Array signal processing using spatial domain infinite impulse response structure and volterra model

Wen, Fuxi

2013

Wen, F. (2013). Array signal processing using spatial domain infinite impulse response structure and volterra model. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/51128>

<https://doi.org/10.32657/10356/51128>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ARRAY SIGNAL PROCESSING USING SPATIAL
DOMAIN INFINITE IMPULSE RESPONSE STRUCTURE
AND VOLTERRA MODEL**

WEN FUXI

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2013

ARRAY SIGNAL PROCESSING USING SPATIAL DOMAIN INFINITE IMPULSE RESPONSE STRUCTURE AND VOLTERRA MODEL

WEN FUXI

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2013

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

2013 / 01 / 16

.....

Date

Fux

.....

Wen Fuxi

Acknowledgments

I would like to thank the following people who have been assisting me during my research.

First of all, I cordially express my sincere gratefulness to my supervisor, Associate Professor Ng Boon Poh, for his great supervision and assiduous instructions to my research, for his patience, suggestion and understanding to my profession growth. Without his broad vision, deep insight, valuable advice and strong encouragement, this work would not have been possible. I am also especially indebted to him for his great concern in matters outside of the academic research.

I also wish to express my sincere gratitude to my fellow graduate students and friends, Huiping, Joni, Aye Aung, Zhang Ying, Vinod, Wang Tao, Tu Lei and Su Yee, for their valuable discussions and sharing their ideas with me during the progress of this research. Their care and friendship have been invaluable to me. I would like to thank all my friends Wang Mingqiang, Wang Lu, Qian Jiang, Pan Junyang, who have given me the emotional support and ideas for my research.

The appreciation is also extended to the technical staff in Information Systems Research Lab, Suat Geok and Eng Teck for their helps, provisions of equipments, their advices and their efforts in providing a friendly research environment. I also would like to acknowledge gratefully to NTU, Singapore, for awarding me the university Research Scholarship for the past four years.

Finally I would like to thank my family members for their unconditional loves. I would like to say special thanks to my wife Danyao, for her unselfishness love. I could do nothing without her trust, support and encouragement. She makes me complete.

Summary

In this thesis, we discuss array processing using infinite impulse response (IIR) and Volterra signal model in spatial domain. Array beamforming can be interpreted as a spatial filter which allows signals from specific directions to pass through it while suppressing the interference signals. This is analogous to a time-domain digital filter which allows signals to pass through within its passband. The equivalence arises because a digital filter samples the signal in time domain whereas an array samples the signal in spatial domain.

With such close relation between array snapshots and discrete-time signals, it is interesting to explore the application of concepts and algorithms developed in different streams of research for array processing. It is well acclaimed that digital filters with desired frequency response can be implemented with either finite impulse response (FIR) structure or IIR structure. In array processing, a one-to-one correspondence of these filter structures with array structures has not been exhaustively studied.

Following the analogy between FIR filter structure and a uniform linear array (ULA), the question in hand is: “what is the array structure in spatial domain that is analogous to IIR filter structure?”. We first present the possibility of using an approximate IIR structure for DOA estimation and beamforming applications. We propose a new DOA estimation technique which incorporates virtual feedback resulting in an IIR structure. While the time delay between the taps of the discrete-time IIR spatial structure in the feedback and feed-forward networks is a constant proportional to the sampling interval, the analogous delay in spatial domain corresponds to the relative propagation delay across the sensor array which is proportional to the source direction. In the IIR array structure, array snapshots can be directly used for the moving average part of the filter. In order to develop the feedback section of the IIR structure, we propose a virtual structure with an approximate propagation delay. The performance of the proposed technique is observed to have improved performance over the Capon method.

Furthermore, we propose a primitive beamformer with an IIR array implementation. Although the configuration presented in this work is not optimal, the scope of this contribution is to prove the concept of IIR structured beamformer. In order to implement a genuine IIR array, it is necessary to have spatial feedback which can be realized using the concept of subarrays. The output of previous subarrays can be used for spatial feedback. The stability of the IIR structure is ensured by using a notch filter structure which constrains the zeros and poles to lie within the unit circle. Although the beamformer capability is achieved, the subarray feedback requires an increased array size.

In previous introduction, linear Gaussian model is considered. The obvious advantage of linear systems is their inherent simplicity. Linear filters are known to be optimal if noise is Gaussian. However, in real applications, a linear model is not always adequate, for example, in communication systems and biological signal processing. The Volterra model is one of the widely used models for nonlinear systems. Both the FIR and IIR Volterra filters in time domain are well studied, however, as far as we know, the FIR and IIR Volterra structure in spatial domain have not been discussed yet. To fill the gaps between Volterra filters in time domain and Volterra model in spatial domain, we extend the FIR and IIR Volterra model in time domain to spatial domain.

Base on the spatial Volterra model, we proposed two narrowband DOA estimation methods, namely, Volterra Capon and Volterra MUSIC. To deal with the correlated or coherent sources, spatial smoothing based Volterra Capon and Volterra MUSIC are proposed.

At last, we generalize our works. The DOA estimator using the IIR notch structure is extended to the generalized IIR structure. The marriage between IIR and Volterra based processings are obtained by using the IIR Volterra model in spatial domain. Joint space and time processing is considered, generalized models consist of an FIR or IIR Volterra structure in spatial domain followed by an FIR or IIR Volterra signal model in time domain are proposed. All the models introduced in the previous Chapters can be classified as special cases of the unified joint spatial- and time-domain Volterra model.

Table of Contents

Acknowledgments	i
Summary	ii
Table of Contents	iv
List of Figures	viii
List of Tables	xiv
List of Abbreviations	xv
List of Notations	xvi
1 Introduction	1
1.1 Background	1
1.2 Signal Models	2
1.3 Volterra Signal Model	6
1.3.1 Volterra Signal Model in Time Domain	7
1.3.2 Volterra Signal Model in Spatial Domain	8
1.4 Motivations and Contribution	10
1.4.1 Array Processing Using Approximate Spatial IIR Structure	10
1.4.2 DOA Estimation using Volterra Signal Model in Spatial Domain	11
1.4.3 Some Extensions for IIR and Volterra Array Processing	12
1.5 Organization of the Thesis	13
2 Narrowband DOA Estimation Techniques	14
2.1 Quadratic Algorithms	15
2.1.1 Beamscan Approach	15
2.1.2 Maximum Entropy	17
2.1.3 Capon	17
2.2 Subspace Based Methods	19
2.2.1 MUSIC	19
2.2.2 ESPRIT	22

2.2.3	Matrix Pencil	24
2.3	Statistical Methods	28
2.3.1	Stochastic Maximum Likelihood Estimator	28
2.3.2	Deterministic Maximum Likelihood Estimator	30
2.4	Adaptive DOA Estimation Techniques	31
3	Volterra Capon DOA Estimation	32
3.1	Introduction	32
3.2	The Volterra Capon DOA Estimation Algorithm	34
3.3	Theoretical Performance Analysis	36
3.3.1	Asymptotic Bias	37
3.3.2	Variance	38
3.4	Simulation Results and Discussions	41
3.4.1	Complex Gaussian Sources	42
3.4.2	Complex Non-Gaussian Sources	47
3.4.3	Detection Probability	52
3.5	Conclusion	56
4	Volterra MUSIC DOA Estimation	58
4.1	Introduction	58
4.2	The Volterra MUSIC DOA Estimation Algorithm	58
4.2.1	Summary of the proposed DOA Estimation Algorithm	61
4.2.2	Comparison with the 2q-MUSIC Algorithm	61
4.3	Numerical Results	63
4.3.1	Simulation Results for the Proposed V-MUSIC Method	63
4.3.2	Simulation Results for the Order Selection Problem	65
4.4	Performance Analysis	68
4.4.1	Variance and Bias of the Proposed Method	69
4.4.2	Probability of Resolution of the Proposed Method	70
4.5	Conclusions	72
5	Spatial Smoothing for Volterra Based Methods	75
5.1	Introduction	75
5.1.1	Forward Spatial Smoothing	76
5.1.2	Forward-Backward Spatial Smoothing	76
5.2	V-Capon and V-MUSIC with Spatial Smoothing	77
5.2.1	Forward spatial smoothing for Volterra based methods	77

5.2.2	Forward-backward spatial smoothing for Volterra based methods . . .	78
5.3	Numerical Results for V-MUSIC with Spatial Smoothing	79
5.3.1	Complex Gaussian Sources	79
5.3.2	Complex Non-Gaussian Sources	83
5.4	Numerical Results for V-Capon with Spatial Smoothing	88
5.4.1	Complex Gaussian Sources	88
5.4.2	Complex Non-Gaussian Sources	93
5.5	Conclusions	97
6	Array Processing Using Spatial IIR Structure	98
6.1	Introduction	98
6.2	Equivalence Between An FIR Filter and A Uniform Linear Array	101
6.3	DOA Estimation Using Spatial Domain IIR Notch Structure	103
6.3.1	Adaptive IIR Notch Filtering	103
6.3.2	The Proposed DOA Estimation Technique	105
6.3.3	Simulation Results	107
6.3.4	Performance Analysis	110
6.4	Implementation of IIR Array Processing Structure	115
6.4.1	Simulation Results for Different SNR	121
6.5	Conclusion	125
7	Extensions for IIR and Volterra Array Processing	126
7.1	Introduction	126
7.2	FIR Volterra Model	127
7.3	IIR Volterra Model	129
7.3.1	IIR Volterra Filter	129
7.3.2	IIR Volterra Model in Spatial Domain	130
7.4	Volterra Based Joint Space and Time Array Processing	132
7.4.1	Spatial FIR Volterra and Time domain FIR Volterra	133
7.4.2	Spatial FIR Volterra and Time domain IIR Volterra	135
7.4.3	Spatial IIR Volterra and Time domain FIR Volterra	135
7.4.4	Spatial IIR Volterra and Time-domain IIR Volterra	137
7.5	Conclusion	139
8	Conclusion and Recommendations	140
8.1	Conclusion	140
8.2	Recommendations for Further Research	141

8.2.1	Array Processing Using Approximate Spatial IIR Structure	141
8.2.2	Volterra based Joint Space and Time Processing	142
Appendix A Mean Squared Error of DOA Estimate		143
Appendix B Performance Analysis of the Proposed Method		144
B.1	Bias Analysis of the Proposed DOA Estimator	149
B.2	MSE Analysis of the Proposed DOA Estimator	150
Appendix C Dimension of the Signal Subspace N_s		153
Author's Publications		156
Bibliography		157

List of Figures

1.1	A uniform linear array with M omni-directional sensor elements.	3
3.1	Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive white Gaussian noise (AWGN) is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	43
3.2	RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise is assumed, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	44
3.3	Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	45
3.4	RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise is assumed, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	46
3.5	Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	48

3.6	RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	49
3.7	Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	50
3.8	RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are form 60° , 80° and 110° , additive complex Gaussian noise, 500 independent trials with 100 snapshots each are averaged to obtain the plots.	51
3.9	Detection probability for Capon and 2nd order Volterra Capon methods with regard to SNR. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex Gaussian and the noise is additive white Gaussian.	53
3.10	Spatial spectrum for Capon and 2nd order Volterra Capon methods. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex Gaussian and the noise is additive white Gaussian, SNR = 22 dB.	54
3.11	Detection probability for Capon and 2nd order Volterra Capon methods with regard to SNR. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex non-Gaussian and the noise is additive white Gaussian.	55
3.12	Spatial spectrum for Capon and 2nd order Volterra Capon methods. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex non-Gaussian and the noise is additive white Gaussian, SNR = 25 dB.	56
4.1	Eigenvalues of $\mathbf{R}_v$. A ULA of 4 elements, 1, 2 and 3 uncorrelated source signals, additive white Gaussian noise, SNR = 30dB, averaged over 100 independent trials.	60
4.2	Snapshots dimension comparison of the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm.	63

4.3	Spatial spectra obtained using the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm. A ULA of 4 elements, SNR=20dB, 50 snapshots are used, 2 non-Gaussian source signals are from 100° and 110° , averaged over 500 independent trials.	64
4.4	Spatial spectra obtained using the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm. A ULA of 4 elements, SNR=20dB, 50 snapshots are used, 2 non-Gaussian source signals are from 100° and 103° , averaged over 500 independent trials.	65
4.5	Spatial spectra obtained using the standard MUSIC algorithm, the dimension of the noise subspace is different. A ULA of 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 10 independent trials.	66
4.6	Spatial spectra obtained using the 2q-MUSIC algorithm, the dimension of the noise subspace are different. A ULA with 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 500 independent trials.	67
4.7	Spatial spectra obtained using the proposed V-MUSIC algorithm, the dimension of the noise subspace are different. A ULA with 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 500 independent trials.	68
4.8	Standard deviations of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.	71
4.9	Bias of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.	72
4.10	RMSE of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.	73
4.11	Probability of Resolution of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR.	74
5.1	Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	80

5.2	RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	81
5.3	Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	82
5.4	RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	83
5.5	Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	84
5.6	RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	85
5.7	Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	86
5.8	RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	87
5.9	Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	89
5.10	RMSEs of Capon and the proposed V-Capon algorithm (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60°, 80° and 120°, the first 2 sources are coherent, averaged over 500 independent trials.	90

5.11	Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	91
5.12	RMSEs of Capon and the proposed V-Capon algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	92
5.13	Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	93
5.14	RMSEs of Capon and the proposed V-Capon algorithm (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	94
5.15	Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	95
5.16	RMSEs of Capon and the proposed V-Capon algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.	96
6.1	Spatial sampling of the incoming wavefront by a ULA.	101
6.2	Tapped-delay line implementation of an FIR filter.	102
6.3	The proposed structure for DOA estimation.	106
6.4	DOA estimation using different methods with sources from 50° and 60°	110
6.5	DOA estimation using different methods with sources from 60° and 90°	111
6.6	DOA estimation using real data, 50 snapshots are used.	112
6.7	DOA estimation using real data, 100 snapshots are used.	113
6.8	Bias of the IIR notch estimator.	114
6.9	Standard deviation of the IIR notch estimator.	115
6.10	The structure of the IIR array.	116

6.11	The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array.	120
6.12	The spatial response and the z-plane of the IIR array.	121
6.13	The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. SNR = 0 dB.	122
6.14	The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. SNR = 10 dB.	123
6.15	The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. SNR = 20 dB.	124
7.1	Second order FIR Volterra model in spatial domain	128
7.2	Second order FIR Volterra module in spatial domain	128
7.3	Second order IIR Volterra model in spatial domain	131
7.4	Second order IIR Volterra module in spatial domain	132
7.5	Spatial FIR Volterra and Time domain FIR Volterra	134
7.6	Spatial FIR Volterra and Time domain IIR Volterra	136
7.7	Spatial IIR Volterra and Time domain FIR Volterra	137
7.8	Spatial IIR Volterra and Time domain IIR Volterra	139

List of Tables

6.1 Analogy between filtering and beamforming. 99

6.2 Complex Adaptive IIR Notch Algorithm 108

C.1 The dimension of the signal subspace and the number of noise eigenvectors. . 155

List of Abbreviations

DOA	Direction of Arrival	4
ULA	Uniform Linear Array	4
FIR	Finite Impulse Response	10
IIR	Infinite Impulse Response	10
LMS	Least Mean Square	11
RLS	Recursive Least Squares	11
RMSE	Root Mean Square Error	11
MUSIC	Multiple Signal Classification	11
ML	Maximum Likelihood	14
WSF	Weighted Subspace Fitting	15
ESPRIT	Estimation Signal Parameter via a Rotational Invariant Technique	15
MP	Matrix Pencil	15
ME	Maximum Entropy	17
MVDR	Minimum Variance Distortionless Response	18
FB	Forward-Backward	19
IQML	Iterative Quadratic Maximum Likelihood	31
MSE	Mean Squared Error	33
AWGN	Additive white Gaussian Noise	43
HOS	Higher Order Statistics	44
SNR	Signal-to-Noise Ratio	44
PDF	Probability Density Function	47

List of Notations

$\mathcal{C}$	Complex number	4
k	Discrete time index	4
$\mathbf{x}(k)$	Array received snapshot	4
$\mathbf{n}(k)$	Array received noise vector	4
M	Number of sensors in the array	4
θ	Direction of arrival	4
$\mathbf{a}(\theta)$	Array steering vector	4
d	Inter-element distance of uniform linear array	4
λ	Wavelength of the signal	4
$\mathbf{A}(\theta)$	Array manifold matrix	4
L	Number of signals	4
$\mathbf{s}(k)$	Signal vector	4
$\mathbf{R}$	Covariance matrix	5
$\mathbf{P}_s$	Autocorrelation matrix of the source signals	5
$\mathbf{P}_n$	Autocorrelation matrix of the noise signals	5
$\mathbb{E}[\cdot]$	Expectation operator	5
$\mathcal{R}$	Real number	5
$\mathbf{I}$	Identity matrix	6
$(\cdot)^H$	Conjugate transposition	5
$(\cdot)^T$	Transposition	8
$\otimes$	Kronecker product	8
$\mathbf{x}_v(k)$	Volterra snapshot	8
$x_i(k)$	The i <i>th</i> element of $\mathbf{x}(k)$	8
$\boxtimes$	Modified Kronecker product	8
$(\cdot)^*$	Complex conjugate	17
$\mathbf{J}$	Exchange matrix	17
$(\cdot)^{-1}$	Matrix inverse	18

$\mathbf{U}_s$	Signal subspace	20
$\mathbf{U}_n$	Noise subspace	20
$\ln(\cdot)$	Natural logarithm	27
$\Im(\cdot)$	Imaginary part of a complex number	27
$(\cdot)^\dagger$	Moore-Penrose pseudoinverse	29
$\text{tr}[\cdot]$	Trace	28
$\det[\cdot]$	Determinant	29
$\mathbf{R}_v$	Volterra covariance	34
$\mathbf{a}_v(\theta)$	Volterra array steering vector	34
$\hat{\mathbf{R}}_v$	Estimated Volterra covariance	36

Chapter 1

Introduction

1.1 Background

Array processing has played an important role in many diverse application areas. Such as radar and sonar, communication systems, radio astronomy, seismic, medical diagnosis and treatment [1,2].

Radar and sonar Most modern radar and sonar systems rely on antenna arrays or hydrophone arrays as an essential component of the system.

Communication Many communication systems utilize phased arrays or multiple beam antennas to achieve their performance objectives. It appears that the third generation of wireless systems will utilize adaptive array processing techniques to achieve the desired system capacity. Smart antenna systems are also a defining characteristic of MIMO systems, such as the IEEE 802.11n standard [3]. Conventionally, a smart antenna is a unit of a wireless communication system and performs spatial signal processing with multiple antennas [4]. Multiple antennas can be used at either the transmitter or receiver. Recently, the technology has been extended to use the multiple antennas at both the transmitter and receiver. Such a system is called a multiple-input multiple-output (MIMO) system.

Radio astronomy Radio astronomy is conducted using large radio antennae referred to as radio telescopes, that are either used singularly, or with multiple linked telescopes utilizing the techniques of radio interferometry and aperture synthesis. The use of interferometry allows radio astronomy to achieve high angular resolution, as the resolving power of an interferometer is set by the distance between its components, rather than the size of its components.

Seismic Seismic arrays can be used to precisely locate, in three dimensions, the source of an earthquake, using the time it takes for seismic waves to propagate away from the hypo center, the initiating point of fault rupture. Interconnected seismic arrays are also used to detect underground nuclear test explosions.

Medical Various medical diagnosis and treatment techniques exploit arrays. For example, diagnostic sonography (ultrasonography) which is an ultrasound-based diagnostic imaging technique used for visualizing subcutaneous body structures including tendons, muscles, joints, vessels and internal organs for possible pathology or lesions. Obstetric sonography is commonly used during pregnancy and is widely recognized by the public.

1.2 Signal Models

Before we introduce the conventional narrowband array signal processing methods, such as narrowband DOA estimation and beamforming, it is necessary to present the mathematical model for the problem. We start with the most basic case, the location of narrowband sources in the far-field of a uniform linear array (ULA). Let the uniform linear array in consideration consist of M omni-directional sensor elements with equal inter-element spacing d , which is illustrated in Figure 1.1.

For convenience, we first assume that there are one desired source, which is narrowband and farfield. The narrowband desired source signal is from direction θ and denoted as

Figure 1.1: A uniform linear array with M omni-directional sensor elements.

$s(t)e^{j\omega t}$, then the snapshot collected at sensor m is

$$x_m(t) = s(t - \tau_m)e^{j\omega(t - \tau_m)}, \quad (1.1)$$

where τ_m is the time delay of the snapshot collected at the m th sensor, with regards to the reference sensor. In the following discussion, the first sensor is selected as the reference sensor.

For simplicity, assuming the desired signal is slow time-varying compared to the carrier $e^{j\omega t}$, and dropping the carrier term $e^{j\omega t}$, a simplified model can be obtained as

$$x_m(t) \cong s(t)e^{-j\omega\tau_m}. \quad (1.2)$$

The time delay of sensor m is obtained by $\tau_m = \frac{(m-1)d\cos(\theta)}{c}$, $1 \leq m \leq M$, where c is the electromagnetic wave velocity. Since $c = \lambda f$, where λ is the wavelength and f is the

frequency of the signal, the signal model in (1.1) is reformulated as

$$\begin{aligned} x_m(t) &= s(t)e^{-j2\pi f \frac{(m-1)d \cos(\theta)}{c}} \\ &= s(t)e^{-j \frac{2\pi d}{\lambda} (m-1) \cos(\theta)}. \end{aligned} \quad (1.3)$$

In real applications, signals are processed in discrete form. Analog-to-digital converters (A/D) are used to convert continuous quantities to a discrete-time digital representations. Without loss of generality, here analog time index t is replaced by discrete-time index k . For a single source, the snapshot at time k is given by

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{a}(\theta)s(k) + \mathbf{n}(k) \\ &= [x_1(k), x_2(k), \dots, x_M(k)]^T \in \mathcal{C}^{M \times 1}, \end{aligned} \quad (1.4)$$

where M is the number of elements of the array and $\mathbf{a}(\theta)$ is the array steering vector defined as

$$\mathbf{a}(\theta) = [a_1(\theta) \ a_2(\theta) \ \dots \ a_M(\theta)]^T \in \mathcal{C}^{M \times 1}. \quad (1.5a)$$

For a uniform linear array (ULA),

$$\mathbf{a}(\theta) = [1 \ e^{-j \frac{2\pi d}{\lambda} \cos(\theta)} \ \dots \ e^{-j \frac{2\pi d}{\lambda} (M-1) \cos(\theta)}]^T, \quad (1.5b)$$

where λ and d are the wavelength of the carrier and inter-element spacing, respectively.

