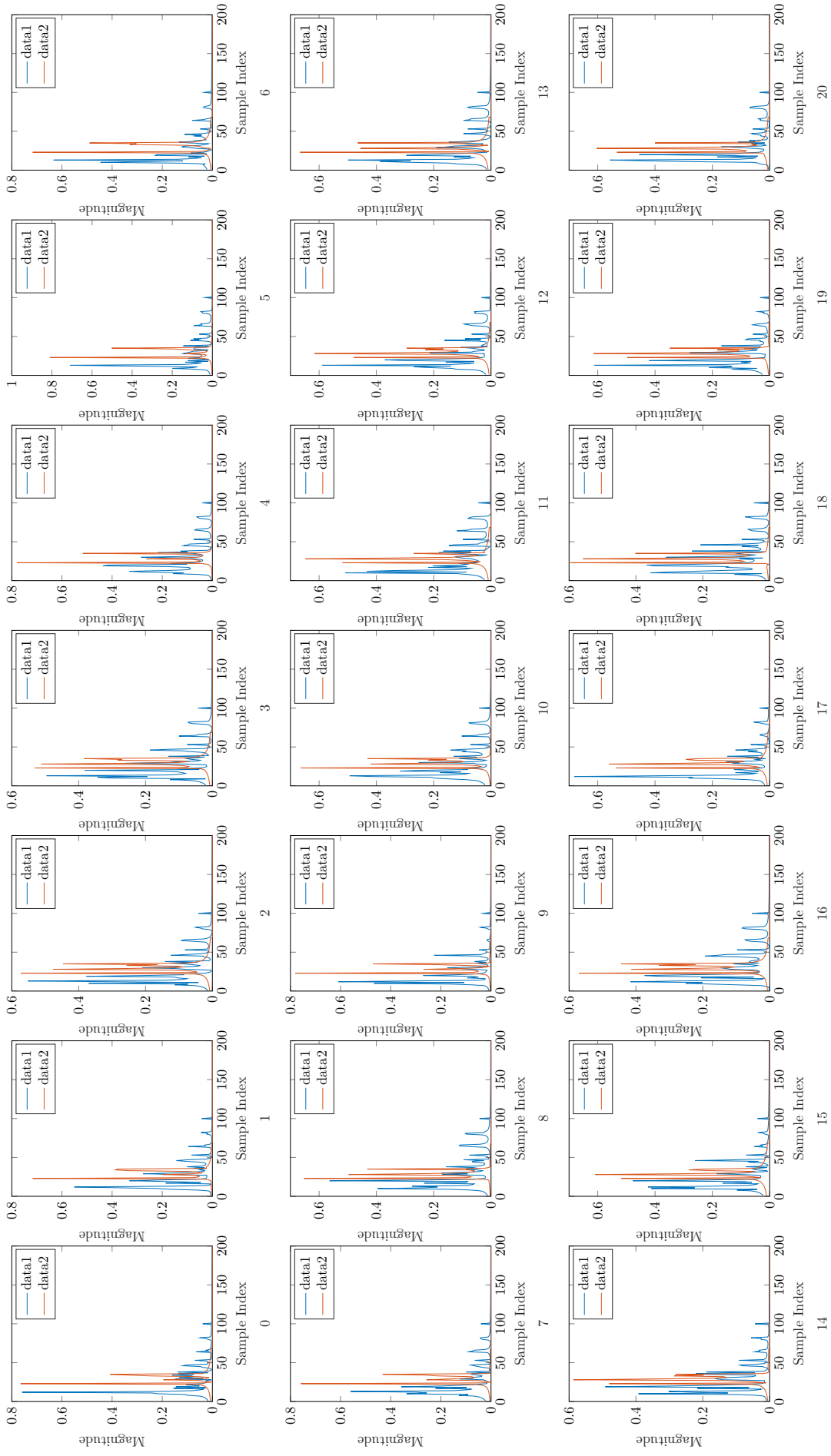


Fig. C.22: Comparison of the 200 MHz transfer functions for impulse response set IR-2 for each seed (0-20) for channels 30 vs 3000

## APPENDIX D

### Further Results

#### D.1 Simulation

As explained in Chapter 3, there were two ways in which simulation results were shown: by figure, and by table. The figures were merely used for visual comparison, to verify the algorithms were, or were not, doing what was expected, so no more figures are included in this thesis. This appendix contains the extended tables (as the full tables could not fit into the bulk of the thesis).

The simulation tables have nine columns with information about (in the following order)

- the simulation state,
- the sample rate,
- the algorithm,
- the actual distance between receivers
- the averaged estimated distance between receivers
- the minimum estimated distance between receivers
- the median estimated distance between receivers,
- the maximum estimated distance between receivers, and
- the averaged error between estimated distance and the actual distance.

For more explanation about what the information is, and how it is calculated see Section 3.3.3.

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg,1,2}$ (m)	$\hat{d}_{min,1,2}$ (m)	$\hat{d}_{med,1,2}$ (m)	$\hat{d}_{max,1,2}$ (m)	$\epsilon_{avg,1,2}$
IR-1: Ideal	20 MHz	SVD	21.772	15	15	15	15	6.772
IR-1: Ideal	20 MHz	AED	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	ModAED	21.772	14.286	0	15	30	7.486
IR-1: Ideal	20 MHz	ModAEDS	21.772	14.286	0	15	30	7.486
IR-1: Ideal	20 MHz	AXIS	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	VAXIS	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	DFPAXIS	21.772	16.429	0	15	45	5.344
IR-1: Ideal	20 MHz	NSAXIS	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	SC	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	VSC	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	DFPSC	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	NSSC	21.772	15	0	15	30	6.772
IR-1: Ideal	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: Ideal	20 MHz	NBF	21.772	18.571	0	15	30	3.201
IR-1: Ideal	20 MHz	ABF	21.772	15	15	15	15	6.772
IR-1: Ideal	20 MHz	VBF	21.772	15	15	15	15	6.772
IR-1: Ideal	20 MHz	DFPBF	21.772	15	15	15	15	6.772
IR-1: Ideal	40 MHz	SVD	21.772	15.714	7.5	15	37.5	6.058
IR-1: Ideal	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: Ideal	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Ideal	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	DFPSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	NSSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Ideal	40 MHz	CC	21.772	15	7.5	15	37.5	6.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	40 MHz	NBF	21.772	22.857	7.5	22.5	45	1.085
IR-1: Ideal	40 MHz	ABF	21.772	15.714	7.5	15	37.5	6.058
IR-1: Ideal	40 MHz	VBF	21.772	15.714	7.5	15	37.5	6.058
IR-1: Ideal	40 MHz	DFPBF	21.772	13.929	0	15	37.5	7.844
IR-1: Ideal	60 MHz	SVD	21.772	16.667	5	20	35	5.105
IR-1: Ideal	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: Ideal	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: Ideal	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: Ideal	60 MHz	AXIS	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	VAXIS	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	NSAXIS	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	SC	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	VSC	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	DFPSC	21.772	17.143	5	20	35	4.629
IR-1: Ideal	60 MHz	NSSC	21.772	16.905	5	20	35	4.867
IR-1: Ideal	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: Ideal	60 MHz	NBF	21.772	15.714	0	20	45	6.058
IR-1: Ideal	60 MHz	ABF	21.772	15.952	5	20	40	5.82
IR-1: Ideal	60 MHz	VBF	21.772	15.952	5	20	40	5.82
IR-1: Ideal	60 MHz	DFPBF	21.772	14.286	0	20	25	7.486
IR-1: Ideal	80 MHz	SVD	21.772	15.536	7.5	18.75	22.5	6.236
IR-1: Ideal	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Ideal	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Ideal	80 MHz	AXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	VAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	DFPAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	NSAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	SC	21.772	14.643	7.5	18.75	22.5	7.129

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	80 MHz	VSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	80 MHz	NSSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Ideal	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Ideal	80 MHz	NBF	21.772	15.893	7.5	18.75	48.75	5.879
IR-1: Ideal	80 MHz	ABF	21.772	14.464	3.75	18.75	22.5	7.308
IR-1: Ideal	80 MHz	VBF	21.772	14.464	3.75	18.75	22.5	7.308
IR-1: Ideal	80 MHz	DFPBF	21.772	15.536	7.5	18.75	22.5	6.236
IR-1: Ideal	100 MHz	SVD	21.772	15.286	3	18	24	6.486
IR-1: Ideal	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: Ideal	100 MHz	ModAED	21.772	16.857	6	18	36	4.915
IR-1: Ideal	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: Ideal	100 MHz	AXIS	21.772	16.714	6	18	36	5.058
IR-1: Ideal	100 MHz	VAXIS	21.772	16.714	6	18	36	5.058
IR-1: Ideal	100 MHz	DFPAXIS	21.772	17.143	6	18	36	4.629
IR-1: Ideal	100 MHz	NSAXIS	21.772	16.714	6	18	36	5.058
IR-1: Ideal	100 MHz	SC	21.772	17.143	6	18	36	4.629
IR-1: Ideal	100 MHz	VSC	21.772	17.143	6	18	36	4.629
IR-1: Ideal	100 MHz	DFPSC	21.772	16.714	6	18	36	5.058
IR-1: Ideal	100 MHz	NSSC	21.772	17.143	6	18	36	4.629
IR-1: Ideal	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: Ideal	100 MHz	NBF	21.772	14.143	3	12	45	7.629
IR-1: Ideal	100 MHz	ABF	21.772	14.571	0	15	30	7.201
IR-1: Ideal	100 MHz	VBF	21.772	14.571	0	15	30	7.201
IR-1: Ideal	100 MHz	DFPBF	21.772	14	3	15	36	7.772
IR-1: Ideal	120 MHz	SVD	21.772	16.19	5	17.5	42.5	5.582
IR-1: Ideal	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	ModAEDS	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: Ideal	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: Ideal	120 MHz	NBF	21.772	13.69	5	12.5	37.5	8.082
IR-1: Ideal	120 MHz	ABF	21.772	13.095	7.5	12.5	42.5	8.677
IR-1: Ideal	120 MHz	VBF	21.772	13.095	7.5	12.5	42.5	8.677
IR-1: Ideal	120 MHz	DFPBF	21.772	12.381	7.5	12.5	20	9.391
IR-1: Ideal	140 MHz	SVD	21.772	15.714	2.143	17.143	34.286	6.058
IR-1: Ideal	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: Ideal	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Ideal	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Ideal	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Ideal	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Ideal	140 MHz	SC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	VSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	DFPSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Ideal	140 MHz	NSSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Ideal	140 MHz	NBF	21.772	15	2.143	15	49.286	6.772
IR-1: Ideal	140 MHz	ABF	21.772	13.571	6.429	12.857	25.714	8.201
IR-1: Ideal	140 MHz	VBF	21.772	13.571	6.429	12.857	25.714	8.201
IR-1: Ideal	140 MHz	DFPBF	21.772	13.265	6.429	12.857	21.429	8.507
IR-1: Ideal	150 MHz	SVD	21.772	17.905	2	18	44	3.867

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: Ideal	150 MHz	ModAED	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	ModAEDS	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	AXIS	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	VAXIS	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	DFPAXIS	21.772	17.143	8	18	36	4.629
IR-1: Ideal	150 MHz	NSAXIS	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	SC	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	VSC	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	DFPSC	21.772	17.143	8	18	36	4.629
IR-1: Ideal	150 MHz	NSSC	21.772	16.667	8	18	36	5.105
IR-1: Ideal	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: Ideal	150 MHz	NBF	21.772	18.381	6	18	42	3.391
IR-1: Ideal	150 MHz	ABF	21.772	14.381	2	14	36	7.391
IR-1: Ideal	150 MHz	VBF	21.772	14.381	2	14	36	7.391
IR-1: Ideal	150 MHz	DFPBF	21.772	15.905	8	18	36	5.867
IR-1: Ideal	160 MHz	SVD	21.772	17.411	7.5	18.75	24.375	4.361
IR-1: Ideal	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Ideal	160 MHz	NSSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Ideal	160 MHz	NBF	21.772	17.232	5.625	15	37.5	4.542

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	160 MHz	ABF	21.772	16.429	1.875	18.75	33.75	5.344
IR-1: Ideal	160 MHz	VBF	21.772	16.429	1.875	18.75	33.75	5.344
IR-1: Ideal	160 MHz	DFPBF	21.772	17.054	7.5	18.75	24.375	4.719
IR-1: Ideal	180 MHz	SVD	21.772	15.238	3.333	16.667	36.667	6.534
IR-1: Ideal	180 MHz	AED	21.772	16.825	6.667	18.333	36.667	4.947
IR-1: Ideal	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Ideal	180 MHz	NBF	21.772	15.635	5	13.333	36.667	6.137
IR-1: Ideal	180 MHz	ABF	21.772	24.048	0	15	58.333	2.275
IR-1: Ideal	180 MHz	VBF	21.772	24.048	0	15	58.333	2.275
IR-1: Ideal	180 MHz	DFPBF	21.772	22.381	1.667	21.667	46.667	0.609
IR-1: Ideal	200 MHz	SVD	21.772	16.786	4.5	18	37.5	4.986
IR-1: Ideal	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	ModAED	21.772	13.5	7.5	15	19.5	8.272
IR-1: Ideal	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: Ideal	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	SC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	VSC	21.772	12.929	7.5	15	19.5	8.844

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Ideal	200 MHz	DFPSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	NSSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Ideal	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: Ideal	200 MHz	NBF	21.772	17.643	0	15	36	4.129
IR-1: Ideal	200 MHz	ABF	21.772	31.429	1.5	19.5	124.5	9.656
IR-1: Ideal	200 MHz	VPF	21.772	31.429	1.5	19.5	124.5	9.656
IR-1: Ideal	200 MHz	DFPBF	21.772	40	3	21	112.5	18.228
IR-1: Narrowband	20 MHz	SVD	21.772	14.286	0	0	60	7.486
IR-1: Narrowband	20 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	20 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	20 MHz	ModAEDS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	20 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	20 MHz	VAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	20 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	20 MHz	NSAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	20 MHz	SC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	20 MHz	VSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	20 MHz	DFPSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	20 MHz	NSSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	20 MHz	CC	21.772	2.321	0	0	15	19.451
IR-1: Narrowband	20 MHz	NBF	21.772	71.429	0	0	240	49.656
IR-1: Narrowband	20 MHz	ABF	21.772	62.857	0	60	240	41.085
IR-1: Narrowband	20 MHz	VPF	21.772	62.857	0	60	240	41.085
IR-1: Narrowband	20 MHz	DFPBF	21.772	NaN	0	NaN	120	NaN
IR-1: Narrowband	40 MHz	SVD	21.772	14.286	0	0	60	7.486
IR-1: Narrowband	40 MHz	AED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	ModAED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	VAXIS	21.772	5.714	0	0	60	16.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	40 MHz	DFPAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	NSAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	40 MHz	SC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	40 MHz	VSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	40 MHz	DFPSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	40 MHz	NSSC	21.772	8.571	0	0	60	13.201
IR-1: Narrowband	40 MHz	CC	21.772	1.339	0	0.937	7.5	20.433
IR-1: Narrowband	40 MHz	NBF	21.772	71.429	0	0	1440	49.656
IR-1: Narrowband	40 MHz	ABF	21.772	68.571	0	0	540	46.799
IR-1: Narrowband	40 MHz	VBF	21.772	68.571	0	0	540	46.799
IR-1: Narrowband	40 MHz	DFPBF	21.772	17.143	0	0	60	4.629
IR-1: Narrowband	60 MHz	SVD	21.772	25.714	0	0	240	3.942
IR-1: Narrowband	60 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	ModAEDS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	AXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	VAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	SC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	VSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	DFPSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	60 MHz	NSSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	60 MHz	CC	21.772	0.813	0	0.417	5	20.959
IR-1: Narrowband	60 MHz	NBF	21.772	508.571	0	0	2640	486.799
IR-1: Narrowband	60 MHz	ABF	21.772	182.857	0	0	1980	161.085
IR-1: Narrowband	60 MHz	VBF	21.772	182.857	0	0	1980	161.085
IR-1: Narrowband	60 MHz	DFPBF	21.772	11.429	0	0	60	10.344
IR-1: Narrowband	80 MHz	SVD	21.772	97.143	0	0	1740	75.371
IR-1: Narrowband	80 MHz	AED	21.772	2.857	0	0	60	18.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	80 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	ModAEDS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	AXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	VAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	SC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	VSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	DFPSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	NSSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	80 MHz	CC	21.772	0.58	0	0.234	3.75	21.192
IR-1: Narrowband	80 MHz	NBF	21.772	120	0	0	1860	98.228
IR-1: Narrowband	80 MHz	ABF	21.772	1531.429	0	1800	3600	1509.656
IR-1: Narrowband	80 MHz	VBF	21.772	1531.429	0	1800	3600	1509.656
IR-1: Narrowband	80 MHz	DFPBF	21.772	NaN	0	NaN	3540	NaN
IR-1: Narrowband	100 MHz	SVD	21.772	25.714	0	0	120	3.942
IR-1: Narrowband	100 MHz	AED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	ModAED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	VAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	DFPAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	NSAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	SC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	VSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	DFPSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	NSSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	100 MHz	CC	21.772	0.493	0	0.3	3	21.279
IR-1: Narrowband	100 MHz	NBF	21.772	260	0	0	2700	238.228
IR-1: Narrowband	100 MHz	ABF	21.772	434.286	0	0	4320	412.514

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	100 MHz	VBF	21.772	434.286	0	0	4320	412.514
IR-1: Narrowband	100 MHz	DFPBF	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	120 MHz	SVD	21.772	22.857	0	0	240	1.085
IR-1: Narrowband	120 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	120 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	120 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	VAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	DFPAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	120 MHz	SC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	VSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	DFPSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	NSSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	120 MHz	CC	21.772	0.377	0	0.208	2.5	21.395
IR-1: Narrowband	120 MHz	NBF	21.772	NaN	0	NaN	420	NaN
IR-1: Narrowband	120 MHz	ABF	21.772	251.429	0	0	4800	229.656
IR-1: Narrowband	120 MHz	VBF	21.772	251.429	0	0	4800	229.656
IR-1: Narrowband	120 MHz	DFPBF	21.772	8.571	0	0	120	13.201
IR-1: Narrowband	140 MHz	SVD	21.772	591.429	0	0	6120	569.656
IR-1: Narrowband	140 MHz	AED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	ModAED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	VAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	DFPAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	NSAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	SC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	VSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	DFPSC	21.772	5.714	0	0	60	16.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	140 MHz	NSSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	140 MHz	CC	21.772	0.324	0	0.153	2.143	21.448
IR-1: Narrowband	140 MHz	NBF	21.772	NaN	0	NaN	0	NaN
IR-1: Narrowband	140 MHz	ABF	21.772	80	0	0	1260	58.228
IR-1: Narrowband	140 MHz	VBF	21.772	80	0	0	1260	58.228
IR-1: Narrowband	140 MHz	DFPBF	21.772	22.857	0	0	120	1.085
IR-1: Narrowband	150 MHz	SVD	21.772	22.857	0	0	180	1.085
IR-1: Narrowband	150 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	ModAEDS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	AXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	VAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	SC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	VSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	DFPSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	NSSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	150 MHz	CC	21.772	0.295	0	0.133	2	21.477
IR-1: Narrowband	150 MHz	NBF	21.772	NaN	60	NaN	60	NaN
IR-1: Narrowband	150 MHz	ABF	21.772	2494.286	0	2580	6120	2472.514
IR-1: Narrowband	150 MHz	VBF	21.772	2494.286	0	2580	6120	2472.514
IR-1: Narrowband	150 MHz	DFPBF	21.772	1134.286	0	60	4800	1112.514
IR-1: Narrowband	160 MHz	SVD	21.772	394.286	0	0	7620	372.514
IR-1: Narrowband	160 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	ModAED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	ModAEDS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	AXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	VAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	160 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	SC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	VSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	DFPSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	NSSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	160 MHz	CC	21.772	0.271	0	0.117	1.875	21.502
IR-1: Narrowband	160 MHz	NBF	21.772	NaN	60	NaN	60	NaN
IR-1: Narrowband	160 MHz	ABF	21.772	2182.857	0	1020	6900	2161.085
IR-1: Narrowband	160 MHz	VBF	21.772	2182.857	0	1020	6900	2161.085
IR-1: Narrowband	160 MHz	DFPBF	21.772	1671.429	0	960	6960	1649.656
IR-1: Narrowband	180 MHz	SVD	21.772	NaN	0	NaN	300	NaN
IR-1: Narrowband	180 MHz	AED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	ModAED	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	AXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	VAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	DFPAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	NSAXIS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	SC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	VSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	DFPSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	NSSC	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	180 MHz	CC	21.772	0.249	0	0.093	1.667	21.523
IR-1: Narrowband	180 MHz	NBF	21.772	NaN	0	NaN	0	NaN
IR-1: Narrowband	180 MHz	ABF	21.772	2117.143	0	660	8820	2095.371
IR-1: Narrowband	180 MHz	VBF	21.772	2117.143	0	660	8820	2095.371
IR-1: Narrowband	180 MHz	DFPBF	21.772	1934.286	0	1260	4920	1912.514
IR-1: Narrowband	200 MHz	SVD	21.772	17.143	0	0	180	4.629
IR-1: Narrowband	200 MHz	AED	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	ModAED	21.772	2.857	0	0	60	18.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Narrowband	200 MHz	ModAEDS	21.772	5.714	0	0	60	16.058
IR-1: Narrowband	200 MHz	AXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	VAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	DFPAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	NSAXIS	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	SC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	VSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	DFPSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	NSSC	21.772	2.857	0	0	60	18.915
IR-1: Narrowband	200 MHz	CC	21.772	0.212	0	0.112	1.5	21.56
IR-1: Narrowband	200 MHz	NBF	21.772	NaN	0	NaN	0	NaN
IR-1: Narrowband	200 MHz	ABF	21.772	5242.857	180	5520	10680	5221.085
IR-1: Narrowband	200 MHz	VBF	21.772	5242.857	180	5520	10680	5221.085
IR-1: Narrowband	200 MHz	DFPBF	21.772	3742.857	0	2820	8580	3721.085
IR-1: Noisy	20 MHz	SVD	21.772	17.143	0	15	45	4.629
IR-1: Noisy	20 MHz	AED	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	ModAED	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	ModAEDS	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	AXIS	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	VAXIS	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	DFPAXIS	21.772	16.429	0	15	45	5.344
IR-1: Noisy	20 MHz	NSAXIS	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	SC	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	VSC	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	DFPSC	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	NSSC	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	CC	21.772	15	0	15	30	6.772
IR-1: Noisy	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	NBF	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	ABF	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	VBF	21.772	14.286	0	15	30	7.486
IR-1: Noisy	20 MHz	VBF	21.772	14.286	0	15	30	7.486

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	20 MHz	DFPBF	21.772	13.571	0	15	15	8.201
IR-1: Noisy	40 MHz	SVD	21.772	22.857	7.5	22.5	67.5	1.085
IR-1: Noisy	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: Noisy	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Noisy	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	DFPSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	NSSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: Noisy	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: Noisy	40 MHz	NBF	21.772	14.286	7.5	15	37.5	7.486
IR-1: Noisy	40 MHz	ABF	21.772	14.643	7.5	15	37.5	7.129
IR-1: Noisy	40 MHz	VBF	21.772	14.643	7.5	15	37.5	7.129
IR-1: Noisy	40 MHz	DFPBF	21.772	15	7.5	15	37.5	6.772
IR-1: Noisy	60 MHz	SVD	21.772	24.762	5	15	140	2.99
IR-1: Noisy	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: Noisy	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: Noisy	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: Noisy	60 MHz	AXIS	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	VAXIS	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	NSAXIS	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	SC	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	VSC	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	DFPSC	21.772	17.143	5	20	35	4.629
IR-1: Noisy	60 MHz	NSSC	21.772	16.905	5	20	35	4.867