For multiple sources, assuming that the L narrowband signals impinge on the array from directions $\{\theta_1, \theta_2, \dots, \theta_L\}$. The snapshot vector $\mathbf{x}(k)$ of the array at time k is defined as

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}(\theta)\mathbf{s}(k) + \mathbf{n}(k) \\ &= [x_1(k), x_2(k), \dots, x_M(k)]^T \in \mathcal{C}^{M \times 1}, \end{aligned} \quad (1.6)$$

where the superscript T denotes transposition, $\mathbf{A}(\theta) \in \mathcal{C}^{M \times L}$ is the array manifold matrix,

consisting of L array steering vectors corresponding to the L directions, $\mathbf{s}(k) \in \mathcal{C}^{L \times 1}$ is the signal vector which consists of L different incidence signals, and $\mathbf{n}(k) \in \mathcal{C}^{M \times 1}$ is the noise vector. The noise component is assumed to be the white Gaussian noise, the mean is zero and variance is δ_n^2 .

$\mathbf{A}(\theta)$, $\mathbf{s}(k)$ and $\mathbf{n}(k)$ can be expressed as

$$\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_L)] \in \mathcal{C}^{M \times L}, \quad (1.7)$$

$$\mathbf{s}(k) = [s_1(k), s_2(k), \dots, s_L(k)]^T \in \mathcal{C}^{L \times 1}, \quad (1.8)$$

and

$$\mathbf{n}(k) = [n_1(k), n_2(k), \dots, n_M(k)]^T \in \mathcal{C}^{M \times 1}, \quad (1.9)$$

respectively. Here $\mathbf{a}(\theta_i)$, $\{i = 1, 2, \dots, L\}$, are the steering vectors of $s_i(k)$ coming from θ_i .

The spatial covariance matrix $\mathbf{R} \in \mathcal{C}^{M \times M}$ is calculated by

$$\begin{aligned} \mathbf{R} &= \text{E} [\mathbf{x}(k)\mathbf{x}^H(k)] \\ &= \mathbf{A}(\theta)\text{E} [\mathbf{s}(k)\mathbf{s}^H(k)] \mathbf{A}^H(\theta) + \text{E} [\mathbf{n}(k)\mathbf{n}^H(k)] \\ &= \mathbf{A}(\theta)\mathbf{P}_s\mathbf{A}^H(\theta) + \mathbf{P}_n, \end{aligned} \quad (1.10)$$

where the superscript H denotes conjugate transposition, $\mathbf{P}_s \in \mathcal{R}^{L \times L}$ and $\mathbf{P}_n \in \mathcal{R}^{M \times M}$ are the autocorrelation matrix of the source and noise signals, respectively.

For additive white Gaussian noise (AWGN),

$$\mathbf{P}_n = \begin{pmatrix} \delta_n^2 & 0 & \cdots & 0 \\ 0 & \delta_n^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \delta_n^2 \end{pmatrix} = \delta_n^2 \mathbf{I}_M, \quad (1.11)$$

where $\mathbf{I}_M$ is the identity matrix or unit matrix of size M and δ_n^2 is the power of the noise.

1.3 Volterra Signal Model

In previous section, linear Gaussian model is considered. The obvious advantage of linear systems is their inherent simplicity. Linear filters are known to be optimal if noise is Gaussian. However, in real applications the linear model is not always adequate. Here are some examples:

1. Communications systems

Bit errors in high-speed communications systems are almost entirely caused by nonlinear mechanisms. Furthermore, satellite communication channel is typically modeled as a memoryless nonlinear channel.

2. Biological signal processing

Biological systems are inherently nonlinear, and modeling such systems such as the human visual system requires nonlinear models.

In these cases, the performance of linear filters may become unacceptable. Volterra model is one of the widely used models for nonlinear systems. Furthermore, the classical formulation of linear adaptive filters can be easily extended to this model, such as LMS Volterra filter [5], RLS Volterra filter [6], QR decomposition based [7] and orthogonal structure [8].

Here we give a short introduction about the Volterra signal model.

1.3.1 Volterra Signal Model in Time Domain

For a general case, the output of a discrete-time nonlinear system $y(k)$ can be represented as a Volterra series expansion of the input signal $x(k)$,

$$\begin{aligned}
 y(k) = & \omega_0 + \sum_{l_1=0}^{\infty} \omega_1(l_1)x_{l_1}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} \sum_{l_3=0}^{\infty} \omega_3(l_1, l_2, l_3)x_{l_1}(k)x_{l_2}(k)x_{l_3}(k) \\
 & + \cdots,
 \end{aligned} \tag{1.12}$$

where ω_0 is meant as the causal impulse response of the zero-order, where $\omega_i(l_1, l_2, \dots, l_i)$ are the coefficients of the nonlinear combiner model based on the Volterra series and $x_{l_i}(k) = x(k - l_i)$ is the input signal at time $k - l_i$, for $\{i = 1, 2, \dots, \infty\}$.

In real applications, truncated Volterra model is used. For second order truncated Volterra series, the output $y_2(k)$ is given by

$$y_2(k) = \sum_{l_1=0}^M \omega_1(l_1)x_{l_1}(k) + \sum_{l_1=0}^M \sum_{l_2=0}^M \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k), \tag{1.13}$$

where M is the number of time delay elements.

For third order truncated Volterra series, the output $y_3(k)$ is given by

$$\begin{aligned}
 y_3(k) = & \sum_{l_1=0}^M \omega_1(l_1)x_{l_1}(k) \\
 & + \sum_{l_1=0}^M \sum_{l_2=0}^M \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k) \\
 & + \sum_{l_1=0}^M \sum_{l_2=0}^M \sum_{l_3=0}^M \omega_3(l_1, l_2, l_3)x_{l_1}(k)x_{l_2}(k)x_{l_3}(k).
 \end{aligned} \tag{1.14}$$

1.3.2 Volterra Signal Model in Spatial Domain

In this section, we extend the Volterra signal model to spatial domain, which is one of our contributions. Furthermore, robust array signal processing methods are proposed base on the extended model. Since Volterra signal model in spatial domain is one of the theoretical foundations of this thesis, most of the methods introduced in the following Chapters will be using this model.

For simplicity, only second-order and third-order Volterra series are considered, as the extension for higher-order case is straightforward.

Second Order Volterra Model in spatial Domain

From (1.13), the new Volterra snapshot at time k is defined as

$$\mathbf{x}_v(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \otimes \mathbf{x}(k))^T \right]^T, \quad (1.15)$$

where $(\cdot)^T$ denotes transpose and $\otimes$ denotes the Kronecker product. The dimension of a Volterra snapshot $\mathbf{x}_v(k)$ is $(M + M^2) \times 1$.

From (1.15), we can see that the second order terms $x_i(k)x_j(k) = x_j(k)x_i(k)$, $1 \leq i, j \leq M$. To reduce the computation complexity, we can choose $x_i(k)x_j(k)$, $1 \leq i \leq j \leq M$ to form the second-order terms.

In order to express the new Volterra snapshot simply, we introduce the operator $\boxtimes$, which is a modified Kronecker product. For a vector $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$, the modified Kronecker product $\mathbf{b} \boxtimes \mathbf{b}$ is defined as

$$\begin{aligned} \mathbf{b} \boxtimes \mathbf{b} &= \left[\underbrace{b_1 b_1, \dots, b_1 b_N}_{N \text{ terms}}, \underbrace{b_2 b_2, \dots, b_2 b_N}_{(N-1) \text{ terms}}, \dots, \underbrace{b_N b_N}_{1 \text{ term}} \right]^T \\ &= [b_1 b_1, b_1 b_2, \dots, b_i b_j, \dots, b_{N-1} b_N, b_N b_N]^T, \end{aligned} \quad (1.16)$$

where i and j are in the lexicographic order ($1 \leq i \leq j \leq N$) and b_i is the i^{th} element of $\mathbf{b}$.

The dimension of $\mathbf{b} \boxtimes \mathbf{b}$ is $\binom{N+1}{N-1} \times 1$.

The new second order Volterra snapshot is given by

$$\begin{aligned} \mathbf{x}_v(k) &= [x_1(k), \dots, x_M(k), \\ &\quad x_1(k)x_1(k), x_1(k)x_2(k), \dots, x_1(k)x_M(k), \\ &\quad x_2(k)x_2(k), \dots, x_2(k)x_M(k), \\ &\quad \vdots \\ &\quad x_M(k)x_M(k)]^T \\ &= \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T, \end{aligned} \quad (1.17)$$

where $1 \leq i \leq M$. $\mathbf{x}_v(k)$ is a $N \times 1$ vector and

$$N = M + \frac{M(M+1)}{2} = \frac{M(M+3)}{2}. \quad (1.18)$$

Then the new Volterra covariance matrix is obtained by

$$\mathbf{R}_v = E [\mathbf{x}_v(k) \mathbf{x}_v^H(k)]. \quad (1.19)$$

The corresponding new Volterra array steering vector is defined as

$$\mathbf{a}_v(\theta) = \left[\mathbf{a}^T(\theta), (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T \right]^T. \quad (1.20)$$

Third Order Volterra Model in spatial Domain

In order to express the new Volterra snapshot simply, the third order modified Kronecker product is introduced.

For a vector $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$, the third order modified Kronecker product $\mathbf{b} \boxtimes \mathbf{b} \boxtimes \mathbf{b}$ is defined as

$$\mathbf{b} \boxtimes \mathbf{b} \boxtimes \mathbf{b} = [b_1 b_1 b_1, \dots, b_i b_j b_k, \dots, b_N b_N b_N]^T, \quad (1.21)$$

where i, j and k are in the lexicographic order ($1 \leq i \leq j \leq k \leq N$). The dimension of $\mathbf{b} \boxtimes \mathbf{b} \boxtimes \mathbf{b}$ is $\binom{N+2}{N-1} \times 1$.

Similar to the second order Volterra snapshot, the new third order Volterra snapshot at time k is defined as

$$\mathbf{x}_v(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T, (\mathbf{x}(k) \boxtimes \mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T. \quad (1.22)$$

Then the new Volterra covariance matrix can be computed using (1.19).

The steering vector for third-order Volterra snapshots is given by

$$\mathbf{a}_v(\theta) = \left[\mathbf{a}^T(\theta), (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T, (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T \right]^T. \quad (1.23)$$

1.4 Motivations and Contribution

1.4.1 Array Processing Using Approximate Spatial IIR Structure

Various array processing techniques applied to uniform linear arrays are involuntarily realized using structures that are analogous to FIR filters. This observation rises the question, “can we extend IIR filtering to IIR array processing?” In Chapter 3, we introduce the concept of IIR array in spatial domain. Note that IIR array here does not mean time-domain IIR filtering for array beamforming which is commonly understood. This part is dedicated to the study of an alternative approach for array signal processing, which defines IIR structure in spatial domain.

To illustrate the applicability of the concept of IIR array, we propose a new DOA estimation technique as well as a primitive beamformer with the spatial domain IIR array implementation. The performance of the proposed methods are comparable to the exiting techniques. These illustrations are intended to introduce a new approach, which potentially, can offer more degrees of freedom to control the performance of the array and reduce complexity of the system for a desired performance.

1.4.2 DOA Estimation using Volterra Signal Model in Spatial Domain

Volterra model is one of the widely used models for nonlinear systems for several reasons. Firstly, it can deal with a general class of nonlinear systems while its output is still linear with respect to various higher order system kernels. Furthermore, the classical formulation of linear adaptive filters can be easily extended to fit this model, such as LMS Volterra filter, RLS Volterra filter, QR decomposition based and orthogonal structure. However, the computational complexity increases when the order of Volterra model increases.

The Volterra polynomial models have been applied to a variety of applications, such as system identification, adaptive filtering, communication, biological system modeling, noise canceling, echo cancellation and prediction modeling. However, the Volterra signal model has not been applied in spatial domain yet. In this thesis, we extend the Volterra signal model in time domain to spatial domain.

1. In Chapter 4, base on the Volterra signal model in spatial domain, we propose a new Capon DOA estimation technique using Volterra signal model in spatial domain. The new technique basically uses the high order terms of Volterra series to produce augmented Volterra snapshots. The proposed technique has an improved resolution and a lower root mean square error (RMSE) when compared with the standard Capon algorithm.
2. In Chapter 5, the Volterra based MUSIC algorithm is also considered. The new technique basically uses additionally the second-order terms of Volterra series to produce augmented Volterra snapshots, the extension for higher-order case is straightforward. The resolution of the proposed method is high comparing with the standard MUSIC algorithm and the 2q- MUSIC algorithm. Simulation results are presented to show the performance of the proposed technique.
3. In Chapter 6, the proposed Volterra Capon and Volterra MUSIC algorithms are further extended to deal with the correlated or coherent sources, by using spatial smooth-

ing, such as forward spatial smoothing or forward-backward spatial smoothing. For Volterra Capon with spatial smoothing, compared with the standard Capon with spatial smoothing, better performance is obtained when considering channel fading effects. Similar conclusions can be drawn for Volterra MUSIC when compared with the standard MUSIC algorithm with spatial smoothing.

1.4.3 Some Extensions for IIR and Volterra Array Processing

In Chapter 7, we discuss some extensions for IIR and Volterra array processing. We first exploit the possibility of DOA estimation technique using approximate generalized spatial IIR structure, the IIR notch constrain is removed. The method discussed in Chapter 3 is further extended to a generalized structure.

The Volterra model discussed in the previous Chapters can be classified as an FIR Volterra structure in spatial domain. The IIR Volterra filter in time domain has been introduced, however, as far as we know the IIR Volterra structure in spatial domain has not been discussed yet. Therefore, the IIR Volterra structure in spatial domain is considered. Now the gap between time domain and spatial domain are filled. In time domain, there are FIR Volterra filter and IIR Volterra filter. While, in spatial domain, we have FIR Volterra structure and IIR Volterra structure.

At last, joint space and time processing base on Volterra signal model is introduced. Since we have FIR and IIR Volterra models in both time and spatial domains. There are four new structures for joint space and time processing based on the Volterra models. FIR or IIR Volterra in spatial domain followed by FIR or IIR Volterra in time domain. These four models are generalized models. The models exploited at the previous Chapters can be classified as special case of the four models.

1.5 Organization of the Thesis

The organization of the thesis is as follows:

Chapter 2 A review of widely used narrowband DOA estimation method is given. such as quadratic, subspace based and statistical methods. Furthermore, a brief comparison and discussion about all the methods are also provided.

Chapter 3 We propose a new Capon DOA estimation technique using Volterra signal model in spatial domain. The proposed technique has an improved resolution and a lower RMSE when compared with the standard Capon algorithm.

Chapter 4 In this chapter, the Volterra based MUSIC algorithm is considered. The resolution of the proposed method is high comparing with the standard MUSIC algorithm and the 2q- MUSIC algorithm. Simulation results are presented to show the performance of the proposed technique.

Chapter 5 We discuss the Volterra Capon and Volterra MUSIC DOA estimators for correlated or coherent sources. Spatial smoothing techniques are used to remove coherency.

Chapter 6 We explore an alternative approach for array signal processing which defines IIR structure in spatial domain. To illustrate the applicability of the concept of IIR array, we propose a new DOA estimation technique as well as a beamformer with the spatial domain IIR array implementation. The performance of the proposed methods are comparable to the existing techniques.

Chapter 7 We exploit the extensions for the IIR and Volterra based array processing techniques, such as IIR Volterra structure in spatial domain, Volterra based joint space and time processing.

Chapter 8 Conclusions are drawn and recommendations are given in this chapter.

Chapter 2

Narrowband DOA Estimation Techniques

DOA estimation using sensor arrays is widely used in many application areas, such as radar, sonar, wireless communications, astronomy, etc [9]. The quadratic algorithms are widely used DOA estimation techniques, such as beamscan approach and Capon. Over the last few decades, many high resolution (HR) DOA estimation methods have been proposed. Among all the classic DOA estimation algorithms, the maximum likelihood (ML) method has been shown to achieve the best estimation performance. However, a multi-dimensional search is needed to obtain the global maximum of the likelihood function, so the computational complexity is high. A number of computationally simpler HR DOA estimation methods have been developed using the concept of signal and noise subspaces. The subspace based DOA estimation algorithms can be classified into two main categories:

1. Searching approaches
2. Searching-free approaches
 - (a) Polynomial rooting
 - (b) Matrix shifting

The MUSIC [10] algorithm is one of the well known searching techniques, which is based on a one dimensional spectral search over the parameter space. The computational complexity is lower compared to the ML method. However, the computational complexity involved in MUSIC may still unacceptably high for real-time applications.

There are two main subgroups for the searching-free techniques, polynomial rooting and matrix shifting. Root-MUSIC [11] and weighted subspace fitting (WSF) [12] are two widely used polynomial rooting methods. For the matrix shifting methods, the shift invariant structure of the signal subspace is used, such as Estimation Signal Parameter via a Rotational Invariant Technique (ESPRIT) [13] and Matrix Pencil (MP) [14].

2.1 Quadratic Algorithms

2.1.1 Beamscan Approach

The beamscan approach relies on scanning the power from different directions by using the conventional beamforming methods. For the far-field case, the array is steered by compensating the delays of the different sensor outputs by appropriately shifting the beampattern. When the weights on all the sensors are unity and no delays are introduced, the array is effectively steered at broadside (perpendicular to the array axis). For waves traveling in that direction, the delays for all the sensors are equal and no compensations are needed. Thus unity weighting achieves the maximum power at broadside among all directions.

The maximum power is achieved by steering at the direction from which the waves are arriving, assuming no aliasing is present. For the narrowband case, the delays can be implemented by using complex weights $\mathbf{w}$ on the sensors. The array output thus becomes:

$$y(k) = \mathbf{w}^H \mathbf{x}(k) = \mathbf{w}^H \mathbf{a}(\theta) s(k). \quad (2.1)$$

To steer the array to angle θ , the weights have to be set as $\mathbf{w} = \mathbf{a}(\theta)$. Due to the linearity of the system, the same approach is used to look for a superposition of plane-waves trav-

eling from different directions, but with identical carrier-frequencies. The spectrum for the beamscan method is represented as

$$P_b(\theta) = \frac{1}{K} \sum_{k=1}^K |\mathbf{a}^H(\theta) \mathbf{x}(k)|^2, \quad (2.2)$$

where K is the number of snapshots. The expression in (2.2) can be rewritten as,

$$\begin{aligned} P_b(\theta) &= \mathbf{a}^H(\theta) \left\{ \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k) \mathbf{x}^H(k) \right\} \mathbf{a}(\theta) \\ &= \mathbf{a}^H(\theta) \mathbf{R} \mathbf{a}(\theta). \end{aligned} \quad (2.3)$$

This estimator is also referred to as the Bartlett beamformer. The performance of the Bartlett beamformer has been studied in [15]. The beamscan method is simple and robust, which is widely used in practice. However, it suffers from the Rayleigh resolution limit [16], which can be mitigated only by increasing the aperture of the array (the number of sensors). Improving SNR or increasing observation time does not improve resolution. The method parallels FIR time-series analysis. For example, to decrease the sidelobes levels, windowing can be used; however, no simple extensions are able to improve resolution. In the wideband case, the processing is usually done in frequency domain using short-time Fourier transforms. To work with wideband signals in the time domain, actual delays have to be implemented instead of phase shifts.

There are various modifications to the Bartlett estimator which are appropriate for various environments, such as the weighted beamscan algorithm. For a conjugate symmetric array, forward-backward (FB) averaging can be used to improve the performance for a modest number of snapshots. The spatial spectrum of the FB averaging method is given by

$$P_{bf}(\theta) = \frac{1}{K} \sum_{k=1}^K |\mathbf{a}^H(\theta) (\mathbf{x}(k) + \mathbf{J} \mathbf{x}^*(k))|^2. \quad (2.4)$$

The expression in (2.4) can be rewritten as

$$\begin{aligned} P_{bf}(\theta) &= \mathbf{a}^H(\theta) \left\{ \frac{1}{2K} \sum_{k=1}^K (\mathbf{x}(k)\mathbf{x}^H(k) + \mathbf{x}^*(k)\mathbf{x}^T(k)) \right\} \mathbf{a}(\theta) \\ &= \mathbf{a}^H(\theta) \mathbf{R}_{fb} \mathbf{a}(\theta), \end{aligned} \quad (2.5)$$

where $\mathbf{J}$ is the exchange matrix, where the counterdiagonal elements are one and all other elements are zero.

2.1.2 Maximum Entropy

The maximum entropy (ME) method was proposed in [17]. The maximized signal entropy is obtained by fitting an auto-regressive (AR) model to the data. The coefficients of the AR model $\mathbf{w}$ is obtained by

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w}, \quad \text{s.t.} \quad \mathbf{w}^H \mathbf{e}_1 = 1, \quad (2.6)$$

where $\mathbf{R}$ is the covariance matrix and $\mathbf{e}_1 = [1 \ 0 \ \cdots \ 0]^T$.

The AR coefficients resulting from this minimization problem may be found via Lagrange multiplier techniques to be

$$\mathbf{w} = \frac{\mathbf{R}^{-1} \mathbf{e}_1}{\mathbf{e}_1^H \mathbf{R}^{-1} \mathbf{e}_1}. \quad (2.7)$$

The DOAs can then be found from the peaks of the spatial spectrum given by

$$P_{me}(\theta) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{w} \mathbf{w}^H \mathbf{a}(\theta)}. \quad (2.8)$$

2.1.3 Capon

As introduced in Section 2.1.1, the weights of the classical beamforming method are independent of the signals and noise. The idea of optimal beamforming is to use the estimated

signal and noise parameters to improve the performance. One of the widely used methods is Capon, also called minimum variance distortionless response (MVDR) [18]. It attempts to minimize the variance due to noise, while keeping the gain in the direction of steering equal to unity:

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w}, \quad \text{s.t.} \quad \mathbf{w}^H \mathbf{a}(\theta) = 1, \quad (2.9)$$

where $\mathbf{w}$ is the complex weight and $\mathbf{R}$ is the covariance matrix. The solution of this optimization problem (2.9) is given by

$$\mathbf{w}_c = \frac{\mathbf{R}^{-1} \mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{R}^{-1} \mathbf{a}(\theta)}. \quad (2.10)$$

The source location estimate is obtained in the same way as for the classical beamscan method by steering the array at a range of directions θ s and looking for maximum powers. The resulting spectrum has an analytic expression:

$$P_c(\theta) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}^{-1} \mathbf{a}(\theta)}. \quad (2.11)$$

The main benefit of this method is a substantial increase in resolution compared with standard beamforming method. In fact, as opposed to the standard beamforming method, the number of sensors does not impose a limit on resolution. Another benefit is the lower amount of ripple in the “sidelobes”. Some of the shortcomings include an increase in the amount of computation compared to the conventional beamforming method, poor performance with small amounts of snapshots (due to the difficulty of estimation of the sensor data covariance matrix) and inability to handle strongly correlated or coherent sources. Nevertheless, the combination of increased resolution, only moderate increase in computational complexity, and the robustness due to model errors which occur in practice (unlike some of the other conventional super-resolution methods) make this method one of the most widely used techniques in practical applications. A more elaborate discussion of all of the above

assertions can be found in [16].

Similar to the forward-backward (FB) averaging method used in the beamscan method, FB averaging of the data can be used to improve performance and reduce computation for conjugate symmetric array. The quadratic algorithm is further extended to root version for linear array, which is computationally simpler and more accurate. The use of rooting algorithm is due to [11]. In [19], a generalized Capon spatial spectrum estimator for localization of multiple incoherently distributed (spread) sources in sensor arrays is developed. In [20], the authors give a general form of the Capon estimator where any unitary matrix can be used to estimate the spectrum. The authors also show how the Capon method can be used to estimate the magnitude squared coherence function.

The Bartlett beamformer, ME and Capon are efficient DOA estimators, although they are not high-resolution. However, they can play a useful role as preliminary processors to indicate the number of plane waves impinging on the array, their approximate locations, and their approximate signal powers.

2.2 Subspace Based Methods

2.2.1 MUSIC

The MUSIC method [10] is the most prominent member of the family of eigen decomposition based source location estimators. The underlying idea is to separate the eigenspace of the covariance matrix of sensor outputs into the signal and noise subspace using the knowledge about the covariance of the noise. The sensor output correlation matrix admits the following decomposition:

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^H + \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^H \in \mathcal{C}^{M \times M}, \quad (2.12)$$

where $\mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_M]$. Without loss of generality, we assume the eigenvalues are in the order of decreasing size.

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq \lambda_{L+1} \geq \dots \geq \lambda_M. \quad (2.13)$$

Since there are L signals, the number of sources L has to be strictly less than the number of sensors M , for the method to work. We refer to the first L eigenvalues $\lambda_i, i = 1, 2, \dots, L$ as the signal subspace eigenvalues and $\mathbf{u}_i, i = 1, 2, \dots, L$ as the signal subspace eigenvectors. Then $\mathbf{\Lambda}_s = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_L]$ and

$$\mathbf{U}_s = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_L] \in \mathcal{C}^{M \times L}. \quad (2.14)$$

Similar to $\mathbf{\Lambda}_s$ and $\mathbf{U}_s$, $\mathbf{\Lambda}_n = \text{diag}[\lambda_{L+1}, \lambda_{L+2}, \dots, \lambda_M]$ and

$$\mathbf{U}_n = [\mathbf{u}_{L+1} \quad \mathbf{u}_{L+2} \quad \dots \quad \mathbf{u}_M] \in \mathcal{C}^{M \times (M-L)}. \quad (2.15)$$

Hence $\mathbf{U}_s$ is the combined signal plus noise subspace, and $\mathbf{U}_n$ is the noise subspace alone. This information can be used to separate the two eigen-subspaces. Due to the orthogonality of eigen-subspaces corresponding to different eigenvalues for Hermitian matrices, the noise subspace is orthogonal to the steering vectors, which are corresponding to the directions of propagation, thus $\mathbf{U}_n^H \mathbf{a}(\theta) = 0$ for all directions from which the signals are impinging. The MUSIC spectrum is obtained by putting the squared norm of this term into the denominator,

$$P_m = \frac{1}{\mathbf{a}(\theta)^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(\theta)}. \quad (2.16)$$

For the noiseless case the peaks of the spectrum approach infinity.

In contrast with the previously discussed techniques, the MUSIC spectrum has no direct relation to power; it simply exhibits sharp peaks at the estimated source directions. Also, it cannot be used as a beamformer directly, since the spectrum is not obtained by steering

the array. Unlike the the methods previously discussed, MUSIC provides a consistent (in the sense of estimation theory) estimation of the locations of the sources, as SNR and the number of sensors go to infinity. Despite the dramatic improvement in resolution, MUSIC suffers from a high sensitivity to model errors, such as sensor position uncertainty. Also, the resolution capabilities decrease when the signals are correlated. When some of the signals are coherent (perfectly correlated), the method fails to work. The computational complexity is dominated by the computation of the eigen decomposition of the covariance matrix.

There are multiple extensions of MUSIC by using a weight matrix in the denominator, one of which is the Min-Norm algorithm [9]. Root-MUSIC [11] is a variant of MUSIC which instead of computing a spectrum, forms a polynomial using the noise subspace, and the source location estimates are the roots of the polynomial. Root-MUSIC relies on the structure of the steering matrix for a uniform linear array (ULA), and cannot be extended to general arrays. The performance for ULAs is very similar to that of MUSIC, except for a somewhat higher robustness at limited numbers of time samples. The forward-backward root MUSIC algorithm can be implemented with real computation by using unitary transformation [21].

MUSIC is based on the second-order statistics of the received data. In [22], they extend the MUSIC algorithm by using the fourth-order cumulants (FO) of the snapshots. One of the main interests in using FO cumulants only instead of second-order (SO) ones in array processing applications is the increasing of both the effective aperture and the number of virtual sensors [23]. The fourth order cumulants based MUSIC is further extended to an arbitrary even order $2q$ ($q \geq 1$), giving rise to the $2q$ -MUSIC methods [24]. The $2q$ -MUSIC algorithm is mainly developed for arrays with identical sensors only and cannot put up with arrays of diversely polarized sensors in the presence of diversely polarized sources. Polarization diversity $2q$ -MUSIC (PD- $2q$ -MUSIC) algorithm is proposed in [25] to solve the diversely polarized problem. A Khatri-Rao (KR) subspace DOA estimation approach is proposed for quasi-stationary signals with less sensors than sources and unknown spatial

noise covariance in [26].

2.2.2 ESPRIT

The ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) algorithm assumes that an antenna array is composed of two identical subarrays [13]. DOAs are estimated by exploiting the property called shift invariance of the array. The subarrays may overlap, that is, an array element may be shared by both subarrays. Since columns of $\mathbf{A}(\theta)$ span the signal subspace, therefore, we have the relationship $\mathbf{U}_s = \mathbf{A}\mathbf{T}$, where $\mathbf{T} \in \mathcal{C}^{L \times L}$ is a non-singular matrix. Let

$$\begin{aligned}\mathbf{U}_{s1} &= \mathbf{J}_1 \mathbf{U}_s = \mathbf{J}_1 \mathbf{A} \mathbf{T} \in \mathcal{C}^{(M-1) \times L}, \\ \mathbf{U}_{s2} &= \mathbf{J}_2 \mathbf{U}_s = \mathbf{J}_2 \mathbf{A} \mathbf{T} \in \mathcal{C}^{(M-1) \times L},\end{aligned}\tag{2.17}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are two selection matrices, and

$$\begin{aligned}\mathbf{J}_1 &= [\mathbf{I}_{M-1} \quad \mathbf{0}_{(M-1) \times 1}] \in \mathcal{R}^{(M-1) \times M}, \\ \mathbf{J}_2 &= [\mathbf{0}_{(M-1) \times 1} \quad \mathbf{I}_{M-1}] \in \mathcal{R}^{(M-1) \times M}.\end{aligned}\tag{2.18}$$

Then

$$\mathbf{U}_{s2} = \mathbf{U}_{s1} (\mathbf{T}^{-1} \mathbf{\Phi} \mathbf{T}) = \mathbf{U}_{s1} \mathbf{\Psi},\tag{2.19}$$

where

$$\mathbf{\Phi} = \text{diag} \left[e^{-j \frac{2\pi d}{\lambda} \cos(\theta_1)} \quad e^{-j \frac{2\pi d}{\lambda} \cos(\theta_2)} \quad \dots \quad e^{-j \frac{2\pi d}{\lambda} \cos(\theta_L)} \right] \in \mathcal{C}^{L \times L},\tag{2.20}$$

and

$$\mathbf{\Psi} = \mathbf{T}^{-1} \mathbf{\Phi} \mathbf{T} \in \mathcal{C}^{L \times L}.\tag{2.21}$$

In practice, the estimated $\hat{\mathbf{U}}_{s1}$ and $\hat{\mathbf{U}}_{s2}$ are used. The matrix $\hat{\mathbf{\Psi}}$ is estimated by using the least square approach

$$\hat{\mathbf{\Psi}}_{LS} = \left[\hat{\mathbf{U}}_{s1}^H \hat{\mathbf{U}}_{s1} \right]^{-1} \hat{\mathbf{U}}_{s1}^H \hat{\mathbf{U}}_{s2}. \quad (2.22)$$

Denoting the eigenvalues of $\hat{\mathbf{\Psi}}_{LS}$ as $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_L$, the directions are estimated by using

$$\theta_l = \arccos \left(-\frac{\lambda}{2\pi d} \arg(\hat{\lambda}_l) \right), l = 1, 2, \dots, L. \quad (2.23)$$

The matrix $\hat{\mathbf{\Psi}}$ can also be estimated by using total least squares (TLS) methods. Structured least squares are used in [27] to improve the performance of ESPRIT-type algorithms.

A summary of the ESPRIT algorithm is as follows:

Algorithm 1 The ESPRIT algorithm

Step 1. Obtain the signal subspace $\hat{\mathbf{U}}_s$ by (2.14).

Step 2. Find $\hat{\mathbf{U}}_{s1}$ and $\hat{\mathbf{U}}_{s2}$ by (2.17).

Step 3. Compute $\mathbf{\Psi}$ using (2.22).

Step 4. Obtain the eigenvalues of $\hat{\mathbf{\Psi}}_{LS}$.

Step 5. The directions are estimated by using (2.23).

The original formulation of ESPRIT assumes that there is only one invariance in the array associated with each dimension of the parameter space. However, in many applications, arrays have multiple invariances (e.g., uniform linear arrays) are employed, a framework for extending the ESPRIT algorithm to exploit arrays with multiple invariances is proposed in [28]. Beamspace transformation is one way to reduce computation and sometimes improve the estimation accuracy. A beamspace ESPRIT algorithm is described in [29]. Unitary ESPRIT incorporates forward-backward averaging, leading to an improved performance compared to the standard ESPRIT algorithm, especially for correlated source signals [30]. The unitary ESPRIT algorithm is further extended to 2D case in [31], which is a new closed-form high

resolution algorithm to provide automatically paired source azimuth and elevation angle. A closed-form multi-dimensional multi-invariance generalization of the ESPRIT algorithm is introduced to exploit the entire invariance structure underlying a (possibly) multiparametric data model, thereby greatly improving estimation performance [32]. Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT is proposed in [33].

The previous introduced ESPRIT algorithms are all matrix based. Recently, tensor based signal processing received extensive attentions. The ESPRIT algorithms are further extended to the tensor based. Tensor-structure based structured least squares (TS-SLS) is proposed in [34] to improve the performance of multi-dimensional ESPRIT-type algorithms. Multidimensional unitary Tensor-ESPRIT for non-circular sources is discussed in [35]. In [36], a R-D unitary Tensor-ESPRIT using an augmentation of the Tensor order for single snapshot is proposed. Tensor-based ESPRIT with spatial smoothing (TB-SS) using multiple snapshots is considered in [37]. Analytical performance assessment for multi-dimensional Tensor-ESPRIT-type parameter estimation algorithms is discussed in [38].

2.2.3 Matrix Pencil

To apply the Matrix Pencil method [39], we first construct a Hankel matrix $\mathbf{X}_M(t) \in \mathcal{C}^{(N-L+1) \times L}$ using the snapshot at time t , which is given by

$$\mathbf{X}_M(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_L(t) \\ x_2(t) & x_3(t) & \cdots & x_{L+1}(t) \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-L}(t) & x_{N-L+1}(t) & \cdots & x_{N-1}(t) \\ x_{N-L+1}(t) & x_{N-L+2}(t) & \cdots & x_N(t) \end{bmatrix}, \quad (2.24)$$

where the pencil parameter L must satisfy

$$\begin{aligned} M \leq L \leq N - L, \text{ for } N \text{ is even,} \\ M \leq L \leq N - L + 1, \text{ for } N \text{ is odd.} \end{aligned} \quad (2.25)$$

L is chosen between $N/3$ and $N/2$ for efficient noise filtering. Let us define two submatrices $\mathbf{A}_1(\theta) \in \mathcal{C}^{(N-L+1) \times M}$ and $\mathbf{A}_2(\theta) \in \mathcal{C}^{L \times M}$ of $\mathbf{A}(\theta)$, where $\mathbf{A}_1(\theta)$ and $\mathbf{A}_2(\theta)$ denote the first $N - L + 1$ and L rows respectively. For the noiseless case

$$\mathbf{X}_M(t) = \mathbf{A}_1(\theta) \begin{bmatrix} s_1(t) & & & \mathbf{0} \\ & s_2(t) & & \\ & & \ddots & \\ \mathbf{0} & & & s_M(t) \end{bmatrix} \mathbf{A}_2^T(\theta). \quad (2.26)$$

Let $z_m = e^{j\frac{2\pi d}{\lambda} \sin(\theta_m)}$, where $\{m = 1, 2, \dots, M\}$ and $\mathbf{Z} = \text{diag}[z_1, z_2, \dots, z_M]$,

$$\mathbf{J}_1 \mathbf{A}_1(\theta) \mathbf{Z} = \mathbf{J}_2 \mathbf{A}_1(\theta), \quad (2.27)$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are two selection matrices, which are defined as

$$\begin{aligned} \mathbf{J}_1 &= [\mathbf{I}_{N-L} \quad \mathbf{0}_{(N-L) \times 1}] \in \mathcal{R}^{(N-L) \times (N-L+1)}, \\ \mathbf{J}_2 &= [\mathbf{0}_{(N-L) \times 1} \quad \mathbf{I}_{N-L}] \in \mathcal{R}^{(N-L) \times (N-L+1)}, \end{aligned} \quad (2.28)$$

where $\mathbf{I}_k$ denotes the $k \times k$ identity matrix. For the noisy data, the singular value decomposition is used to reduce the noise effect:

$$\mathbf{X}_M = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{V}_s + \mathbf{U}_n \mathbf{\Sigma}_n \mathbf{V}_n, \quad (2.29)$$

where $\mathbf{U}_s \in \mathcal{C}^{(N-L+1) \times M}$ contains the M principle left singular vectors that spans the signal subspace and $\mathbf{U}_n$ contains the remaining left singular vectors that spans the noise subspace. Since $\mathbf{U}_s = \mathbf{A}_1(\theta)\mathbf{T}$, where $\mathbf{T} \in \mathcal{C}^{M \times M}$ is a square non-singular matrix,

$$\mathbf{J}_1 \mathbf{U}_s \mathbf{Q} = \mathbf{J}_2 \mathbf{U}_s, \quad (2.30)$$

where $\mathbf{Q} = \mathbf{T}^{-1} \mathbf{Z} \mathbf{T}$. Now, we define the matrices $\mathbf{U}_{s1}$ and $\mathbf{U}_{s2}$,

$$\begin{aligned} \mathbf{U}_{s1} &= \mathbf{J}_1 \mathbf{U}_s \in \mathcal{C}^{(N-L) \times M}, \\ \mathbf{U}_{s2} &= \mathbf{J}_2 \mathbf{U}_s \in \mathcal{C}^{(N-L) \times M}, \end{aligned} \quad (2.31)$$

then

$$\mathbf{U}_{s1} \mathbf{Q} = \mathbf{U}_{s2}. \quad (2.32)$$

For noisy data, $\hat{\mathbf{X}}_M(t) = \mathbf{X}_M(t) + \mathbf{N}(t)$,

$$\hat{\mathbf{X}}_M = \hat{\mathbf{U}}_s \hat{\mathbf{\Sigma}}_s \hat{\mathbf{V}}_s + \hat{\mathbf{U}}_n \hat{\mathbf{\Sigma}}_n \hat{\mathbf{V}}_n, \quad (2.33)$$

and

$$\hat{\mathbf{U}}_{s1} \mathbf{Q} \approx \hat{\mathbf{U}}_{s2}, \quad (2.34)$$

where $\hat{\mathbf{U}}_s = \mathbf{U}_s + \Delta \mathbf{U}_s$.

For the LS based Matrix Pencil method, DOAs are estimated using the M eigenvalues $\{z_1, z_2, \dots, z_M\}$ of $\mathbf{Q}_{LS}$, where

$$\mathbf{Q}_{LS} = \hat{\mathbf{U}}_{s1}^\dagger \hat{\mathbf{U}}_{s2} = \left[\hat{\mathbf{U}}_{s1}^H \hat{\mathbf{U}}_{s1} \right]^H \hat{\mathbf{U}}_{s1}^H \hat{\mathbf{U}}_{s2}, \quad (2.35)$$

where $\dagger$ denotes pseudoinverse. The DOAs are obtained by

$$\theta_m = \arcsin\left(\frac{\lambda \Im(\ln z_m)}{2\pi d}\right), \quad m = 1, 2, \dots, M. \quad (2.36)$$

where $\Im(z)$ denotes the imaginary part of z .

A summary of the MP algorithm is as follows:

Algorithm 2 Matrix Pencil

Step 1. Given N and M , choose L to satisfy (2.25). Form $\mathbf{X}_M(t)$ by (2.24).

Step 2. Calculate the signal subspace $\mathbf{U}_s$ by (2.29).

Step 3. Obtain $\mathbf{U}_{s1}$ and $\mathbf{U}_{s2}$ by (2.31).

Step 4. Obtain $\mathbf{Q}_{LS}$ by (2.35) using LS or TLS.

Step 5. Find the eigenvalues z_m of the matrix $\mathbf{Q}_{LS}$.

Step 6. Obtain θ_m by (2.36).

In [40], the performance of the matrix pencil (MP) method and the estimation of signal parameters using rotational invariance technique (ESPRIT) method have been compared under varying number of snapshots taken. The performance of matrix pencil and root-MUSIC (with and without spatial smoothing) algorithms for DOA estimation for a uniform linear array using multiple snapshots have been studied in [41]. An optimally weighted matrix pencil approach for 2-D harmonic retrieval is proposed in [42]. In [43], the authors use the matrix pencil (MP) method to compute the directions of arrival (DOAs) of the signals using a very efficient computational procedure in which the complexity of the computation can be reduced significantly by using a unitary matrix transformation. A DFT beamspace matrix pencil method is proposed in [44], the DFT beamspace approach has the advantage of reduced computational complexity due to converting complex computations (in Matrix Pencil method) into real ones along with the inherent ability to operate in a reduced beamspace in situations where some priori information is available about the DOAs. The matrix

pencil method is further extended to 3D case in [45]. The 2D DOA for coherently distributed sources could be estimated from the enhanced matrix by using the 3D matrix pencil method.

2.3 Statistical Methods

In this chapter, we focus on the maximum likelihood (ML) approach for estimating the direction of arrivals in the presence of additive Gaussian noise [1]. The array outputs can be described by probability density functions under two different assumptions on signals. The first estimate assumes the source signals are sample functions from Gaussian random processes. We refer to this estimate as the unconditional (or stochastic) UML estimate. The second estimate assumes the source signals are unknown, but nonrandom. The resulting ML estimator is referred to as the conditional (or deterministic) ML estimate.

2.3.1 Stochastic Maximum Likelihood Estimator

Case 1: δ^2 known

The Stochastic ML (SML) method models the signals as stationary jointly Gaussian stochastic processes with covariance matrix given by $\mathbf{R}_s = E[\mathbf{x}(k)\mathbf{x}(k)^H]$.

Under the assumptions that the observation process $\mathbf{x}(k)$ is a stationary, zero-mean, Gaussian process, it is completely described by its second order covariance matrix

$$\mathbf{R} = \mathbf{A}(\theta)\mathbf{R}_s\mathbf{A}^H(\theta) + \delta_n^2\mathbf{I}. \quad (2.37)$$

The likelihood function

$$\begin{aligned} L(\theta, \mathbf{R}_s) &= -\ln \det \mathbf{R} - \frac{1}{K} \sum_{k=1}^K \mathbf{x}^H(k)\mathbf{R}^{-1}\mathbf{x}(k) \\ &= -\ln \det \mathbf{R} - \text{tr}[\mathbf{R}^{-1}\mathbf{C}_{xx}], \end{aligned} \quad (2.38)$$

where $\mathbf{C}_{xx} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k)\mathbf{x}^H(k)$ is the sample covariance matrix.

With known δ_w^2 , $\mathbf{R}_s$ can be estimated by

$$\hat{\mathbf{R}}_s = \mathbf{A}^\dagger(\theta) [\mathbf{C}_{xx} - \delta_w^2 \mathbf{I}] [\mathbf{A}^\dagger(\theta)]^H, \quad (2.39)$$

where $\mathbf{A}^\dagger(\theta) = [\mathbf{A}^H(\theta)\mathbf{A}(\theta)]^{-1} \mathbf{A}^H$ is the Moore-Penrose pseudoinverse. The relationship in (2.37) can be written as

$$\hat{\mathbf{R}} = \mathbf{P}_A [\mathbf{C}_{xx} - \delta_w^2 \mathbf{I}] \mathbf{P}_A + \delta_w^2 \mathbf{I}, \quad (2.40)$$

where $\mathbf{P}_A$ is the projection matrix onto the range of $\mathbf{A}(\theta)$. Substituting (2.40) into the log-likelihood function (2.38), we obtain the following

$$L(\theta) = -\ln \det \hat{\mathbf{R}} - \text{tr} [\hat{\mathbf{R}}^{-1} \mathbf{C}_{xx}]. \quad (2.41)$$

The ML estimate of θ is obtained by maximizing $L(\theta)$ over θ

$$\hat{\theta} = \arg \max_{\theta} \left\{ -\ln \det [\mathbf{P}_A \mathbf{C}_{xx} \mathbf{P}_A + \delta_n^2 \mathbf{P}_A^\perp] - \frac{1}{\delta_n^2} \text{tr} [\mathbf{P}_A^\perp \mathbf{C}_{xx}] \right\}, \quad (2.42)$$

where $\mathbf{P}_A^\perp = \mathbf{I} - \mathbf{P}_A$.