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: Noisy	60 MHz	NBF	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	ABF	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	VBF	21.772	16.905	5	20	35	4.867
IR-1: Noisy	60 MHz	DFPBF	21.772	16.429	5	15	35	5.344
IR-1: Noisy	80 MHz	SVD	21.772	33.571	3.75	18.75	127.5	11.799
IR-1: Noisy	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Noisy	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Noisy	80 MHz	AXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	VAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	DFPAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	NSAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	SC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	VSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	80 MHz	NSSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: Noisy	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Noisy	80 MHz	NBF	21.772	16.607	7.5	18.75	37.5	5.165
IR-1: Noisy	80 MHz	ABF	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	80 MHz	VBF	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	80 MHz	DFPBF	21.772	15.714	7.5	18.75	30	6.058
IR-1: Noisy	100 MHz	SVD	21.772	22.143	6	18	78	0.371
IR-1: Noisy	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: Noisy	100 MHz	ModAED	21.772	16.857	6	18	36	4.915
IR-1: Noisy	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: Noisy	100 MHz	AXIS	21.772	16.714	6	18	36	5.058
IR-1: Noisy	100 MHz	VAXIS	21.772	16.714	6	18	36	5.058
IR-1: Noisy	100 MHz	DFPAXIS	21.772	17.143	6	18	36	4.629
IR-1: Noisy	100 MHz	NSAXIS	21.772	16.714	6	18	36	5.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	100 MHz	SC	21.772	17.143	6	18	36	4.629
IR-1: Noisy	100 MHz	VSC	21.772	17.143	6	18	36	4.629
IR-1: Noisy	100 MHz	DFPSC	21.772	16.714	6	18	36	5.058
IR-1: Noisy	100 MHz	NSSC	21.772	17.143	6	18	36	4.629
IR-1: Noisy	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: Noisy	100 MHz	NBF	21.772	15.857	6	18	36	5.915
IR-1: Noisy	100 MHz	ABF	21.772	15.857	6	18	36	5.915
IR-1: Noisy	100 MHz	VBF	21.772	15.857	6	18	36	5.915
IR-1: Noisy	100 MHz	DFPBF	21.772	23.714	6	18	195	1.942
IR-1: Noisy	120 MHz	SVD	21.772	37.857	2.5	17.5	145	16.085
IR-1: Noisy	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	ModAEDS	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: Noisy	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: Noisy	120 MHz	NBF	21.772	12.738	7.5	15	22.5	9.034
IR-1: Noisy	120 MHz	ABF	21.772	14.167	7.5	15	37.5	7.605
IR-1: Noisy	120 MHz	VBF	21.772	14.167	7.5	15	37.5	7.605
IR-1: Noisy	120 MHz	DFPBF	21.772	14.167	7.5	15	37.5	7.605
IR-1: Noisy	140 MHz	SVD	21.772	44.694	4.286	38.571	173.571	22.922
IR-1: Noisy	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: Noisy	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Noisy	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	SC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Noisy	140 MHz	VSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Noisy	140 MHz	DFPSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	NSSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Noisy	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Noisy	140 MHz	NBF	21.772	17.041	6.429	19.286	36.429	4.731
IR-1: Noisy	140 MHz	ABF	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	VBF	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Noisy	140 MHz	DFPBF	21.772	16.327	2.143	17.143	36.429	5.446
IR-1: Noisy	150 MHz	SVD	21.772	61.333	4	40	170	39.561
IR-1: Noisy	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: Noisy	150 MHz	ModAED	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	ModAEDS	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	AXIS	21.772	17.143	8	18	36	4.629
IR-1: Noisy	150 MHz	VAXIS	21.772	17.143	8	18	36	4.629
IR-1: Noisy	150 MHz	DFPAXIS	21.772	17.143	8	18	36	4.629
IR-1: Noisy	150 MHz	NSAXIS	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	SC	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	VSC	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	DFPSC	21.772	17.143	8	18	36	4.629
IR-1: Noisy	150 MHz	NSSC	21.772	16.667	8	18	36	5.105
IR-1: Noisy	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: Noisy	150 MHz	NBF	21.772	16.095	8	18	36	5.677
IR-1: Noisy	150 MHz	ABF	21.772	15.619	8	18	22	6.153
IR-1: Noisy	150 MHz	VBF	21.772	15.619	8	18	22	6.153
IR-1: Noisy	150 MHz	DFPBF	21.772	15.143	6	16	36	6.629

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	160 MHz	SVD	21.772	56.25	1.875	45	151.875	34.478
IR-1: Noisy	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	NSSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Noisy	160 MHz	NBF	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: Noisy	160 MHz	ABF	21.772	16.429	7.5	16.875	37.5	5.344
IR-1: Noisy	160 MHz	VBF	21.772	16.429	7.5	16.875	37.5	5.344
IR-1: Noisy	160 MHz	DFPBF	21.772	20.714	7.5	16.875	148.125	1.058
IR-1: Noisy	180 MHz	SVD	21.772	NaN	0	NaN	130	NaN
IR-1: Noisy	180 MHz	AED	21.772	16.825	6.667	18.333	36.667	4.947
IR-1: Noisy	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Noisy	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Noisy	180 MHz	NBF	21.772	16.111	6.667	18.333	36.667	5.661
IR-1: Noisy	180 MHz	ABF	21.772	78.492	8.333	46.667	216.667	56.72
IR-1: Noisy	180 MHz	VBF	21.772	78.492	8.333	46.667	216.667	56.72
IR-1: Noisy	180 MHz	DFPBF	21.772	94.444	8.333	88.333	221.667	72.672
IR-1: Noisy	200 MHz	SVD	21.772	NaN	6	NaN	160.5	NaN
IR-1: Noisy	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	ModAED	21.772	13.5	7.5	15	19.5	8.272
IR-1: Noisy	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: Noisy	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	SC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	VSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	DFPSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	NSSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: Noisy	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: Noisy	200 MHz	NBF	21.772	13.643	7.5	16.5	19.5	8.129
IR-1: Noisy	200 MHz	ABF	21.772	48.857	7.5	37.5	159	27.085
IR-1: Noisy	200 MHz	VBF	21.772	48.857	7.5	37.5	159	27.085
IR-1: Noisy	200 MHz	DFPBF	21.772	54.357	9	39	153	32.585
IR-1: Fewer Samples	20 MHz	SVD	21.772	15	15	15	15	6.772
IR-1: Fewer Samples	20 MHz	AED	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	ModAED	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	ModAEDS	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	AXIS	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	VAXIS	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	DFPAXIS	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	NSAXIS	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	SC	21.772	14.286	0	15	30	7.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	20 MHz	VSC	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	DFPSC	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	NSSC	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: Fewer Samples	20 MHz	NBF	21.772	15	15	15	15	6.772
IR-1: Fewer Samples	20 MHz	ABF	21.772	15	15	15	15	6.772
IR-1: Fewer Samples	20 MHz	VBF	21.772	15	15	15	15	6.772
IR-1: Fewer Samples	20 MHz	DFPBF	21.772	15	15	15	15	6.772
IR-1: Fewer Samples	40 MHz	SVD	21.772	15.714	7.5	15	37.5	6.058
IR-1: Fewer Samples	40 MHz	AED	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	AXIS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	VAXIS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	DFPAXIS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	NSAXIS	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	SC	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	VSC	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	DFPSC	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	NSSC	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: Fewer Samples	40 MHz	NBF	21.772	15.357	7.5	15	37.5	6.415
IR-1: Fewer Samples	40 MHz	ABF	21.772	15.714	7.5	15	37.5	6.058
IR-1: Fewer Samples	40 MHz	VBF	21.772	15.714	7.5	15	37.5	6.058
IR-1: Fewer Samples	40 MHz	DFPBF	21.772	14.643	7.5	15	37.5	7.129
IR-1: Fewer Samples	60 MHz	SVD	21.772	14.762	5	20	25	7.01
IR-1: Fewer Samples	60 MHz	AED	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	AXIS	21.772	16.19	5	15	35	5.582

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	60 MHz	VAXIS	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	DFPAXIS	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	NSAXIS	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	SC	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	VSC	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	DFPSC	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	NSSC	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: Fewer Samples	60 MHz	NBF	21.772	16.667	5	20	40	5.105
IR-1: Fewer Samples	60 MHz	ABF	21.772	15.952	5	20	40	5.82
IR-1: Fewer Samples	60 MHz	VBF	21.772	15.952	5	20	40	5.82
IR-1: Fewer Samples	60 MHz	DFPBF	21.772	14.762	0	20	25	7.01
IR-1: Fewer Samples	80 MHz	SVD	21.772	15.536	7.5	18.75	22.5	6.236
IR-1: Fewer Samples	80 MHz	AED	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	ModAED	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	ModAEDS	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	AXIS	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	VAXIS	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	DFPAXIS	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	NSAXIS	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	SC	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	VSC	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	DFPSC	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	NSSC	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	CC	21.772	15.536	7.5	18.75	37.5	6.236
IR-1: Fewer Samples	80 MHz	NBF	21.772	15.357	7.5	18.75	22.5	6.415
IR-1: Fewer Samples	80 MHz	ABF	21.772	15	7.5	18.75	22.5	6.772
IR-1: Fewer Samples	80 MHz	VBF	21.772	15	7.5	18.75	22.5	6.772
IR-1: Fewer Samples	80 MHz	DFPBF	21.772	15.536	7.5	18.75	22.5	6.236
IR-1: Fewer Samples	100 MHz	SVD	21.772	16.143	3	18	36	5.628

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	100 MHz	AED	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	ModAED	21.772	17.429	6	18	36	4.344
IR-1: Fewer Samples	100 MHz	ModAEDS	21.772	17.429	6	18	36	4.344
IR-1: Fewer Samples	100 MHz	AXIS	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	VAXIS	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	DFPAXIS	21.772	17.429	6	18	36	4.344
IR-1: Fewer Samples	100 MHz	NSAXIS	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	SC	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	VSC	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	DFPSC	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	NSSC	21.772	16.857	6	18	36	4.915
IR-1: Fewer Samples	100 MHz	CC	21.772	17.429	6	18	36	4.344
IR-1: Fewer Samples	100 MHz	NBF	21.772	16.429	6	18	39	5.344
IR-1: Fewer Samples	100 MHz	ABF	21.772	16.143	6	15	30	5.629
IR-1: Fewer Samples	100 MHz	VBF	21.772	16.143	6	15	30	5.629
IR-1: Fewer Samples	100 MHz	DFPBF	21.772	14	6	12	33	7.772
IR-1: Fewer Samples	120 MHz	SVD	21.772	27.5	2.5	17.5	95	5.728
IR-1: Fewer Samples	120 MHz	AED	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	ModAED	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	ModAEDS	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	AXIS	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	VAXIS	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	DFPAXIS	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	NSAXIS	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	SC	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	VSC	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	DFPSC	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	NSSC	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	CC	21.772	12.143	7.5	7.5	20	9.629
IR-1: Fewer Samples	120 MHz	NBF	21.772	16.548	5	17.5	37.5	5.225

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	120 MHz	ABF	21.772	14.881	7.5	12.5	42.5	6.891
IR-1: Fewer Samples	120 MHz	VBF	21.772	14.881	7.5	12.5	42.5	6.891
IR-1: Fewer Samples	120 MHz	DFPBF	21.772	15.595	7.5	12.5	75	6.177
IR-1: Fewer Samples	140 MHz	SVD	21.772	99.796	8.571	62.143	272.143	78.024
IR-1: Fewer Samples	140 MHz	AED	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	ModAED	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	ModAEDS	21.772	17.959	6.429	19.286	36.429	3.813
IR-1: Fewer Samples	140 MHz	AXIS	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	VAXIS	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	DFPAXIS	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	NSAXIS	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	SC	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	VSC	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	DFPSC	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	NSSC	21.772	18.163	6.429	19.286	36.429	3.609
IR-1: Fewer Samples	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Fewer Samples	140 MHz	NBF	21.772	17.041	4.286	17.143	40.714	4.731
IR-1: Fewer Samples	140 MHz	ABF	21.772	14.796	8.571	12.857	40.714	6.976
IR-1: Fewer Samples	140 MHz	VBF	21.772	14.796	8.571	12.857	40.714	6.976
IR-1: Fewer Samples	140 MHz	DFPBF	21.772	16.122	6.429	15	36.429	5.65
IR-1: Fewer Samples	150 MHz	SVD	21.772	41.143	2	12	196	19.371
IR-1: Fewer Samples	150 MHz	AED	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	ModAED	21.772	16.19	8	18	36	5.582
IR-1: Fewer Samples	150 MHz	ModAEDS	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	AXIS	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	VAXIS	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	DFPAXIS	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	NSAXIS	21.772	16.19	8	18	36	5.582
IR-1: Fewer Samples	150 MHz	SC	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	VSC	21.772	15.429	8	18	20	6.344

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	150 MHz	DFPSC	21.772	15.429	8	18	20	6.344
IR-1: Fewer Samples	150 MHz	NSSC	21.772	16.19	8	18	36	5.582
IR-1: Fewer Samples	150 MHz	CC	21.772	15.905	8	18	20	5.867
IR-1: Fewer Samples	150 MHz	NBF	21.772	17.143	2	16	44	4.629
IR-1: Fewer Samples	150 MHz	ABF	21.772	14.667	2	16	36	7.105
IR-1: Fewer Samples	150 MHz	VBF	21.772	14.667	2	16	36	7.105
IR-1: Fewer Samples	150 MHz	DFPBF	21.772	16	8	18	36	5.772
IR-1: Fewer Samples	160 MHz	SVD	21.772	55.625	7.5	7.5	255	33.853
IR-1: Fewer Samples	160 MHz	AED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	AXIS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	VAXIS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	DFPAXIS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	NSAXIS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	DFPSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	NSSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Fewer Samples	160 MHz	NBF	21.772	14.732	7.5	15	37.5	7.04
IR-1: Fewer Samples	160 MHz	ABF	21.772	16.875	7.5	16.875	37.5	4.897
IR-1: Fewer Samples	160 MHz	VBF	21.772	16.875	7.5	16.875	37.5	4.897
IR-1: Fewer Samples	160 MHz	DFPBF	21.772	19.375	7.5	18.75	24.375	2.397
IR-1: Fewer Samples	180 MHz	SVD	21.772	147.143	6.667	158.333	248.333	125.371
IR-1: Fewer Samples	180 MHz	AED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Fewer Samples	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Fewer Samples	180 MHz	NBF	21.772	12.857	0	13.333	33.333	8.915
IR-1: Fewer Samples	180 MHz	ABF	21.772	18.81	3.333	16.667	55	2.963
IR-1: Fewer Samples	180 MHz	VBF	21.772	18.81	3.333	16.667	55	2.963
IR-1: Fewer Samples	180 MHz	DFPBF	21.772	20.556	1.667	21.667	53.333	1.217
IR-1: Fewer Samples	200 MHz	SVD	21.772	172.357	7.5	190.5	250.5	150.585
IR-1: Fewer Samples	200 MHz	AED	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	ModAED	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	ModAEDS	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	AXIS	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	VAXIS	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	DFPAXIS	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	NSAXIS	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	SC	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	VSC	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	DFPSC	21.772	14.214	7.5	15	19.5	7.558
IR-1: Fewer Samples	200 MHz	NSSC	21.772	14.429	7.5	16.5	19.5	7.344
IR-1: Fewer Samples	200 MHz	CC	21.772	13.5	7.5	15	19.5	8.272
IR-1: Fewer Samples	200 MHz	NBF	21.772	16.714	6	16.5	36	5.058
IR-1: Fewer Samples	200 MHz	ABF	21.772	42.929	0	31.5	153	21.156
IR-1: Fewer Samples	200 MHz	VBF	21.772	42.929	0	31.5	153	21.156
IR-1: Fewer Samples	200 MHz	DFPBF	21.772	53.286	3	55.5	136.5	31.514
IR-1: L/2	20 MHz	SVD	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	AED	21.772	NaN	0	NaN	30	NaN

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	20 MHz	ModAED	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	ModAEDS	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	AXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	VAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	DFPAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	NSAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	SC	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	VSC	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	DFPSC	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	NSSC	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: L/2	20 MHz	NBF	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	ABF	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	VBF	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	20 MHz	DFPBF	21.772	NaN	0	NaN	30	NaN
IR-1: L/2	40 MHz	SVD	21.772	9.643	0	7.5	30	12.129
IR-1: L/2	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	ModAED	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: L/2	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L/2	40 MHz	DFPSC	21.772	15	7.5	15	37.5	6.772
IR-1: L/2	40 MHz	NSSC	21.772	14.286	7.5	15	37.5	7.486
IR-1: L/2	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: L/2	40 MHz	NBF	21.772	16.071	7.5	15	37.5	5.701
IR-1: L/2	40 MHz	ABF	21.772	16.071	7.5	15	37.5	5.701

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	40 MHz	VBF	21.772	16.071	7.5	15	37.5	5.701
IR-1: L/2	40 MHz	DFPBF	21.772	17.5	0	15	30	4.272
IR-1: L/2	60 MHz	SVD	21.772	10.714	0	10	35	11.058
IR-1: L/2	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: L/2	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: L/2	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: L/2	60 MHz	AXIS	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	VAXIS	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867
IR-1: L/2	60 MHz	NSAXIS	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	SC	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	VSC	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	DFPSC	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	NSSC	21.772	17.143	5	20	35	4.629
IR-1: L/2	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: L/2	60 MHz	NBF	21.772	16.429	5	20	35	5.344
IR-1: L/2	60 MHz	ABF	21.772	16.429	5	20	35	5.344
IR-1: L/2	60 MHz	VBF	21.772	16.429	5	20	35	5.344
IR-1: L/2	60 MHz	DFPBF	21.772	16.667	5	20	35	5.105
IR-1: L/2	80 MHz	SVD	21.772	10.536	0	7.5	33.75	11.236
IR-1: L/2	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L/2	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L/2	80 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	SC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	VSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	80 MHz	NSSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L/2	80 MHz	NBF	21.772	17.5	0	18.75	45	4.272
IR-1: L/2	80 MHz	ABF	21.772	15.893	0	18.75	33.75	5.879
IR-1: L/2	80 MHz	VBF	21.772	15.893	0	18.75	33.75	5.879
IR-1: L/2	80 MHz	DFPBF	21.772	13.393	7.5	15	22.5	8.379
IR-1: L/2	100 MHz	SVD	21.772	12.714	3	9	33	9.058
IR-1: L/2	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: L/2	100 MHz	ModAED	21.772	16.857	6	18	36	4.915
IR-1: L/2	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: L/2	100 MHz	AXIS	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	VAXIS	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	DFPAXIS	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	NSAXIS	21.772	16	6	18	36	5.772
IR-1: L/2	100 MHz	SC	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	VSC	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	DFPSC	21.772	16	6	18	36	5.772
IR-1: L/2	100 MHz	NSSC	21.772	16.714	6	18	36	5.058
IR-1: L/2	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: L/2	100 MHz	NBF	21.772	19.429	6	18	45	2.344
IR-1: L/2	100 MHz	ABF	21.772	17.857	6	18	45	3.915
IR-1: L/2	100 MHz	VBF	21.772	17.857	6	18	45	3.915
IR-1: L/2	100 MHz	DFPBF	21.772	13.571	6	15	24	8.201
IR-1: L/2	120 MHz	SVD	21.772	11.429	0	7.5	35	10.344
IR-1: L/2	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	ModAEDS	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: L/2	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: L/2	120 MHz	NBF	21.772	14.762	7.5	17.5	22.5	7.01
IR-1: L/2	120 MHz	ABF	21.772	14.762	7.5	17.5	22.5	7.01
IR-1: L/2	120 MHz	VBF	21.772	14.762	7.5	17.5	22.5	7.01
IR-1: L/2	120 MHz	DFPBF	21.772	13.452	7.5	15	22.5	8.32
IR-1: L/2	140 MHz	SVD	21.772	12.041	2.143	10.714	30	9.731
IR-1: L/2	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: L/2	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L/2	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L/2	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	SC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	VSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	DFPSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	NSSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L/2	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L/2	140 MHz	NBF	21.772	15.918	6.429	17.143	47.143	5.854
IR-1: L/2	140 MHz	ABF	21.772	18.265	6.429	19.286	42.857	3.507
IR-1: L/2	140 MHz	VBF	21.772	18.265	6.429	19.286	42.857	3.507
IR-1: L/2	140 MHz	DFPBF	21.772	16.02	2.143	17.143	38.571	5.752
IR-1: L/2	150 MHz	SVD	21.772	12.095	2	8	38	9.677
IR-1: L/2	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	ModAED	21.772	16.667	8	18	36	5.106

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	150 MHz	ModAEDS	21.772	16.667	8	18	36	5.105
IR-1: L/2	150 MHz	AXIS	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	VAXIS	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	DFPAXIS	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	NSAXIS	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	SC	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	VSC	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	DFPSC	21.772	17.143	8	18	36	4.629
IR-1: L/2	150 MHz	NSSC	21.772	16.667	8	18	36	5.105
IR-1: L/2	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: L/2	150 MHz	NBF	21.772	14.952	6	18	22	6.82
IR-1: L/2	150 MHz	ABF	21.772	16.857	6	16	36	4.915
IR-1: L/2	150 MHz	VBF	21.772	16.857	6	16	36	4.915
IR-1: L/2	150 MHz	DFPBF	21.772	16.952	8	18	38	4.82
IR-1: L/2	160 MHz	SVD	21.772	10.536	1.875	7.5	30	11.236
IR-1: L/2	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L/2	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L/2	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	SC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	VSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	NSSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L/2	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L/2	160 MHz	NBF	21.772	15.536	7.5	16.875	20.625	6.236
IR-1: L/2	160 MHz	ABF	21.772	17.5	7.5	16.875	43.125	4.272
IR-1: L/2	160 MHz	VBF	21.772	17.5	7.5	16.875	43.125	4.272

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	160 MHz	DFPBF	21.772	14.375	1.875	16.875	37.5	7.397
IR-1: L/2	180 MHz	SVD	21.772	9.921	3.333	6.667	30	11.851
IR-1: L/2	180 MHz	AED	21.772	16.905	6.667	18.333	36.667	4.867
IR-1: L/2	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L/2	180 MHz	NBF	21.772	17.46	6.667	18.333	46.667	4.312
IR-1: L/2	180 MHz	ABF	21.772	16.349	1.667	15	53.333	5.423
IR-1: L/2	180 MHz	VBF	21.772	16.349	1.667	15	53.333	5.423
IR-1: L/2	180 MHz	DFPBF	21.772	13.81	6.667	16.667	21.667	7.963
IR-1: L/2	200 MHz	SVD	21.772	12.214	4.5	9	28.5	9.558
IR-1: L/2	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	ModAED	21.772	13.5	7.5	15	19.5	8.272
IR-1: L/2	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L/2	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	SC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	VSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	DFPSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L/2	200 MHz	NSSC	21.772	12.929	7.5	15	19.5	8.844

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L/2	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L/2	200 MHz	NBF	21.772	18.714	7.5	16.5	46.5	3.058
IR-1: L/2	200 MHz	ABF	21.772	13.786	1.5	13.5	42	7.986
IR-1: L/2	200 MHz	VBF	21.772	13.786	1.5	13.5	42	7.986
IR-1: L/2	200 MHz	DFPBF	21.772	13.929	6	15	28.5	7.844
IR-1: L-10	20 MHz	SVD	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	AED	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	ModAED	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	ModAEDS	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	AXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	VAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	DFPAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	NSAXIS	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	SC	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	VSC	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	DFPSC	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	NSSC	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: L-10	20 MHz	NBF	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	ABF	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	VBF	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	20 MHz	DFPBF	21.772	NaN	0	NaN	30	NaN
IR-1: L-10	40 MHz	SVD	21.772	16.429	0	15	37.5	5.344
IR-1: L-10	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: L-10	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: L-10	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	DFPSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	NSSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L-10	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: L-10	40 MHz	NBF	21.772	14.286	7.5	15	37.5	7.486
IR-1: L-10	40 MHz	ABF	21.772	14.643	7.5	15	37.5	7.129
IR-1: L-10	40 MHz	VBF	21.772	14.643	7.5	15	37.5	7.129
IR-1: L-10	40 MHz	DFPBF	21.772	16.786	7.5	15	37.5	4.986
IR-1: L-10	60 MHz	SVD	21.772	15	5	15	35	6.772
IR-1: L-10	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: L-10	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: L-10	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: L-10	60 MHz	AXIS	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	VAXIS	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	NSAXIS	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	SC	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	VSC	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	DFPSC	21.772	17.143	5	20	35	4.629
IR-1: L-10	60 MHz	NSSC	21.772	16.905	5	20	35	4.867
IR-1: L-10	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: L-10	60 MHz	NBF	21.772	16.667	5	20	35	5.105
IR-1: L-10	60 MHz	ABF	21.772	18.571	5	15	95	3.201
IR-1: L-10	60 MHz	VBF	21.772	18.571	5	15	95	3.201
IR-1: L-10	60 MHz	DFPBF	21.772	15.714	5	15	35	6.058
IR-1: L-10	80 MHz	SVD	21.772	15.714	0	15	37.5	6.058
IR-1: L-10	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L-10	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	80 MHz	AXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L-10	80 MHz	VAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L-10	80 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	80 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	80 MHz	SC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L-10	80 MHz	VSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L-10	80 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	80 MHz	NSSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L-10	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L-10	80 MHz	NBF	21.772	15.179	7.5	18.75	37.5	6.594
IR-1: L-10	80 MHz	ABF	21.772	14.821	3.75	15	37.5	6.951
IR-1: L-10	80 MHz	VBF	21.772	14.821	3.75	15	37.5	6.951
IR-1: L-10	80 MHz	DFPBF	21.772	15.536	3.75	15	37.5	6.236
IR-1: L-10	100 MHz	SVD	21.772	16.143	6	18	36	5.629
IR-1: L-10	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: L-10	100 MHz	ModAED	21.772	16.857	6	18	36	4.915
IR-1: L-10	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: L-10	100 MHz	AXIS	21.772	16.714	6	18	36	5.058
IR-1: L-10	100 MHz	VAXIS	21.772	16.714	6	18	36	5.058
IR-1: L-10	100 MHz	DFPAXIS	21.772	17.143	6	18	36	4.629
IR-1: L-10	100 MHz	NSAXIS	21.772	16.714	6	18	36	5.058
IR-1: L-10	100 MHz	SC	21.772	17.143	6	18	36	4.629
IR-1: L-10	100 MHz	VSC	21.772	17.143	6	18	36	4.629
IR-1: L-10	100 MHz	DFPSC	21.772	16.714	6	18	36	5.058
IR-1: L-10	100 MHz	NSSC	21.772	17.143	6	18	36	4.629
IR-1: L-10	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: L-10	100 MHz	NBF	21.772	20.143	6	18	117	1.629
IR-1: L-10	100 MHz	ABF	21.772	15.857	6	18	36	5.915
IR-1: L-10	100 MHz	VBF	21.772	15.857	6	18	36	5.915
IR-1: L-10	100 MHz	DFPBF	21.772	16.429	6	18	36	5.344

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	120 MHz	SVD	21.772	15.476	5	17.5	35	6.296
IR-1: L-10	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	ModAEDS	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: L-10	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: L-10	120 MHz	NBF	21.772	15.714	7.5	17.5	35	6.058
IR-1: L-10	120 MHz	ABF	21.772	15.714	5	17.5	37.5	6.058
IR-1: L-10	120 MHz	VBF	21.772	15.714	5	17.5	37.5	6.058
IR-1: L-10	120 MHz	DFPBF	21.772	17.143	7.5	17.5	37.5	4.629
IR-1: L-10	140 MHz	SVD	21.772	15	6.429	17.143	36.429	6.772
IR-1: L-10	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: L-10	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L-10	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L-10	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L-10	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L-10	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L-10	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L-10	140 MHz	SC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L-10	140 MHz	VSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L-10	140 MHz	DFPSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L-10	140 MHz	NSSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L-10	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	140 MHz	NBF	21.772	23.673	2.143	17.143	117.857	1.901
IR-1: L-10	140 MHz	ABF	21.772	13.265	6.429	15	21.429	8.507
IR-1: L-10	140 MHz	VBF	21.772	13.265	6.429	15	21.429	8.507
IR-1: L-10	140 MHz	DFPBF	21.772	16.224	2.143	17.143	36.429	5.548
IR-1: L-10	150 MHz	SVD	21.772	15.524	6	18	36	6.248
IR-1: L-10	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: L-10	150 MHz	ModAED	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	ModAEDS	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	AXIS	21.772	17.143	8	18	36	4.629
IR-1: L-10	150 MHz	VAXIS	21.772	17.143	8	18	36	4.629
IR-1: L-10	150 MHz	DFPAXIS	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	NSAXIS	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	SC	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	VSC	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	DFPSC	21.772	17.143	8	18	36	4.629
IR-1: L-10	150 MHz	NSSC	21.772	16.667	8	18	36	5.105
IR-1: L-10	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: L-10	150 MHz	NBF	21.772	15.905	4	18	38	5.867
IR-1: L-10	150 MHz	ABF	21.772	15.429	4	18	36	6.344
IR-1: L-10	150 MHz	VBF	21.772	15.429	4	18	36	6.344
IR-1: L-10	150 MHz	DFPBF	21.772	15.905	6	18	36	5.867
IR-1: L-10	160 MHz	SVD	21.772	14.375	5.625	16.875	35.625	7.397
IR-1: L-10	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L-10	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L-10	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L-10	160 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	NSSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L-10	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L-10	160 MHz	NBF	21.772	14.554	1.875	15	37.5	7.219
IR-1: L-10	160 MHz	ABF	21.772	14.375	3.75	15	35.625	7.397
IR-1: L-10	160 MHz	VBF	21.772	14.375	3.75	15	35.625	7.397
IR-1: L-10	160 MHz	DFPBF	21.772	14.464	3.75	16.875	24.375	7.308
IR-1: L-10	180 MHz	SVD	21.772	14.603	6.667	16.667	35	7.169
IR-1: L-10	180 MHz	AED	21.772	16.825	6.667	18.333	36.667	4.947
IR-1: L-10	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L-10	180 MHz	NBF	21.772	13.81	1.667	15	25	7.963
IR-1: L-10	180 MHz	ABF	21.772	14.286	5	15	23.333	7.486
IR-1: L-10	180 MHz	VBF	21.772	14.286	5	15	23.333	7.486
IR-1: L-10	180 MHz	DFPBF	21.772	15.317	1.667	15	36.667	6.455
IR-1: L-10	200 MHz	SVD	21.772	14.214	6	16.5	36	7.558
IR-1: L-10	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: L-10	200 MHz	ModAED	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L-10	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L-10	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L-10	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L-10	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L-10	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L-10	200 MHz	SC	21.772	13.5	7.5	15	19.5	8.272
IR-1: L-10	200 MHz	VSC	21.772	13.5	7.5	15	19.5	8.272
IR-1: L-10	200 MHz	DFPSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L-10	200 MHz	NSSC	21.772	13.5	7.5	15	19.5	8.272
IR-1: L-10	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L-10	200 MHz	NBF	21.772	14.929	7.5	15	34.5	6.844
IR-1: L-10	200 MHz	ABF	21.772	14.357	1.5	15	27	7.415
IR-1: L-10	200 MHz	VBF	21.772	14.357	1.5	15	27	7.415
IR-1: L-10	200 MHz	DFPBF	21.772	22.643	0	19.5	57	0.871
IR-1: 2*L	20 MHz	SVD	21.772	90.714	0	90	255	68.942
IR-1: 2*L	20 MHz	AED	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	ModAED	21.772	14.286	0	15	30	7.486
IR-1: 2*L	20 MHz	ModAEDS	21.772	14.286	0	15	30	7.486
IR-1: 2*L	20 MHz	AXIS	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	VAXIS	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	DFPAXIS	21.772	16.429	0	15	45	5.344
IR-1: 2*L	20 MHz	NSAXIS	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	SC	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	VSC	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	DFPSC	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	NSSC	21.772	15	0	15	30	6.772
IR-1: 2*L	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: 2*L	20 MHz	NBF	21.772	162.143	15	225	285	140.371
IR-1: 2*L	20 MHz	ABF	21.772	152.143	0	90	345	130.371
IR-1: 2*L	20 MHz	VBF	21.772	152.143	0	90	345	130.371
IR-1: 2*L	20 MHz	DFPBF	21.772	13.571	0	15	15	8.201
IR-1: 2*L	40 MHz	SVD	21.772	138.214	7.5	142.5	285	116.442

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: 2*L	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: 2*L	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	DFPSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	NSSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: 2*L	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: 2*L	40 MHz	NBF	21.772	181.786	0	262.5	315	160.014
IR-1: 2*L	40 MHz	ABF	21.772	145.714	7.5	150	322.5	123.942
IR-1: 2*L	40 MHz	VBF	21.772	145.714	7.5	150	322.5	123.942
IR-1: 2*L	40 MHz	DFPBF	21.772	14.643	7.5	15	37.5	7.129
IR-1: 2*L	60 MHz	SVD	21.772	90.714	5	75	275	68.942
IR-1: 2*L	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: 2*L	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	AXIS	21.772	16.905	5	20	35	4.867
IR-1: 2*L	60 MHz	VAXIS	21.772	16.905	5	20	35	4.867
IR-1: 2*L	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867
IR-1: 2*L	60 MHz	NSAXIS	21.772	16.905	5	20	35	4.867
IR-1: 2*L	60 MHz	SC	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	VSC	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	DFPSC	21.772	16.905	5	20	35	4.867
IR-1: 2*L	60 MHz	NSSC	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: 2*L	60 MHz	NBF	21.772	135.714	0	115	290	113.942

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	60 MHz	ABF	21.772	115.238	5	105	280	93.466
IR-1: 2*L	60 MHz	VBF	21.772	115.238	5	105	280	93.466
IR-1: 2*L	60 MHz	DFPBF	21.772	NaN	5	NaN	35	NaN
IR-1: 2*L	80 MHz	SVD	21.772	100.893	3.75	63.75	296.25	79.121
IR-1: 2*L	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: 2*L	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	AXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: 2*L	80 MHz	VAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: 2*L	80 MHz	DFPAXIS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	NSAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: 2*L	80 MHz	SC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	VSC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	DFPSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: 2*L	80 MHz	NSSC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	80 MHz	NBF	21.772	140.179	0	116.25	281.25	118.406
IR-1: 2*L	80 MHz	ABF	21.772	83.75	3.75	48.75	285	61.978
IR-1: 2*L	80 MHz	VBF	21.772	83.75	3.75	48.75	285	61.978
IR-1: 2*L	80 MHz	DFPBF	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: 2*L	100 MHz	SVD	21.772	84	3	63	279	62.228
IR-1: 2*L	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: 2*L	100 MHz	ModAED	21.772	16.857	6	18	36	4.915
IR-1: 2*L	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: 2*L	100 MHz	AXIS	21.772	17.143	6	18	36	4.629
IR-1: 2*L	100 MHz	VAXIS	21.772	17.143	6	18	36	4.629
IR-1: 2*L	100 MHz	DFPAXIS	21.772	17.143	6	18	36	4.629
IR-1: 2*L	100 MHz	NSAXIS	21.772	17.143	6	18	36	4.629
IR-1: 2*L	100 MHz	SC	21.772	16.857	6	18	36	4.915
IR-1: 2*L	100 MHz	VSC	21.772	16.857	6	18	36	4.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	100 MHz	DFPSC	21.772	17.143	6	18	36	4.629
IR-1: 2*L	100 MHz	NSSC	21.772	16.857	6	18	36	4.915
IR-1: 2*L	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: 2*L	100 MHz	NBF	21.772	133.143	6	99	285	111.371
IR-1: 2*L	100 MHz	ABF	21.772	61.714	0	30	234	39.942
IR-1: 2*L	100 MHz	VBF	21.772	61.714	0	30	234	39.942
IR-1: 2*L	100 MHz	DFPBF	21.772	92.286	6	60	270	70.514
IR-1: 2*L	120 MHz	SVD	21.772	NaN	0	NaN	215	NaN
IR-1: 2*L	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	ModAEDS	21.772	12.143	7.5	7.5	20	9.629
IR-1: 2*L	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: 2*L	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: 2*L	120 MHz	NBF	21.772	NaN	7.5	NaN	320	NaN
IR-1: 2*L	120 MHz	ABF	21.772	120.476	2.5	77.5	420	98.704
IR-1: 2*L	120 MHz	VBF	21.772	120.476	2.5	77.5	420	98.704
IR-1: 2*L	120 MHz	DFPBF	21.772	189.167	7.5	152.5	425	167.395
IR-1: 2*L	140 MHz	SVD	21.772	NaN	0	NaN	222.857	NaN
IR-1: 2*L	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: 2*L	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: 2*L	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: 2*L	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: 2*L	140 MHz	SC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	VSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	DFPSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	NSSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: 2*L	140 MHz	NBF	21.772	NaN	2.143	NaN	308.571	NaN
IR-1: 2*L	140 MHz	ABF	21.772	188.673	8.571	154.286	527.143	166.901
IR-1: 2*L	140 MHz	VBF	21.772	188.673	8.571	154.286	527.143	166.901
IR-1: 2*L	140 MHz	DFPBF	21.772	221.224	15	188.571	576.429	199.452
IR-1: 2*L	150 MHz	SVD	21.772	NaN	2	NaN	256	NaN
IR-1: 2*L	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: 2*L	150 MHz	ModAED	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	ModAEDS	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	AXIS	21.772	16.667	8	18	36	5.105
IR-1: 2*L	150 MHz	VAXIS	21.772	16.667	8	18	36	5.105
IR-1: 2*L	150 MHz	DFPAXIS	21.772	16.667	8	18	36	5.105
IR-1: 2*L	150 MHz	NSAXIS	21.772	16.667	8	18	36	5.105
IR-1: 2*L	150 MHz	SC	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	VSC	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	DFPSC	21.772	16.667	8	18	36	5.105
IR-1: 2*L	150 MHz	NSSC	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: 2*L	150 MHz	NBF	21.772	NaN	4	NaN	314	NaN
IR-1: 2*L	150 MHz	ABF	21.772	220.476	20	226	452	198.704
IR-1: 2*L	150 MHz	VBF	21.772	220.476	20	226	452	198.704
IR-1: 2*L	150 MHz	DFPBF	21.772	189.143	22	200	420	167.371
IR-1: 2*L	160 MHz	SVD	21.772	NaN	3.75	NaN	249.375	NaN
IR-1: 2*L	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: 2*L	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: 2*L	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: 2*L	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: 2*L	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	DFPSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	NSSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: 2*L	160 MHz	NBF	21.772	NaN	3.75	NaN	294.375	NaN
IR-1: 2*L	160 MHz	ABF	21.772	NaN	9.375	NaN	506.25	NaN
IR-1: 2*L	160 MHz	VBF	21.772	NaN	9.375	NaN	506.25	NaN
IR-1: 2*L	160 MHz	DFPBF	21.772	NaN	5.625	NaN	528.75	NaN
IR-1: 2*L	180 MHz	SVD	21.772	NaN	6.667	NaN	288.333	NaN
IR-1: 2*L	180 MHz	AED	21.772	16.825	6.667	18.333	36.667	4.947
IR-1: 2*L	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: 2*L	180 MHz	NBF	21.772	NaN	5	NaN	283.333	NaN
IR-1: 2*L	180 MHz	ABF	21.772	NaN	3.333	NaN	483.333	NaN

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: 2*L	180 MHz	VBF	21.772	NaN	3.333	NaN	483.333	NaN
IR-1: 2*L	180 MHz	DFPBF	21.772	NaN	1.667	NaN	421.667	NaN
IR-1: 2*L	200 MHz	SVD	21.772	NaN	12	NaN	117	NaN
IR-1: 2*L	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: 2*L	200 MHz	ModAED	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: 2*L	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: 2*L	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: 2*L	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: 2*L	200 MHz	SC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	VSC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	DFPSC	21.772	13.5	7.5	15	19.5	8.272
IR-1: 2*L	200 MHz	NSSC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: 2*L	200 MHz	NBF	21.772	NaN	6	NaN	168	NaN
IR-1: 2*L	200 MHz	ABF	21.772	NaN	93	NaN	340.5	NaN
IR-1: 2*L	200 MHz	VBF	21.772	NaN	93	NaN	340.5	NaN
IR-1: 2*L	200 MHz	DFPBF	21.772	NaN	15	NaN	286.5	NaN
IR-1: L+10	20 MHz	SVD	21.772	31.429	0	15	120	9.656
IR-1: L+10	20 MHz	AED	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	ModAED	21.772	14.286	0	15	30	7.486
IR-1: L+10	20 MHz	ModAEDS	21.772	14.286	0	15	30	7.486
IR-1: L+10	20 MHz	AXIS	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	VAXIS	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	DFPAXIS	21.772	16.429	0	15	45	5.344
IR-1: L+10	20 MHz	NSAXIS	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	SC	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	VSC	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	DFPSC	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	DFPSC	21.772	15	0	15	30	6.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	20 MHz	NSSC	21.772	15	0	15	30	6.772
IR-1: L+10	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: L+10	20 MHz	NBF	21.772	65.714	0	60	135	43.942
IR-1: L+10	20 MHz	ABF	21.772	100.714	0	135	195	78.942
IR-1: L+10	20 MHz	VBF	21.772	100.714	0	135	195	78.942
IR-1: L+10	20 MHz	DFPBF	21.772	NaN	0	NaN	15	NaN
IR-1: L+10	40 MHz	SVD	21.772	27.857	7.5	15	82.5	6.085
IR-1: L+10	40 MHz	AED	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	ModAED	21.772	14.643	7.5	15	37.5	7.129
IR-1: L+10	40 MHz	ModAEDS	21.772	14.643	7.5	15	37.5	7.129
IR-1: L+10	40 MHz	AXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	VAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	DFPAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	NSAXIS	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	SC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	VSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	DFPSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	NSSC	21.772	13.929	7.5	15	37.5	7.844
IR-1: L+10	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: L+10	40 MHz	NBF	21.772	42.143	7.5	37.5	105	20.371
IR-1: L+10	40 MHz	ABF	21.772	48.214	7.5	45	97.5	26.442
IR-1: L+10	40 MHz	VBF	21.772	48.214	7.5	45	97.5	26.442
IR-1: L+10	40 MHz	DFPBF	21.772	15	7.5	15	37.5	6.772
IR-1: L+10	60 MHz	SVD	21.772	20.714	5	20	60	1.058
IR-1: L+10	60 MHz	AED	21.772	16.429	5	20	35	5.344
IR-1: L+10	60 MHz	ModAED	21.772	16.19	5	15	35	5.582
IR-1: L+10	60 MHz	ModAEDS	21.772	16.19	5	15	35	5.582
IR-1: L+10	60 MHz	AXIS	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	VAXIS	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	DFPAXIS	21.772	16.905	5	20	35	4.867

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	60 MHz	NSAXIS	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	SC	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	VSC	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	DFPSC	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	NSSC	21.772	16.905	5	20	35	4.867
IR-1: L+10	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: L+10	60 MHz	NBF	21.772	27.619	0	25	70	5.847
IR-1: L+10	60 MHz	ABF	21.772	38.571	0	35	85	16.799
IR-1: L+10	60 MHz	VBF	21.772	38.571	0	35	85	16.799
IR-1: L+10	60 MHz	DFPBF	21.772	16.667	5	15	35	5.105
IR-1: L+10	80 MHz	SVD	21.772	16.786	0	15	41.25	4.986
IR-1: L+10	80 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	80 MHz	ModAED	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L+10	80 MHz	ModAEDS	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L+10	80 MHz	AXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	VAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	DFPAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	NSAXIS	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	SC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	VSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	DFPSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	NSSC	21.772	14.643	7.5	18.75	22.5	7.129
IR-1: L+10	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: L+10	80 MHz	NBF	21.772	22.143	0	18.75	67.5	0.371
IR-1: L+10	80 MHz	ABF	21.772	35.179	7.5	37.5	60	13.406
IR-1: L+10	80 MHz	VBF	21.772	35.179	7.5	37.5	60	13.406
IR-1: L+10	80 MHz	DFPBF	21.772	15.357	7.5	15	33.75	6.415
IR-1: L+10	100 MHz	SVD	21.772	13	0	9	42	8.772
IR-1: L+10	100 MHz	AED	21.772	15.429	6	18	36	6.344
IR-1: L+10	100 MHz	ModAED	21.772	16.857	6	18	36	4.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	100 MHz	ModAEDS	21.772	16.857	6	18	36	4.915
IR-1: L+10	100 MHz	AXIS	21.772	16.714	6	18	36	5.058
IR-1: L+10	100 MHz	VAXIS	21.772	16.714	6	18	36	5.058
IR-1: L+10	100 MHz	DFPAXIS	21.772	17.143	6	18	36	4.629
IR-1: L+10	100 MHz	NSAXIS	21.772	16.714	6	18	36	5.058
IR-1: L+10	100 MHz	SC	21.772	17.143	6	18	36	4.629
IR-1: L+10	100 MHz	VSC	21.772	17.143	6	18	36	4.629
IR-1: L+10	100 MHz	DFPSC	21.772	16.714	6	18	36	5.058
IR-1: L+10	100 MHz	NSSC	21.772	17.143	6	18	36	4.629
IR-1: L+10	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: L+10	100 MHz	NBF	21.772	21.429	3	18	66	0.344
IR-1: L+10	100 MHz	ABF	21.772	18.857	0	18	45	2.915
IR-1: L+10	100 MHz	VBF	21.772	18.857	0	18	45	2.915
IR-1: L+10	100 MHz	DFPBF	21.772	15.143	0	18	36	6.629
IR-1: L+10	120 MHz	SVD	21.772	15.476	0	15	37.5	6.296
IR-1: L+10	120 MHz	AED	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	ModAED	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	ModAEDS	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	AXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	VAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	DFPAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	NSAXIS	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	SC	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	VSC	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	DFPSC	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	NSSC	21.772	12.738	7.5	15	20	9.034
IR-1: L+10	120 MHz	CC	21.772	14.286	7.5	15	37.5	7.486
IR-1: L+10	120 MHz	NBF	21.772	21.548	2.5	17.5	62.5	0.225
IR-1: L+10	120 MHz	ABF	21.772	15.119	2.5	12.5	65	6.653
IR-1: L+10	120 MHz	VBF	21.772	15.119	2.5	12.5	65	6.653

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	120 MHz	DFPBF	21.772	14.405	7.5	17.5	37.5	7.367
IR-1: L+10	140 MHz	SVD	21.772	16.837	6.429	15	42.857	4.935
IR-1: L+10	140 MHz	AED	21.772	16.939	6.429	19.286	36.429	4.833
IR-1: L+10	140 MHz	ModAED	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	ModAEDS	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L+10	140 MHz	VAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L+10	140 MHz	DFPAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L+10	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L+10	140 MHz	SC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	VSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	DFPSC	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: L+10	140 MHz	NSSC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: L+10	140 MHz	NBF	21.772	19.184	0	15	60	2.588
IR-1: L+10	140 MHz	ABF	21.772	19.082	0	17.143	64.286	2.691
IR-1: L+10	140 MHz	VBF	21.772	19.082	0	17.143	64.286	2.691
IR-1: L+10	140 MHz	DFPBF	21.772	16.939	6.429	17.143	36.429	4.833
IR-1: L+10	150 MHz	SVD	21.772	16.571	0	18	36	5.201
IR-1: L+10	150 MHz	AED	21.772	17.143	8	18	36	4.629
IR-1: L+10	150 MHz	ModAED	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	ModAEDS	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	AXIS	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	VAXIS	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	DFPAXIS	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	NSAXIS	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	SC	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	VSC	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	DFPSC	21.772	16.667	8	18	36	5.105
IR-1: L+10	150 MHz	NSSC	21.772	16.667	8	18	36	5.105