Case 2: δ^2 unknown

If the noise variance of δ_w^2 is unknown, then the likelihood function in (2.38) becomes

$$L(\theta, \mathbf{R}_s, \delta_w^2) = -\ln \det \mathbf{R} - \text{tr} [\mathbf{R}^{-1} \mathbf{C}_{xx}]. \quad (2.43)$$

The δ_w^2 can be estimated by

$$\hat{\delta}_w^2 = \frac{1}{M-L} \text{tr} [\mathbf{P}_A^\perp \mathbf{C}_{xx}]. \quad (2.44)$$

Substituting (2.44) into (2.42), we obtain

$$\hat{\theta} = \arg \min_{\theta} \left\{ \det \left[\mathbf{P}_A \mathbf{C}_{xx} \mathbf{P}_A + \frac{1}{M-L} \text{tr} \left[\mathbf{P}_A^\perp \mathbf{C}_{xx} \right] \mathbf{P}_A^\perp \right] \right\}. \quad (2.45)$$

2.3.2 Deterministic Maximum Likelihood Estimator

Case 1: δ^2 known

For deterministic ML estimator, the log-likelihood is given by

$$L(\theta, \mathbf{s}) = -KM \ln \delta_n^2 - \frac{1}{\delta_n^2} \sum_{k=1}^K |\mathbf{x}(k) - \mathbf{A}(\theta) \mathbf{s}(k)|^2. \quad (2.46)$$

For a fixed and unknown θ , the ML estimate for the signal vector $\mathbf{s}(k)$ is the solution to the least square problem

$$\min_{\mathbf{s}(k)} \|\mathbf{x}(k) - \mathbf{A}(\theta) \mathbf{s}(k)\|^2, \quad (2.47)$$

then

$$\mathbf{s}(\hat{k}) = \mathbf{A}^\dagger(\theta) \mathbf{x}(k). \quad (2.48)$$

Substituting (2.48) into (2.46), we obtain

$$L(\theta) = -KM \ln \delta_n^2 - \frac{K}{\delta_n^2} \text{tr} \left[\mathbf{P}_A^\perp \mathbf{C}_{xx} \right]. \quad (2.49)$$

The ML estimate is obtained by

$$\hat{\theta} = \arg \max_{\theta} \{ \text{tr} [\mathbf{P}_A \mathbf{C}_{xx}] \}. \quad (2.50)$$

Case 2: δ_n^2 unknown

If the variance δ_n^2 is unknown, then the following equation can be used to estimate δ_n^2

$$\hat{\delta}_n^2 = \frac{1}{M} \text{tr} \left[\mathbf{P}_A^\perp \mathbf{C}_{xx} \right]. \quad (2.51)$$

Substituting (2.51) into (2.49) and ignoring the factor independent of θ , an estimator which is identical to (2.50) is obtained.

Gradient techniques, such as Quasi-Newton methods can be used to find statistic and deterministic ML estimators. Alternating maximization [46] and expectation maximization [47] [48] are two widely used relaxation algorithms.

For standard linear array, two computational efficient algorithms are applicable. Iterative quadratic maximum likelihood (IQML) [49] and iterative MODE (IMODE). Please refer to [1] for more details.

2.4 Adaptive DOA Estimation Techniques

Array beamforming can be considered as a spatial filter, which is analogous to a time-domain digital filter. The response of the array beamformer is equivalent to the frequency response of the time-domain digital filter. This equivalence arises because a digital filter samples the signal in time-domain whereas an array samples the signal in spatial domain. With such close relation between array snapshots and discrete-time signals, the concepts and algorithms developed in adaptive filtering can be applied for array processing. For example, instead of using the covariance matrix directly to perform DOA estimation, the DOAs can be estimated adaptively by using the adaptive filtering techniques. The detailed discussions are given in Chapter 6.

Chapter 3

Volterra Capon DOA Estimation

3.1 Introduction

Smart antennas have been widely used in many applications such as radar, sonar and communication systems, which can significantly increase the data rate and improve the quality of wireless transmission. However, the performance of smart antennas relies heavily on the accurate estimation of the direction of arrival of each signal.

As we know, the subspace based techniques, such as the standard MUSIC algorithm, are formulated using the underlying principle that the source signals belong to a low rank subspace in the presence of spatially uncorrelated noise. The performance of the technique is vulnerable to the identification of the signal subspace. Capon estimator does not require us to know either the signal subspace or the number of sources present. The trade-off is its inferior resolution capabilities at the low SNR region.

Due to the above property, the Capon method has received a lot of attentions for both beamforming applications and DOA estimation. To deal with variations or uncertainties in the array response, such as imprecise knowledge of the DOAs and uncertainties in the array manifold, the standard Capon beamformer has been extended to robust Capon beamformer [50–54] . In [55], a generalized Capon spatial spectrum estimator has been proposed for

the localization of multiple incoherently distributed (spread) sources. It involves a 2-D search over the parameter space and shows a substantially improved performance relative to several popular spread source localization techniques. A study of mean squared error (MSE) threshold SNR prediction and probability of resolution of the Capon algorithm is given in [56]. In [57], a DOA estimation technique based on the spectral Capon rooting algorithm is developed. The developed method uses real polynomial rooting technique rather than grid searching. Lower computational complexity and better performance are the advantages of this technique over the standard Capon algorithm.

In this chapter, we propose a new Capon DOA estimation technique using Volterra signal model in spatial domain. The new technique basically uses the high order terms of Volterra series to produce augmented Volterra snapshots. The proposed technique has an improved resolution and a lower RMSE when compared with the standard Capon algorithm. The better performance is explained from the statistical point of view, instead of augmented virtual array.

When the signals are Gaussian, a similar performance is obtained for the standard Capon algorithm and the proposed Volterra Capon algorithm, because Gaussian distribution is completely characterized by its mean and variance. However, non-Gaussian signals arise in a wide variety of applications, including sonar, digital communications, seismology, and radio astronomy. When the signals are non-Gaussian, or the channel or antenna attenuation effect is considered, better performance is obtained for the proposed Volterra Capon algorithm by using the higher order statistics information of the signals. Simulation results are presented to show the performance of the proposed technique.

The organization of this Chapter is as follows. The proposed DOA estimation technique using Volterra signal model is presented in Section 3.2. The performance analysis of the proposed method is given in Section 3.3. Numerical results of Volterra Capon method are shown in Section 3.4. Conclusions are drawn in Section 3.5.

3.2 The Volterra Capon DOA Estimation Algorithm

The Volterra signal model in spatial domain is introduced in Section 1.3.2. To describe the proposed method easily, a simple description about the related variables is given here. Please refer to Section 1.3.2 for a detailed definition.

Second order Volterra model in spatial domain

The new second order Volterra snapshot is given by

$$\mathbf{x}_v(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T. \quad (3.1)$$

Then the new Volterra covariance matrix is given by

$$\mathbf{R}_v = E \left[\mathbf{x}_v(k) \mathbf{x}_v^H(k) \right]. \quad (3.2)$$

The corresponding new Volterra array steering vector is defined as

$$\mathbf{a}_v(\theta) = \left[\mathbf{a}^T(\theta), (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T \right]^T. \quad (3.3)$$

Third order Volterra model in spatial domain

The new third order Volterra snapshot at time k is defined as

$$\mathbf{x}_v(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T (\mathbf{x}(k) \boxtimes \mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T. \quad (3.4)$$

Then the new Volterra covariance matrix can be computed using (3.2).

The steering vector for third-order Volterra snapshots is given by

$$\mathbf{a}_v(\theta) = \left[\mathbf{a}^T(\theta), (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T, (\mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta) \boxtimes \mathbf{a}(\theta))^T \right]^T. \quad (3.5)$$

Volterra Capon DOA Estimation

Now, the Capon algorithm is used to estimate the directions of the sources based on the new covariance matrix $\mathbf{R}_v$ and the new array steering vector $\mathbf{a}_v(\theta)$.

Similar to the standard Capon method, the Volterra Capon DOA estimator is formulated as follows:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^H \mathbf{R}_v \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^H \mathbf{a}_v(\theta) = 1. \end{aligned} \quad (3.6)$$

It can be solved by using the Lagrange multiplier method. The function that we minimized is

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \mathbf{w}^H \mathbf{R}_v \mathbf{w} - \lambda [\mathbf{w}^H \mathbf{a}_v(\theta) - 1], \quad (3.7)$$

where λ is the multiplier. Taking the complex gradient with respect to $\mathbf{w}$ and let

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = 0. \quad (3.8)$$

After solving (3.8), we obtain

$$\mathbf{w}_{\text{opt}} = \mu \mathbf{R}_v^{-1} \mathbf{a}(\theta). \quad (3.9)$$

Since $\mathbf{w}^H \mathbf{a}_v(\theta) = 1$, then

$$\mu = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}_v^{-1} \mathbf{a}(\theta)}. \quad (3.10)$$

The DOAs of the multiple incident signals can be estimated by locating the peaks of

the Volterra Capon (V-Capon) spatial spectrum, which is determined by

$$\begin{aligned} P_{\text{V-Capon}} &= \mathbf{w}_{\text{opt}}^H \mathbf{R}_v \mathbf{w}_{\text{opt}} \\ &= \frac{1}{\mathbf{a}_v^H(\theta) \mathbf{R}_v^{-1} \mathbf{a}_v(\theta)}. \end{aligned} \quad (3.11)$$

A summary of the proposed V-Capon DOA estimation algorithm is as follows:

Algorithm 3 Volterra Capon DOA estimation algorithm

Step 1. Obtain the new array steering vector $\mathbf{a}_v(\theta)$.

Step 2. Compute the new Volterra snapshots $\{\mathbf{x}_v(k), k = 1, 2, \dots, K\}$.

Step 3. Estimate the data covariance matrix as

$$\hat{\mathbf{R}}_v = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_v(k) \mathbf{x}_v^H(k), \quad (3.12)$$

where K is the number of snapshots.

Step 4. Compute the V-Capon spectrum using (3.11), find the peaks to obtain estimates of the DOAs.

3.3 Theoretical Performance Analysis

The performance of the standard Capon method has been theoretically analyzed in [58] and [59]. The bias and variance of the proposed V-Capon method can be calculated similar to that in [59] as follows.

The DOAs $\{\hat{\theta}_k, k = 1, 2, \dots, L\}$ estimated by Capon are given by the locations of the L smallest minima of the following function

$$f(\theta) = \mathbf{a}_v^H(\theta) \mathbf{R}_v^{-1} \mathbf{a}_v(\theta). \quad (3.13)$$

Let θ_k and $\bar{\theta}_k$ denotes the true and asymptotic DOAs, respectively. It is well known that

$\bar{\theta}_k$ is not necessarily equals to θ_k for all SNR. Thus, the mean square error (MSE) of the estimated $\hat{\theta}_k$ is evaluated by

$$\begin{aligned} \text{MSE}(\hat{\theta}_k) &= \left[E \left\{ \hat{\theta}_k - \theta_k \right\} \right]^2 + E \left\{ (\hat{\theta}_k - \bar{\theta}_k)^2 \right\} \\ &= \left[E \left\{ \hat{\theta}_k - \bar{\theta}_k \right\} + (\bar{\theta}_k - \theta_k) \right]^2 + E \left\{ (\hat{\theta}_k - \bar{\theta}_k)^2 \right\} \\ &= \left[E \{ \Delta \bar{\theta}_k \} + \Delta \theta_k \right]^2 + E \left\{ (\Delta \bar{\theta}_k)^2 \right\}, \end{aligned} \quad (3.14)$$

where $\Delta \bar{\theta}_k = \hat{\theta}_k - \bar{\theta}_k$ and $\Delta \theta_k = \bar{\theta}_k - \theta_k$. A detailed discussion is given in Appendix A.

Then we have the first three derivatives of the array response with respect to the direction θ , as

$$\begin{aligned} \dot{\mathbf{a}}_v(\theta) &= \frac{\partial \mathbf{a}_v(\theta)}{\partial \theta}, \quad \ddot{\mathbf{a}}_v(\theta) = \frac{\partial^2 \mathbf{a}_v(\theta)}{\partial \theta^2}, \\ \ddot{\mathbf{a}}_v(\theta) &= \frac{\partial^3 \mathbf{a}_v(\theta)}{\partial \theta^3}. \end{aligned} \quad (3.15)$$

The first, second and third order derivatives of $f(\theta)$ are respectively given by

$$\begin{aligned} \dot{f}(\theta) &= 2\text{Re} \left[\dot{\mathbf{a}}_v^H(\theta) \mathbf{R}_v^{-1} \mathbf{a}_v(\theta) \right], \\ \ddot{f}(\theta) &= 2\text{Re} \left[\dot{\mathbf{a}}_v^H(\theta) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\theta) + \mathbf{a}_v^H(\theta) \mathbf{R}_v^{-1} \ddot{\mathbf{a}}_v(\theta) \right], \\ \ddot{f}(\theta) &= 2\text{Re} \left[3\dot{\mathbf{a}}_v^H(\theta) \mathbf{R}_v^{-1} \ddot{\mathbf{a}}_v(\theta) + \mathbf{a}_v^H(\theta) \mathbf{R}_v^{-1} \ddot{\mathbf{a}}_v(\theta) \right]. \end{aligned} \quad (3.16)$$

3.3.1 Asymptotic Bias

The first-order Taylor series expansion of $\dot{f}(\bar{\theta}_k)$ around θ_k is given by

$$\dot{f}(\bar{\theta}_k) \simeq \dot{f}(\theta_k) + \ddot{f}(\theta_k)(\bar{\theta}_k - \theta_k). \quad (3.17)$$

Since Capon algorithm is asymptotically biased, therefore $\dot{f}(\bar{\theta}_k) = 0$, and

$$\begin{aligned}\Delta \theta_k &\simeq -\frac{\dot{f}(\theta_k)}{\ddot{f}(\theta_k)} \\ &\simeq -\frac{Re [\dot{\mathbf{a}}_v^H(\theta) \mathbf{R}_v^{-1} \mathbf{a}_v(\theta)]}{Re [\dot{\mathbf{a}}_v^H(\theta) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\theta) + \mathbf{a}_v^H(\theta) \mathbf{R}_v^{-1} \ddot{\mathbf{a}}_v(\theta)]}.\end{aligned}\quad (3.18)$$

3.3.2 Variance

First-order Taylor series expansion is used to analyze the variance. Since the estimator is biased, the expansion is done around the asymptotic estimate $\bar{\theta}_k$ and $\mathbf{R}_v^{-1}$.

The first Taylor expansion of $\dot{f}(\hat{\theta}_k, \hat{\mathbf{R}}_v^{-1})$ evaluated at $(\hat{\theta}_k, \hat{\mathbf{R}}_v^{-1})$ is given by

$$\begin{aligned}\dot{f}(\hat{\theta}_k, \hat{\mathbf{R}}_v^{-1}) &= \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) + \ddot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) \Delta \bar{\theta}_k \\ &\quad + Tr \left\{ \left[\frac{\partial}{\partial \mathbf{R}_v^{-1}} \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) \right]^T \Delta \mathbf{R}_v^{-1} \right\}.\end{aligned}\quad (3.19)$$

Since

$$\left[\frac{\partial}{\partial \mathbf{R}_v^{-1}} \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) \right]^T = \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) + \mathbf{a}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k), \quad (3.20)$$

the second term in (3.19) can be simplified as

$$\begin{aligned}&Tr \left\{ \left[\frac{\partial}{\partial \mathbf{R}_v^{-1}} \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) \right]^T \Delta \mathbf{R}_v^{-1} \right\} \\ &= Tr \left\{ \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} + \mathbf{a}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \right\} \\ &= \mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) + \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}_k) \\ &= 2Re [\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)].\end{aligned}\quad (3.21)$$

Substitute (3.21) into (3.19), therefore

$$\begin{aligned}\dot{f}(\hat{\theta}_k, \hat{\mathbf{R}}_v^{-1}) &= \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) + \ddot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) \Delta \bar{\theta}_k \\ &\quad + 2Re [\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)].\end{aligned}\quad (3.22)$$

Since the minima of the function in (3.13) yield the DOA estimates, then the first derivative of $f(\theta)$ equals 0. Therefore

$$\begin{aligned}\dot{f}(\hat{\theta}_k, \hat{\mathbf{R}}_v^{-1}) &= 2Re \left[\dot{\mathbf{a}}_v^H(\hat{\theta}) \hat{\mathbf{R}}_v^{-1} \mathbf{a}_v(\hat{\theta}) \right] = 0, \\ \dot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1}) &= 2Re \left[\dot{\mathbf{a}}_v^H(\bar{\theta}) \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}) \right] = 0.\end{aligned}\tag{3.23}$$

Substitute (3.23) into (3.22), we obtain the following expression for the bias,

$$\Delta \bar{\theta}_k = - \frac{2Re \left[\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \right]}{\ddot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1})}.\tag{3.24}$$

Since $\hat{\mathbf{R}}_v^{-1}$ is an unbiased estimation of $\mathbf{R}_v^{-1}$, then

$$\begin{aligned}E \{ \Delta \bar{\theta}_k \} &= - \frac{2E \{ Re \left[\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \right] \}}{\ddot{f}(\bar{\theta}_k, \mathbf{R}_v^{-1})} \\ &= 0.\end{aligned}\tag{3.25}$$

The variance of the proposed method is evaluated by

$$E \{ (\Delta \bar{\theta}_k)^2 \} \simeq \frac{4E \left\{ Re \left[\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \right]^2 \right\}}{\ddot{f}^2(\bar{\theta}_k, \mathbf{R}_v^{-1})}.\tag{3.26}$$

Let u and v be two complex scalars, then

$$Re(u)Re(v) = \frac{1}{2}Re(uv + uv^*) = \frac{1}{2}Re(uv + uv^H).\tag{3.27}$$

According to (3.27), the variance is given by

$$\begin{aligned}E \{ (\Delta \bar{\theta}_k)^2 \} &\simeq \frac{2}{\ddot{f}^2(\bar{\theta}_k, \mathbf{R}_v^{-1})} \\ &\times E \{ Re[\mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \\ &+ \mathbf{a}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \Delta \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}_k)] \}.\end{aligned}\tag{3.28}$$

Let $\{\mathbf{w}_k, k = 1, 2, \dots, 4\}$ denote constant complex vectors. For simplicity, a real scalar C is used, C equals to $K - M$, where K is the number of snapshots used to perform DOA estimation and M is the number of sensors. The following equation is obtained [59] ,

$$\begin{aligned} E \{ & \mathbf{w}_1^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_2 \mathbf{w}_3^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_4 \} \\ &= \frac{1}{(C+1)(C-1)} [\mathbf{w}_1^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_2 \mathbf{w}_3^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_4] \\ &+ \frac{C}{(C+1)(C-1)} [\mathbf{w}_1^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_4 \mathbf{w}_3^H \Delta \mathbf{R}_v^{-1} \mathbf{w}_2] , \end{aligned} \quad (3.29)$$

then

$$\begin{aligned} E \{ (\Delta \bar{\theta}_k)^2 \} &\simeq \frac{2}{(C+1)(C-1) \dot{f}^2(\bar{\theta}_k, \mathbf{R}_v^{-1})} \\ &\times Re \{ \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \\ &+ \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}_k) \\ &+ C \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \\ &+ C \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \}. \end{aligned} \quad (3.30)$$

Let N denote a real scalar and N equal to $\frac{2C}{(C+1)(C-1)}$. From (3.23), since

$$[\dot{\mathbf{a}}_v^H(\bar{\theta}) \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta})]^H = [\mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)] , \quad (3.31)$$

we obtain

$$Re [\mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)] = 0, \quad (3.32)$$

therefore

$$\begin{aligned}
E \{(\Delta \bar{\theta}_k)^2\} &\simeq \frac{N}{\ddot{f}^2(\bar{\theta}_k, \mathbf{R}_v^{-1})} \\
&\quad \times \text{Re}\{\mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k) \\
&\quad + \mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \mathbf{a}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)\} \\
&\simeq N \frac{\mathbf{a}_v^H(\bar{\theta}_k) \mathbf{R}_v^{-1} \mathbf{B} \mathbf{R}_v^{-1} \dot{\mathbf{a}}_v(\bar{\theta}_k)}{\ddot{f}^2(\bar{\theta}_k, \mathbf{R}_v^{-1})}, \tag{3.33}
\end{aligned}$$

where

$$\mathbf{B} = \dot{\mathbf{a}}_v(\bar{\theta}_k) \mathbf{a}_v^H(\bar{\theta}_k) + \mathbf{a}_v(\bar{\theta}_k) \dot{\mathbf{a}}_v^H(\bar{\theta}_k). \tag{3.34}$$

To summarize, the bias and variance of the V-Capon DOA estimation are given in (3.18) and (3.33), respectively. The MSE of the proposed method can be evaluated using (3.14).

3.4 Simulation Results and Discussions

In this section, the proposed method is compared with the standard Capon algorithm. Simulations are performed for 4 test cases. In Experiment I, the source signals are complex Gaussian, in Experiment II, the source signals are complex Gaussian and a fading Channel is considered.

Fading channel models are often used to model the effects of electromagnetic transmission of information over the air in cellular networks and broadcasting. Fading channel models are also used in underwater acoustic communications to model the distortion caused by the water. Since the fading may vary with time, geographical position and/or radio frequency, mathematically, fading is usually modeled as a time-varying random change in the amplitude and phase of the transmitted signal [60, 61].

In Experiment III, the source signals are complex non-Gaussian, in Experiment IV, the source signals are complex non-Gaussian and a fading Channel is considered. These four cases validate the performance for Gaussian and non-Gaussian sources. The sources are

uncorrelated in all the four cases. The used non-Gaussian source signals follow exponential distributions. Channel fading effects are not considered in Experiment I and III.

The simulation setup is as follows: a ULA made of 4 sensors with half-wavelength inter-element spacing is used, with three sources are at 60° , 80° and 110° , additive Gaussian noise is assumed, 500 independent trials with 100 snapshots each are simulated. The root mean square angle error (RMSE) is used as a performance measure and it is defined as

$$RMSE = \sqrt{E \left\{ \frac{1}{L} \sum_{k=1}^L |\theta_k - \hat{\theta}_k|^2 \right\}}, \quad (3.35)$$

where θ_k and $\hat{\theta}_k$ denote the true and estimated DOAs, respectively.