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	150 MHz	CC	21.772	16.19	8	18	36	5.582
IR-1: L+10	150 MHz	NBF	21.772	20.19	2	14	62	1.582
IR-1: L+10	150 MHz	ABF	21.772	15.905	0	14	38	5.867
IR-1: L+10	150 MHz	VBF	21.772	15.905	0	14	38	5.867
IR-1: L+10	150 MHz	DFPBF	21.772	13.238	0	16	22	8.534
IR-1: L+10	160 MHz	SVD	21.772	15	0	15	37.5	6.772
IR-1: L+10	160 MHz	AED	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	ModAED	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	ModAEDS	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	AXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	VAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	DFPAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	NSAXIS	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	SC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	VSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	DFPSC	21.772	15.179	7.5	18.75	22.5	6.594
IR-1: L+10	160 MHz	NSSC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: L+10	160 MHz	NBF	21.772	16.518	3.75	11.25	60	5.254
IR-1: L+10	160 MHz	ABF	21.772	19.732	1.875	18.75	39.375	2.04
IR-1: L+10	160 MHz	VBF	21.772	19.732	1.875	18.75	39.375	2.04
IR-1: L+10	160 MHz	DFPBF	21.772	NaN	1.875	NaN	39.375	NaN
IR-1: L+10	180 MHz	SVD	21.772	14.603	3.333	15	28.333	7.169
IR-1: L+10	180 MHz	AED	21.772	16.825	6.667	18.333	36.667	4.947
IR-1: L+10	180 MHz	ModAED	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	ModAEDS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	AXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	VAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	DFPAXIS	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	NSAXIS	21.772	16.587	6.667	16.667	36.667	5.185

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: L+10	180 MHz	SC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	VSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	DFPSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	NSSC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: L+10	180 MHz	NBF	21.772	18.333	5	13.333	63.333	3.439
IR-1: L+10	180 MHz	ABF	21.772	19.603	1.667	15	95	2.169
IR-1: L+10	180 MHz	VBF	21.772	19.603	1.667	15	95	2.169
IR-1: L+10	180 MHz	DFPBF	21.772	NaN	0	NaN	91.667	NaN
IR-1: L+10	200 MHz	SVD	21.772	16.286	1.5	16.5	33	5.486
IR-1: L+10	200 MHz	AED	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	ModAED	21.772	13.5	7.5	15	19.5	8.272
IR-1: L+10	200 MHz	ModAEDS	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L+10	200 MHz	AXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	VAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	DFPAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	NSAXIS	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	SC	21.772	13.5	7.5	15	19.5	8.272
IR-1: L+10	200 MHz	VSC	21.772	13.5	7.5	15	19.5	8.272
IR-1: L+10	200 MHz	DFPSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	NSSC	21.772	12.929	7.5	15	19.5	8.844
IR-1: L+10	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: L+10	200 MHz	NBF	21.772	20.286	1.5	13.5	63	1.486
IR-1: L+10	200 MHz	ABF	21.772	76.5	6	66	136.5	54.728
IR-1: L+10	200 MHz	VBF	21.772	76.5	6	66	136.5	54.728
IR-1: L+10	200 MHz	DFPBF	21.772	43.5	1.5	33	145.5	21.728
IR-1: Random Start	20 MHz	AED	21.772	125.714	0	135	300	103.942
IR-1: Random Start	20 MHz	ModAED	21.772	189.286	15	195	405	167.514
IR-1: Random Start	20 MHz	ModAEDS	21.772	189.286	15	195	405	167.514
IR-1: Random Start	20 MHz	AXIS	21.772	15	0	15	30	6.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	20 MHz	VAXIS	21.772	75	15	45	195	53.228
IR-1: Random Start	20 MHz	DFPAXIS	21.772	59.286	15	15	195	37.514
IR-1: Random Start	20 MHz	NSAXIS	21.772	15	0	15	30	6.772
IR-1: Random Start	20 MHz	SC	21.772	125.714	0	135	300	103.942
IR-1: Random Start	20 MHz	VSC	21.772	125.714	0	135	300	103.942
IR-1: Random Start	20 MHz	DFPSC	21.772	107.857	0	60	300	86.085
IR-1: Random Start	20 MHz	NSSC	21.772	125.714	0	135	300	103.942
IR-1: Random Start	20 MHz	CC	21.772	14.286	0	15	30	7.486
IR-1: Random Start	20 MHz	NBF	21.772	33.571	0	30	90	11.799
IR-1: Random Start	20 MHz	ABF	21.772	112.143	0	135	195	90.371
IR-1: Random Start	20 MHz	VBF	21.772	112.143	0	135	195	90.371
IR-1: Random Start	20 MHz	DFPBF	21.772	15	0	15	30	6.772
IR-1: Random Start	40 MHz	AED	21.772	77.143	7.5	37.5	225	55.371
IR-1: Random Start	40 MHz	ModAED	21.772	134.286	22.5	112.5	315	112.514
IR-1: Random Start	40 MHz	ModAEDS	21.772	132.143	22.5	112.5	315	110.371
IR-1: Random Start	40 MHz	AXIS	21.772	15.714	7.5	15	37.5	6.058
IR-1: Random Start	40 MHz	VAXIS	21.772	40.357	7.5	22.5	150	18.585
IR-1: Random Start	40 MHz	DFPAXIS	21.772	40.357	7.5	22.5	150	18.585
IR-1: Random Start	40 MHz	NSAXIS	21.772	15.714	7.5	15	37.5	6.058
IR-1: Random Start	40 MHz	SC	21.772	109.643	0	105	232.5	87.871
IR-1: Random Start	40 MHz	VSC	21.772	109.643	0	105	232.5	87.871
IR-1: Random Start	40 MHz	DFPSC	21.772	76.071	7.5	37.5	225	54.299
IR-1: Random Start	40 MHz	NSSC	21.772	109.643	0	105	232.5	87.871
IR-1: Random Start	40 MHz	CC	21.772	15	7.5	15	37.5	6.772
IR-1: Random Start	40 MHz	NBF	21.772	23.571	7.5	22.5	67.5	1.799
IR-1: Random Start	40 MHz	ABF	21.772	43.214	7.5	37.5	97.5	21.442
IR-1: Random Start	40 MHz	VBF	21.772	43.214	7.5	37.5	97.5	21.442
IR-1: Random Start	40 MHz	DFPBF	21.772	15.357	7.5	15	37.5	6.415
IR-1: Random Start	60 MHz	AED	21.772	136.905	10	165	265	115.133
IR-1: Random Start	60 MHz	ModAED	21.772	131.667	25	105	280	109.895

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	60 MHz	ModAEDS	21.772	131.667	25	105	280	109.895
IR-1: Random Start	60 MHz	AXIS	21.772	15.952	5	15	35	5.82
IR-1: Random Start	60 MHz	VAXIS	21.772	65	5	65	160	43.228
IR-1: Random Start	60 MHz	DFPAXIS	21.772	58.095	5	60	115	36.323
IR-1: Random Start	60 MHz	NSAXIS	21.772	15.952	5	15	35	5.82
IR-1: Random Start	60 MHz	SC	21.772	121.905	0	110	265	100.133
IR-1: Random Start	60 MHz	VSC	21.772	121.905	0	110	265	100.133
IR-1: Random Start	60 MHz	DFPSC	21.772	131.19	0	145	265	109.418
IR-1: Random Start	60 MHz	NSSC	21.772	121.905	0	110	265	100.133
IR-1: Random Start	60 MHz	CC	21.772	16.19	5	15	35	5.582
IR-1: Random Start	60 MHz	NBF	21.772	22.143	0	20	60	0.371
IR-1: Random Start	60 MHz	ABF	21.772	35.714	0	20	75	13.942
IR-1: Random Start	60 MHz	VBF	21.772	35.714	0	20	75	13.942
IR-1: Random Start	60 MHz	DFPBF	21.772	15.952	5	15	35	5.82
IR-1: Random Start	80 MHz	AED	21.772	59.286	3.75	22.5	206.25	37.514
IR-1: Random Start	80 MHz	ModAED	21.772	113.036	3.75	90	315	91.264
IR-1: Random Start	80 MHz	ModAEDS	21.772	111.607	3.75	90	315	89.835
IR-1: Random Start	80 MHz	AXIS	21.772	16.607	7.5	18.75	37.5	5.165
IR-1: Random Start	80 MHz	VAXIS	21.772	40.536	3.75	18.75	153.75	18.764
IR-1: Random Start	80 MHz	DFPAXIS	21.772	40.536	3.75	18.75	153.75	18.764
IR-1: Random Start	80 MHz	NSAXIS	21.772	16.607	7.5	18.75	37.5	5.165
IR-1: Random Start	80 MHz	SC	21.772	90.893	0	75	236.25	69.121
IR-1: Random Start	80 MHz	VSC	21.772	90.893	0	75	236.25	69.121
IR-1: Random Start	80 MHz	DFPSC	21.772	75	3.75	37.5	225	53.228
IR-1: Random Start	80 MHz	NSSC	21.772	97.143	0	90	236.25	75.371
IR-1: Random Start	80 MHz	CC	21.772	16.071	7.5	18.75	37.5	5.701
IR-1: Random Start	80 MHz	NBF	21.772	20.536	0	18.75	71.25	1.236
IR-1: Random Start	80 MHz	ABF	21.772	34.107	0	37.5	63.75	12.335
IR-1: Random Start	80 MHz	VBF	21.772	34.107	0	37.5	63.75	12.335
IR-1: Random Start	80 MHz	DFPBF	21.772	16.786	3.75	18.75	37.5	4.986

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	100 MHz	AED	21.772	64.286	6	30	258	42.514
IR-1: Random Start	100 MHz	ModAED	21.772	138.714	3	135	321	116.942
IR-1: Random Start	100 MHz	ModAEDS	21.772	149.571	3	147	321	127.799
IR-1: Random Start	100 MHz	AXIS	21.772	16	6	18	36	5.772
IR-1: Random Start	100 MHz	VAXIS	21.772	58.571	6	33	159	36.799
IR-1: Random Start	100 MHz	DFPAXIS	21.772	58.571	6	33	159	36.799
IR-1: Random Start	100 MHz	NSAXIS	21.772	16	6	18	36	5.772
IR-1: Random Start	100 MHz	SC	21.772	117.286	3	114	303	95.514
IR-1: Random Start	100 MHz	VSC	21.772	117.286	3	114	303	95.514
IR-1: Random Start	100 MHz	DFPSC	21.772	75.286	6	51	258	53.514
IR-1: Random Start	100 MHz	NSSC	21.772	114.143	3	111	303	92.371
IR-1: Random Start	100 MHz	CC	21.772	16.857	6	18	36	4.915
IR-1: Random Start	100 MHz	NBF	21.772	21.714	6	21	36	0.058
IR-1: Random Start	100 MHz	ABF	21.772	20.571	0	21	48	1.201
IR-1: Random Start	100 MHz	VBF	21.772	20.571	0	21	48	1.201
IR-1: Random Start	100 MHz	DFPBF	21.772	15.571	6	18	36	6.201
IR-1: Random Start	120 MHz	AED	21.772	66.071	0	55	197.5	44.299
IR-1: Random Start	120 MHz	ModAED	21.772	97.5	0	87.5	272.5	75.728
IR-1: Random Start	120 MHz	ModAEDS	21.772	101.905	0	92.5	272.5	80.133
IR-1: Random Start	120 MHz	AXIS	21.772	14.286	7.5	15	37.5	7.486
IR-1: Random Start	120 MHz	VAXIS	21.772	68.69	10	72.5	147.5	46.918
IR-1: Random Start	120 MHz	DFPAXIS	21.772	68.69	10	72.5	147.5	46.918
IR-1: Random Start	120 MHz	NSAXIS	21.772	14.286	7.5	15	37.5	7.486
IR-1: Random Start	120 MHz	SC	21.772	84.048	0	65	222.5	62.275
IR-1: Random Start	120 MHz	VSC	21.772	84.048	0	65	222.5	62.275
IR-1: Random Start	120 MHz	DFPSC	21.772	86.905	0	67.5	197.5	65.133
IR-1: Random Start	120 MHz	NSSC	21.772	81.786	0	65	222.5	60.014
IR-1: Random Start	120 MHz	CC	21.772	13.571	7.5	15	37.5	8.201
IR-1: Random Start	120 MHz	NBF	21.772	19.524	0	17.5	52.5	2.248
IR-1: Random Start	120 MHz	ABF	21.772	11.071	0	10	37.5	10.706

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	120 MHz	VBF	21.772	11.071	0	10	37.5	10.701
IR-1: Random Start	120 MHz	DFPBF	21.772	15.119	7.5	17.5	37.5	6.653
IR-1: Random Start	140 MHz	AED	21.772	90.816	2.143	66.429	250.714	69.044
IR-1: Random Start	140 MHz	ModAED	21.772	113.571	8.571	100.714	308.571	91.799
IR-1: Random Start	140 MHz	ModAEDS	21.772	113.571	8.571	100.714	308.571	91.799
IR-1: Random Start	140 MHz	AXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Random Start	140 MHz	VAXIS	21.772	47.959	6.429	23.571	135	26.187
IR-1: Random Start	140 MHz	DFPAXIS	21.772	47.959	6.429	23.571	135	26.187
IR-1: Random Start	140 MHz	NSAXIS	21.772	17.653	6.429	19.286	36.429	4.119
IR-1: Random Start	140 MHz	SC	21.772	113.571	8.571	100.714	308.571	91.799
IR-1: Random Start	140 MHz	VSC	21.772	113.571	8.571	100.714	308.571	91.799
IR-1: Random Start	140 MHz	DFPSC	21.772	94.796	2.143	100.714	263.571	73.024
IR-1: Random Start	140 MHz	NSSC	21.772	102.857	2.143	92.143	308.571	81.085
IR-1: Random Start	140 MHz	CC	21.772	18.061	6.429	19.286	36.429	3.711
IR-1: Random Start	140 MHz	NBF	21.772	15.714	0	15	51.429	6.058
IR-1: Random Start	140 MHz	ABF	21.772	17.041	2.143	15	45	4.731
IR-1: Random Start	140 MHz	VBF	21.772	17.041	2.143	15	45	4.731
IR-1: Random Start	140 MHz	DFPBF	21.772	16.122	6.429	17.143	36.429	5.65
IR-1: Random Start	150 MHz	AED	21.772	50.667	0	18	202	28.895
IR-1: Random Start	150 MHz	ModAED	21.772	122.952	6	126	258	101.18
IR-1: Random Start	150 MHz	ModAEDS	21.772	119.333	6	112	258	97.561
IR-1: Random Start	150 MHz	AXIS	21.772	16.476	8	18	36	5.296
IR-1: Random Start	150 MHz	VAXIS	21.772	42.286	8	20	124	20.514
IR-1: Random Start	150 MHz	DFPAXIS	21.772	44	8	20	124	22.228
IR-1: Random Start	150 MHz	NSAXIS	21.772	16.476	8	18	36	5.296
IR-1: Random Start	150 MHz	SC	21.772	116.667	6	102	258	94.895
IR-1: Random Start	150 MHz	VSC	21.772	116.667	6	102	258	94.895
IR-1: Random Start	150 MHz	DFPSC	21.772	112.476	6	116	224	90.704
IR-1: Random Start	150 MHz	NSSC	21.772	116.667	6	102	258	94.895
IR-1: Random Start	150 MHz	CC	21.772	16.19	8	18	36	5.582

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	150 MHz	NBF	21.772	23.81	2	20	68	2.037
IR-1: Random Start	150 MHz	ABF	21.772	20.667	2	18	44	1.105
IR-1: Random Start	150 MHz	VBF	21.772	20.667	2	18	44	1.105
IR-1: Random Start	150 MHz	DFPBF	21.772	17.905	4	18	60	3.867
IR-1: Random Start	160 MHz	AED	21.772	89.732	3.75	78.75	230.625	67.96
IR-1: Random Start	160 MHz	ModAED	21.772	91.518	11.25	73.125	234.375	69.746
IR-1: Random Start	160 MHz	ModAEDS	21.772	91.518	11.25	73.125	234.375	69.746
IR-1: Random Start	160 MHz	AXIS	21.772	14.911	7.5	16.875	18.75	6.861
IR-1: Random Start	160 MHz	VAXIS	21.772	41.607	0	18.75	114.375	19.835
IR-1: Random Start	160 MHz	DFPAXIS	21.772	41.607	0	18.75	114.375	19.835
IR-1: Random Start	160 MHz	NSAXIS	21.772	14.911	7.5	16.875	18.75	6.861
IR-1: Random Start	160 MHz	SC	21.772	94.375	11.25	97.5	213.75	72.603
IR-1: Random Start	160 MHz	VSC	21.772	94.375	11.25	97.5	213.75	72.603
IR-1: Random Start	160 MHz	DFPSC	21.772	93.75	9.375	97.5	213.75	71.978
IR-1: Random Start	160 MHz	NSSC	21.772	96.25	11.25	97.5	213.75	74.478
IR-1: Random Start	160 MHz	CC	21.772	14.821	7.5	16.875	18.75	6.951
IR-1: Random Start	160 MHz	NBF	21.772	17.232	1.875	15	54.375	4.54
IR-1: Random Start	160 MHz	ABF	21.772	15.893	3.75	15	35.625	5.879
IR-1: Random Start	160 MHz	VBF	21.772	15.893	3.75	15	35.625	5.879
IR-1: Random Start	160 MHz	DFPBF	21.772	18.393	1.875	16.875	37.5	3.379
IR-1: Random Start	180 MHz	AED	21.772	94.683	8.333	73.333	245	72.91
IR-1: Random Start	180 MHz	ModAED	21.772	111.905	5	95	281.667	90.133
IR-1: Random Start	180 MHz	ModAEDS	21.772	113.81	5	95	281.667	92.037
IR-1: Random Start	180 MHz	AXIS	21.772	15.556	6.667	16.667	21.667	6.217
IR-1: Random Start	180 MHz	VAXIS	21.772	63.651	6.667	66.667	128.333	41.879
IR-1: Random Start	180 MHz	DFPAXIS	21.772	63.651	6.667	66.667	128.333	41.879
IR-1: Random Start	180 MHz	NSAXIS	21.772	15.556	6.667	16.667	21.667	6.217
IR-1: Random Start	180 MHz	SC	21.772	111.27	5	91.667	281.667	89.498
IR-1: Random Start	180 MHz	VSC	21.772	111.27	5	91.667	281.667	89.498
IR-1: Random Start	180 MHz	DFPSC	21.772	96.27	3.333	80	281.667	74.498

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-1: Random Start	180 MHz	NSSC	21.772	111.27	5	91.667	281.667	89.498
IR-1: Random Start	180 MHz	CC	21.772	16.587	6.667	16.667	36.667	5.185
IR-1: Random Start	180 MHz	NBF	21.772	16.032	0	13.333	55	5.74
IR-1: Random Start	180 MHz	ABF	21.772	23.73	3.333	18.333	71.667	1.958
IR-1: Random Start	180 MHz	VBF	21.772	23.73	3.333	18.333	71.667	1.958
IR-1: Random Start	180 MHz	DFPBF	21.772	NaN	0	NaN	56.667	NaN
IR-1: Random Start	200 MHz	AED	21.772	71.143	9	66	201	49.371
IR-1: Random Start	200 MHz	ModAED	21.772	85.714	0	69	295.5	63.942
IR-1: Random Start	200 MHz	ModAEDS	21.772	85.714	0	69	295.5	63.942
IR-1: Random Start	200 MHz	AXIS	21.772	13.429	7.5	15	19.5	8.344
IR-1: Random Start	200 MHz	VAXIS	21.772	43.214	1.5	21	126	21.442
IR-1: Random Start	200 MHz	DFPAXIS	21.772	43.214	1.5	21	126	21.442
IR-1: Random Start	200 MHz	NSAXIS	21.772	13.429	7.5	15	19.5	8.344
IR-1: Random Start	200 MHz	SC	21.772	85.714	0	69	295.5	63.942
IR-1: Random Start	200 MHz	VSC	21.772	85.714	0	69	295.5	63.942
IR-1: Random Start	200 MHz	DFPSC	21.772	95.571	9	72	219	73.799
IR-1: Random Start	200 MHz	NSSC	21.772	85.714	0	69	295.5	63.942
IR-1: Random Start	200 MHz	CC	21.772	14.071	7.5	16.5	19.5	7.701
IR-1: Random Start	200 MHz	NBF	21.772	17.286	7.5	15	61.5	4.486
IR-1: Random Start	200 MHz	ABF	21.772	48.286	4.5	27	150	26.514
IR-1: Random Start	200 MHz	VBF	21.772	48.286	4.5	27	150	26.514
IR-1: Random Start	200 MHz	DFPBF	21.772	64.286	7.5	69	172.5	42.514
IR-2: Ideal	20 MHz	SVD	21.772	17.857	0	15	30	3.915
IR-2: Ideal	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	ModAED	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	DFPAXIS	21.772	20	0	15	30	1.772
IR-2: Ideal	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	VSC	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: Ideal	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: Ideal	20 MHz	NBF	21.772	16.429	0	15	30	5.344
IR-2: Ideal	20 MHz	ABF	21.772	17.857	0	15	30	3.915
IR-2: Ideal	20 MHz	VBF	21.772	17.857	0	15	30	3.915
IR-2: Ideal	20 MHz	DFPBF	21.772	17.857	0	15	30	3.915
IR-2: Ideal	40 MHz	SVD	21.772	21.429	0	22.5	37.5	0.344
IR-2: Ideal	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: Ideal	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: Ideal	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: Ideal	40 MHz	AXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	VAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: Ideal	40 MHz	NSAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	SC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	VSC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	DFPSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: Ideal	40 MHz	NSSC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Ideal	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Ideal	40 MHz	NBF	21.772	19.643	0	22.5	37.5	2.129
IR-2: Ideal	40 MHz	ABF	21.772	21.429	0	22.5	37.5	0.344
IR-2: Ideal	40 MHz	VBF	21.772	21.429	0	22.5	37.5	0.344
IR-2: Ideal	40 MHz	DFPBF	21.772	22.143	0	22.5	37.5	0.371
IR-2: Ideal	60 MHz	SVD	21.772	22.381	5	25	35	0.609
IR-2: Ideal	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: Ideal	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: Ideal	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	60 MHz	AXIS	21.772	20.476	5	20	35	1.296
IR-2: Ideal	60 MHz	VAXIS	21.772	20.476	5	20	35	1.296
IR-2: Ideal	60 MHz	DFPAXIS	21.772	20.952	5	25	35	0.82
IR-2: Ideal	60 MHz	NSAXIS	21.772	20.714	5	20	35	1.058
IR-2: Ideal	60 MHz	SC	21.772	20.714	5	20	35	1.058
IR-2: Ideal	60 MHz	VSC	21.772	20.714	5	20	35	1.058
IR-2: Ideal	60 MHz	DFPSC	21.772	20.714	5	20	35	1.058
IR-2: Ideal	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: Ideal	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: Ideal	60 MHz	NBF	21.772	19.524	0	20	35	2.248
IR-2: Ideal	60 MHz	ABF	21.772	23.333	10	25	35	1.561
IR-2: Ideal	60 MHz	VBF	21.772	23.333	10	25	35	1.561
IR-2: Ideal	60 MHz	DFPBF	21.772	24.286	10	25	35	2.514
IR-2: Ideal	80 MHz	SVD	21.772	17.321	3.75	18.75	33.75	4.451
IR-2: Ideal	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: Ideal	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Ideal	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Ideal	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	SC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	VSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	DFPSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	NSSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Ideal	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: Ideal	80 MHz	NBF	21.772	18.75	3.75	18.75	33.75	3.022
IR-2: Ideal	80 MHz	ABF	21.772	19.286	3.75	18.75	48.75	2.486
IR-2: Ideal	80 MHz	VBF	21.772	19.286	3.75	18.75	48.75	2.486
IR-2: Ideal	80 MHz	DFPBF	21.772	17.679	3.75	15	30	4.094