3.4.1 Complex Gaussian Sources

Case 1: Complex Gaussian sources

In this simulation, both source signals and noise are complex Gaussian.

Figure 3.1: Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive white Gaussian noise (AWGN) is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

Figure 3.2: RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise is assumed, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The spatial spectrum and RMSE of Capon, 2nd and 3rd order Volterra Capon are shown in Figure 3.1 and Figure 3.2, respectively. From Figure 3.1, we observe that, all the methods works. The peaks of the spectrum are in good agreement with the true DOAs. Compared with Capon, the peaks of the spatial spectrum of Volterra Capon is sharper. From Figure 3.2, we can see that the performance of Capon and Volterra Capon methods are similar with regard to the RMSE, when the SNR is larger than 20 dB.

Discussion

Any Gaussian signal is completely characterized by its mean and variance. Consequently the higher order statistics (HOS) of Gaussian signals are either zero (e.g. the 3rd order

moment of a Gaussian signal is zero), or contain redundant information. For the complex Gaussian signals and noise, Volterra based Capon is the same as the standard Capon. The similar performance of all the three methods are obtained.

Case 2: Complex Gaussian sources, consider channel fading effects.

Both source signals and noise are complex Gaussian, consider channel fading effects.

Figure 3.3: Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

Figure 3.4: RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise is assumed, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The spatial spectrum and RMSE of Capon, 2nd and 3rd order Volterra Capon are shown in Figure 3.3 and Figure 3.4, respectively. From Figure 3.3, we observe that, all the methods work. The peaks of the spectrum are matched with the true DOAs. Compared with Capon, Volterra Capon shows sharper peaks.

From Figure 3.4, we can see that the RMSE of Volterra Capon is lower than Capon in the low SNR region. The performance of 3rd order Volterra Capon is better than 2nd order Volterra Capon. Similar performance is obtained for all the methods when SNR is larger than 18 dB.

Discussion

Due to the channel fading effect, the source signals received are non-Gaussian, even the Gaussian signals are transmitted. The higher order statistics (HOS) for non-Gaussian signals are non-zero. In principle the Volterra based Capon are less affected by Gaussian background noise than the standard Capon algorithm. The better performance is obtained for Volterra Capon by using the higher order statistics information.

3.4.2 Complex Non-Gaussian Sources

Case 1: Complex non-Gaussian sources

Complex non-Gaussian source signals and complex Gaussian noise are considered in this simulation. The non-Gaussian source signals are assumed to follow exponential distributions. In statistics and probability theory, the exponential distribution is used to describe the time between events in a Poisson process. The probability density function (pdf) of an exponential distribution is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0, \end{cases} \quad (3.36)$$

where $\lambda > 0$ is the rate, or inverse scale parameter. The pdf in equation (3.36) is used to generate the real part and imaginary part of the complex source signals.

Figure 3.5: Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

Figure 3.6: RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The spatial spectrum and RMSE of Capon, 2nd and 3rd order Volterra Capon are shown in Figure 3.5 and Figure 3.6, respectively. From Figure 3.5, we observe that, all the methods work. The peaks of the spectrum are in good agreement with the true DOAs. Compared with Capon, Volterra Capon shows sharper peaks.

From Figure 3.6, we can see that the RMSE of Volterra Capon is lower than Capon in the low SNR region. The performance of 3rd order Volterra Capon is better than 2nd order Volterra Capon. Similar performance is obtained for all the methods when SNR is larger than 25 dB.

Discussion

The better performance is obtained for Volterra Capon when non-Gaussian signals are transmitted. It can be explained from statistical point of view. The higher order statistics (HOS) for non-Gaussian signals are non-zero. The performance is improved by using both the second order and higher order statistics information.

Case 2: Complex non-Gaussian sources, consider channel fading effects

In this simulation, complex non-Gaussian source signals and complex Gaussian noise are considered, the channel fading effects are also considered.

Figure 3.7: Spatial spectrum obtained using Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

Figure 3.8: RMSE of Capon, 2nd and 3rd order V-Capon. A ULA made of 4 sensors with half-wavelength inter-element spacing is used, three sources are from 60° , 80° and 110° , additive complex Gaussian noise, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The spatial spectrum and RMSE of Capon, 2nd and 3rd order Volterra Capon are shown in Figure 3.7 and Figure 3.8, respectively. From Figure 3.7, we observe that, all the methods work. The peaks of the spectrum are matched with the true DOAs. Compared with Capon, Volterra Capon shows sharper peaks. For Volterra Capon, the spatial spectrum on the sources directions is sharper when using higher order.

From Figure 3.8, we can see that the RMSE of Volterra Capon is lower than Capon even in the low SNR region. The performance of 3rd order Volterra Capon is better than 2nd order Volterra Capon. Similar performance is obtained for all the methods when SNR is larger than 25 dB.

Discussion

The better performance is obtained for Volterra Capon by using the higher order statistics information, when the source signals received are non-Gaussian and considering the channel fading effect. It can be explained from two different ways. First, using both the second order and higher order statistics (HOS) information of the non-Gaussian sources increases the estimation accuracy. Second, compared with the widely used linear model, Volterra model is a generalized model. The system can be described more accurately by using Volterra model, which will improve the performance of the DOA estimators.

3.4.3 Detection Probability

To verify the detection probability of the proposed Volterra Capon method, the following simulations are implemented. A ULA with 5 omni-directional sensors are considered. The inter-element spacing is half-wavelength. There are three uncorrelated sources from 60° , 80° and 110° . Furthermore, the signal powers are different and the channel fading effects are considered. Additive white Gaussian noise is assumed, 500 independent trials with 200 snapshots each are simulated.

From Figure 3.9 and Figure 3.11, we can see that higher detection probability is obtained for Volterra Capon method. Furthermore, the resolution is also improved for Volterra Capon method, which can be seen from Figure 3.10 and Figure 3.12. The sources from 60° and 80° are distinguishable for Volterra Capon method. These two sources are undistinguished for the conventional Capon method.

Figure 3.9: Detection probability for Capon and 2nd order Volterra Capon methods with regard to SNR. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex Gaussian and the noise is additive white Gaussian.

Figure 3.10: Spatial spectrum for Capon and 2nd order Volterra Capon methods. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex Gaussian and the noise is additive white Gaussian, $\text{SNR} = 22$ dB.

Figure 3.11: Detection probability for Capon and 2nd order Volterra Capon methods with regard to SNR. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex non-Gaussian and the noise is additive white Gaussian.

Figure 3.12: Spatial spectrum for Capon and 2nd order Volterra Capon methods. A ULA with 5 omni-directional sensors are considered. There are three uncorrelated sources from 60° , 80° and 110° . The sources are complex non-Gaussian and the noise is additive white Gaussian, $\text{SNR} = 25$ dB.

3.5 Conclusion

A new DOA estimation technique bases on the Volterra signal model have been proposed. In the new technique, the higher order Volterra model is used to compute the new augmented Volterra snapshots. Then the DOAs of the sources are estimated using the Capon algorithm base on the new Volterra snapshots. By adding the higher order terms of the Volterra series, the performance of the Capon algorithm is improved. Numerical results show that the proposed approach leads to higher resolution, lower RMSE in DOA estimation relative to those achieved with the standard Capon algorithm.

Here are some of our observations:

1. Complex Gaussian sources

For the complex Gaussian sources and noise, Volterra based Capon is reduced to the standard Capon. The similar performance of all the three methods are obtained.

2. Complex non-Gaussian sources

The higher order statistics (HOS) for the received signals are non-zero when the transmitted signals are non-Gaussian. In principle the Volterra based Capon are less affected by Gaussian background noise than the standard Capon algorithm. The better performance is obtained for Volterra Capon by using the higher order statistics.

3. Complex Gaussian or non-Gaussian sources

Consider the channel fading effect, the better performance is obtained for Volterra Capon by using both the second order and higher order statistics information.

Chapter 4

Volterra MUSIC DOA Estimation

4.1 Introduction

In this Chapter, the spatial Volterra model based MUSIC algorithm is considered. The new technique basically uses additionally the second-order terms of Volterra series to produce augmented Volterra snapshots, the extension for higher-order case is straightforward. The resolution of the proposed method is high comparing with the standard MUSIC algorithm and the 2q-MUSIC algorithm. Simulation results are presented to show the performance of the proposed technique.

The organization of this Chapter is as follows. The proposed DOA estimation algorithm using spatial Volterra model is presented in Section 5.2. In Section IV, the performance of the proposed method is compared with the 2q-MUSIC algorithm. Numerical results of Volterra MUSIC are shown in Section V. In Section VI, the performance analysis of the proposed method is given. Conclusions are drawn in Section VII.

4.2 The Volterra MUSIC DOA Estimation Algorithm

Please refer to Section 1.3.2 for a detailed introduction about the spatial Volterra model.

Voltterra MUSIC DOA Estimation

Now, the MUSIC algorithm is used to estimate the directions of the sources based on the new covariance matrix $\mathbf{R}_v$ and the new array steering vector $\mathbf{a}_v(\theta)$, which are defined in (1.19) and (1.20), respectively.

We obtain N eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ by performing eigen decomposition on $\mathbf{R}_v$. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, the unit-norm eigenvectors associated with $\lambda_1, \dots, \lambda_N$ are denoted by $\mathbf{u}_1, \dots, \mathbf{u}_N$, respectively.

Assume that the L source signals are uncorrelated, for second order case, the maximum dimension of the signal subspace N_s is given by

$$N_s = L + L^2. \quad (4.1)$$

The analysis of N_s in detail is given in the Appendix B.

Figure 4.1 shows a numerical illustration of (4.1). The eigenvalues in group 1 correspond to the original source signals and the cross product term of the source signals. The eigenvalues in group 2 correspond to the noise signals and the cross product between the source and noise signals. While, the eigenvalues in group 3 only correspond to the cross product of the noise signals. Since N_s is the maximum dimension of the signal subspace, then the number of eigenvalues in group 1 is less than or equal to N_s .

The signal and the noise subspaces are spanned by

$$\begin{aligned} \mathbf{U}_s &= [\mathbf{u}_1, \dots, \mathbf{u}_{N_s}] \quad (N \times N_s), \\ \mathbf{U}_n &= [\mathbf{u}_{N_s+1}, \dots, \mathbf{u}_N] \quad (N \times N_e), \end{aligned} \quad (4.2)$$

where $N_e = N - N_s$.

Since the steering vectors corresponding to signal components are orthogonal to the noise

Figure 4.1: Eigenvalues of $\mathbf{R}_v$. A ULA of 4 elements, 1, 2 and 3 uncorrelated source signals, additive white Gaussian noise, $\text{SNR} = 30\text{dB}$, averaged over 100 independent trials.

subspace eigenvectors, we have

$$\mathbf{a}_v^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}_v(\theta) = 0. \quad (4.3)$$

Then the DOAs of the multiple incident signals can be estimated by locating the peaks of the Volterra MUSIC (V-MUSIC) spatial spectrum determined by

$$P_{V-MUSIC} = \frac{1}{\mathbf{a}_v^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}_v(\theta)}. \quad (4.4)$$

The L largest peaks in the V-MUSIC spectrum correspond to the DOAs of the signals imping on the array.

4.2.1 Summary of the proposed DOA Estimation Algorithm

A summary of the proposed V-MUSIC DOA estimation algorithm is given in Algorithm 4.

Algorithm 4 Volterra MUSIC DOA estimation algorithm

Step 1. Obtain the new array steering vector $\mathbf{a}_v(\theta)$.

Step 2. Compute the new Volterra snapshots $\{\mathbf{x}_v(k), k = 1, 2, \dots, K\}$ by (4.4) and estimate the input covariance matrix as

$$\hat{\mathbf{R}}_v = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_v(k) \mathbf{x}_v^H(k). \quad (4.5)$$

where K is the number of snapshots, L is assumed to be known.

Step 3. Perform eigen decomposition on $\hat{\mathbf{R}}_v$, estimate the noise subspace $\mathbf{U}_n$ by (4.2).

Step 4. Compute the V-MUSIC spectrum using (4.4), find the L largest peaks to obtain estimates of the DOAs.

4.2.2 Comparison with the 2q-MUSIC Algorithm

In this Section, the proposed method is compared with the 2q-MUSIC algorithm [24], which is an approach based on higher-order statistics.

The new virtual snapshot of the 2q-MUSIC algorithm is given by

$$\mathbf{x}_q(k) = [\mathbf{x}(k) \otimes \dots \otimes \mathbf{x}^*(k)]_q, \quad (4.6)$$

where $[\]_q$ denotes the number of Kronecker product equal to $q - 1$ and $\mathbf{x}(k)$ is the original snapshot. For $q = 2$, the new snapshot is $\mathbf{x}_q(k) = [\mathbf{x}(k) \otimes \mathbf{x}^*(k)]$, since M array elements are used, then the dimension of the new snapshot is M^2 .

The cumulants matrix of the 2q-MUSIC algorithm $\mathbf{C}_{2q}$ is defined as

$$\mathbf{C}_{2q} = \text{Cum}[\mathbf{x}_q(k) \mathbf{x}_q^H(k)]. \quad (4.7)$$

The corresponding array steering vector is

$$\mathbf{a}_q(\theta) = [\mathbf{a}(\theta) \otimes \cdots \otimes \mathbf{a}^*(\theta)]_q, \quad (4.8)$$

where $\mathbf{a}(\theta)$ is the original array steering vector, which is defined at (1.5a).

Based on the cumulants matrix $\mathbf{C}_{2q}$, the algorithm similar to MUSIC can be used to estimate the DOAs with eigen decomposition on $\mathbf{C}_{2q}$. In some cases, the number of sources L is known, such that $L < M$, but their statistical dependence is not known. In such a case, a conservative approach is to use only the $M^q - L^q$ eigenvectors associated with the smallest eigenvalues to form the 2q-MUSIC noise eigenmatrix $\mathbf{U}_{2q,n}$. This means that the dimension of the signal subspace is L^q . For $q = 2$, the dimension of the signal subspace is L^2 .

The spatial spectrum of the 2q-MUSIC algorithm is given by

$$P_{2q-MUSIC} = \frac{1}{\mathbf{a}_q^H(\theta) \mathbf{U}_{2q,n} \mathbf{U}_{2q,n}^H \mathbf{a}_q(\theta)}. \quad (4.9)$$

The standard MUSIC algorithm is a special case of 2q-MUSIC for $q = 1$.

Form the above analysis, we can see that there are two main differences between the 2q-MUSIC algorithm and the proposed method.

First, for 2q-MUSIC when $q > 1$, only the new higher-order term of the snapshot is used, the original snapshot is only used to construct the the new higher order snapshots. For the proposed V-MUSIC algorithm, both the original snapshot and the constructed higher-order term of the snapshot are used together.

Second, the redundancy of the new higher order snapshot is removed in our method. For simplicity, consider the second order case, which means $q = 2$ for 2q-MUSIC, then all the M^2 second order terms are used to construct the new snapshot (4.6). For the proposed method, only $\frac{M(M+1)}{2}$ second order terms are used to construct the new Volterra snapshots (1.17), as $x_i(k)x_j(k) = x_j(k)x_i(k), 1 \leq i, j \leq 1$.

To evaluate the computational complexity of 2q-MUSIC and V-MUSIC, the dimension

of a snapshot of the two methods is compared. Figure 4.2 shows the dimension comparison of the proposed method and the 2q-MUSIC algorithm for the second order ($q = 2$) and third order ($q = 3$) case. From Figure 4.2, we can see that the dimension of the Volterra snapshot is lower than the snapshot of the 2q-MUSIC algorithm for the same order, which means the lower computational complexity of the proposed method.

Figure 4.2: Snapshots dimension comparison of the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm.

4.3 Numerical Results

4.3.1 Simulation Results for the Proposed V-MUSIC Method

In this section, the proposed method is compared with the standard MUSIC and 2q-MUSIC algorithms. Simulations are performed for 2 test cases. In Case I, two sources are from 100° and 110° (w.r.t array axis) and in Case II, the two sources are at 100° and 103° . These two cases validate the performance for widely spaced and closely spaced sources. The sources

are uncorrelated in both cases.

The simulation setup is as follows: a uniform linear array made of 4 sensors with half-wavelength inter-element spacing is used, additive exponential noise is assumed to be 20 dB below the zero reference, 500 independent trials with 50 snapshots each are averaged to obtain the plots shown in Figure 4.3 and Figure 4.4. From Figure 4.3, we observe that the resolution of the 2q-MUSIC and the proposed method are high compared with the standard MUSIC algorithm in the widely spaced case. Figure 4.4 shows the performance of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed method when sources are closely spaced, the proposed method also exhibits higher resolution. A more detailed analysis will be discussed in Section 4.4.

Figure 4.3: Spatial spectra obtained using the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm. A ULA of 4 elements, SNR=20dB, 50 snapshots are used, 2 non-Gaussian source signals are from 100° and 110° , averaged over 500 independent trials.

Figure 4.4: Spatial spectra obtained using the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm. A ULA of 4 elements, $\text{SNR}=20\text{dB}$, 50 snapshots are used, 2 non-Gaussian source signals are from 100° and 103° , averaged over 500 independent trials.

4.3.2 Simulation Results for the Order Selection Problem

In this simulation, the order selection problem of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the V-MUSIC algorithm is considered. For each method, different number of noise eigenvectors correspond to the smaller eigenvalues are used to perform DOA estimation.

For the standard MUSIC algorithm, the $M - L$ eigenvectors correspond to the smaller eigenvalues of $\mathbf{R}$ are chosen as the noise eigenvectors. Here M is the number of array elements and L is the number of source signals. To estimate the DOAs of the source signals accurately, the information of the number of source signals is needed.

Figure 4.5 shows the spatial spectrum of the standard MUSIC algorithm using different noise subspace. The legend $N \times n$ denotes that the first n noise eigenvectors corresponding

to the n smaller eigenvalues are used to perform DOA estimation. For this simulation, since 4 sensors are used and 2 source signals exist, then the dimension of the noise subspace is 2. Form Figure 4.5, we can see that the standard MUSIC algorithm is sensitive to the dimension selection of the noise subspace. The spatial spectrum gives the wrong direction information of the source signals when the wrong dimension of the noise subspace is selected.

Figure 4.5: Spatial spectra obtained using the standard MUSIC algorithm, the dimension of the noise subspace is different. A ULA of 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 10 independent trials.

Figure 4.6 shows the spatial spectrum of the 2q-MUSIC algorithm using different number of noise eigenvectors. The legend $N \times n$ denotes that the first n noise eigenvectors corresponding to the n smaller eigenvalues are used to perform DOA estimation. From Figure 4.6, we can see that false peaks arise when the dimension of the noise subspace is not selected properly. Similar to the standard MUSIC algorithm, it suffers from the order mismatch problem.

For the proposed Volterra method, the dimension of the noise subspace is $\frac{M(M+3)}{2} -$

$\frac{L(L+3)}{2}$ for second order Volterra case. In this simulation, since $M = 4$ and $L = 2$, the dimension of the noise subspace is 9. Since the dimension of the noise subspace of the proposed method is higher than the standard MUSIC algorithm, there are more choices available to select the noise subspace. Figure 4.7 shows the spatial spectrum of the proposed method using different dimensions of the noise subspace. The legend $N \times n$ denotes that the first n noise eigenvectors corresponding to the n smaller eigenvalues are used to perform DOA estimation. From Figure 4.7, we can see that the proposed method is more robust against the error in selecting the dimension of the noise subspace comparing with the standard MUSIC algorithm and the 2q-MUSIC algorithm.

Figure 4.6: Spatial spectra obtained using the 2q-MUSIC algorithm, the dimension of the noise subspace are different. A ULA with 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 500 independent trials.

Figure 4.7: Spatial spectra obtained using the proposed V-MUSIC algorithm, the dimension of the noise subspace are different. A ULA with 4 elements, SNR=10dB, 50 snapshots are used, 2 non-Gaussian source signals from 60° and 80° , averaged over 500 independent trials.

4.4 Performance Analysis

The performance analysis of the standard MUSIC algorithm were well studied in the last few decades [1]. The variance and Cramer Rao bound of the standard MUSIC algorithm were derived in [62] and [62]. The bias analysis of the standard MUSIC algorithm was carried out by [63]. The first comprehensive study of the resolution capacity of the MUSIC algorithm was due to [64], and [65] derived the probability of resolution without using the Gaussian approximation and obtained better results in the low SNR region. Following from the above literature, statistical analysis is carried out as follows for the proposed V-MUSIC algorithm.

4.4.1 Variance and Bias of the Proposed Method

Define the array manifold of the Volterra array $\mathbf{A}_v(\theta)$ as

$$\mathbf{A}_v(\theta) = [\mathbf{a}_v(\theta_1), \mathbf{a}_v(\theta_2), \dots, \mathbf{a}_v(\theta_L)], \quad (4.10)$$

where $\mathbf{a}_v(\theta)$ is given in (3.3).

Following the results given in [66], let

$$h(\theta) = \dot{\mathbf{a}}_v^H(\theta) [\mathbf{I} - \mathbf{A}_v(\mathbf{A}_v^H \mathbf{A}_v)^{-1} \mathbf{A}_v^H] \dot{\mathbf{a}}_v(\theta), \quad (4.11)$$

where $\dot{\mathbf{a}}_v(\theta) = d\mathbf{a}_v(\theta)/d\theta$. Then the variance of V-MUSIC $V_v(\hat{\theta})$ is evaluated by

$$V_v(\hat{\theta}_i) = \frac{\delta^2}{2K} \left\{ [\mathbf{P}^{-1}]_{ii} + \delta^2 [\mathbf{P}^{-1} (\mathbf{A}_v^H \mathbf{A}_v)^{-1} \mathbf{P}^{-1}]_{ii} \right\} / h(\theta_i), \quad (4.12)$$

where K is the number of snapshots, $\mathbf{P}$ is the covariance matrix of the source waveforms and $(\cdot)_{ij}$ denotes the i, j element of the matrix, θ_i is the direction of the i^{th} signal and δ^2 is the power of the noise.

Here we consider the uncorrelated signals. In such a case, the matrix $\mathbf{P}$ is diagonal, and (4.12) reduces to

$$V_v(\hat{\theta}_i) = \frac{1}{2K \cdot SNR_i} \left[1 + \frac{[(\mathbf{A}_v^H \mathbf{A}_v)^{-1}]_{ii}}{SNR_i} \right] / h(\theta_i), \quad (4.13)$$

where $SNR_i = \mathbf{P}_{ii}/\delta^2$.