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	100 MHz	SVD	21.772	17.286	3	18	30	4.486
IR-2: Ideal	100 MHz	AED	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	ModAED	21.772	21.857	6	24	33	0.085
IR-2: Ideal	100 MHz	ModAEDS	21.772	21.857	6	24	33	0.085
IR-2: Ideal	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	SC	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	VSC	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	NSSC	21.772	21.714	6	24	33	0.058
IR-2: Ideal	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: Ideal	100 MHz	NBF	21.772	20.714	3	21	33	1.058
IR-2: Ideal	100 MHz	ABF	21.772	20.714	3	15	72	1.058
IR-2: Ideal	100 MHz	VBF	21.772	20.714	3	15	72	1.058
IR-2: Ideal	100 MHz	DFPBF	21.772	15.571	3	15	33	6.201
IR-2: Ideal	120 MHz	SVD	21.772	19.524	5	20	32.5	2.248
IR-2: Ideal	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: Ideal	120 MHz	ModAED	21.772	17.143	5	15	35	4.629
IR-2: Ideal	120 MHz	ModAEDS	21.772	17.143	5	15	35	4.629
IR-2: Ideal	120 MHz	AXIS	21.772	17.738	5	15	35	4.034
IR-2: Ideal	120 MHz	VAXIS	21.772	17.738	5	15	35	4.034
IR-2: Ideal	120 MHz	DFPAXIS	21.772	17.738	5	15	35	4.034
IR-2: Ideal	120 MHz	NSAXIS	21.772	17.738	5	15	35	4.034
IR-2: Ideal	120 MHz	SC	21.772	17.143	5	15	35	4.629
IR-2: Ideal	120 MHz	VSC	21.772	17.143	5	15	35	4.629
IR-2: Ideal	120 MHz	DFPSC	21.772	17.738	5	15	35	4.034
IR-2: Ideal	120 MHz	NSSC	21.772	17.143	5	15	35	4.629
IR-2: Ideal	120 MHz	CC	21.772	16.19	5	15	27.5	5.582

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	120 MHz	NBF	21.772	22.262	12.5	20	42.5	0.49
IR-2: Ideal	120 MHz	ABF	21.772	17.262	0	17.5	35	4.51
IR-2: Ideal	120 MHz	VBF	21.772	17.262	0	17.5	35	4.51
IR-2: Ideal	120 MHz	DFPBF	21.772	15.595	0	15	32.5	6.177
IR-2: Ideal	140 MHz	SVD	21.772	16.633	2.143	17.143	34.286	5.139
IR-2: Ideal	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: Ideal	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Ideal	140 MHz	ModAEDS	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Ideal	140 MHz	AXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Ideal	140 MHz	VAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Ideal	140 MHz	DFPAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Ideal	140 MHz	NSAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Ideal	140 MHz	SC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Ideal	140 MHz	VSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Ideal	140 MHz	DFPSC	21.772	18.776	4.286	17.143	34.286	2.997
IR-2: Ideal	140 MHz	NSSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Ideal	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: Ideal	140 MHz	NBF	21.772	23.878	2.143	21.429	57.857	2.105
IR-2: Ideal	140 MHz	ABF	21.772	16.327	0	17.143	27.857	5.446
IR-2: Ideal	140 MHz	VBF	21.772	16.327	0	17.143	27.857	5.446
IR-2: Ideal	140 MHz	DFPBF	21.772	18.98	4.286	21.429	34.286	2.793
IR-2: Ideal	150 MHz	SVD	21.772	16.381	0	18	24	5.391
IR-2: Ideal	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: Ideal	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: Ideal	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963
IR-2: Ideal	150 MHz	AXIS	21.772	18.476	4	18	34	3.296
IR-2: Ideal	150 MHz	VAXIS	21.772	18.476	4	18	34	3.296
IR-2: Ideal	150 MHz	DFPAXIS	21.772	18.476	4	18	34	3.296
IR-2: Ideal	150 MHz	NSAXIS	21.772	18.476	4	18	34	3.296
IR-2: Ideal	150 MHz	SC	21.772	19.81	4	20	34	1.963

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	150 MHz	VSC	21.772	19.81	4	20	34	1.963
IR-2: Ideal	150 MHz	DFPSC	21.772	18.476	4	18	34	3.296
IR-2: Ideal	150 MHz	NSSC	21.772	19.81	4	20	34	1.963
IR-2: Ideal	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: Ideal	150 MHz	NBF	21.772	20.095	2	22	36	1.677
IR-2: Ideal	150 MHz	ABF	21.772	22.571	6	22	42	0.799
IR-2: Ideal	150 MHz	VBF	21.772	22.571	6	22	42	0.799
IR-2: Ideal	150 MHz	DFPBF	21.772	18.571	4	22	34	3.201
IR-2: Ideal	160 MHz	SVD	21.772	20.536	5.625	20.625	35.625	1.236
IR-2: Ideal	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: Ideal	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	AXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Ideal	160 MHz	VAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Ideal	160 MHz	DFPAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Ideal	160 MHz	NSAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Ideal	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	DFPSC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Ideal	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Ideal	160 MHz	NBF	21.772	23.036	3.75	22.5	76.875	1.264
IR-2: Ideal	160 MHz	ABF	21.772	20.357	7.5	18.75	35.625	1.415
IR-2: Ideal	160 MHz	VBF	21.772	20.357	7.5	18.75	35.625	1.415
IR-2: Ideal	160 MHz	DFPBF	21.772	19.821	3.75	18.75	35.625	1.951
IR-2: Ideal	180 MHz	SVD	21.772	18.175	5	18.333	33.333	3.598
IR-2: Ideal	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: Ideal	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: Ideal	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: Ideal	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Ideal	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Ideal	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Ideal	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Ideal	180 MHz	SC	21.772	21.508	5	23.333	35	0.264
IR-2: Ideal	180 MHz	VSC	21.772	21.508	5	23.333	35	0.264
IR-2: Ideal	180 MHz	DFPSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: Ideal	180 MHz	NSSC	21.772	21.508	5	23.333	35	0.264
IR-2: Ideal	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: Ideal	180 MHz	NBF	21.772	21.429	1.667	20	53.333	0.344
IR-2: Ideal	180 MHz	ABF	21.772	23.73	3.333	21.667	48.333	1.958
IR-2: Ideal	180 MHz	VBF	21.772	23.73	3.333	21.667	48.333	1.958
IR-2: Ideal	180 MHz	DFPBF	21.772	24.444	1.667	21.667	58.333	2.672
IR-2: Ideal	200 MHz	SVD	21.772	21.357	4.5	19.5	64.5	0.415
IR-2: Ideal	200 MHz	AED	21.772	19.786	4.5	22.5	34.5	1.986
IR-2: Ideal	200 MHz	ModAED	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Ideal	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Ideal	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	DFPAXIS	21.772	20.143	4.5	19.5	34.5	1.629
IR-2: Ideal	200 MHz	NSAXIS	21.772	19.857	4.5	19.5	34.5	1.915
IR-2: Ideal	200 MHz	SC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	VSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	DFPSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	NSSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Ideal	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Ideal	200 MHz	NBF	21.772	23.357	0	24	57	1.585
IR-2: Ideal	200 MHz	ABF	21.772	53.071	0	52.5	133.5	31.299
IR-2: Ideal	200 MHz	VBF	21.772	53.071	0	52.5	133.5	31.299
IR-2: Ideal	200 MHz	DFPBF	21.772	52.357	0	48	145.5	30.585
IR-2: Narrowband	20 MHz	SVD	21.772	22.857	0	0	60	1.085

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	20 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	20 MHz	CC	21.772	5.178	0	3.749	15	16.594
IR-2: Narrowband	20 MHz	NBF	21.772	122.857	0	60	300	101.085
IR-2: Narrowband	20 MHz	ABF	21.772	68.571	0	0	420	46.799
IR-2: Narrowband	20 MHz	VBF	21.772	68.571	0	0	420	46.799
IR-2: Narrowband	20 MHz	DFPBF	21.772	NaN	0	NaN	60	NaN
IR-2: Narrowband	40 MHz	SVD	21.772	102.857	0	60	1020	81.085
IR-2: Narrowband	40 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	CC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	40 MHz	CC	21.772	2.723	0	1.874	7.5	19.049
IR-2: Narrowband	40 MHz	NBF	21.772	148.571	0	60	1440	126.799

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	40 MHz	ABF	21.772	334.286	0	60	1200	312.514
IR-2: Narrowband	40 MHz	VBF	21.772	334.286	0	60	1200	312.514
IR-2: Narrowband	40 MHz	DFPBF	21.772	NaN	0	NaN	180	NaN
IR-2: Narrowband	60 MHz	SVD	21.772	254.286	0	60	2220	232.514
IR-2: Narrowband	60 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	60 MHz	CC	21.772	1.845	0	1.25	5	19.927
IR-2: Narrowband	60 MHz	NBF	21.772	871.429	0	60	2580	849.656
IR-2: Narrowband	60 MHz	ABF	21.772	362.857	0	60	1980	341.085
IR-2: Narrowband	60 MHz	VBF	21.772	362.857	0	60	1980	341.085
IR-2: Narrowband	60 MHz	DFPBF	21.772	20	0	0	60	1.772
IR-2: Narrowband	80 MHz	SVD	21.772	94.286	0	0	840	72.514
IR-2: Narrowband	80 MHz	AED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	ModAED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	ModAEDS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	AXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	VAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	DFPAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	NSAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	SC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	VSC	21.772	14.286	0	0	60	7.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	80 MHz	DFPSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	NSSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	80 MHz	CC	21.772	1.35	0	0.937	3.75	20.422
IR-2: Narrowband	80 MHz	NBF	21.772	860	0	60	3720	838.228
IR-2: Narrowband	80 MHz	ABF	21.772	1208.571	60	900	3480	1186.799
IR-2: Narrowband	80 MHz	VBF	21.772	1208.571	60	900	3480	1186.799
IR-2: Narrowband	80 MHz	DFPBF	21.772	1142.857	0	480	4500	1121.085
IR-2: Narrowband	100 MHz	SVD	21.772	245.714	0	0	2880	223.942
IR-2: Narrowband	100 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	100 MHz	CC	21.772	1.1	0	0.9	3	20.672
IR-2: Narrowband	100 MHz	NBF	21.772	597.143	0	0	4860	575.371
IR-2: Narrowband	100 MHz	ABF	21.772	200	0	60	3060	178.228
IR-2: Narrowband	100 MHz	VBF	21.772	200	0	60	3060	178.228
IR-2: Narrowband	100 MHz	DFPBF	21.772	NaN	0	NaN	60	NaN
IR-2: Narrowband	120 MHz	SVD	21.772	51.429	0	60	480	29.656
IR-2: Narrowband	120 MHz	AED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	ModAED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	ModAEDS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	AXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	VAXIS	21.772	14.286	0	0	60	7.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	120 MHz	DFPAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	NSAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	SC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	VSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	DFPSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	NSSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	120 MHz	CC	21.772	0.858	0	0.521	2.5	20.914
IR-2: Narrowband	120 MHz	NBF	21.772	291.429	0	60	5520	269.656
IR-2: Narrowband	120 MHz	ABF	21.772	242.857	0	0	4320	221.085
IR-2: Narrowband	120 MHz	VBF	21.772	242.857	0	0	4320	221.085
IR-2: Narrowband	120 MHz	DFPBF	21.772	25.714	0	0	120	3.942
IR-2: Narrowband	140 MHz	SVD	21.772	34.286	0	60	120	12.514
IR-2: Narrowband	140 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	140 MHz	CC	21.772	0.758	0	0.612	2.143	21.014
IR-2: Narrowband	140 MHz	NBF	21.772	NaN	0	NaN	7080	NaN
IR-2: Narrowband	140 MHz	ABF	21.772	114.286	0	0	1260	92.514
IR-2: Narrowband	140 MHz	VBF	21.772	114.286	0	0	1260	92.514
IR-2: Narrowband	140 MHz	DFPBF	21.772	34.286	0	60	60	12.514
IR-2: Narrowband	150 MHz	SVD	21.772	NaN	0	NaN	120	NaN
IR-2: Narrowband	150 MHz	AED	21.772	14.286	0	0	60	7.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	150 MHz	ModAED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	ModAEDS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	AXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	VAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	DFPAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	NSAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	SC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	VSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	DFPSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	NSSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	150 MHz	CC	21.772	0.714	0	0.533	2	21.058
IR-2: Narrowband	150 MHz	NBF	21.772	NaN	0	NaN	60	NaN
IR-2: Narrowband	150 MHz	ABF	21.772	2302.857	0	2100	5940	2281.085
IR-2: Narrowband	150 MHz	VBF	21.772	2302.857	0	2100	5940	2281.085
IR-2: Narrowband	150 MHz	DFPBF	21.772	1980	0	2100	6840	1958.228
IR-2: Narrowband	160 MHz	SVD	21.772	NaN	0	NaN	360	NaN
IR-2: Narrowband	160 MHz	AED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	ModAED	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	ModAEDS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	AXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	VAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	DFPAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	NSAXIS	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	SC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	VSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	DFPSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	NSSC	21.772	17.143	0	0	60	4.629
IR-2: Narrowband	160 MHz	CC	21.772	0.656	0	0.41	1.875	21.117
IR-2: Narrowband	160 MHz	NBF	21.772	NaN	0	NaN	60	NaN
IR-2: Narrowband	160 MHz	ABF	21.772	1477.143	0	660	7560	1455.351

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	160 MHz	VBF	21.772	1477.143	0	660	7560	1455.371
IR-2: Narrowband	160 MHz	DFPBF	21.772	2825.714	0	2040	8880	2803.942
IR-2: Narrowband	180 MHz	SVD	21.772	NaN	0	NaN	120	NaN
IR-2: Narrowband	180 MHz	AED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	ModAED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	ModAEDS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	AXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	VAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	DFPAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	NSAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	SC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	VSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	DFPSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	NSSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	180 MHz	CC	21.772	0.589	0.046	0.463	1.667	21.184
IR-2: Narrowband	180 MHz	NBF	21.772	NaN	9420	NaN	9420	NaN
IR-2: Narrowband	180 MHz	ABF	21.772	2388.571	0	180	9960	2366.799
IR-2: Narrowband	180 MHz	VBF	21.772	2388.571	0	180	9960	2366.799
IR-2: Narrowband	180 MHz	DFPBF	21.772	2425.714	0	900	9240	2403.942
IR-2: Narrowband	200 MHz	SVD	21.772	NaN	0	NaN	1620	NaN
IR-2: Narrowband	200 MHz	AED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	ModAED	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	ModAEDS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	AXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	VAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	DFPAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	NSAXIS	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	SC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	VSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	DFPSC	21.772	14.286	0	0	60	7.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Narrowband	200 MHz	NSSC	21.772	14.286	0	0	60	7.486
IR-2: Narrowband	200 MHz	CC	21.772	0.527	0	0.337	1.5	21.245
IR-2: Narrowband	200 MHz	NBF	21.772	NaN	NaN	NaN	NaN	NaN
IR-2: Narrowband	200 MHz	ABF	21.772	3697.143	60	3060	11280	3675.371
IR-2: Narrowband	200 MHz	VBF	21.772	3697.143	60	3060	11280	3675.371
IR-2: Narrowband	200 MHz	DFPBF	21.772	NaN	0	NaN	10440	NaN
IR-2: Noisy	20 MHz	SVD	21.772	27.143	0	30	75	5.371
IR-2: Noisy	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	ModAED	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	DFPAXIS	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	VSC	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: Noisy	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: Noisy	20 MHz	NBF	21.772	20.714	0	30	45	1.058
IR-2: Noisy	20 MHz	ABF	21.772	20	0	15	45	1.772
IR-2: Noisy	20 MHz	VBF	21.772	20	0	15	45	1.772
IR-2: Noisy	20 MHz	DFPBF	21.772	20	0	15	60	1.772
IR-2: Noisy	40 MHz	SVD	21.772	30.357	0	30	75	8.585
IR-2: Noisy	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: Noisy	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: Noisy	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: Noisy	40 MHz	AXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	VAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	40 MHz	NSAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	SC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	VSC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	DFPSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: Noisy	40 MHz	NSSC	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: Noisy	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Noisy	40 MHz	NBF	21.772	20	7.5	22.5	37.5	1.772
IR-2: Noisy	40 MHz	ABF	21.772	20.357	7.5	22.5	37.5	1.415
IR-2: Noisy	40 MHz	VBF	21.772	20.357	7.5	22.5	37.5	1.415
IR-2: Noisy	40 MHz	DFPBF	21.772	16.429	0	15	30	5.344
IR-2: Noisy	60 MHz	SVD	21.772	38.333	5	35	90	16.561
IR-2: Noisy	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: Noisy	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: Noisy	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058
IR-2: Noisy	60 MHz	AXIS	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	VAXIS	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	DFPAXIS	21.772	21.19	5	25	35	0.582
IR-2: Noisy	60 MHz	NSAXIS	21.772	20.714	5	20	35	1.058
IR-2: Noisy	60 MHz	SC	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	VSC	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	DFPSC	21.772	20.714	5	20	35	1.058
IR-2: Noisy	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: Noisy	60 MHz	NBF	21.772	21.429	5	25	35	0.344
IR-2: Noisy	60 MHz	ABF	21.772	20.476	5	25	35	1.296
IR-2: Noisy	60 MHz	VBF	21.772	20.476	5	25	35	1.296
IR-2: Noisy	60 MHz	DFPBF	21.772	20.714	5	20	35	1.058
IR-2: Noisy	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: Noisy	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: Noisy	60 MHz	NBF	21.772	21.429	5	25	35	0.344
IR-2: Noisy	60 MHz	ABF	21.772	20.476	5	25	35	1.296
IR-2: Noisy	60 MHz	VBF	21.772	20.476	5	25	35	1.296
IR-2: Noisy	60 MHz	DFPBF	21.772	20.238	5	20	35	1.534
IR-2: Noisy	80 MHz	SVD	21.772	41.607	0	37.5	135	19.835
IR-2: Noisy	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: Noisy	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.954

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Noisy	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	SC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	VSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	DFPSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	NSSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: Noisy	80 MHz	NBF	21.772	20	3.75	22.5	33.75	1.772
IR-2: Noisy	80 MHz	ABF	21.772	18.75	3.75	18.75	33.75	3.022
IR-2: Noisy	80 MHz	VBF	21.772	18.75	3.75	18.75	33.75	3.022
IR-2: Noisy	80 MHz	DFPBF	21.772	19.643	3.75	18.75	37.5	2.129
IR-2: Noisy	100 MHz	SVD	21.772	36.571	0	27	99	14.799
IR-2: Noisy	100 MHz	AED	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	ModAED	21.772	21.857	6	24	33	0.085
IR-2: Noisy	100 MHz	ModAEDS	21.772	21.857	6	24	33	0.085
IR-2: Noisy	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	SC	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	VSC	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	NSSC	21.772	21.714	6	24	33	0.058
IR-2: Noisy	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: Noisy	100 MHz	NBF	21.772	20.857	3	24	33	0.915
IR-2: Noisy	100 MHz	ABF	21.772	19.857	3	21	33	1.915
IR-2: Noisy	100 MHz	VBF	21.772	19.857	3	21	33	1.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	100 MHz	DFPBF	21.772	21.286	3	24	39	0.486
IR-2: Noisy	120 MHz	SVD	21.772	48.095	5	35	125	26.323
IR-2: Noisy	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: Noisy	120 MHz	ModAED	21.772	17.143	5	15	35	4.629
IR-2: Noisy	120 MHz	ModAEDS	21.772	16.905	5	15	35	4.867
IR-2: Noisy	120 MHz	AXIS	21.772	17.738	5	15	35	4.034
IR-2: Noisy	120 MHz	VAXIS	21.772	17.738	5	15	35	4.034
IR-2: Noisy	120 MHz	DFPAXIS	21.772	17.738	5	15	35	4.034
IR-2: Noisy	120 MHz	NSAXIS	21.772	17.738	5	15	35	4.034
IR-2: Noisy	120 MHz	SC	21.772	17.143	5	15	35	4.629
IR-2: Noisy	120 MHz	VSC	21.772	17.143	5	15	35	4.629
IR-2: Noisy	120 MHz	DFPSC	21.772	17.738	5	15	35	4.034
IR-2: Noisy	120 MHz	NSSC	21.772	17.143	5	15	35	4.629
IR-2: Noisy	120 MHz	CC	21.772	16.19	5	15	27.5	5.582
IR-2: Noisy	120 MHz	NBF	21.772	17.143	5	17.5	32.5	4.629
IR-2: Noisy	120 MHz	ABF	21.772	17.619	2.5	17.5	32.5	4.153
IR-2: Noisy	120 MHz	VPF	21.772	17.619	2.5	17.5	32.5	4.153
IR-2: Noisy	120 MHz	DFPBF	21.772	18.095	2.5	17.5	35	3.677
IR-2: Noisy	140 MHz	SVD	21.772	43.878	8.571	30	139.286	22.105
IR-2: Noisy	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: Noisy	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Noisy	140 MHz	ModAEDS	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Noisy	140 MHz	AXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Noisy	140 MHz	VAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Noisy	140 MHz	DFPAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Noisy	140 MHz	NSAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Noisy	140 MHz	SC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Noisy	140 MHz	VSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: Noisy	140 MHz	DFPSC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: Noisy	140 MHz	NSSC	21.772	17.653	4.286	17.143	34.286	4.119

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: Noisy	140 MHz	NBF	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: Noisy	140 MHz	ABF	21.772	20.51	4.286	21.429	34.286	1.262
IR-2: Noisy	140 MHz	VBF	21.772	20.51	4.286	21.429	34.286	1.262
IR-2: Noisy	140 MHz	DFPBF	21.772	19.388	4.286	17.143	34.286	2.384
IR-2: Noisy	150 MHz	SVD	21.772	35.429	4	20	136	13.656
IR-2: Noisy	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: Noisy	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: Noisy	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963
IR-2: Noisy	150 MHz	AXIS	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	VAXIS	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	DFPAXIS	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	NSAXIS	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	SC	21.772	19.81	4	20	34	1.963
IR-2: Noisy	150 MHz	VSC	21.772	19.81	4	20	34	1.963
IR-2: Noisy	150 MHz	DFPSC	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	NSSC	21.772	19.81	4	20	34	1.963
IR-2: Noisy	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: Noisy	150 MHz	NBF	21.772	19.143	4	22	34	2.629
IR-2: Noisy	150 MHz	ABF	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	VBF	21.772	18.476	4	18	34	3.296
IR-2: Noisy	150 MHz	DFPBF	21.772	20.857	0	18	74	0.915
IR-2: Noisy	160 MHz	SVD	21.772	43.482	1.875	24.375	133.125	21.71
IR-2: Noisy	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: Noisy	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	AXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Noisy	160 MHz	VAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Noisy	160 MHz	DFPAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: Noisy	160 MHz	NSAXIS	21.772	17.679	3.75	16.875	33.75	4.094