Define

$$f(\theta) = \mathbf{a}_v^H(\theta) \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}_v(\theta). \quad (4.14)$$

Taking first and second derivative of $f(\theta)$ with respect to θ at θ_i ,

$$f'(\theta_i) = 2Re[\dot{\mathbf{a}}_v^H(\theta_i) \mathbf{U}_n \mathbf{U}_n^H \dot{\mathbf{a}}_v(\theta_i)], \quad (4.15)$$

and

$$f''(\theta_i) = 2\text{Re}[\dot{\mathbf{a}}_v^H(\theta_i)\mathbf{U}_n\mathbf{U}_n^H\dot{\mathbf{a}}_v(\theta_i) + \mathbf{a}_v^H(\theta_i)\mathbf{U}_n\mathbf{U}_n^H\ddot{\mathbf{a}}_v(\theta_i)]. \quad (4.16)$$

As $\hat{\theta}_i$ is a minimum point of $f(\theta)$, then $f'(\hat{\theta}_i) = 0$. Taking a first-order Taylor expansion of $f'(\hat{\theta}_i)$ around $\theta = \theta_i$,

$$f'(\hat{\theta}_i) \simeq f'(\theta_i) + f''(\theta_i)(\hat{\theta}_i - \theta_i). \quad (4.17)$$

Substitute $f'(\theta_i)$ and $f''(\theta_i)$ into (4.17), we obtain an approximate expression for the bias of the proposed method $B_v(\hat{\theta}_i)$,

$$B_v(\hat{\theta}_i) = -\frac{f'(\theta_i)}{f''(\theta_i)}. \quad (4.18)$$

To show the performance of the proposed V-MUSIC method, the standard deviation, bias and root mean square error (RMSE) of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed method are compared. The simulation setup is as follows: a ULA with 4 elements and half-wavelength inter-element spacing is used, two non-Gaussian sources from 60° and 70° , 50 snapshots are used.

Figure 4.8 shows the standard deviation of the proposed algorithm, the standard MUSIC algorithm and the 2q-MUSIC algorithm w.r.t the SNR. Figure 4.9 shows the bias of the three methods w.r.t the SNR. From Figure 4.8 and Figure 4.9, we observe that the standard deviation and bias of the proposed method are small compared with the standard MUSIC algorithm and the 2q-MUSIC algorithm. Figure 4.10 shows the RMSE of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed method.

4.4.2 Probability of Resolution of the Proposed Method

In order to analyze the performance of V-MUSIC in terms of the resolution probability, we use the following random inequality [65]:

$$P_v(\theta_m) - \frac{1}{2}(P_v(\theta_1) + P_v(\theta_2)) < 0, \quad (4.19)$$

Figure 4.8: Standard deviations of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.

where θ_1 and θ_2 are the directions of arrival of the two signals and θ_m denotes their mean. $P_v(\theta)$ is the spatial spectrum defined in (4.4) as a function of the angle of arrival θ .

To illustrate the performance of the proposed algorithm, the following experiment is conducted.

Figure 4.11 shows the probability of resolution of the MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR. In the simulation, we assume a ULA of 4 elements and an inter-element spacing of half-wavelength, two equal power, narrow band uncorrelated non-Gaussian sources from 100° and 103° impinging on the array. The number of snapshots is 50. From Figure 4.11, we see that, the probability of resolution is increasing with the order of the Volterra snapshots.

The increase in resolution can be explained from two different ways. First, from (1.12), it can be seen that a signal can be expanded in Volterra series. The MUSIC algorithm only uses the first order terms of the series, which is a special case of Volterra series. Since the

Figure 4.9: Bias of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.

higher order terms of the Volterra series also contains the information of the signals. Higher resolution is obtained in DOA estimation by using both the original and the higher order Volterra snapshots. We can also understand this based on the virtual array concept. The higher order terms are virtual snapshots, which increase the effective virtual aperture.

4.5 Conclusions

A new DOA estimation technique based on the Volterra signal model has been proposed. In the new technique, the second order Volterra model is used to compute the new augmented Volterra snapshots, the extension for higher-order case is straightforward. Then the DOAs of the signals are estimated using the MUSIC algorithm based on the new Volterra snapshots and the Volterra steering vector. By adding the second order terms of the Volterra series, the performance of the MUSIC algorithms is improved. Numerical results show that the

Figure 4.10: RMSE of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR, 500 independent trials.

proposed approach leads to higher resolution, small bias and variance in DOA estimation relative to those achieved with the standard MUSIC algorithm and the 2q-MUSIC algorithm.

Figure 4.11: Probability of Resolution of the standard MUSIC algorithm, the 2q-MUSIC algorithm and the proposed V-MUSIC algorithm w.r.t SNR.

Chapter 5

Spatial Smoothing for Volterra Based Methods

5.1 Introduction

The standard Capon and MUSIC type methods encounter difficulties when the sources are correlated or coherent. However, in practice such a phenomenon is common. Such as the multi-path propagation effect in wireless communication and the intentional unfriendly jamming in radar applications. Multipath models in the sensor and radar environment are discussed in [67]. Correlated signals are signals whose correlation coefficient is non-zero. The magnitude of the correlation coefficient for coherent signals equals to one.

The covariance matrix of the correlated sources is rank deficient. Data preprocessing is needed to remove the coherency. There are four coherency removing techniques:

1. Spatial smoothing which is introduced in [68].
2. Array moving approach which is discussed in [69] and is analyzed in detail in [70].
3. Constrained ML techniques are described in [67].

4. Generalized eigenvector approach proposed in [71] that requires the solution of a non-linear programming problem.

Since spatial smoothing is one of the widely used techniques for correlated or coherent sources, in this chapter, we are focusing on DOA estimation methods with spatial smoothing. The M sensors are divided into overlapping Q subarrays of size m_s , with sensors $\{1, 2, \dots, m_s\}$ forming the first subarray, sensors $\{2, 3, \dots, m_s + 1\}$ forming the second subarray. Let $\mathbf{x}_{(n)}(k)$ denote the snapshots of the n^{th} subarray given by $\mathbf{x}_{(n)}(k) = [x_n(k), \dots, x_{m_s+n-1}(k)]$.

5.1.1 Forward Spatial Smoothing

For forward spatial smoothing [68], the covariance matrix of the n_{th} subarray is given by

$$\mathbf{R}_{(n)} = E \left[\mathbf{x}_{(n)}(k) \mathbf{x}_{(n)}^H(k) \right]. \quad (5.1)$$

Then the new covariance matrix after forward spatial smoothing is obtained by

$$\mathbf{R}_f = \frac{1}{Q} \sum_{n=1}^Q \mathbf{R}_{(n)}. \quad (5.2)$$

The DOAs can be estimated based on the new covariance matrix $\mathbf{R}_{SS}$.

5.1.2 Forward-Backward Spatial Smoothing

For forward-only smoothing scheme more sensor elements are used compared with the conventional ones without spatial smoothing, and in particular $2L$ sensor elements are required to estimate any L DOAs. In [72], the authors propose an improved spatial smoothing scheme called the forward-backward smoothing scheme and prove that at most $\lceil 3L/2 \rceil$ elements are needed to estimate L DOAs. In addition to the forward subarrays, this scheme makes use of complex conjugated backward subarrays of the original array to achieve superior performance.

Then the new covariance matrix for forward-backward spatial smoothing is obtained by

$$\mathbf{R}_{fb} = \frac{1}{2Q} \sum_{n=1}^Q \left[\mathbf{R}_{(n)} + \mathbf{J} \mathbf{R}_{(n)}^* \mathbf{J} \right]. \quad (5.3)$$

where $*$ denotes the complex conjugated and $\mathbf{J}$ is the exchange matrix. The exchange matrix is a special case of a permutation matrix, where the elements on the counterdiagonal are one and all other elements are zero,

$$\mathbf{J} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}. \quad (5.4)$$

The DOAs are estimated based on the new covariance matrix $\mathbf{R}_{fb}$.

5.2 V-Capon and V-MUSIC with Spatial Smoothing

5.2.1 Forward spatial smoothing for Volterra based methods

Let $\mathbf{x}_{(n)}(k)$ denote the snapshots of the n_{th} subarray. Then the new snapshots of the n_{th} subarray is given by

$$\mathbf{x}_{v_{(n)}}(k) = \left[(\mathbf{x}_{(n)}(k))^T, (\mathbf{x}_{(n)}(k) \otimes \mathbf{x}_{(n)}(k))^T \right]^T. \quad (5.5)$$

The Volterra covariance matrix of the n^{th} subarray is given by

$$\mathbf{R}_{v_{(n)}} = E \left[\mathbf{x}_{v_{(n)}}(k) \mathbf{x}_{v_{(n)}}^H(k) \right]. \quad (5.6)$$

Then the new Volterra covariance matrix after forward spatial smoothing is obtained by

$$\mathbf{R}_{fv} = \frac{1}{Q} \sum_{n=1}^Q \mathbf{R}_{v(n)}. \quad (5.7)$$

The DOAs can be estimated based on the new covariance matrix $\mathbf{R}_{fv}$ using the proposed Volterra MUSIC and Volterra Capon algorithms.

5.2.2 Forward-backward spatial smoothing for Volterra based methods

Let $\mathbf{x}_{(n)}(k)$ denotes the snapshots of the n_{th} subarray. Similar to (5.5) and (5.6), the new snapshots of the n^{th} subarray is given by

$$\mathbf{x}_{v(n)}(k) = \left[(\mathbf{x}_{(n)}(k))^T, (\mathbf{x}_{(n)}(k) \otimes \mathbf{x}_{(n)}(k))^T \right]^T. \quad (5.8)$$

The Volterra covariance matrix of the n^{th} subarray is given by

$$\mathbf{R}_{v(n)} = E \left[\mathbf{x}_{v(n)}(k) \mathbf{x}_{v(n)}^H(k) \right]. \quad (5.9)$$

Then the new Volterra covariance matrix after forward-backward spatial smoothing is obtained by

$$\mathbf{R}_{fbv} = \frac{1}{2Q} \sum_{n=1}^Q \left[\mathbf{R}_{v(n)} + \mathbf{J}_v \mathbf{R}_{v(n)}^* \mathbf{J}_v \right], \quad (5.10)$$

where $\mathbf{J}_v = \begin{pmatrix} \mathbf{J}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 \end{pmatrix}$, $\mathbf{J}_1 \in \mathcal{R}^{M_s \times M_s}$ and $\mathbf{J}_2 \in \mathcal{R}^{M_s^2 \times M_s^2}$ are two exchange matrices, respectively. For $\mathbf{J}_1$ and $\mathbf{J}_2$, the elements on the counterdiagonal are one and all other elements are zero. The DOAs are estimated based on the new covariance matrix $\mathbf{R}_{fbv}$ using the proposed Volterra MUSIC and Volterra Capon algorithms.

5.3 Numerical Results for V-MUSIC with Spatial Smoothing

In this section, the proposed method is compared with the standard MUSIC algorithm with forward spatial smoothing.

The simulation setup is as follows: a ULA made of 8 sensors with half-wavelength inter-element spacing is used, the 8 sensors are divided into overlapping 4 subarrays of size 5, with sensors $\{1, 2, \dots, 5\}$ forming the first subarray, sensors $\{2, 3, \dots, 6\}$ forming the second subarray, etc. Additive white Gaussian noise is assumed to be 20 dB below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The root mean square angle error (RMSE) is used as a performance measure and it is defined as

$$RMSE = \sqrt{E \left\{ \frac{1}{L} \sum_{k=1}^L |\theta_k - \hat{\theta}_k|^2 \right\}}, \quad (5.11)$$

where E is the expectation operator, θ_k and $\hat{\theta}_k$ denote the true and estimated DOAs, respectively.

5.3.1 Complex Gaussian Sources

Case 1: Complex Gaussian Sources

From Figure 5.1 and Figure 5.2, we observe that the resolution of the proposed method is high and the RMSE is low comparing with the standard MUSIC algorithm.

Figure 5.1: Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.2: RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of MUSIC and 2nd order Volterra MUSIC with spatial smoothing are shown in Figure 5.1 and Figure 5.2, respectively. From Figure 5.1, we observe that spatial smoothing methods works. The peaks of the spectrum are in good agreement with the true DOAs. For the complex Gaussian signals and noise, Volterra based MUSIC is the same as the standard MUSIC method. From Figure 5.2, we can see that the performance of MUSIC and V-MUSIC methods are similar with regard to the RMSE.

Case 2: Complex Gaussian Sources, Consider Channel Fading Effects.

The spatial spectrum and RMSE of MUSIC, 2nd Volterra MUSIC are shown in Figure 5.3 and Figure 5.4, respectively.

From Figure 5.3, we observe that the peaks of the MUSIC and V-MUSIC spectrum are

Figure 5.3: Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

matched with the true DOAs. Similar to the cases without considering the channel fading affects, Volterra MUSIC shows sharper peaks.

Consider the channel fading effect, the better performance is obtained for Volterra MUSIC. From Figure 5.4, we can see that the RMSE of Volterra MUSIC is lower than MUSIC in the low SNR region. The RMSE of V-MUSIC is about 0.2 degree lower than MUSIC, when the $\text{SNR} = 0\text{ dB}$. The improvement becomes smaller with the SNR increasing. Similar performance is obtained for MUSIC and V-MUSIC methods when SNR is larger than 25 dB .

Figure 5.4: RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

5.3.2 Complex Non-Gaussian Sources

For the following simulations, the pdf of the non-Gaussian sources can be referred to in Section 3.4.2.

Case 1: Complex Non-Gaussian sources

Figure 5.5: Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.6: RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of MUSIC, 2nd Volterra MUSIC are shown in Figure 5.5 and Figure 5.6, respectively. From Figure 5.5, we observe that the peaks of the MUSIC and Volterra MUSIC spectrum are in good agreement with the true DOAs. Compared with MUSIC, Volterra MUSIC shows sharper peaks.

Since the sources are non-Gaussian, the better performance is obtained for Volterra MUSIC by using both the second order and higher order statistics. From Figure 5.6, we can see that the RMSE of Volterra MUSIC is lower than MUSIC in the low SNR region. When $\text{SNR} = 0\text{ dB}$, the RMSE of V-MUSIC is about 0.2 degree lower than MUSIC. The improvement is becoming smaller with increasing SNR. Similar performance is obtained when SNR is larger than 30 dB.

Case 2: Complex Non-Gaussian Sources, Consider Channel Fading Effects

Figure 5.7: Spatial spectra obtained using MUSIC and the proposed V-MUSIC algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.8: RMSEs of MUSIC and the proposed V-MUSIC algorithm. (with spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of MUSIC, 2nd order Volterra MUSIC are shown in Figure 5.7 and Figure 5.8, respectively. From Figure 5.7, we observe that the peaks of the MUSIC and V-MUSIC spectrum are matched with the true DOAs. The peaks of the Volterra MUSIC are sharper when compared with MUSIC.

From Figure 5.8, we can see that the RMSE of Volterra MUSIC is lower than MUSIC. It is about 0.2 degree improvement for V-MUSIC when $\text{SNR} = 0 \text{ dB}$. For V-MUSIC, the RMSE is about 0.05 degree lower than MUSIC, even when the SNR is 30 dB .

5.4 Numerical Results for V-Capon with Spatial Smoothing

In this section, the proposed method is compared with the standard Capon algorithm with forward spatial smoothing.

The simulation setup is as follows: a uniform linear array made of 8 sensors with half-wavelength inter-element spacing is used, the 8 sensors are divided into overlapping 4 subarrays of size 5, with sensors $\{1, 2, \dots, 5\}$ forming the first subarray, sensors $\{2, 3, \dots, 6\}$ forming the second subarray, etc. Additive Gaussian noise is assumed to be 20 *dB* below the zero reference, 500 independent trials with 100 snapshots each are averaged to obtain the plots.

The root mean square angle error (RMSE) is used as performance measure and it is defined as

$$RMSE = \sqrt{E \left\{ \frac{1}{L} \sum_{k=1}^L |\theta_k - \hat{\theta}_k|^2 \right\}}, \quad (5.12)$$

where E is the expectation operator, θ_k and $\hat{\theta}_k$ denote the true and estimated DOAs, respectively.

5.4.1 Complex Gaussian Sources

Complex Gaussian Sources

From Figure 5.9 and Figure 5.10, we observe that the resolution of the proposed method is high and the RMSE is low comparing with the standard Capon algorithm.

Figure 5.9: Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.10: RMSEs of Capon and the proposed V-Capon algorithm (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of Capon and 2nd order Volterra Capon are shown in Figure 5.9 and Figure 5.10, respectively. From Figure 5.9, we observe that, all the spatial smoothing method works for Volterra Capon. The peaks of the spectrum are in good agreement with the true DOAs. Compared with Capon, the peaks of Volterra Capon spectrum are sharper.

Since both the sources and noise are Gaussian, Volterra based Capon is the same as the standard Capon method. From Figure 5.10, we can see that the performance of the Capon and Volterra Capon methods are similar with regard to the RMSE.

Case 2: Complex Gaussian Sources, Consider Channel Fading Effects.

In this simulation, the channel fading effects is considered. To simulate the fading affects, different random attenuation factors are used for sources from different directions.

Figure 5.11: Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.12: RMSEs of Capon and the proposed V-Capon algorithm. (with spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of Capon and 2nd order Volterra Capon are shown in Figure 5.11 and Figure 5.12, respectively. From Figure 5.11, we observe that, spatial smoothing method works for Volterra Capon when considering the channel fading affects. The DOAs can be estimated from the peaks of the spectrum.

From Figure 5.12, we can see that the RMSE of Volterra Capon is lower than Capon in the low SNR region. The RMSE of Volterra Capon is about 0.15 degree lower than Capon, when the $\text{SNR} = 0\text{ dB}$. The improvement becomes smaller with the SNR increasing. Similar performance is obtained for Capon and Volterra Capon methods when SNR is larger than 30 dB.

5.4.2 Complex Non-Gaussian Sources

In the following simulations, the non-Gaussian signal sources used follow the exponential distributions as in Section 5.3.2.

Case 1: Complex Non-Gaussian Sources

Figure 5.13: Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.14: RMSEs of Capon and the proposed V-Capon algorithm (with spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of Capon and 2nd order Volterra Capon are shown in Figure 5.13 and Figure 5.14, respectively. From Figure 5.13, we observe that, all the spatial smoothing method works when the sources are non-Gaussian. the peaks of the spectrum are matched with the true DOAs.

Since the sources are non-Gaussian, the better performance is obtained for Volterra Capon by using both the second order and higher order statistics. From Figure 5.14, we can see that the RMSE of Volterra Capon is lower than Capon even in the low SNR region. It is about 0.2 degree improvement for Volterra Capon when $\text{SNR} = 0\text{ dB}$. Similar performance is obtained when SNR is larger than 24 dB.

Case 2: Complex Non-Gaussian Sources, Consider Channel Fading Effects.

Figure 5.15: Spatial spectra obtained using Capon and the proposed V-Capon algorithm (with and without spatial smoothing). A ULA of 8 elements, $\text{SNR}=20\text{dB}$, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

Figure 5.16: RMSEs of Capon and the proposed V-Capon algorithm. (with spatial smoothing). A ULA of 8 elements, SNR=20dB, 100 snapshots are used, 3 source signals are from 60° , 80° and 120° , the first 2 sources are coherent, averaged over 500 independent trials.

The spatial spectrum and RMSE of Capon and 2nd order Volterra Capon are shown in Figure 5.15 and Figure 5.16, respectively. From Figure 5.15, we observe that, all the methods works. the peaks of the spectrum are in good agreement with the true DOAs. Compared with Capon, Volterra Capon shows sharper peaks. For Volterra Capon, the spatial spectrum on the sources directions is sharper when using higher order.

From Figure 5.16, we can see that the RMSE of Volterra Capon is lower than Capon even in the low SNR region. It is about 0.25 degree improvement for Volterra Capon when SNR = 0 dB. For Volterra Capon, the RMSE is about 0.05 degree lower than Capon, even when the SNR is 30 dB.

5.5 Conclusions

A new DOA estimation technique based on Volterra signal model with spatial smoothing has been proposed to remove the coherence of the sources. In the new technique, the higher order Volterra model is used to compute the new augmented Volterra snapshots. Then the DOAs of the signals are estimated based on the new Volterra snapshots by using the standard DOA estimation techniques, such as MUSIC or Capon. By adding the high order terms of the Volterra series, the performance of the standard DOA estimation algorithm is improved. Numerical results show that lower RMSE in DOA estimation relative to those achieved with standard MUSIC or Capon.

Chapter 6

Array Processing Using Spatial IIR Structure

6.1 Introduction

Array beamforming can be interpreted as a spatial filter which allows signals from specific directions to pass through it while suppressing the interference signals. This is analogous to a time-domain digital filter which allows signals within its passband. Therefore, the response of the array beamformer (beampattern) is equivalent to the frequency response of the time-domain digital filter as will be shown in Section 6.2. This equivalence arises because a digital filter samples the signal in time-domain whereas an array samples the signal in spatial domain.

With such close relation between array snapshots and discrete-time signals, it is interesting to explore the application of concepts and algorithms developed in different streams of research for array processing. It is well acclaimed that digital filters with desired frequency response can be implemented with either finite impulse response (FIR) structure or infinite impulse response (IIR) structure. In array processing, a one-to-one correspondence of these filter structures with array structures has not been exhaustively studied. While an

FIR filter structure processes uniformly sampled discrete-time signal, a uniform linear array (ULA) structure samples the signal at uniform spatial displacement and the corresponding weights process similar to an FIR filter [73]. Therefore, the FIR filter response is analogous to the array beampattern. The feasibility of applying FIR filter design techniques in array processing has been explored recently by considering special types of FIR filter design techniques, such as frequency-response masking (FRM) [74,75] and interpolated FIR (IFIR) [76]. In [77] a wideband DOA technique is proposed based on the correlation between the sensor measurements of a uniform linear array (ULA) and a sine or cosine waveform with known frequency.

Table 6.1: Analogy between filtering and beamforming.

Digital Filter (Time Domain)	Array Beamformer (Spatial Domain)
FIR	Uniform Linear Array
IIR	? ?

Following the analogy between FIR filter structure and ULA, the question in hand is: what is the array structure in spatial domain that is analogous to the IIR filter structure? Table 6.1 shows the correspondence between the structures available in temporal and spatial processing. Completion of this table with equivalent array structure enables us to extend several filter designs for the design of beamformer. This can potentially extend the advantages of IIR structure to array processing in the following ways:

1. IIR processing provides increased degrees of freedom over conventional array processing to control the array performance. Moreover, the well-established IIR filter design theory can be used to design an IIR array.
2. In comparison with FIR filters, the number of IIR filter taps required for a desired performance is very less. Likewise, for a desired array response, we hope that an array designed with IIR structure requires a smaller array aperture than the conventional array. This reflects in a significant reduction in cost and complexity of the entire

system.

In this chapter, we present the possibility of replacing FIR structure with approximated IIR structure for DOA estimation and beamforming applications. We propose a new DOA estimation technique which incorporates a virtual spatial feedback resulting in an IIR structure. While the time delay between the taps of the discrete-time IIR spatial structure in the feedback and feedforward networks is a constant proportional to sampling interval, the analogous delay in spatial domain corresponds to the relative propagation delay across the sensor array which is proportional to the source direction. In the IIR array structure, array snapshots can be directly used for the moving average part of the filter. In order to develop the feedback section of the IIR structure, we propose a virtual structure with an approximate propagation delay (which is dependent on the source direction as discussed). The performance of the proposed technique is observed to have improved performance over the Capon method. This work may provide a platform for a possible subspace based DOA estimation for IIR array structure.

Furthermore, we propose a primitive beamformer with an IIR array implementation. Although the configuration presented in this work is not optimal, the scope of this contribution is to prove the concept of IIR structured beamformer. In order to implement a genuine IIR array, it is necessary to have spatial feedback which can be realized using the concept of subarrays. The output of previous subarrays can be used for spatial feedback. The stability of the IIR structure is ensured by using a notch filter structure [78–85] which constrains the zeros and poles to lie within the unit circle. Although the beamformer capability is achieved, the subarray feedback requires an increased array size.

This chapter is organized as follows. The equivalence between an FIR filter and a linear array is introduced in Section 6.2. In Section 6.3, application of spatial IIR notch structure in DOA estimation is described. Problem and possibility of IIR structure in spatial domain and implementation of IIR array processing are discussed in Section 6.4. Conclusions are drawn in Section 6.5.

6.2 Equivalence Between An FIR Filter and A Uniform Linear Array

In this section, we review the equivalence observed between a temporal FIR filter structure and a ULA [73].

Figure 6.1: Spatial sampling of the incoming wavefront by a ULA.

Figure 6.1 depicts a ULA of m sensors used for processing narrowband signals. The propagating waves in space are sampled by the sensors and linearly combined to give the sensor output,

$$y(t) = \sum_{k=0}^{m-1} w_k^* x_k(t) = \mathbf{w}^H \mathbf{x}(t), \quad (6.1)$$

where $\mathbf{w} = [w_0, w_1, \dots, w_{m-1}]^T$ (for the case of adaptive array, $\mathbf{w}$ is data dependent and is updated as a function of time) and $\mathbf{x}(t) = [x_0(t), x_1(t), \dots, x_{m-1}(t)]^T$ are the weight and data vectors, respectively. It is conventional to multiply the data by conjugates of the weights to simplify notation.

Let θ denote the direction measured w.r.t. the array axis, $\tau_k(\theta)$ is the time delay on

sensor k w.r.t. the reference and ω_0 is the temporal frequency. The beamformer response is given by

$$H(\theta, \omega_0) = \sum_{k=0}^{m-1} w_k^* e^{j\omega_0 \tau_k(\theta)} = \mathbf{w}^H \mathbf{d}(\theta, \omega_0), \quad (6.2)$$

where $\mathbf{d}(\theta, \omega_0)$ is known as the steering vector. The correspondence between FIR filtering and beamforming is closest when operated at a single temporal frequency ω_0 and the array geometry is linear and equi-spaced. Now

$$\tau_k(\theta) = \frac{d_k \cos(\theta)}{c} = k \frac{d \cos(\theta)}{c}, \quad (6.3)$$

where $\{d_k, k = 0, 1, \dots, m-1\}$ is the distance between the k^{th} element and the endfire array element chosen as the reference, d is the inter-element spacing, c is the propagation speed.

Figure 6.2: Tapped-delay line implementation of an FIR filter.

For comparison, we now study a temporal $(m-1)^{th}$ order FIR filter shown in Figure 6.2. The sampled input signal is delayed by a factor $z^{-1} = e^{-j\omega}$ where ω is the normalized frequency and the delayed samples are linearly combined to get the filter output,

$$y(n) = \sum_{k=0}^{m-1} w_k^* x(n-k). \quad (6.4)$$

The FIR filter response is given by

$$H(\omega) = \sum_{k=0}^{m-1} w_k^* e^{-j\omega k} = \mathbf{w}^H \mathbf{d}(\omega). \quad (6.5)$$

Here, the vector $\mathbf{d}(\omega)$ describes the phase of the complex sinusoid at each tap in the FIR filter.

From (6.2) and (6.5), the similarity in the structure of a ULA and an FIR filter is apparent. Thus, temporal frequency in an FIR filter corresponds to the cosine of direction in a narrowband, linear, equi-spaced array.

6.3 DOA Estimation Using Spatial Domain IIR Notch Structure

This section provides a study of adaptive IIR notch filter followed by a new DOA estimation technique incorporating adaptive IIR notch filter structure. Comparisons with the Capon and ME methods are shown along with the performance analysis.

6.3.1 Adaptive IIR Notch Filtering

A complex m_{th} order IIR notch filter is characterized by the following transfer function [86],

$$H(z) = \frac{1 + a_1^* z^{-1} + a_2^* z^{-2} + \dots + a_m^* z^{-m}}{1 + r a_1^* z^{-1} + r^2 a_2^* z^{-2} + \dots + r^m a_m^* z^{-m}}, \quad (6.6)$$

where $0 < r < 1$, is a constant close to 1, which controls the notch bandwidth and $\{a_1, \dots, a_m\}$ are the weights of the filter. The stability of the IIR notch filter is ensured if all the poles are present within the unit circle.

The complex adaptive IIR notch filtering algorithm [78] can be used to estimate the

weights of the filter. For a complex input signal $x(n)$, the output of the filter is given by

$$y(n) = \frac{A(z^{-1})}{A(rz^{-1})}x(n), \quad (6.7)$$

where z^{-1} is the unit delay operator, i.e., $z^{-1}x(n) = x(n-1)$. From (6.7), it is possible to write the following recursive difference equation:

$$y(n) = x(n) + \sum_{k=1}^m a_k^* x(n-k) - \sum_{k=1}^m r^k a_k^* y(n-k). \quad (6.8)$$

With

$$\boldsymbol{\varphi}(n) = \begin{bmatrix} x(n-1) - ry(n-1) \\ \vdots \\ x(n-m) - r^m y(n-m) \end{bmatrix},$$

the output can be rewritten as

$$y(n) = x(n) + \boldsymbol{\psi}^H(n)\boldsymbol{\varphi}(n), \quad (6.9)$$

where $\boldsymbol{\psi}(n) = [a_1(n), \dots, a_m(n)]^T$ are the adaptive weights of the IIR notch filter when it is implemented adaptively.

The purpose of using a notch filter is to block the signal present in a particular band. This can be achieved by minimizing the output power of the filter

$$F = E[|y(n)|^2]. \quad (6.10)$$

The adaptation of filter weights using the gradient descent algorithm [87] is given by

$$\boldsymbol{\psi}(n+1) = \boldsymbol{\psi}(n) - \mu \boldsymbol{\phi}(n) y^*(n), \quad (6.11)$$

where $\boldsymbol{\phi}(n) = [\frac{\partial y(n)}{\partial a_1^*(n)}, \dots, \frac{\partial y(n)}{\partial a_m^*(n)}]^T$ and $0 < \mu \leq 1$ is the stepsize. Since this algorithm has

a very slow convergence to the optimal solution, the inverse of the data correlation matrix $\mathbf{R} = E\{\mathbf{xx}^H\}$, can be incorporated to accelerate the convergence as follows:

$$\psi(n+1) = \psi(n) - \mu \mathbf{R}^{-1}(n) \phi(n) y^*(n). \quad (6.12)$$

In order to reduce the computation complexity in evaluating $\mathbf{R}^{-1}$, $\mathbf{P}(n) = \mathbf{R}^{-1}(n)$ is directly updated using the matrix inversion lemma [88]:

$$\mathbf{P}(n) = \frac{1}{\lambda} \left[\mathbf{P}(n-1) - \frac{\mathbf{P}(n-1) \phi(n) \phi^H(n) \mathbf{P}(n-1)}{\lambda + \phi^H(n) \mathbf{P}(n-1) \phi(n)} \right]. \quad (6.13)$$

where λ is the forgetting factor, $0 < \lambda < 1$.

6.3.2 The Proposed DOA Estimation Technique

With the equivalence between a discrete-time FIR filter and a ULA studied in Section 6.2, we now introduce an array whose output is obtained from an IIR structure as shown in Figure 6.3. In order to ensure stability, the above discussed adaptive IIR notch filters is used. The sensor outputs are weighted with the coefficients of the numerator of $H(j\omega)$ in (6.14). The desired time delay for the auto regressor is obtained approximately which will be discussed in Section 6.4.

To explain this new structure, we consider a ULA with 3 elements which can identify a maximum of 2 directions. We can write an equivalent transfer function for this structure as

$$H(j\omega) = \frac{1 + a_1^* e^{-j\omega\tau(\theta)} + a_2^* e^{-j2\omega\tau(\theta)}}{1 + r a_1^* e^{-j\omega\tau(\theta)} + r^2 a_2^* e^{-j2\omega\tau(\theta)}}, \quad (6.14)$$

where $\tau(\theta)$ is the inter-element time delay for the source from direction θ .

Considering the first sensor to be the reference, the snapshots at time index t can be

Figure 6.3: The proposed structure for DOA estimation.

written as

$$x_k(t) = x(t - k\tau(\theta)), \quad k = 0, 1, 2. \quad (6.15)$$

In order to obtain the feedback component, the array output

$$y_k(t) = y(t - k\tau(\theta)), \quad k = 0, 1, 2 \quad (6.16)$$

is passed through the virtual delay elements shown in Figure 6.3. The output signal at time t , based on (6.8), is given by

$$y(t) = x(t) + \sum_{k=1}^2 \left(a_k^* x(t - k\tau(\theta)) - r^k a_k^* y(t - k\tau(\theta)) \right). \quad (6.17)$$

Since the real $\tau(\theta)$ is unknown at the beginning, a fixed τ is used to obtain $y(t)$. Similar to (6.9), we can write the above equation (6.17) as

$$y(t) = x(t) + \boldsymbol{\psi}^H(t)\boldsymbol{\varphi}(t). \quad (6.18)$$

For fast convergence, we use the adaptive algorithm shown in (6.12) as

$$\boldsymbol{\psi}(t+1) = \boldsymbol{\psi}(t) - \mu\boldsymbol{\phi}(t)\mathbf{P}(t)y^*(t), \quad (6.19)$$

where $t+1$ denotes one sample delay, $\mathbf{P}(t)$ is the inverse correlation matrix which can be updated using (6.13) with $\mu = 1$.

For the IIR notch structure used, the coefficients in the numerator and denominator have to be constrained such that the structure is stable. This requires the zeros of the filter structure to be situated on the unit circle and the poles to be placed inside the unit circle [80]. While the DOAs of sources can be estimated by plotting the spatial spectrum, a non-search procedure is to estimate source DOAs from the phase of complex roots of (6.14), i.e., by setting,

$$1 + a_1^* \exp(-j\omega\tau(\theta)) + a_2^* \exp(-j2\omega\tau(\theta)) = 0. \quad (6.20)$$

The relationship between the phase of the complex roots α and DOAs θ are $\alpha = 2\pi d \cos(\theta)$, where d is the inter-element spacing of the ULA with regard to the signal wavelength. The complex adaptive notch filter algorithm explained above is summarized in Table 6.2.

6.3.3 Simulation Results

In this section, we evaluate the proposed method with simulated and experimental data. A ULA formed by three omni-directional sensors with half-wavelength inter-element spacing

Table 6.2: Complex Adaptive IIR Notch Algorithm

Notations:

$$[x_0(t), \dots, x_k(t)] = [x(t) \dots x(t - k\tau(\theta))]$$

$$[y_0(t) \dots y_k(t)] = [y(t) \dots y(t - k\tau(\theta))]$$

$$[\phi_0(t) \dots \phi_k(t)] = [\phi(t) \dots \phi(t - k\tau(\theta))]$$

$$[\mathbf{P}_0(t) \mathbf{P}_1(t)] = [\mathbf{P}(t) \mathbf{P}(t - \tau(\theta))]$$

$$\boldsymbol{\psi}(t) = [a_1(t) \ a_2(t) \ \dots \ a_k(t)]^T$$

Initialization:

For $t = 1, \dots, k$

$$\lambda(t) = 0.1, \lambda_1 = 0.55, \lambda_\infty = 0.95$$

$$r(t) = 0.1, r_1 = 0.55, r_\infty = 0.95$$

$$y(t), \phi(t), \boldsymbol{\psi}(t) = 0, \mathbf{P}(t) = \mathbf{I}$$

Main iteration loop:

For $t = k + 1, \dots, N$, where N is the number of snapshots.

$$\boldsymbol{\varphi}(t) = [\varphi_1(t), \dots, \varphi_k(t)]^T$$

$$\varphi_i(t) = x_i(t) - r^i y_i(t), i = 1, 2, \dots, k$$

$$y(t) = x(t) + \boldsymbol{\psi}^H(t) \boldsymbol{\varphi}(t)$$

$$\phi(t) = \boldsymbol{\varphi}(t) + r a_1^* \phi_1(t) + \dots + r^k a_k^* \phi_k(t)$$

$$\mathbf{P}(t) = \frac{1}{\lambda} [\mathbf{P}_1(t) - \frac{\mathbf{P}_1(t) \phi(t) \phi^H(t) \mathbf{P}_1(t)}{\lambda + \phi^H(t) \mathbf{P}_1(t) \phi(t)}]$$

$$\boldsymbol{\psi}(t+1) = \boldsymbol{\psi}(t) - \mathbf{P}(t) \phi(t) y^*(t)$$

$$\lambda(t+1) = \lambda_1 \lambda(t) + (1 - \lambda_1) \lambda_\infty$$

$$r(t+1) = r_1 r(t) + (1 - r_1) r_\infty$$

is used to estimate two source directions. Additive white Gaussian noise (AWGN) is added to the simulated sensor outputs at an SNR of 30 dB.

In the first simulation, the array acquires data from two closely-spaced sources situated at 50° and 60° direction with respect to the array axis. The proposed algorithm operates on 1000 array snapshots to obtain DOA estimates. Figure 6.4 shows the spatial spectrum of the proposed technique averaged over 500 trials in comparison with the Capon and ME methods. Since the concept of subspace is not yet defined for IIR structure, we limit our

comparison to conventional methods such as the Capon and ME methods. In Figure 6.4, it is interesting to note that the resolution of the proposed method is better than the Capon method which fails to resolve the two sources and is comparable with that of ME method. We notice that the spectrum of the proposed technique in source-free directions is better than the other two techniques. This phenomenon can be attributed to the increase in degrees of freedom obtained from the IIR structure over the FIR structure adopted by the other techniques. The degrees of freedom available for Capon and ME methods are utilized to give distinct peaks along the source directions. However, besides providing the distinct source direction peaks in the spatial spectrum, the proposed technique can suppress the spectrum floor along all other directions due to the increased degrees of freedom.

In order to study this “side-lobe” suppression performance, we simulate another example where the sources are located at 60° and 90° directions with respect to the array axis. Since the source are farther apart, the Capon method and the ME method are expected to have better response along the source-free directions. Figure 6.5 asserts this understanding, where the spatial spectrum of Capon method and the ME method have improved spatial spectrum floor. However, we notice that the proposed technique has a better flat response along the source-free directions than the other techniques.

We next subject experimentally acquired data from 3 sensors to the proposed method, the Capon method and the MUSIC algorithm. The sources are situated in the far-field region at angles 44° and 90° from the array axis. Since the angular separation between the sources is large, Figure 6.6 shows that all the techniques succeed in estimating accurate directions with as low as 50 snapshots. We observe that all the techniques have similar performance. In order to study the effect of increase in the number of snapshots, we plot the spatial spectrum for the same scenario for 100 snapshots in Figure 6.7. The comparison of the spatial spectra in Figure 6.6 and Figure 6.7 unveils that the performance of all the techniques improve with the number of snapshots. It is interesting to know that the floor of spatial spectrum for the proposed technique is better than the other two techniques which

can be credited to the increase in the degrees of freedom obtained by the IIR structure.

Figure 6.4: DOA estimation using different methods with sources from 50° and 60° .

6.3.4 Performance Analysis

The performance of the proposed DOA estimator can be analyzed on the same lines as that provided in [82] and [84]. We here obtain a closed form solutions of bias and MSE for the proposed estimator. For convenience, a first order IIR notch filter is considered. Its transfer function is given by

$$H_N(z) = \frac{1 + a^* z^{-1}}{1 + r a^* z^{-1}}, \quad (6.21)$$

where $r \in (0, 1)$ is a pole contraction factor which controls the notch bandwidth of the filter and a is the filter coefficient. Here, this first order transfer function corresponds to a 2 sensor array. For the positions of zeros along the source directions, it is necessary that $a = -\exp(-j\pi \cos(\theta)) = -\exp(-j\omega)$, with θ being the source direction. The output of the

Figure 6.5: DOA estimation using different methods with sources from 60° and 90° .

sensors (a snapshot) is given by

$$\mathbf{x}(t) = A(\theta)s(t) + v(t), \quad (6.22)$$

where θ is the direction of the signal, $A(\theta) = [1 \ a]^T$ is the steering vector of the array corresponds to a source at direction θ , $v(t)$ is an additive white Gaussian noise with variance δ_v^2 .

Using the steepest descent algorithm explained in Section III(B)(6.11), the coefficient is updated adaptively as follows:

$$\hat{a}(t+1) = \hat{a}(t) - \mu\phi(t)y^*(t), \quad (6.23)$$

where $\hat{a}(t)$ is the filter (DOA estimator) coefficient estimated at time instant t , the gradient signal $\phi(t)$ is the output of the gradient filter [78]

Figure 6.6: DOA estimation using real data, 50 snapshots are used.

$$H_S(z) \simeq z^{-1} - rz^{-1}H(z) = \frac{(1-r)z^{-1}}{1+raz^{-1}}. \quad (6.24)$$

The filter coefficient $\hat{a}(t)$ will converge to its ideal value a_0 when

$$f(\hat{a}) = E[\phi(t)y(t)] = 0. \quad (6.25)$$

Using the Taylor series expansion, $f(\hat{a})$ can be approximated in the vicinity of a_0 as follows:

$$\begin{aligned} f(\hat{a}) &\simeq f(a_0) + (\hat{a} - a_0)f'(a)|_{a=a_0} + (\hat{a} - a_0)^2 f''(a)|_{a=a_0} \\ &= f(a_0) + \delta_a f'(a)|_{a=a_0} + \delta_a^2 f''(a)|_{a=a_0}. \end{aligned} \quad (6.26)$$

Here, $\delta_a = \hat{a} - a_0$, is the estimation error of the filter coefficient. The expansion terms with

Figure 6.7: DOA estimation using real data, 100 snapshots are used.

$\{\delta_a^k \mid k \geq 3\}$ are neglected.

The bias of the DOA estimator is given by

$$E[\delta_a(\infty)] = \frac{\eta_{12}E[\delta_a^2(\infty)]}{\eta_{11}} - \frac{R_{12}}{\eta_{11}}, \quad (6.27)$$

where $E[\delta_a^2(\infty)]$ is given in (6.28), R_{12} is given in (B.33), η_{11} and η_{12} are given in (B.35) and (B.36), respectively. The detailed derivation is given in the Appendix A. The first term of (6.27) is associated with the estimated mean square error (MSE) which converges to a small value by choosing a small and proper step size. However the second term is a constant indicating that the DOA estimate obtained by the proposed method is biased.

Figure 6.8 shows the plot of bias as a function of the number of snapshots for two SNR values of 20dB and 30dB. We observe that the bias estimated over 500 trials for the proposed method follows closely with the theoretical values. The bias of ME estimator is observed to be better than the bias of DOA estimates obtained by the proposed technique.

Figure 6.8: Bias of the IIR notch estimator.

The MSE of the algorithm is given by:

$$E [\delta_a^2(\infty)] = \frac{\frac{R_{12}^2}{\eta_{11}} - \mu(\frac{\eta_{23}}{\eta_{11}} R_{12} - \eta_{24})}{\eta_{21} + \frac{\eta_{12} R_{12}}{\eta_{11}} - \mu(\eta_{22} + \frac{\eta_{12} \eta_{23}}{\eta_{11}})}, \quad (6.28)$$

where R_{12} is given in (B.33), η_{11} and η_{12} are given in (B.35) and (B.36), respectively. η_{21} , η_{22} , η_{23} and η_{24} are given in (B.42), (B.44), (B.45) and (B.46), respectively. The detailed derivation is given in the Appendix A.

Figure 6.9 shows the standard deviation for these two methods along with the Capon algorithm. It can be observed that the variance of the proposed method is better than the other two methods.

In summary, the proposed method has better noise suppression than the other conventional methods such as Capon and ME. Resolution of MUSIC is observed to be better than the proposed method. The reason being that the proposed method does not use subspace-

Figure 6.9: Standard deviation of the IIR notch estimator.

based mechanism. The order selection problem observed in other DOA estimation techniques is also observed in this method. The order mismatch issue can be solved by considering more snapshots.

Though the performance of the IIR notch method is limited, the purpose here is to introduce the IIR structure (in spatial domain) for array processing which has not been used in the literature. It is now appropriate to question whether the structure in Figure 6.3 is truly an IIR array. The next section explains the approximation made for this structure.

6.4 Implementation of IIR Array Processing Structure

With the array assumed to be situated in the farfield of directional sources, the planar wavefront is spatially sampled by the sensors. The sensor output captures the effect of signal propagation as composite time delay $\tau(\theta)$ obtained due to multiple sources in the scenario. For the case where a ULA with 3 sensors acquires signal from a single source, the

time delays with respect to the reference sensor are $\tau(\theta)$ and $2\tau(\theta)$.

The discrete-time IIR filter is characterized with poles and zeros. The positioning of zeros, or equivalently the nulls along the signal direction blocks the signal from that direction resulting in minimum array output power. In order to provide feedback, the array output has to be spatially delayed by a factor of $\tau(\theta)$. However, the output $y(t)$ is obtained by compensating for all the spatial delays and there is no means to preserve this information. Therefore, in Section 6.3, we have provided an approximate temporal delay instead of a delay corresponding to the source direction to obtain the feedback components $y_1(t)$ and $y_2(t)$, which are given in (6.16).