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$\hat{d}_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	DFPSC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: Noisy	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Noisy	160 MHz	NBF	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: Noisy	160 MHz	ABF	21.772	21.161	1.875	18.75	91.875	0.611
IR-2: Noisy	160 MHz	VBF	21.772	21.161	1.875	18.75	91.875	0.611
IR-2: Noisy	160 MHz	DFPBF	21.772	27.321	3.75	16.875	157.5	5.549
IR-2: Noisy	180 MHz	SVD	21.772	38.413	0	30	113.333	16.641
IR-2: Noisy	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: Noisy	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: Noisy	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: Noisy	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Noisy	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Noisy	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Noisy	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: Noisy	180 MHz	SC	21.772	21.508	5	23.333	35	0.264
IR-2: Noisy	180 MHz	VSC	21.772	21.508	5	23.333	35	0.264
IR-2: Noisy	180 MHz	DFPSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: Noisy	180 MHz	NSSC	21.772	21.508	5	23.333	35	0.264
IR-2: Noisy	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: Noisy	180 MHz	NBF	21.772	21.587	6.667	23.333	35	0.185
IR-2: Noisy	180 MHz	ABF	21.772	34.683	0	30	110	12.91
IR-2: Noisy	180 MHz	VBF	21.772	34.683	0	30	110	12.91
IR-2: Noisy	180 MHz	DFPBF	21.772	NaN	1.667	NaN	148.333	NaN
IR-2: Noisy	200 MHz	SVD	21.772	NaN	3	NaN	160.5	NaN
IR-2: Noisy	200 MHz	AED	21.772	19.786	4.5	22.5	34.5	1.986
IR-2: Noisy	200 MHz	ModAED	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Noisy	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Noisy	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Noisy	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Noisy	200 MHz	DFPAXIS	21.772	20.143	4.5	19.5	34.5	1.629
IR-2: Noisy	200 MHz	NSAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Noisy	200 MHz	SC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Noisy	200 MHz	VSC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Noisy	200 MHz	DFPSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Noisy	200 MHz	NSSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: Noisy	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Noisy	200 MHz	NBF	21.772	19.929	4.5	22.5	34.5	1.844
IR-2: Noisy	200 MHz	ABF	21.772	48	1.5	39	148.5	26.228
IR-2: Noisy	200 MHz	VBF	21.772	48	1.5	39	148.5	26.228
IR-2: Noisy	200 MHz	DFPBF	21.772	63.571	1.5	51	159	41.799
IR-2: Fewer Samples	20 MHz	SVD	21.772	17.857	0	15	30	3.915
IR-2: Fewer Samples	20 MHz	AED	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	ModAED	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	ModAEDS	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	AXIS	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	VAXIS	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	DFPAXIS	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	NSAXIS	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	SC	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	VSC	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	DFPSC	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	NSSC	21.772	19.286	0	15	30	2.486
IR-2: Fewer Samples	20 MHz	CC	21.772	18.571	0	15	30	3.201
IR-2: Fewer Samples	20 MHz	NBF	21.772	17.857	0	15	30	3.915
IR-2: Fewer Samples	20 MHz	ABF	21.772	17.857	0	15	30	3.915
IR-2: Fewer Samples	20 MHz	VBF	21.772	17.857	0	15	30	3.915
IR-2: Fewer Samples	20 MHz	DFPBF	21.772	17.857	0	15	30	3.915

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	40 MHz	SVD	21.772	21.429	0	22.5	37.5	0.344
IR-2: Fewer Samples	40 MHz	AED	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	ModAED	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	ModAEDS	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	AXIS	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	VAXIS	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	DFPAXIS	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	NSAXIS	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	SC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	VSC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	DFPSC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	NSSC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Fewer Samples	40 MHz	NBF	21.772	22.857	0	22.5	37.5	1.085
IR-2: Fewer Samples	40 MHz	ABF	21.772	21.786	0	22.5	37.5	0.014
IR-2: Fewer Samples	40 MHz	VBF	21.772	21.786	0	22.5	37.5	0.014
IR-2: Fewer Samples	40 MHz	DFPBF	21.772	22.5	0	22.5	37.5	0.728
IR-2: Fewer Samples	60 MHz	SVD	21.772	22.619	5	25	40	0.847
IR-2: Fewer Samples	60 MHz	AED	21.772	20.476	5	20	35	1.296
IR-2: Fewer Samples	60 MHz	ModAED	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	ModAEDS	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	AXIS	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	VAXIS	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	DFPAXIS	21.772	20.476	5	20	35	1.296
IR-2: Fewer Samples	60 MHz	NSAXIS	21.772	20.476	5	20	35	1.296
IR-2: Fewer Samples	60 MHz	SC	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	VSC	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	DFPSC	21.772	21.429	5	25	35	0.344
IR-2: Fewer Samples	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: Fewer Samples	60 MHz	CC	21.772	22.143	5	25	35	0.374

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	60 MHz	NBF	21.772	23.81	5	25	60	2.037
IR-2: Fewer Samples	60 MHz	ABF	21.772	24.048	10	25	35	2.275
IR-2: Fewer Samples	60 MHz	VBF	21.772	24.048	10	25	35	2.275
IR-2: Fewer Samples	60 MHz	DFPBF	21.772	23.571	10	25	35	1.799
IR-2: Fewer Samples	80 MHz	SVD	21.772	17.857	3.75	18.75	33.75	3.915
IR-2: Fewer Samples	80 MHz	AED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	ModAEDS	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: Fewer Samples	80 MHz	AXIS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	VAXIS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	DFPAXIS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	NSAXIS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	SC	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	VSC	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	DFPSC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: Fewer Samples	80 MHz	NSSC	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: Fewer Samples	80 MHz	CC	21.772	20.714	3.75	22.5	33.75	1.058
IR-2: Fewer Samples	80 MHz	NBF	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: Fewer Samples	80 MHz	ABF	21.772	19.107	3.75	18.75	30	2.665
IR-2: Fewer Samples	80 MHz	VBF	21.772	19.107	3.75	18.75	30	2.665
IR-2: Fewer Samples	80 MHz	DFPBF	21.772	18.75	3.75	18.75	37.5	3.022
IR-2: Fewer Samples	100 MHz	SVD	21.772	20	3	18	54	1.772
IR-2: Fewer Samples	100 MHz	AED	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	ModAED	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	ModAEDS	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	AXIS	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	VAXIS	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	DFPAXIS	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	NSAXIS	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	SC	21.772	20.857	3	24	33	0.915

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	100 MHz	VSC	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	DFPSC	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	NSSC	21.772	20.857	3	24	33	0.915
IR-2: Fewer Samples	100 MHz	CC	21.772	21.286	3	24	33	0.486
IR-2: Fewer Samples	100 MHz	NBF	21.772	20.286	6	21	33	1.486
IR-2: Fewer Samples	100 MHz	ABF	21.772	19	3	18	45	2.772
IR-2: Fewer Samples	100 MHz	VBF	21.772	19	3	18	45	2.772
IR-2: Fewer Samples	100 MHz	DFPBF	21.772	16.286	3	15	33	5.486
IR-2: Fewer Samples	120 MHz	SVD	21.772	35	0	32.5	80	13.228
IR-2: Fewer Samples	120 MHz	AED	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	ModAED	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	ModAEDS	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	AXIS	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	VAXIS	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	DFPAXIS	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	NSAXIS	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	SC	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	VSC	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	DFPSC	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	NSSC	21.772	16.905	5	15	35	4.867
IR-2: Fewer Samples	120 MHz	CC	21.772	16.19	5	15	27.5	5.582
IR-2: Fewer Samples	120 MHz	NBF	21.772	21.548	7.5	20	42.5	0.225
IR-2: Fewer Samples	120 MHz	ABF	21.772	21.19	2.5	17.5	80	0.582
IR-2: Fewer Samples	120 MHz	VBF	21.772	21.19	2.5	17.5	80	0.582
IR-2: Fewer Samples	120 MHz	DFPBF	21.772	16.071	0	15	32.5	5.701
IR-2: Fewer Samples	140 MHz	SVD	21.772	77.041	2.143	32.143	240	55.269
IR-2: Fewer Samples	140 MHz	AED	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	ModAED	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	ModAEDS	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	AXIS	21.772	18.061	4.286	17.143	34.286	3.711

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	140 MHz	VAXIS	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	DFPAXIS	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	NSAXIS	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	SC	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	VSC	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	DFPSC	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	NSSC	21.772	18.061	4.286	17.143	34.286	3.711
IR-2: Fewer Samples	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: Fewer Samples	140 MHz	NBF	21.772	24.592	2.143	23.571	57.857	2.82
IR-2: Fewer Samples	140 MHz	ABF	21.772	16.429	2.143	17.143	30	5.344
IR-2: Fewer Samples	140 MHz	VBF	21.772	16.429	2.143	17.143	30	5.344
IR-2: Fewer Samples	140 MHz	DFPBF	21.772	18.265	4.286	21.429	27.857	3.507
IR-2: Fewer Samples	150 MHz	SVD	21.772	77.429	0	50	242	55.656
IR-2: Fewer Samples	150 MHz	AED	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	ModAED	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	ModAEDS	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	AXIS	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	VAXIS	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	DFPAXIS	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	NSAXIS	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	SC	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	VSC	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	DFPSC	21.772	19.429	4	20	34	2.344
IR-2: Fewer Samples	150 MHz	NSSC	21.772	18.952	4	18	34	2.82
IR-2: Fewer Samples	150 MHz	CC	21.772	19.714	4	20	34	2.058
IR-2: Fewer Samples	150 MHz	NBF	21.772	19.524	0	22	34	2.248
IR-2: Fewer Samples	150 MHz	ABF	21.772	23.524	4	24	58	1.752
IR-2: Fewer Samples	150 MHz	VBF	21.772	23.524	4	24	58	1.752
IR-2: Fewer Samples	150 MHz	DFPBF	21.772	17.905	2	20	34	3.867
IR-2: Fewer Samples	160 MHz	SVD	21.772	120.446	35.625	84.375	251.25	98.674

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	160 MHz	AED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	AXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	VAXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	DFPAXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	NSAXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	DFPSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Fewer Samples	160 MHz	CC	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: Fewer Samples	160 MHz	NBF	21.772	22.411	3.75	22.5	67.5	0.639
IR-2: Fewer Samples	160 MHz	ABF	21.772	21.339	3.75	20.625	39.375	0.433
IR-2: Fewer Samples	160 MHz	VBF	21.772	21.339	3.75	20.625	39.375	0.433
IR-2: Fewer Samples	160 MHz	DFPBF	21.772	26.161	9.375	22.5	69.375	4.389
IR-2: Fewer Samples	180 MHz	SVD	21.772	175.238	101.667	165	251.667	153.466
IR-2: Fewer Samples	180 MHz	AED	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	ModAED	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	ModAEDS	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	AXIS	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	VAXIS	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	DFPAXIS	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	NSAXIS	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	SC	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	VSC	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	DFPSC	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	NSSC	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: Fewer Samples	180 MHz	NBF	21.772	20.317	1.667	23.333	51.667	1.455

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Fewer Samples	180 MHz	ABF	21.772	25.635	1.667	20	81.667	3.863
IR-2: Fewer Samples	180 MHz	VBF	21.772	25.635	1.667	20	81.667	3.863
IR-2: Fewer Samples	180 MHz	DFPBF	21.772	28.016	3.333	30	60	6.244
IR-2: Fewer Samples	200 MHz	SVD	21.772	198.929	153	199.5	235.5	177.156
IR-2: Fewer Samples	200 MHz	AED	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	ModAED	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	ModAEDS	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	AXIS	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	VAXIS	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	DFPAXIS	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	NSAXIS	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	SC	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	VSC	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	DFPSC	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	NSSC	21.772	19.143	4.5	16.5	34.5	2.629
IR-2: Fewer Samples	200 MHz	CC	21.772	19.286	4.5	19.5	34.5	2.486
IR-2: Fewer Samples	200 MHz	NBF	21.772	23.286	0	24	46.5	1.514
IR-2: Fewer Samples	200 MHz	ABF	21.772	33.357	1.5	30	84	11.585
IR-2: Fewer Samples	200 MHz	VBF	21.772	33.357	1.5	30	84	11.585
IR-2: Fewer Samples	200 MHz	DFPBF	21.772	52.714	4.5	37.5	126	30.942
IR-2: L/2	20 MHz	SVD	21.772	23.571	0	15	45	1.799
IR-2: L/2	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	ModAED	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	DFPAXIS	21.772	20	0	15	30	1.772
IR-2: L/2	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	VSC	21.772	18.571	0	15	30	3.201

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: L/2	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: L/2	20 MHz	NBF	21.772	23.571	0	30	45	1.799
IR-2: L/2	20 MHz	ABF	21.772	23.571	0	30	45	1.799
IR-2: L/2	20 MHz	VBF	21.772	23.571	0	30	45	1.799
IR-2: L/2	20 MHz	DFPBF	21.772	NaN	0	NaN	30	NaN
IR-2: L/2	40 MHz	SVD	21.772	15	0	15	30	6.772
IR-2: L/2	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: L/2	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L/2	40 MHz	AXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	VAXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L/2	40 MHz	NSAXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	SC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	VSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	DFPSC	21.772	19.286	7.5	22.5	37.5	2.486
IR-2: L/2	40 MHz	NSSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L/2	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: L/2	40 MHz	NBF	21.772	27.5	0	30	45	5.728
IR-2: L/2	40 MHz	ABF	21.772	27.143	0	30	45	5.371
IR-2: L/2	40 MHz	VBF	21.772	27.143	0	30	45	5.371
IR-2: L/2	40 MHz	DFPBF	21.772	19.286	7.5	22.5	37.5	2.486
IR-2: L/2	60 MHz	SVD	21.772	17.143	0	20	30	4.629
IR-2: L/2	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: L/2	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: L/2	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058
IR-2: L/2	60 MHz	AXIS	21.772	20.714	5	20	35	1.058
IR-2: L/2	60 MHz	VAXIS	21.772	20.714	5	20	35	1.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	60 MHz	DFPAXIS	21.772	21.19	5	25	35	0.582
IR-2: L/2	60 MHz	NSAXIS	21.772	20.238	5	20	35	1.534
IR-2: L/2	60 MHz	SC	21.772	20.714	5	20	35	1.058
IR-2: L/2	60 MHz	VSC	21.772	20.714	5	20	35	1.058
IR-2: L/2	60 MHz	DFPSC	21.772	20.714	5	25	35	1.058
IR-2: L/2	60 MHz	NSSC	21.772	20.238	5	20	35	1.534
IR-2: L/2	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: L/2	60 MHz	NBF	21.772	24.048	5	25	50	2.275
IR-2: L/2	60 MHz	ABF	21.772	24.762	5	25	50	2.99
IR-2: L/2	60 MHz	VBF	21.772	24.762	5	25	50	2.99
IR-2: L/2	60 MHz	DFPBF	21.772	16.19	5	15	35	5.582
IR-2: L/2	80 MHz	SVD	21.772	16.786	3.75	15	33.75	4.986
IR-2: L/2	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: L/2	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L/2	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L/2	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	SC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	VSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	DFPSC	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: L/2	80 MHz	NSSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L/2	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: L/2	80 MHz	NBF	21.772	23.036	0	22.5	41.25	1.264
IR-2: L/2	80 MHz	ABF	21.772	23.214	0	22.5	45	1.442
IR-2: L/2	80 MHz	VBF	21.772	23.214	0	22.5	45	1.442
IR-2: L/2	80 MHz	DFPBF	21.772	18.036	3.75	18.75	33.75	3.736
IR-2: L/2	100 MHz	SVD	21.772	15.714	0	15	36	6.058
IR-2: L/2	100 MHz	AED	21.772	21.714	6	24	33	0.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	100 MHz	ModAED	21.772	22.286	6	24	33	0.514
IR-2: L/2	100 MHz	ModAEDS	21.772	22.286	6	24	33	0.514
IR-2: L/2	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	SC	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	VSC	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	NSSC	21.772	21.714	6	24	33	0.058
IR-2: L/2	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: L/2	100 MHz	NBF	21.772	25.857	3	24	51	4.085
IR-2: L/2	100 MHz	ABF	21.772	26.429	0	27	51	4.656
IR-2: L/2	100 MHz	VBF	21.772	26.429	0	27	51	4.656
IR-2: L/2	100 MHz	DFPBF	21.772	17.571	3	18	33	4.201
IR-2: L/2	120 MHz	SVD	21.772	16.31	2.5	15	35	5.463
IR-2: L/2	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: L/2	120 MHz	ModAED	21.772	17.143	5	15	35	4.629
IR-2: L/2	120 MHz	ModAEDS	21.772	17.143	5	15	35	4.629
IR-2: L/2	120 MHz	AXIS	21.772	17.381	5	15	35	4.391
IR-2: L/2	120 MHz	VAXIS	21.772	17.381	5	15	35	4.391
IR-2: L/2	120 MHz	DFPAXIS	21.772	15.952	5	15	27.5	5.82
IR-2: L/2	120 MHz	NSAXIS	21.772	17.381	5	15	35	4.391
IR-2: L/2	120 MHz	SC	21.772	17.738	5	15	35	4.034
IR-2: L/2	120 MHz	VSC	21.772	17.738	5	15	35	4.034
IR-2: L/2	120 MHz	DFPSC	21.772	16.429	5	15	27.5	5.344
IR-2: L/2	120 MHz	NSSC	21.772	17.381	5	15	35	4.391
IR-2: L/2	120 MHz	CC	21.772	16.19	5	15	27.5	5.582
IR-2: L/2	120 MHz	NBF	21.772	24.881	2.5	25	50	3.109
IR-2: L/2	120 MHz	ABF	21.772	21.548	2.5	22.5	37.5	0.225

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	120 MHz	VBF	21.772	21.548	2.5	22.5	37.5	0.225
IR-2: L/2	120 MHz	DFPBF	21.772	15.952	0	17.5	35	5.82
IR-2: L/2	140 MHz	SVD	21.772	16.837	2.143	17.143	27.857	4.935
IR-2: L/2	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: L/2	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L/2	140 MHz	ModAEDS	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L/2	140 MHz	AXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	VAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	DFPAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	NSAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	SC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	VSC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	DFPSC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	NSSC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L/2	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L/2	140 MHz	NBF	21.772	24.184	2.143	23.571	45	2.412
IR-2: L/2	140 MHz	ABF	21.772	24.184	0	27.857	45	2.412
IR-2: L/2	140 MHz	VBF	21.772	24.184	0	27.857	45	2.412
IR-2: L/2	140 MHz	DFPBF	21.772	13.878	4.286	15	34.286	7.895
IR-2: L/2	150 MHz	SVD	21.772	16.667	4	16	32	5.105
IR-2: L/2	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: L/2	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: L/2	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963
IR-2: L/2	150 MHz	AXIS	21.772	18.19	4	18	34	3.582
IR-2: L/2	150 MHz	VAXIS	21.772	18.19	4	18	34	3.582
IR-2: L/2	150 MHz	DFPAXIS	21.772	18.19	4	18	34	3.582
IR-2: L/2	150 MHz	NSAXIS	21.772	18.19	4	18	34	3.582
IR-2: L/2	150 MHz	SC	21.772	18.476	4	18	34	3.296
IR-2: L/2	150 MHz	VSC	21.772	18.476	4	18	34	3.296
IR-2: L/2	150 MHz	DFPSC	21.772	18.19	4	18	34	3.582

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	150 MHz	NSSC	21.772	18.476	4	18	34	3.296
IR-2: L/2	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: L/2	150 MHz	NBF	21.772	22.857	0	22	52	1.085
IR-2: L/2	150 MHz	ABF	21.772	21.238	0	16	50	0.534
IR-2: L/2	150 MHz	VBF	21.772	21.238	0	16	50	0.534
IR-2: L/2	150 MHz	DFPBF	21.772	13.81	2	14	32	7.963
IR-2: L/2	160 MHz	SVD	21.772	16.429	3.75	15	30	5.344
IR-2: L/2	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: L/2	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L/2	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L/2	160 MHz	AXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L/2	160 MHz	VAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L/2	160 MHz	DFPAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L/2	160 MHz	NSAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L/2	160 MHz	SC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L/2	160 MHz	VSC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L/2	160 MHz	DFPSC	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L/2	160 MHz	NSSC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L/2	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L/2	160 MHz	NBF	21.772	25	0	30	56.25	3.228
IR-2: L/2	160 MHz	ABF	21.772	26.071	0	31.875	48.75	4.299
IR-2: L/2	160 MHz	VBF	21.772	26.071	0	31.875	48.75	4.299
IR-2: L/2	160 MHz	DFPBF	21.772	18.214	3.75	16.875	31.875	3.558
IR-2: L/2	180 MHz	SVD	21.772	15.159	3.333	16.667	30	6.613
IR-2: L/2	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: L/2	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: L/2	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: L/2	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L/2	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	SC	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	VSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	DFPSC	21.772	21.587	6.667	23.333	35	0.185
IR-2: L/2	180 MHz	NSSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: L/2	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: L/2	180 MHz	NBF	21.772	26.984	1.667	33.333	45	5.212
IR-2: L/2	180 MHz	ABF	21.772	22.778	1.667	20	45	1.006
IR-2: L/2	180 MHz	VBF	21.772	22.778	1.667	20	45	1.006
IR-2: L/2	180 MHz	DFPBF	21.772	14.762	1.667	15	36.667	7.01
IR-2: L/2	200 MHz	SVD	21.772	14.643	0	16.5	27	7.129
IR-2: L/2	200 MHz	AED	21.772	20	4.5	22.5	34.5	1.772
IR-2: L/2	200 MHz	ModAED	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L/2	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	DFPAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	NSAXIS	21.772	19.857	4.5	19.5	34.5	1.915
IR-2: L/2	200 MHz	SC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	VSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	DFPSC	21.772	20.143	4.5	19.5	34.5	1.629
IR-2: L/2	200 MHz	NSSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L/2	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L/2	200 MHz	NBF	21.772	20.857	1.5	18	54	0.915
IR-2: L/2	200 MHz	ABF	21.772	19.143	1.5	16.5	49.5	2.629
IR-2: L/2	200 MHz	VBF	21.772	19.143	1.5	16.5	49.5	2.629
IR-2: L/2	200 MHz	DFPBF	21.772	13.571	3	15	27	8.201
IR-2: L-10	20 MHz	SVD	21.772	23.571	0	15	45	1.799
IR-2: L-10	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	ModAED	21.772	18.571	0	15	30	3.201

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	DFPAXIS	21.772	20	0	15	30	1.772
IR-2: L-10	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	VSC	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: L-10	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: L-10	20 MHz	NBF	21.772	23.571	0	30	45	1.799
IR-2: L-10	20 MHz	ABF	21.772	22.857	0	30	45	1.085
IR-2: L-10	20 MHz	VBF	21.772	22.857	0	30	45	1.085
IR-2: L-10	20 MHz	DFPBF	21.772	NaN	0	NaN	30	NaN
IR-2: L-10	40 MHz	SVD	21.772	20.357	0	22.5	37.5	1.415
IR-2: L-10	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: L-10	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L-10	40 MHz	AXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	VAXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L-10	40 MHz	NSAXIS	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	SC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	VSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	DFPSC	21.772	18.571	7.5	15	37.5	3.201
IR-2: L-10	40 MHz	NSSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L-10	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: L-10	40 MHz	NBF	21.772	26.071	7.5	30	67.5	4.299
IR-2: L-10	40 MHz	ABF	21.772	26.429	7.5	30	52.5	4.656
IR-2: L-10	40 MHz	VBF	21.772	26.429	7.5	30	52.5	4.656

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	40 MHz	DFPBF	21.772	19.286	7.5	15	37.5	2.486
IR-2: L-10	60 MHz	SVD	21.772	15.238	0	10	35	6.534
IR-2: L-10	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: L-10	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: L-10	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058
IR-2: L-10	60 MHz	AXIS	21.772	20.714	5	20	35	1.058
IR-2: L-10	60 MHz	VAXIS	21.772	20.714	5	20	35	1.058
IR-2: L-10	60 MHz	DFPAXIS	21.772	21.19	5	25	35	0.582
IR-2: L-10	60 MHz	NSAXIS	21.772	20.238	5	20	35	1.534
IR-2: L-10	60 MHz	SC	21.772	20.476	5	20	35	1.296
IR-2: L-10	60 MHz	VSC	21.772	20.476	5	20	35	1.296
IR-2: L-10	60 MHz	DFPSC	21.772	20.714	5	25	35	1.058
IR-2: L-10	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: L-10	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: L-10	60 MHz	NBF	21.772	33.81	5	25	85	12.037
IR-2: L-10	60 MHz	ABF	21.772	19.048	0	20	45	2.725
IR-2: L-10	60 MHz	VBF	21.772	19.048	0	20	45	2.725
IR-2: L-10	60 MHz	DFPBF	21.772	13.333	0	15	25	8.439
IR-2: L-10	80 MHz	SVD	21.772	18.393	3.75	18.75	33.75	3.379
IR-2: L-10	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: L-10	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L-10	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L-10	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	SC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	VSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	DFPSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L-10	80 MHz	NSSC	21.772	20	3.75	22.5	33.75	1.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: L-10	80 MHz	NBF	21.772	30.893	7.5	26.25	78.75	9.121
IR-2: L-10	80 MHz	ABF	21.772	18.393	3.75	15	52.5	3.379
IR-2: L-10	80 MHz	VBF	21.772	18.393	3.75	15	52.5	3.379
IR-2: L-10	80 MHz	DFPBF	21.772	16.964	3.75	15	30	4.808
IR-2: L-10	100 MHz	SVD	21.772	15	3	15	42	6.772
IR-2: L-10	100 MHz	AED	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	ModAED	21.772	21.857	6	24	33	0.085
IR-2: L-10	100 MHz	ModAEDS	21.772	21.857	6	24	33	0.085
IR-2: L-10	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	SC	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	VSC	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	NSSC	21.772	21.714	6	24	33	0.058
IR-2: L-10	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: L-10	100 MHz	NBF	21.772	21	3	15	57	0.772
IR-2: L-10	100 MHz	ABF	21.772	14.143	0	15	27	7.629
IR-2: L-10	100 MHz	VBF	21.772	14.143	0	15	27	7.629
IR-2: L-10	100 MHz	DFPBF	21.772	17.143	3	18	33	4.629
IR-2: L-10	120 MHz	SVD	21.772	16.071	0	15	37.5	5.701
IR-2: L-10	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: L-10	120 MHz	ModAED	21.772	17.143	5	15	35	4.629
IR-2: L-10	120 MHz	ModAEDS	21.772	16.905	5	15	35	4.867
IR-2: L-10	120 MHz	AXIS	21.772	17.738	5	15	35	4.034
IR-2: L-10	120 MHz	VAXIS	21.772	17.738	5	15	35	4.034
IR-2: L-10	120 MHz	DFPAXIS	21.772	17.738	5	15	35	4.034
IR-2: L-10	120 MHz	NSAXIS	21.772	17.381	5	15	35	4.394

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	120 MHz	SC	21.772	17.143	5	15	35	4.629
IR-2: L-10	120 MHz	VSC	21.772	17.143	5	15	35	4.629
IR-2: L-10	120 MHz	DFPSC	21.772	17.738	5	15	35	4.034
IR-2: L-10	120 MHz	NSSC	21.772	17.143	5	15	35	4.629
IR-2: L-10	120 MHz	CC	21.772	15.833	5	15	27.5	5.939
IR-2: L-10	120 MHz	NBF	21.772	20.119	0	17.5	57.5	1.653
IR-2: L-10	120 MHz	ABF	21.772	16.786	0	15	57.5	4.986
IR-2: L-10	120 MHz	VBF	21.772	16.786	0	15	57.5	4.986
IR-2: L-10	120 MHz	DFPBF	21.772	16.071	2.5	17.5	45	5.701
IR-2: L-10	140 MHz	SVD	21.772	13.061	0	15	25.714	8.711
IR-2: L-10	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: L-10	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L-10	140 MHz	ModAEDS	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L-10	140 MHz	AXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L-10	140 MHz	VAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L-10	140 MHz	DFPAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L-10	140 MHz	NSAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L-10	140 MHz	SC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L-10	140 MHz	VSC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L-10	140 MHz	DFPSC	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L-10	140 MHz	NSSC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L-10	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L-10	140 MHz	NBF	21.772	25.408	0	19.286	62.143	3.636
IR-2: L-10	140 MHz	ABF	21.772	15.408	4.286	15	25.714	6.364
IR-2: L-10	140 MHz	VBF	21.772	15.408	4.286	15	25.714	6.364
IR-2: L-10	140 MHz	DFPBF	21.772	14.592	4.286	15	32.143	7.18
IR-2: L-10	150 MHz	SVD	21.772	15.905	4	16	34	5.867
IR-2: L-10	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: L-10	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: L-10	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	150 MHz	AXIS	21.772	18.476	4	18	34	3.296
IR-2: L-10	150 MHz	VAXIS	21.772	18.476	4	18	34	3.296
IR-2: L-10	150 MHz	DFPAXIS	21.772	18.476	4	18	34	3.296
IR-2: L-10	150 MHz	NSAXIS	21.772	19.048	4	18	34	2.725
IR-2: L-10	150 MHz	SC	21.772	19.81	4	20	34	1.963
IR-2: L-10	150 MHz	VSC	21.772	19.81	4	20	34	1.963
IR-2: L-10	150 MHz	DFPSC	21.772	18.476	4	18	34	3.296
IR-2: L-10	150 MHz	NSSC	21.772	19.81	4	20	34	1.963
IR-2: L-10	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: L-10	150 MHz	NBF	21.772	19.905	2	20	54	1.867
IR-2: L-10	150 MHz	ABF	21.772	17.905	2	18	50	3.867
IR-2: L-10	150 MHz	VBF	21.772	17.905	2	18	50	3.867
IR-2: L-10	150 MHz	DFPBF	21.772	16.19	4	16	42	5.582
IR-2: L-10	160 MHz	SVD	21.772	17.857	3.75	16.875	56.25	3.915
IR-2: L-10	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: L-10	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	AXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L-10	160 MHz	VAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L-10	160 MHz	DFPAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L-10	160 MHz	NSAXIS	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L-10	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	DFPSC	21.772	18.125	3.75	16.875	33.75	3.647
IR-2: L-10	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L-10	160 MHz	NBF	21.772	23.393	0	16.875	80.625	1.621
IR-2: L-10	160 MHz	ABF	21.772	19.018	1.875	16.875	54.375	2.754
IR-2: L-10	160 MHz	VBF	21.772	19.018	1.875	16.875	54.375	2.754
IR-2: L-10	160 MHz	DFPBF	21.772	16.071	1.875	15	35.625	5.704

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	180 MHz	SVD	21.772	11.27	1.667	8.333	35	10.502
IR-2: L-10	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: L-10	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: L-10	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: L-10	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L-10	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L-10	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L-10	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L-10	180 MHz	SC	21.772	21.508	5	23.333	35	0.264
IR-2: L-10	180 MHz	VSC	21.772	21.508	5	23.333	35	0.264
IR-2: L-10	180 MHz	DFPSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: L-10	180 MHz	NSSC	21.772	21.587	6.667	23.333	35	0.185
IR-2: L-10	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: L-10	180 MHz	NBF	21.772	24.127	1.667	18.333	118.333	2.355
IR-2: L-10	180 MHz	ABF	21.772	16.746	0	13.333	53.333	5.026
IR-2: L-10	180 MHz	VBF	21.772	16.746	0	13.333	53.333	5.026
IR-2: L-10	180 MHz	DFPBF	21.772	16.984	0	15	35	4.788
IR-2: L-10	200 MHz	SVD	21.772	12.643	0	15	28.5	9.129
IR-2: L-10	200 MHz	AED	21.772	19.786	4.5	22.5	34.5	1.986
IR-2: L-10	200 MHz	ModAED	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L-10	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L-10	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	DFPAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	NSAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	SC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	VSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	DFPSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	NSSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L-10	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L-10	200 MHz	NBF	21.772	22.357	1.5	22.5	63	0.585
IR-2: L-10	200 MHz	ABF	21.772	33.786	4.5	27	120	12.014
IR-2: L-10	200 MHz	VBF	21.772	33.786	4.5	27	120	12.014
IR-2: L-10	200 MHz	DFPBF	21.772	38.214	3	31.5	109.5	16.442
IR-2: 2*L	20 MHz	SVD	21.772	124.286	0	60	285	102.514
IR-2: 2*L	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	ModAED	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	DFPAXIS	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	VSC	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: 2*L	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: 2*L	20 MHz	NBF	21.772	150	0	180	285	128.228
IR-2: 2*L	20 MHz	ABF	21.772	146.429	0	120	315	124.656
IR-2: 2*L	20 MHz	VBF	21.772	146.429	0	120	315	124.656
IR-2: 2*L	20 MHz	DFPBF	21.772	19.286	0	15	30	2.486
IR-2: 2*L	40 MHz	SVD	21.772	94.286	7.5	67.5	255	72.514
IR-2: 2*L	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: 2*L	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	AXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	VAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	NSAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	SC	21.772	20	7.5	22.5	37.5	1.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	40 MHz	VSC	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	DFPSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: 2*L	40 MHz	NSSC	21.772	20	7.5	22.5	37.5	1.772
IR-2: 2*L	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: 2*L	40 MHz	NBF	21.772	171.071	0	225	315	149.299
IR-2: 2*L	40 MHz	ABF	21.772	128.214	7.5	97.5	330	106.442
IR-2: 2*L	40 MHz	VBF	21.772	128.214	7.5	97.5	330	106.442
IR-2: 2*L	40 MHz	DFPBF	21.772	NaN	7.5	NaN	30	NaN
IR-2: 2*L	60 MHz	SVD	21.772	80.476	0	45	220	58.704
IR-2: 2*L	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: 2*L	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	AXIS	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	VAXIS	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	DFPAXIS	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	NSAXIS	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	SC	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	VSC	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	DFPSC	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	NSSC	21.772	20.714	5	20	35	1.058
IR-2: 2*L	60 MHz	CC	21.772	23.333	5	25	35	1.561
IR-2: 2*L	60 MHz	NBF	21.772	154.762	0	175	360	132.99
IR-2: 2*L	60 MHz	ABF	21.772	96.19	5	70	290	74.418
IR-2: 2*L	60 MHz	VBF	21.772	96.19	5	70	290	74.418
IR-2: 2*L	60 MHz	DFPBF	21.772	21.19	5	25	35	0.582
IR-2: 2*L	80 MHz	SVD	21.772	79.821	3.75	52.5	206.25	58.049
IR-2: 2*L	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: 2*L	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: 2*L	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: 2*L	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: 2*L	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: 2*L	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: 2*L	80 MHz	SC	21.772	20.179	3.75	22.5	33.75	1.594
IR-2: 2*L	80 MHz	VSC	21.772	20.179	3.75	22.5	33.75	1.594
IR-2: 2*L	80 MHz	DFPSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: 2*L	80 MHz	NSSC	21.772	20.179	3.75	22.5	33.75	1.594
IR-2: 2*L	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: 2*L	80 MHz	NBF	21.772	134.107	0	108.75	315	112.335
IR-2: 2*L	80 MHz	ABF	21.772	96.607	11.25	67.5	213.75	74.835
IR-2: 2*L	80 MHz	VBF	21.772	96.607	11.25	67.5	213.75	74.835
IR-2: 2*L	80 MHz	DFPBF	21.772	NaN	3.75	NaN	183.75	NaN
IR-2: 2*L	100 MHz	SVD	21.772	109.429	3	114	264	87.656
IR-2: 2*L	100 MHz	AED	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	ModAED	21.772	21.857	6	24	33	0.085
IR-2: 2*L	100 MHz	ModAEDS	21.772	21.857	6	24	33	0.085
IR-2: 2*L	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	SC	21.772	22.286	6	24	33	0.514
IR-2: 2*L	100 MHz	VSC	21.772	22.286	6	24	33	0.514
IR-2: 2*L	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058
IR-2: 2*L	100 MHz	NSSC	21.772	22.286	6	24	33	0.514
IR-2: 2*L	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: 2*L	100 MHz	NBF	21.772	132.857	3	111	303	111.085
IR-2: 2*L	100 MHz	ABF	21.772	96.429	0	93	267	74.656
IR-2: 2*L	100 MHz	VBF	21.772	96.429	0	93	267	74.656
IR-2: 2*L	100 MHz	DFPBF	21.772	NaN	3	NaN	300	NaN
IR-2: 2*L	120 MHz	SVD	21.772	NaN	0	NaN	295	NaN

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: 2*L	120 MHz	ModAED	21.772	16.905	5	15	35	4.867
IR-2: 2*L	120 MHz	ModAEDS	21.772	16.905	5	15	35	4.867
IR-2: 2*L	120 MHz	AXIS	21.772	17.143	5	15	35	4.629
IR-2: 2*L	120 MHz	VAXIS	21.772	17.143	5	15	35	4.629
IR-2: 2*L	120 MHz	DFPAXIS	21.772	17.738	5	15	35	4.034
IR-2: 2*L	120 MHz	NSAXIS	21.772	17.143	5	15	35	4.629
IR-2: 2*L	120 MHz	SC	21.772	16.905	5	15	35	4.867
IR-2: 2*L	120 MHz	VSC	21.772	16.905	5	15	35	4.867
IR-2: 2*L	120 MHz	DFPSC	21.772	17.143	5	15	35	4.629
IR-2: 2*L	120 MHz	NSSC	21.772	16.905	5	15	35	4.867
IR-2: 2*L	120 MHz	CC	21.772	15.833	5	15	27.5	5.939
IR-2: 2*L	120 MHz	NBF	21.772	NaN	0	NaN	317.5	NaN
IR-2: 2*L	120 MHz	ABF	21.772	187.976	2.5	177.5	470	166.204
IR-2: 2*L	120 MHz	VBF	21.772	187.976	2.5	177.5	470	166.204
IR-2: 2*L	120 MHz	DFPBF	21.772	NaN	12.5	NaN	512.5	NaN
IR-2: 2*L	140 MHz	SVD	21.772	NaN	0	NaN	261.429	NaN
IR-2: 2*L	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: 2*L	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: 2*L	140 MHz	ModAEDS	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	AXIS	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	VAXIS	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	DFPAXIS	21.772	18.776	4.286	17.143	34.286	2.997
IR-2: 2*L	140 MHz	NSAXIS	21.772	18.776	4.286	17.143	34.286	2.997
IR-2: 2*L	140 MHz	SC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	VSC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	DFPSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: 2*L	140 MHz	NSSC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: 2*L	140 MHz	NBF	21.772	NaN	2.143	NaN	319.286	NaN

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	140 MHz	ABF	21.772	134.184	15	94.286	334.286	112.412
IR-2: 2*L	140 MHz	VBF	21.772	134.184	15	94.286	334.286	112.412
IR-2: 2*L	140 MHz	DFPBF	21.772	249.184	6.429	267.857	495	227.412
IR-2: 2*L	150 MHz	SVD	21.772	NaN	16	NaN	294	NaN
IR-2: 2*L	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: 2*L	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	AXIS	21.772	19.524	4	18	34	2.248
IR-2: 2*L	150 MHz	VAXIS	21.772	19.524	4	18	34	2.248
IR-2: 2*L	150 MHz	DFPAXIS	21.772	19.048	4	18	34	2.725
IR-2: 2*L	150 MHz	NSAXIS	21.772	19.524	4	18	34	2.248
IR-2: 2*L	150 MHz	SC	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	VSC	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	DFPSC	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	NSSC	21.772	19.81	4	20	34	1.963
IR-2: 2*L	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: 2*L	150 MHz	NBF	21.772	NaN	4	NaN	302	NaN
IR-2: 2*L	150 MHz	ABF	21.772	NaN	4	NaN	446	NaN
IR-2: 2*L	150 MHz	VBF	21.772	NaN	4	NaN	446	NaN
IR-2: 2*L	150 MHz	DFPBF	21.772	189.714	34	92	512	167.942
IR-2: 2*L	160 MHz	SVD	21.772	NaN	1.875	NaN	273.75	NaN
IR-2: 2*L	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: 2*L	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	AXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	VAXIS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	DFPAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: 2*L	160 MHz	NSAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: 2*L	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	160 MHz	DFPSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: 2*L	160 MHz	NBF	21.772	NaN	30	NaN	300	NaN
IR-2: 2*L	160 MHz	ABF	21.772	213.661	69.375	168.75	511.875	191.889
IR-2: 2*L	160 MHz	VBF	21.772	213.661	69.375	168.75	511.875	191.889
IR-2: 2*L	160 MHz	DFPBF	21.772	NaN	1.875	NaN	511.875	NaN
IR-2: 2*L	180 MHz	SVD	21.772	NaN	10	NaN	36.667	NaN
IR-2: 2*L	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: 2*L	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: 2*L	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: 2*L	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: 2*L	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: 2*L	180 MHz	SC	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	VSC	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	DFPSC	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	NSSC	21.772	21.508	5	23.333	35	0.264
IR-2: 2*L	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: 2*L	180 MHz	NBF	21.772	NaN	31.667	NaN	31.667	NaN
IR-2: 2*L	180 MHz	ABF	21.772	NaN	0	NaN	500	NaN
IR-2: 2*L	180 MHz	VBF	21.772	NaN	0	NaN	500	NaN
IR-2: 2*L	180 MHz	DFPBF	21.772	NaN	25	NaN	563.333	NaN
IR-2: 2*L	200 MHz	SVD	21.772	NaN	37.5	NaN	279	NaN
IR-2: 2*L	200 MHz	AED	21.772	19.786	4.5	22.5	34.5	1.986
IR-2: 2*L	200 MHz	ModAED	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: 2*L	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: 2*L	200 MHz	DFPAXIS	21.772	20.143	4.5	19.5	34.5	1.629
IR-2: 2*L	200 MHz	NSAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: 2*L	200 MHz	SC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	VSC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	DFPSC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	NSSC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: 2*L	200 MHz	NBF	21.772	NaN	NaN	NaN	NaN	NaN
IR-2: 2*L	200 MHz	ABF	21.772	NaN	109.5	NaN	507	NaN
IR-2: 2*L	200 MHz	VBF	21.772	NaN	109.5	NaN	507	NaN
IR-2: 2*L	200 MHz	DFPBF	21.772	NaN	7.5	NaN	534	NaN
IR-2: L+10	20 MHz	SVD	21.772	78.571	0	60	165	56.799
IR-2: L+10	20 MHz	AED	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	ModAED	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	ModAEDS	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	VAXIS	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	DFPAXIS	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	SC	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	VSC	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	DFPSC	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	NSSC	21.772	18.571	0	15	30	3.201
IR-2: L+10	20 MHz	CC	21.772	19.286	0	15	30	2.486
IR-2: L+10	20 MHz	NBF	21.772	70	0	75	150	48.228
IR-2: L+10	20 MHz	ABF	21.772	56.429	0	30	165	34.656
IR-2: L+10	20 MHz	VBF	21.772	56.429	0	30	165	34.656
IR-2: L+10	20 MHz	DFPBF	21.772	20	0	15	30	1.772
IR-2: L+10	40 MHz	SVD	21.772	26.429	0	22.5	90	4.656
IR-2: L+10	40 MHz	AED	21.772	18.929	7.5	22.5	37.5	2.844

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	40 MHz	ModAED	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	ModAEDS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	AXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: L+10	40 MHz	VAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: L+10	40 MHz	DFPAXIS	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	NSAXIS	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: L+10	40 MHz	SC	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	VSC	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	DFPSC	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: L+10	40 MHz	NSSC	21.772	20	7.5	22.5	37.5	1.772
IR-2: L+10	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: L+10	40 MHz	NBF	21.772	36.429	0	30	90	14.656
IR-2: L+10	40 MHz	ABF	21.772	65	7.5	67.5	112.5	43.228
IR-2: L+10	40 MHz	VBF	21.772	65	7.5	67.5	112.5	43.228
IR-2: L+10	40 MHz	DFPBF	21.772	19.643	7.5	22.5	37.5	2.129
IR-2: L+10	60 MHz	SVD	21.772	23.333	0	15	60	1.561
IR-2: L+10	60 MHz	AED	21.772	20.714	5	25	35	1.058
IR-2: L+10	60 MHz	ModAED	21.772	20.714	5	20	35	1.058
IR-2: L+10	60 MHz	ModAEDS	21.772	20.714	5	20	35	1.058
IR-2: L+10	60 MHz	AXIS	21.772	20.476	5	20	35	1.296
IR-2: L+10	60 MHz	VAXIS	21.772	20.476	5	20	35	1.296
IR-2: L+10	60 MHz	DFPAXIS	21.772	21.19	5	25	35	0.582
IR-2: L+10	60 MHz	NSAXIS	21.772	20.476	5	20	35	1.296
IR-2: L+10	60 MHz	SC	21.772	20.714	5	20	35	1.058
IR-2: L+10	60 MHz	VSC	21.772	20.714	5	20	35	1.058
IR-2: L+10	60 MHz	DFPSC	21.772	20.476	5	20	35	1.296
IR-2: L+10	60 MHz	NSSC	21.772	20.476	5	20	35	1.296
IR-2: L+10	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: L+10	60 MHz	NBF	21.772	28.333	0	25	110	6.561
IR-2: L+10	60 MHz	ABF	21.772	30.238	0	25	120	8.466

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	60 MHz	VBF	21.772	30.238	0	25	120	8.466
IR-2: L+10	60 MHz	DFPBF	21.772	19.762	5	25	35	2.01
IR-2: L+10	80 MHz	SVD	21.772	28.036	3.75	22.5	101.25	6.264
IR-2: L+10	80 MHz	AED	21.772	19.286	3.75	18.75	33.75	2.486
IR-2: L+10	80 MHz	ModAED	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L+10	80 MHz	ModAEDS	21.772	19.821	3.75	18.75	33.75	1.951
IR-2: L+10	80 MHz	AXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	VAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	DFPAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	NSAXIS	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	SC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	VSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	DFPSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	NSSC	21.772	20	3.75	22.5	33.75	1.772
IR-2: L+10	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: L+10	80 MHz	NBF	21.772	26.786	0	26.25	60	5.014
IR-2: L+10	80 MHz	ABF	21.772	24.286	3.75	11.25	75	2.514
IR-2: L+10	80 MHz	VBF	21.772	24.286	3.75	11.25	75	2.514
IR-2: L+10	80 MHz	DFPBF	21.772	18.75	3.75	18.75	48.75	3.022
IR-2: L+10	100 MHz	SVD	21.772	20.714	0	18	45	1.058
IR-2: L+10	100 MHz	AED	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	ModAED	21.772	21.857	6	24	33	0.085
IR-2: L+10	100 MHz	ModAEDS	21.772	21.857	6	24	33	0.085
IR-2: L+10	100 MHz	AXIS	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	VAXIS	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	DFPAXIS	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	NSAXIS	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	SC	21.772	22.143	6	24	33	0.371
IR-2: L+10	100 MHz	VSC	21.772	22.143	6	24	33	0.371
IR-2: L+10	100 MHz	DFPSC	21.772	21.714	6	24	33	0.058

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	100 MHz	NSSC	21.772	21.714	6	24	33	0.058
IR-2: L+10	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: L+10	100 MHz	NBF	21.772	26.571	3	24	57	4.799
IR-2: L+10	100 MHz	ABF	21.772	26.143	0	18	66	4.371
IR-2: L+10	100 MHz	VBF	21.772	26.143	0	18	66	4.371
IR-2: L+10	100 MHz	DFPBF	21.772	19.857	6	15	39	1.915
IR-2: L+10	120 MHz	SVD	21.772	21.31	2.5	22.5	45	0.463
IR-2: L+10	120 MHz	AED	21.772	16.429	5	15	27.5	5.344
IR-2: L+10	120 MHz	ModAED	21.772	17.143	5	15	35	4.629
IR-2: L+10	120 MHz	ModAEDS	21.772	17.143	5	15	35	4.629
IR-2: L+10	120 MHz	AXIS	21.772	17.738	5	15	35	4.034
IR-2: L+10	120 MHz	VAXIS	21.772	17.738	5	15	35	4.034
IR-2: L+10	120 MHz	DFPAXIS	21.772	17.738	5	15	35	4.034
IR-2: L+10	120 MHz	NSAXIS	21.772	17.738	5	15	35	4.034
IR-2: L+10	120 MHz	SC	21.772	17.143	5	15	35	4.629
IR-2: L+10	120 MHz	VSC	21.772	17.143	5	15	35	4.629
IR-2: L+10	120 MHz	DFPSC	21.772	17.738	5	15	35	4.034
IR-2: L+10	120 MHz	NSSC	21.772	17.143	5	15	35	4.629
IR-2: L+10	120 MHz	CC	21.772	15.833	5	15	27.5	5.939
IR-2: L+10	120 MHz	NBF	21.772	21.786	5	22.5	50	0.014
IR-2: L+10	120 MHz	ABF	21.772	18.81	0	15	55	2.963
IR-2: L+10	120 MHz	VBF	21.772	18.81	0	15	55	2.963
IR-2: L+10	120 MHz	DFPBF	21.772	17.381	2.5	15	47.5	4.391
IR-2: L+10	140 MHz	SVD	21.772	16.327	2.143	15	40.714	5.446
IR-2: L+10	140 MHz	AED	21.772	19.082	4.286	19.286	34.286	2.691
IR-2: L+10	140 MHz	ModAED	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L+10	140 MHz	ModAEDS	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L+10	140 MHz	AXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L+10	140 MHz	VAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L+10	140 MHz	DFPAXIS	21.772	18.265	4.286	17.143	34.286	3.507

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	140 MHz	NSAXIS	21.772	18.265	4.286	17.143	34.286	3.507
IR-2: L+10	140 MHz	SC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L+10	140 MHz	VSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L+10	140 MHz	DFPSC	21.772	18.776	4.286	17.143	34.286	2.997
IR-2: L+10	140 MHz	NSSC	21.772	17.653	4.286	17.143	34.286	4.119
IR-2: L+10	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: L+10	140 MHz	NBF	21.772	24.796	2.143	23.571	45	3.024
IR-2: L+10	140 MHz	ABF	21.772	17.143	0	15	42.857	4.629
IR-2: L+10	140 MHz	VBF	21.772	17.143	0	15	42.857	4.629
IR-2: L+10	140 MHz	DFPBF	21.772	18.776	4.286	17.143	47.143	2.997
IR-2: L+10	150 MHz	SVD	21.772	20.476	2	20	38	1.296
IR-2: L+10	150 MHz	AED	21.772	18.19	4	18	34	3.582
IR-2: L+10	150 MHz	ModAED	21.772	19.81	4	20	34	1.963
IR-2: L+10	150 MHz	ModAEDS	21.772	19.81	4	20	34	1.963
IR-2: L+10	150 MHz	AXIS	21.772	18.476	4	18	34	3.296
IR-2: L+10	150 MHz	VAXIS	21.772	18.476	4	18	34	3.296
IR-2: L+10	150 MHz	DFPAXIS	21.772	18.476	4	18	34	3.296
IR-2: L+10	150 MHz	NSAXIS	21.772	18.476	4	18	34	3.296
IR-2: L+10	150 MHz	SC	21.772	19.81	4	20	34	1.963
IR-2: L+10	150 MHz	VSC	21.772	19.81	4	20	34	1.963
IR-2: L+10	150 MHz	DFPSC	21.772	18.476	4	18	34	3.296
IR-2: L+10	150 MHz	NSSC	21.772	19.81	4	20	34	1.963
IR-2: L+10	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: L+10	150 MHz	NBF	21.772	24.952	0	28	56	3.18
IR-2: L+10	150 MHz	ABF	21.772	18.667	4	16	44	3.105
IR-2: L+10	150 MHz	VBF	21.772	18.667	4	16	44	3.105
IR-2: L+10	150 MHz	DFPBF	21.772	14.857	2	14	52	6.915
IR-2: L+10	160 MHz	SVD	21.772	19.732	0	16.875	75	2.04
IR-2: L+10	160 MHz	AED	21.772	18.929	3.75	22.5	33.75	2.844
IR-2: L+10	160 MHz	ModAED	21.772	18.304	3.75	16.875	33.75	3.469

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	160 MHz	ModAEDS	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L+10	160 MHz	AXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L+10	160 MHz	VAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L+10	160 MHz	DFPAXIS	21.772	17.857	3.75	16.875	33.75	3.915
IR-2: L+10	160 MHz	NSAXIS	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L+10	160 MHz	SC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L+10	160 MHz	VSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L+10	160 MHz	DFPSC	21.772	17.679	3.75	16.875	33.75	4.094
IR-2: L+10	160 MHz	NSSC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L+10	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: L+10	160 MHz	NBF	21.772	24.911	0	24.375	84.375	3.139
IR-2: L+10	160 MHz	ABF	21.772	18.482	3.75	16.875	43.125	3.29
IR-2: L+10	160 MHz	VBF	21.772	18.482	3.75	16.875	43.125	3.29
IR-2: L+10	160 MHz	DFPBF	21.772	22.768	7.5	20.625	50.625	0.996
IR-2: L+10	180 MHz	SVD	21.772	22.222	0	23.333	41.667	0.45
IR-2: L+10	180 MHz	AED	21.772	21.587	6.667	23.333	35	0.185
IR-2: L+10	180 MHz	ModAED	21.772	21.508	5	23.333	35	0.264
IR-2: L+10	180 MHz	ModAEDS	21.772	21.508	5	23.333	35	0.264
IR-2: L+10	180 MHz	AXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L+10	180 MHz	VAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L+10	180 MHz	DFPAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L+10	180 MHz	NSAXIS	21.772	21.984	6.667	23.333	35	0.212
IR-2: L+10	180 MHz	SC	21.772	21.508	5	23.333	35	0.264
IR-2: L+10	180 MHz	VSC	21.772	21.508	5	23.333	35	0.264
IR-2: L+10	180 MHz	DFPSC	21.772	21.984	6.667	23.333	35	0.212
IR-2: L+10	180 MHz	NSSC	21.772	21.508	5	23.333	35	0.264
IR-2: L+10	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: L+10	180 MHz	NBF	21.772	22.619	1.667	23.333	61.667	0.847
IR-2: L+10	180 MHz	ABF	21.772	26.825	1.667	26.667	70	5.053
IR-2: L+10	180 MHz	VBF	21.772	26.825	1.667	26.667	70	5.053

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: L+10	180 MHz	DFPBF	21.772	23.095	0	15	78.333	1.323
IR-2: L+10	200 MHz	SVD	21.772	22.643	3	22.5	51	0.871
IR-2: L+10	200 MHz	AED	21.772	19.786	4.5	22.5	34.5	1.986
IR-2: L+10	200 MHz	ModAED	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L+10	200 MHz	ModAEDS	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L+10	200 MHz	AXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	VAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	DFPAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	NSAXIS	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	SC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L+10	200 MHz	VSC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L+10	200 MHz	DFPSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	NSSC	21.772	19.643	4.5	19.5	34.5	2.129
IR-2: L+10	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: L+10	200 MHz	NBF	21.772	26.214	1.5	27	55.5	4.442
IR-2: L+10	200 MHz	ABF	21.772	53.714	6	49.5	120	31.942
IR-2: L+10	200 MHz	VBF	21.772	53.714	6	49.5	120	31.942
IR-2: L+10	200 MHz	DFPBF	21.772	58.429	3	55.5	124.5	36.656
IR-2: Random Start	20 MHz	AED	21.772	90.714	0	75	285	68.942
IR-2: Random Start	20 MHz	ModAED	21.772	172.143	15	180	435	150.371
IR-2: Random Start	20 MHz	ModAEDS	21.772	172.143	15	180	435	150.371
IR-2: Random Start	20 MHz	AXIS	21.772	18.571	0	15	30	3.201
IR-2: Random Start	20 MHz	VAXIS	21.772	85	0	30	255	63.228
IR-2: Random Start	20 MHz	DFPAXIS	21.772	97.143	0	90	285	75.371
IR-2: Random Start	20 MHz	NSAXIS	21.772	18.571	0	15	30	3.201
IR-2: Random Start	20 MHz	SC	21.772	97.143	0	75	345	75.371
IR-2: Random Start	20 MHz	VSC	21.772	97.143	0	75	345	75.371
IR-2: Random Start	20 MHz	DFPSC	21.772	85.714	0	60	240	63.942
IR-2: Random Start	20 MHz	NSSC	21.772	97.143	0	75	345	75.371
IR-2: Random Start	20 MHz	CC	21.772	19.286	0	15	30	2.486

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	20 MHz	NBF	21.772	35	0	30	105	13.228
IR-2: Random Start	20 MHz	ABF	21.772	84.286	15	105	180	62.514
IR-2: Random Start	20 MHz	VBF	21.772	84.286	15	105	180	62.514
IR-2: Random Start	20 MHz	DFPBF	21.772	17.857	0	15	30	3.915
IR-2: Random Start	40 MHz	AED	21.772	85	0	60	225	63.228
IR-2: Random Start	40 MHz	ModAED	21.772	85	7.5	60	262.5	63.228
IR-2: Random Start	40 MHz	ModAEDS	21.772	89.286	7.5	60	262.5	67.514
IR-2: Random Start	40 MHz	AXIS	21.772	18.214	7.5	15	37.5	3.558
IR-2: Random Start	40 MHz	VAXIS	21.772	57.5	7.5	45	150	35.728
IR-2: Random Start	40 MHz	DFPAXIS	21.772	53.929	7.5	37.5	150	32.156
IR-2: Random Start	40 MHz	NSAXIS	21.772	17.143	7.5	15	37.5	4.629
IR-2: Random Start	40 MHz	SC	21.772	85	0	60	225	63.228
IR-2: Random Start	40 MHz	VSC	21.772	85	0	60	225	63.228
IR-2: Random Start	40 MHz	DFPSC	21.772	85	0	60	225	63.228
IR-2: Random Start	40 MHz	NSSC	21.772	85	0	60	225	63.228
IR-2: Random Start	40 MHz	CC	21.772	18.571	7.5	22.5	30	3.201
IR-2: Random Start	40 MHz	NBF	21.772	33.571	0	30	67.5	11.799
IR-2: Random Start	40 MHz	ABF	21.772	50	0	37.5	112.5	28.228
IR-2: Random Start	40 MHz	VBF	21.772	50	0	37.5	112.5	28.228
IR-2: Random Start	40 MHz	DFPBF	21.772	18.929	7.5	22.5	37.5	2.844
IR-2: Random Start	60 MHz	AED	21.772	61.429	0	35	195	39.656
IR-2: Random Start	60 MHz	ModAED	21.772	117.619	5	105	285	95.847
IR-2: Random Start	60 MHz	ModAEDS	21.772	117.619	5	105	285	95.847
IR-2: Random Start	60 MHz	AXIS	21.772	20.952	5	20	35	0.82
IR-2: Random Start	60 MHz	VAXIS	21.772	87.857	10	90	145	66.085
IR-2: Random Start	60 MHz	DFPAXIS	21.772	69.048	5	70	135	47.275
IR-2: Random Start	60 MHz	NSAXIS	21.772	20	5	20	35	1.772
IR-2: Random Start	60 MHz	SC	21.772	100.952	5	100	285	79.18
IR-2: Random Start	60 MHz	VSC	21.772	100.952	5	100	285	79.18
IR-2: Random Start	60 MHz	DFPSC	21.772	58.095	0	25	195	36.323

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	60 MHz	NSSC	21.772	99.524	5	100	285	77.752
IR-2: Random Start	60 MHz	CC	21.772	22.381	5	25	35	0.609
IR-2: Random Start	60 MHz	NBF	21.772	24.762	0	25	75	2.99
IR-2: Random Start	60 MHz	ABF	21.772	34.524	0	35	85	12.752
IR-2: Random Start	60 MHz	VBF	21.772	34.524	0	35	85	12.752
IR-2: Random Start	60 MHz	DFPBF	21.772	20.714	5	25	35	1.058
IR-2: Random Start	80 MHz	AED	21.772	72.143	3.75	63.75	210	50.371
IR-2: Random Start	80 MHz	ModAED	21.772	130.179	3.75	112.5	303.75	108.406
IR-2: Random Start	80 MHz	ModAEDS	21.772	130.179	3.75	112.5	303.75	108.406
IR-2: Random Start	80 MHz	AXIS	21.772	20.536	3.75	22.5	33.75	1.236
IR-2: Random Start	80 MHz	VAXIS	21.772	54.107	3.75	33.75	131.25	32.335
IR-2: Random Start	80 MHz	DFPAXIS	21.772	54.464	3.75	33.75	131.25	32.692
IR-2: Random Start	80 MHz	NSAXIS	21.772	20.714	3.75	22.5	33.75	1.058
IR-2: Random Start	80 MHz	SC	21.772	107.321	18.75	82.5	303.75	85.549
IR-2: Random Start	80 MHz	VSC	21.772	107.321	18.75	82.5	303.75	85.549
IR-2: Random Start	80 MHz	DFPSC	21.772	65.357	3.75	52.5	191.25	43.585
IR-2: Random Start	80 MHz	NSSC	21.772	98.929	7.5	67.5	236.25	77.156
IR-2: Random Start	80 MHz	CC	21.772	20.179	3.75	18.75	33.75	1.594
IR-2: Random Start	80 MHz	NBF	21.772	20.893	3.75	22.5	52.5	0.879
IR-2: Random Start	80 MHz	ABF	21.772	30.179	3.75	26.25	75	8.406
IR-2: Random Start	80 MHz	VBF	21.772	30.179	3.75	26.25	75	8.406
IR-2: Random Start	80 MHz	DFPBF	21.772	18.036	3.75	15	33.75	3.736
IR-2: Random Start	100 MHz	AED	21.772	72.143	6	42	186	50.371
IR-2: Random Start	100 MHz	ModAED	21.772	121	12	108	246	99.228
IR-2: Random Start	100 MHz	ModAEDS	21.772	118.857	12	108	246	97.085
IR-2: Random Start	100 MHz	AXIS	21.772	19.857	3	21	33	1.915
IR-2: Random Start	100 MHz	VAXIS	21.772	61.429	3	54	135	39.656
IR-2: Random Start	100 MHz	DFPAXIS	21.772	61.429	3	54	135	39.656
IR-2: Random Start	100 MHz	NSAXIS	21.772	19.857	3	21	33	1.915
IR-2: Random Start	100 MHz	SC	21.772	113.571	9	93	249	91.799

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	100 MHz	VSC	21.772	113.571	9	93	249	91.799
IR-2: Random Start	100 MHz	DFPSC	21.772	83.571	6	63	273	61.799
IR-2: Random Start	100 MHz	NSSC	21.772	113.571	9	93	249	91.799
IR-2: Random Start	100 MHz	CC	21.772	21.857	6	24	33	0.085
IR-2: Random Start	100 MHz	NBF	21.772	21.286	0	18	48	0.486
IR-2: Random Start	100 MHz	ABF	21.772	30.857	6	27	81	9.085
IR-2: Random Start	100 MHz	VBF	21.772	30.857	6	27	81	9.085
IR-2: Random Start	100 MHz	DFPBF	21.772	19	3	21	33	2.772
IR-2: Random Start	120 MHz	AED	21.772	56.31	5	27.5	132.5	34.537
IR-2: Random Start	120 MHz	ModAED	21.772	74.405	0	50	210	52.633
IR-2: Random Start	120 MHz	ModAEDS	21.772	74.405	0	50	210	52.633
IR-2: Random Start	120 MHz	AXIS	21.772	18.571	5	17.5	35	3.201
IR-2: Random Start	120 MHz	VAXIS	21.772	50.952	0	32.5	132.5	28.823
IR-2: Random Start	120 MHz	DFPAXIS	21.772	50.952	0	32.5	140	29.18
IR-2: Random Start	120 MHz	NSAXIS	21.772	18.571	5	17.5	35	3.201
IR-2: Random Start	120 MHz	SC	21.772	55.952	0	40	210	34.18
IR-2: Random Start	120 MHz	VSC	21.772	55.952	0	40	210	34.18
IR-2: Random Start	120 MHz	DFPSC	21.772	67.619	5	62.5	162.5	45.847
IR-2: Random Start	120 MHz	NSSC	21.772	55.952	0	40	210	34.18
IR-2: Random Start	120 MHz	CC	21.772	15.833	5	15	27.5	5.939
IR-2: Random Start	120 MHz	NBF	21.772	24.286	5	25	52.5	2.514
IR-2: Random Start	120 MHz	ABF	21.772	16.667	2.5	15	45	5.105
IR-2: Random Start	120 MHz	VBF	21.772	16.667	2.5	15	45	5.105
IR-2: Random Start	120 MHz	DFPBF	21.772	13.81	2.5	15	27.5	7.963
IR-2: Random Start	140 MHz	AED	21.772	65.918	17.143	51.429	150	44.146
IR-2: Random Start	140 MHz	ModAED	21.772	106.531	4.286	96.429	246.429	84.758
IR-2: Random Start	140 MHz	ModAEDS	21.772	106.531	4.286	96.429	246.429	84.758
IR-2: Random Start	140 MHz	AXIS	21.772	16.531	4.286	17.143	27.857	5.242
IR-2: Random Start	140 MHz	VAXIS	21.772	61.837	0	57.857	122.143	40.065
IR-2: Random Start	140 MHz	DFPAXIS	21.772	65.408	0	83.571	122.143	43.636

Continued on next page



Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	140 MHz	NSAXIS	21.772	17.143	4.286	17.143	27.857	4.629
IR-2: Random Start	140 MHz	SC	21.772	90.204	4.286	85.714	216.429	68.432
IR-2: Random Start	140 MHz	VSC	21.772	90.204	4.286	85.714	216.429	68.432
IR-2: Random Start	140 MHz	DFPSC	21.772	82.245	17.143	85.714	147.857	60.473
IR-2: Random Start	140 MHz	NSSC	21.772	88.673	4.286	85.714	214.286	66.901
IR-2: Random Start	140 MHz	CC	21.772	18.367	4.286	17.143	34.286	3.405
IR-2: Random Start	140 MHz	NBF	21.772	18.571	0	21.429	30	3.201
IR-2: Random Start	140 MHz	ABF	21.772	25.204	0	27.857	55.714	3.432
IR-2: Random Start	140 MHz	VBF	21.772	25.204	0	27.857	55.714	3.432
IR-2: Random Start	140 MHz	DFPBF	21.772	18.367	4.286	17.143	30	3.405
IR-2: Random Start	150 MHz	AED	21.772	91.333	10	94	226	69.561
IR-2: Random Start	150 MHz	ModAED	21.772	109.714	10	92	228	87.942
IR-2: Random Start	150 MHz	ModAEDS	21.772	120.476	10	92	282	98.704
IR-2: Random Start	150 MHz	AXIS	21.772	19.81	6	18	34	1.963
IR-2: Random Start	150 MHz	VAXIS	21.772	70.381	2	82	152	48.609
IR-2: Random Start	150 MHz	DFPAXIS	21.772	70.381	2	82	152	48.609
IR-2: Random Start	150 MHz	NSAXIS	21.772	19.81	6	18	34	1.963
IR-2: Random Start	150 MHz	SC	21.772	112.381	8	92	228	90.609
IR-2: Random Start	150 MHz	VSC	21.772	112.381	8	92	228	90.609
IR-2: Random Start	150 MHz	DFPSC	21.772	118.476	8	108	238	96.704
IR-2: Random Start	150 MHz	NSSC	21.772	112.381	8	92	228	90.609
IR-2: Random Start	150 MHz	CC	21.772	19.429	4	20	34	2.344
IR-2: Random Start	150 MHz	NBF	21.772	22.19	2	22	50	0.418
IR-2: Random Start	150 MHz	ABF	21.772	24	4	22	44	2.228
IR-2: Random Start	150 MHz	VBF	21.772	24	4	22	44	2.228
IR-2: Random Start	150 MHz	DFPBF	21.772	21.714	0	22	58	0.058
IR-2: Random Start	160 MHz	AED	21.772	75.179	15	60	243.75	53.406
IR-2: Random Start	160 MHz	ModAED	21.772	112.5	22.5	106.875	249.375	90.728
IR-2: Random Start	160 MHz	ModAEDS	21.772	113.214	22.5	106.875	249.375	91.442
IR-2: Random Start	160 MHz	AXIS	21.772	20.268	5.625	18.75	33.75	1.504

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	160 MHz	VAXIS	21.772	65.446	9.375	63.75	153.75	43.674
IR-2: Random Start	160 MHz	DFPAXIS	21.772	65.446	9.375	63.75	153.75	43.674
IR-2: Random Start	160 MHz	NSAXIS	21.772	20.268	5.625	18.75	33.75	1.504
IR-2: Random Start	160 MHz	SC	21.772	92.679	5.625	46.875	249.375	70.906
IR-2: Random Start	160 MHz	VSC	21.772	92.679	5.625	46.875	249.375	70.906
IR-2: Random Start	160 MHz	DFPSC	21.772	105.268	16.875	105	249.375	83.496
IR-2: Random Start	160 MHz	NSSC	21.772	97.5	5.625	67.5	249.375	75.728
IR-2: Random Start	160 MHz	CC	21.772	18.304	3.75	16.875	33.75	3.469
IR-2: Random Start	160 MHz	NBF	21.772	24.018	0	24.375	45	2.246
IR-2: Random Start	160 MHz	ABF	21.772	24.286	1.875	24.375	39.375	2.514
IR-2: Random Start	160 MHz	VBF	21.772	24.286	1.875	24.375	39.375	2.514
IR-2: Random Start	160 MHz	DFPBF	21.772	20.893	1.875	16.875	45	0.879
IR-2: Random Start	180 MHz	AED	21.772	91.905	0	108.333	165	70.133
IR-2: Random Start	180 MHz	ModAED	21.772	93.81	0	71.667	291.667	72.037
IR-2: Random Start	180 MHz	ModAEDS	21.772	94.048	0	71.667	291.667	72.275
IR-2: Random Start	180 MHz	AXIS	21.772	20.556	6.667	23.333	33.333	1.217
IR-2: Random Start	180 MHz	VAXIS	21.772	55.397	8.333	58.333	140	33.625
IR-2: Random Start	180 MHz	DFPAXIS	21.772	51.508	8.333	51.667	140	29.736
IR-2: Random Start	180 MHz	NSAXIS	21.772	20.556	6.667	23.333	33.333	1.217
IR-2: Random Start	180 MHz	SC	21.772	99.127	0	81.667	291.667	77.355
IR-2: Random Start	180 MHz	VSC	21.772	99.127	0	81.667	291.667	77.355
IR-2: Random Start	180 MHz	DFPSC	21.772	83.492	0	71.667	223.333	61.72
IR-2: Random Start	180 MHz	NSSC	21.772	103.81	0	88.333	291.667	82.037
IR-2: Random Start	180 MHz	CC	21.772	21.111	5	23.333	35	0.661
IR-2: Random Start	180 MHz	NBF	21.772	26.111	0	28.333	41.667	4.339
IR-2: Random Start	180 MHz	ABF	21.772	36.19	0	23.333	110	14.418
IR-2: Random Start	180 MHz	VBF	21.772	36.19	0	23.333	110	14.418
IR-2: Random Start	180 MHz	DFPBF	21.772	41.746	1.667	45	95	19.974
IR-2: Random Start	200 MHz	AED	21.772	105.857	6	115.5	234	84.085
IR-2: Random Start	200 MHz	ModAED	21.772	117.571	9	84	280.5	95.799

Continued on next page

Table D.1 – Continued from previous page

Simulation State	Sample Rate	Algorithm	$d_{1,2}$ (m)	$\hat{d}_{avg_{1,2}}$ (m)	$\hat{d}_{min_{1,2}}$ (m)	$\hat{d}_{med_{1,2}}$ (m)	$\hat{d}_{max_{1,2}}$ (m)	$\epsilon_{avg_{1,2}}$
IR-2: Random Start	200 MHz	ModAEDS	21.772	114.143	9	84	280.5	92.371
IR-2: Random Start	200 MHz	AXIS	21.772	18.357	4.5	16.5	40.5	3.415
IR-2: Random Start	200 MHz	VAXIS	21.772	70.143	4.5	70.5	132	48.371
IR-2: Random Start	200 MHz	DFPAXIS	21.772	70.143	4.5	70.5	132	48.371
IR-2: Random Start	200 MHz	NSAXIS	21.772	18.357	4.5	16.5	40.5	3.415
IR-2: Random Start	200 MHz	SC	21.772	117.643	9	84	280.5	95.871
IR-2: Random Start	200 MHz	VSC	21.772	117.643	9	84	280.5	95.871
IR-2: Random Start	200 MHz	DFPSC	21.772	98.857	3	66	270	77.085
IR-2: Random Start	200 MHz	NSSC	21.772	117.643	9	84	280.5	95.871
IR-2: Random Start	200 MHz	CC	21.772	19.357	4.5	19.5	34.5	2.415
IR-2: Random Start	200 MHz	NBF	21.772	22.286	4.5	21	84	0.514
IR-2: Random Start	200 MHz	ABF	21.772	50.143	13.5	42	108	28.371
IR-2: Random Start	200 MHz	VBF	21.772	50.143	13.5	42	108	28.371
IR-2: Random Start	200 MHz	DFPBF	21.772	57.786	1.5	51	145.5	36.014