In order to implement an IIR array, spatial feedback has to be provided. This can be realized using the concept of subarrays. The output of preceding subarray can be interpreted as a spatially delayed output of the subarray of interest and can be used for the IIR structure. The following details the implementation of this structure.

In Section 6.2, the equivalence between the output of a ULA and an FIR filter output is discussed. Likewise, we here present the design of an IIR structure using ULA for beamforming application. This provides a prototype of the spatial IIR array.

Figure 6.10: The structure of the IIR array.

Consider a ULA with sufficiently large number of identical sensors. We divide this array into overlapping subarrays of 3 sensors each, such that the sensors $\{0, 1, 2\}$ form the first sub array, sensors $\{1, 2, 3\}$ form the second subarray and so on as shown in Figure 6.10. For

an array of n elements, $n - 2$ subarrays of 3-element ULA can be formed. For each subarray, a second order constrained IIR notch array processing structure is implemented. The output of the $(n - 2)_{th}$ subarray (the current subarray with the snapshot $[x_{n-2}(t) \ x_{n-1}(t) \ x_n(t)]^T$ being processed) at time t is given by

$$y_n(t) = x_n(t) + a_1^* x_{n-1}(t) + a_2^* x_{n-2}(t) - r a_1^* y_{n-1}(t) - r^2 a_2^* y_{n-2}(t). \quad (6.29)$$

For the first subarrays, the feedback components $y_1(t)$ and $y_2(t)$ are zeros reducing $y_3(t)$ to conventional FIR structured beamformer outputs. For the second subarrays, only the feedback component $y_2(t)$ is zero. For any $n > 4$, the output of the $(n - 2)_{th}$ subarray is obtained using spatial feedback from the previous subarrays unlike the structure adopted for DOA estimation where approximate temporal delay was used instead.

It is important to note that $y_n(t)$ is obtained using $y_{n-1}(t)$ and $y_{n-2}(t)$, likewise, $y_{n-1}(t)$ from $y_{n-2}(t)$ and $y_{n-3}(t)$, and so on. This effectively makes the impulse response of the last subarray to depend on all proceeding subarray outputs and hence establishes the IIR structure. The output of the last subarray is thus the output of the entire IIR structured array. It is apparent from (6.29) that, a second order IIR array can be implemented with a minimum of 5 sensors. From other experiments, it was noticed that an array with larger aperture is necessary for the algorithm to converge.

In order to develop an IIR beamformer using spatial feedback, consider the response of one of the 3-element subarrays given by

$$H(f, \theta) = \frac{1 + a_1^* e^{-j2\pi f \tau(\theta)} + a_2^* e^{-j2\pi f 2\tau(\theta)}}{1 + r a_1^* e^{-j2\pi f \tau(\theta)} + r^2 a_2^* e^{-j2\pi f 2\tau(\theta)}}. \quad (6.30)$$

The position of zeros in the transfer function is critical in either suppressing the signal or filtering it. If the zero corresponding to a source direction is positioned within the unit circle, the array will introduce a deep null along the direction, thus blocking the signal. On

the contrary, if the zero corresponding to a source is located outside the unit circle, the signal passes through the filter without any distortion. Therefore, a beamformer can be designed by positioning the zeros of the IIR filter such that only the desired source is beamformed which suppressing all the interferences. With $z = e^{j2\pi f\tau(\theta)}$, the zeros for the notch filter can be evaluated with

$$\begin{aligned} 1 + a_1^* z^{-1} + a_2^* z^{-2} &= (1 - c_1 z^{-1})(1 - c_2 z^{-1}) \\ &= 1 - (c_1 + c_2)z^{-1} + c_1 c_2 z^{-2} = 0. \end{aligned} \quad (6.31)$$

Since the zero in the desired look direction θ_0 should be placed outside the unit circle,

$$|1 - c_1 z^{-1}| > 0 \Rightarrow |c_1| > 1. \quad (6.32)$$

For a half-wavelength spacing ULA, we have $z = e^{j\pi \cos(\theta)}$ and hence $c_1 = r_c e^{j\pi \cos(\theta)}$, where $r_c > 1$ is a constant which ensures that the zero is situated outside the unit circle. From (6.31) and (6.32), it can be shown that $a_1^* = -(c_1 + c_2)$ and $a_2^* = c_1 c_2$, then

$$a_2^* = -c_1 a_1^* - c_1^2. \quad (6.33)$$

The weight a_1 can be adaptively estimated by following the algorithm in Table 6.2. This results in suppressing the interference and bypassing the desired signal.

We validate this structure for beamforming with an experiment where the narrowband desired signal impinges the array from the broadside of the 32-element ULA in the presence of an interference from 60° . The desired signal is a complex amplitude modulated (AM) signal, while the interference is a random signal with Gaussian distribution. Sensor noise is simulated with AWGN at an SNR of 30dB. The array is divided into overlapping subarrays with each subarray forming a ULA of 3 sensors. The real part of the desired signal and one of the sensor outputs are shown in Figure 6.11(a) and Figure 6.11(b), respectively. The

IIR structured beamformer output is shown in Figure 6.11(c). It can be observed that the desired signal is reconstructed at the array output with significant interference suppression.

The array directional response in Figure 6.12(a) shows that the IIR structure introduces a deep null along the interference direction and simultaneously maintains a flat response along the look direction. The poles and zeros of the IIR notch beamformer are shown in Figure 6.12(b). We note that one of the zeros of the system transfer function is outside the unit circle while the other zero is situated on the unit circle. While the position of the zero corresponding to the source direction is constrained, the other zero as well as the positions of poles are adaptively regulated by the remaining parameters. The phase of the zeros and poles implicitly relate to the signal direction which requires further investigation.

We observe that the constrained notch beamformer does not have any signal cancellation problem. Initial tests reveal that the method is more robust when compared to conventional arrays such as the Capon beamformer. As is evident, one of the limitations of this structure is the necessity of recursive spatial components, thus requiring a large aperture. However, the main aim of this work is to identify the possibility of extending IIR filtering to IIR array beamforming. With afore explained structure, we infer that IIR arrays are indeed realizable for practical applications. This provides the scope for further research towards the development of efficient IIR array structure for array processing which can improve the performance and cost of the system.

Figure 6.11: The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array.

Figure 6.12: The spatial response and the z-plane of the IIR array.

6.4.1 Simulation Results for Different SNR

In this section, the simulation results of the proposed method for different SNR are given. In Figure 6.13, $\text{SNR} = 0$ dB is considered. Similarly, in Figure 6.14 and Figure 6.15, we consider $\text{SNR} = 10$ dB and 20 dB.

Figure 6.13: The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. $\text{SNR} = 0$ dB.

Figure 6.14: The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. SNR = 10 dB.

Figure 6.15: The (a) desired signal, (b) input signal on sensor 1 and (c) output signal of the IIR array. $\text{SNR} = 20 \text{ dB}$.

6.5 Conclusion

This work studies the possibility of using IIR structure for array processing. A new IIR structure-based DOA estimation technique has been proposed. The performance of the proposed method is better than the conventional DOA estimation methods, such as Capon and Maximum Entropy (ME). Furthermore, an IIR array beamformer is implemented in spatial domain using subarrays. The above illustrations demonstrate the possibility of extending IIR filtering to IIR array processing, but with the following challenges:

1. Choice of optimal IIR structure for array processing.
2. Extension of subspace techniques to IIR structure to enhance the performance of the IIR structure based DOA estimation techniques.
3. Algorithm development for IIR beamformer synthesis.

Chapter 7

Extensions for IIR and Volterra Array Processing

7.1 Introduction

In Section 3.3, we propose a DOA estimation technique using approximate spatial IIR Notch structure. To release the constraints of using Notch structure, one possibility is extending the IIR Notch structure to a generalized IIR structure.

In Chapter 4, 5 and 6, we proposed three DOA estimation algorithms based on the spatial Volterra structure. Second and third order Volterra structures were considered. The models used can be considered as an FIR Volterra structure in spatial domain. IIR Volterra filter in time domain has been discussed and studied. Here we give a short review for IIR Volterra filter in time domain.

A general introduction and review about principles and applications of quadratic filters for signal processing is given in [89], IIR Volterra filtering with application to bilinear systems is discussed in [90] and [91]. Theory and applications of adaptive second order IIR Volterra filters are considered in [92]. Multisource discrimination method using IIR Volterra filtering is proposed in [93].

An interesting question is “what happens when IIR meets Volterra in spatial domain?”. In this Chapter, we explore the possibility of DOA estimation using approximate spatial IIR structure and IIR Volterra structure in spacial domain.

7.2 FIR Volterra Model

In Section 1.3, second order and third order Volterra snapshots are introduced.

The second order Volterra snapshot at time k is given by

$$\mathbf{x}_{2v}(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T, \quad (7.1)$$

where $\mathbf{x}_{2v}(k) \in \mathcal{C}^{\left(M + \frac{M(M+1)}{2}\right) \times 1}$.

Based on the Volterra snapshots, it is straightforward to propose FIR Volterra models in spatial domain. The output of the array at time k is obtained by

$$y_{2v}(k) = \mathbf{w}_{2v}^H(k) \mathbf{x}_{2v}(k), \quad (7.2)$$

where $\mathbf{w}_{2v}(k) \in \mathcal{C}^{\left(M + \frac{M(M+1)}{2}\right) \times 1}$ is the complex weights vector for second order FIR Volterra structure. The first M elements of $\mathbf{w}_{2v}(k)$ correspond to the first order term of the snapshots. The rest of the $\frac{M(M+1)}{2}$ elements are for the second order term of the snapshots. For convenience, the second order FIR Volterra model in Figure 7.1 is simplified using the following module in Figure 7.2. Similar to (7.1), the third order Volterra snapshot at time k is defined as

$$\mathbf{x}_{3v}(k) = \left[\mathbf{x}^T(k), (\mathbf{x}(k) \boxtimes \mathbf{x}(k))^T (\mathbf{x}(k) \boxtimes \mathbf{x}(k) \boxtimes \mathbf{x}(k))^T \right]^T. \quad (7.3)$$

The output of the array at time k is obtained by

$$y_{3v}(k) = \mathbf{w}_{3v}^H(k) \mathbf{x}_{3v}(k), \quad (7.4)$$

Figure 7.1: Second order FIR Volterra model in spatial domain

Figure 7.2: Second order FIR Volterra module in spatial domain

where $\mathbf{w}_{3v}(k) \in \mathcal{C}^{\left(M + \frac{M(M+1)}{2}\right) \times 1}$ is the complex weights vector for third order FIR Volterra structure.

7.3 IIR Volterra Model

7.3.1 IIR Volterra Filter

Here we give a short introduction for IIR Volterra filter. For a general case, the output of a discrete-time nonlinear system $y(k)$ can be represented as an IIR Volterra series expansion of the input signal $x(k)$ and $y(k)$,

$$\begin{aligned}
 y(k) = & \omega_0 + \sum_{l_1=0}^{\infty} \omega_1(l_1)x_{l_1}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} \sum_{l_3=0}^{\infty} \omega_3(l_1, l_2, l_3)x_{l_1}(k)x_{l_2}(k)x_{l_3}(k) + \cdots \\
 & + \sum_{l_1=0}^{\infty} g_1(l_1)y_{l_1}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} g_2(l_1, l_2)y_{l_1}(k)y_{l_2}(k) \\
 & + \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} \sum_{l_3=0}^{\infty} g_3(l_1, l_2, l_3)y_{l_1}(k)y_{l_2}(k)y_{l_3}(k) + \cdots, \tag{7.5}
 \end{aligned}$$

where ω_0 is meant as the causal impulse response of the zero-order, where $\omega_i(l_1, l_2, \dots, l_i)$ are the coefficients of the nonlinear combiner model based on the Volterra series and $x_{l_i}(k) = x(k - l_i)$ is the input signal at time $k - l_i$, for $\{i = 1, 2, \dots, \infty\}$. Similarly, $g_i(l_1, l_2, \dots, l_i)$ are the feedback coefficients of the nonlinear combiner model based on the IIR Volterra series and $y_{l_i}(k) = y(k - l_i)$ is the output signal at time $k - l_i$, for $\{i = 1, 2, \dots, \infty\}$.

In real applications, truncated Volterra model is used. For second order truncated

Volterra series, the output $y_2(k)$ is given by

$$\begin{aligned}
 y_2(k) = & \sum_{l_1=0}^M \omega_1(l_1)x_{l_1}(k) + \sum_{l_1=0}^M \sum_{l_2=0}^M \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k) \\
 & + \sum_{l_1=0}^N g_1(l_1)y_{l_1}(k) + \sum_{l_1=0}^N \sum_{l_2=0}^N g_2(l_1, l_2)y_{l_1}(k)y_{l_2}(k),
 \end{aligned} \tag{7.6}$$

where M and N are the number of time delay elements for input signal $x(k)$ and output signal $y(k)$.

For third order truncated Volterra series, the output $y_3(k)$ is given by

$$\begin{aligned}
 y_3(k) = & \sum_{l_1=0}^M \omega_1(l_1)x_{l_1}(k) \\
 & + \sum_{l_1=0}^M \sum_{l_2=0}^M \omega_2(l_1, l_2)x_{l_1}(k)x_{l_2}(k) \\
 & + \sum_{l_1=0}^M \sum_{l_2=0}^M \sum_{l_3=0}^M \omega_3(l_1, l_2, l_3)x_{l_1}(k)x_{l_2}(k)x_{l_3}(k) \\
 & + \sum_{l_1=0}^N g_1(l_1)y_{l_1}(k) \\
 & + \sum_{l_1=0}^N \sum_{l_2=0}^N g_2(l_1, l_2)y_{l_1}(k)y_{l_2}(k) \\
 & + \sum_{l_1=0}^N \sum_{l_2=0}^N \sum_{l_3=0}^N g_3(l_1, l_2, l_3)y_{l_1}(k)y_{l_2}(k)y_{l_3}(k).
 \end{aligned} \tag{7.7}$$

7.3.2 IIR Volterra Model in Spatial Domain

In Section 3.4, implementation of IIR array using spatial smoothing is discussed. In this section, we exploit some possible extensions for the method we proposed.

Consider a ULA with sufficiently large number of identical sensors. We divide this array into overlapping subarrays of 3 sensors each, such that the sensors $\{0, 1, 2\}$ form the first subarray, sensors $\{1, 2, 3\}$ form the second subarray. For an array of n elements, $n - 2$

subarrays of 3-element ULA can be formed. For each subarray, a second order IIR array processing structure is implemented. Assume that both time delay elements for input signal and output signal is 2. The output of the $(n-2)^{th}$ subarray (the current subarray with the snapshot $[x_{n-2}(k) \ x_{n-1}(k) \ x_n(k)]^T$ being processed) at time k is given by

$$\begin{aligned}
 y_n(k) = & w_1 x_n(k) + w_2 x_{n-1}(k) + w_3 x_{n-2}(k) \\
 & + w_{11} x_n(k) x_n(k) + w_{12} x_n(k) x_{n-1}(k) + w_{13} x_{n-1}(k) x_{n-2}(k) \\
 & + w_{22} x_{n-1}(k) x_{n-1}(k) + w_{23} x_{n-1}(k) x_{n-2}(k) + w_{33} x_{n-2}(k) x_{n-2}(k) \\
 & + g_1 y_{n-1}(k) + g_2 y_{n-2}(k) \\
 & + g_{11} y_{n-1}(k) y_{n-1}(k) + g_{12} y_{n-1}(k) y_{n-2}(k) + g_{22} y_{n-2}(k) y_{n-2}(k). \quad (7.8)
 \end{aligned}$$

For convenience, the second order IIR Volterra model in Figure 7.3 is simplified using the

Figure 7.3: Second order IIR Volterra model in spatial domain

following module in Figure 7.4 For the first subarrays, the feedback components $y_1(t)$ and

Figure 7.4: Second order IIR Volterra module in spatial domain

$y_2(t)$ are zeros reducing $y_3(t)$ to conventional FIR structured beamformer outputs. For the second subarrays, the feedback components $y_2(t)$ is zero. For any $n > 4$, the output of the $(n - 2)^{th}$ subarray is obtained using spatial feedback from the previous subarrays unlike the structure adopted for DOA estimation where approximate temporal delay was used instead.

It is important to note that $y_n(t)$ is obtained using $y_{n-1}(t)$ and $y_{n-2}(t)$, likewise, $y_{n-1}(t)$ from $y_{n-2}(t)$ and $y_{n-3}(t)$, and so on. This effectively makes the impulse response of the last subarray to depend on all the subarray outputs and hence establishes the IIR structure. The output of the last subarray is thus the output of the entire IIR structured array. It is apparent from (6.29) that, a second order IIR array can be implemented with a minimum of 5 sensors. From other experiments, it was noticed that a larger array than this is necessary for the algorithm to converge.

7.4 Volterra Based Joint Space and Time Array Processing

In Section 7.3 and 7.4, we introduce the FIR and IIR Volterra model in both time and spatial domain. In these models, the signals are processed either in time domain or in spatial domain. A natural question arises here is “can we further extend the Volterra model by jointly processing in both spacial and time domain?”

The technique of joint space and time processing has been widely used in wireless com-

munication systems. Space-time coding is a method employed to improve the reliability of data transmission in wireless communication systems using multiple transmit antennas. Space-time coding relies on transmitting multiple, redundant copies of a data stream to the receiver in the hope that at least some of them may survive the physical path between transmission and reception in a good enough state to allow reliable decoding. Recently, joint space-time transmitter-receiver beamforming is discussed in [94]. Performance gain in a mobile radio environment can be achieved in [95] by using joint space-time equalization and decoding.

Extending the Volterra model by jointly processing in both spatial and time domain is one of our motivations. Since there are two structures in time domain, FIR Volterra and IIR Volterra. Similarly, there are FIR Volterra and IIR Volterra structures in spatial domain. By doing joint space and time processing, four new structures are obtained in the following sections.

7.4.1 Spatial FIR Volterra and Time domain FIR Volterra

In spatial domain, the second order FIR Volterra is used to obtain the sensor output $y(k)$ at different time index k using (7.2). To apply the FIR Volterra in time domain, the delayed outputs are used. The final output $y(k)$ at time k is obtained by (7.6).

In Figure 7.5, a linear array with three sensors is considered, $\mathbf{x}(k) = [x_1(k) \ x_2(k) \ x_3(k)]^T$ is the snapshot of the array at time k , for illustration $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ are used to obtain the final output of the FIR Volterra filter in time domain. There is no constraints on the number of delayed output signals used, however, the computational complexity is increased when more delayed output signals are used. In spatial domain, we obtain $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ by using (7.2).

The final output $y(k)$ at time k is given by

$$\begin{aligned}
 y(k) = & w_1 y_{2v}(k) + w_2 y_{2v}(k-1) + w_3 y_{2v}(k-2) \\
 & + w_{11} y_{2v}(k) y_{2v}(k) + w_{12} y_{2v}(k) y_{2v}(k-1) + w_{13} y_{2v}(k) y_{2v}(k-2) \\
 & + w_{22} y_{2v}(k-1) y_{2v}(k-1) + w_{23} y_{2v}(k-1) y_{2v}(k-2) \\
 & + w_{33} y_{2v}(k-2) y_{2v}(k-2),
 \end{aligned} \tag{7.9}$$

where $\mathbf{w} = [w_1, w_2, w_3, w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}]^T$ is the forward weights of the time-domain FIR Volterra filter.

Figure 7.5: Spatial FIR Volterra and Time domain FIR Volterra

7.4.2 Spatial FIR Volterra and Time domain IIR Volterra

Similar to the structure introduced in the previous section, the second order FIR Volterra is used to obtain the sensor output $y(k)$ at different time index k using (7.2). Based on the delayed outputs, IIR Volterra filter in time domain is used to obtain the final output $y(k)$.

In Figure 7.7, a linear array with three sensors is considered, $\mathbf{x}(k) = [x_1(k) \ x_2(k) \ x_3(k)]^T$ is the snapshot of the array at time k , for illustration $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ are used to obtain the final output of the FIR Volterra filter in time domain. In spatial domain, we obtain $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ by using (7.2).

For simplicity, two delayed outputs $y(k-1)$ and $y(k-2)$ are used to obtain the final output. The final output $y(k)$ at time k is given by

$$\begin{aligned}
y(k) = & w_1 y_{2v}(k) + w_2 y_{2v}(k-1) + w_3 y_{2v}(k-2) \\
& + w_{11} y_{2v}(k) y_{2v}(k) + w_{12} y_{2v}(k) y_{2v}(k-1) + w_{13} y_{2v}(k) y_{2v}(k-2) \\
& + w_{22} y_{2v}(k-1) y_{2v}(k-1) + w_{23} y_{2v}(k-1) y_{2v}(k-2) \\
& + w_{33} y_{2v}(k-2) y_{2v}(k-2) \\
& + g_1 y(k-1) + g_2 y(k-2) \\
& + g_{11} y(k-1) y(k-1) + g_{12} y(k-1) y(k-2) \\
& + g_{22} y(k-2) y(k-2),
\end{aligned} \tag{7.10}$$

where $\mathbf{w} = [w_1, w_2, w_3, w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}]^T$ and $\mathbf{g} = [g_1, g_2, g_{11}, g_{12}, g_{22}]^T$ are the forward and backward weights of the time domain IIR Volterra filter.

7.4.3 Spatial IIR Volterra and Time domain FIR Volterra

In Figure 7.8, a linear array with five sensors is considered, $\mathbf{x}(k) = [x_1(k) \ x_2(k) \ x_3(k) \ x_4(k) \ x_5(k)]^T$ is the snapshot of the array at time k , for illustration $y_{2v}(k)$ and $y_{2v}(k-1)$ are used to obtain the final output of the FIR Volterra filter in time domain. In spatial domain, we obtain

Figure 7.6: Spatial FIR Volterra and Time domain IIR Volterra

$y_{2v}(k)$ and $y_{2v}(k-1)$ by using (7.8).

The final output $y(k)$ at time k is given by

$$\begin{aligned}
 y(k) = & w_1 y_{2v}(k) + w_2 y_{2v}(k-1) \\
 & + w_{11} y_{2v}(k) y_{2v}(k) + w_{12} y_{2v}(k) y_{2v}(k-1) \\
 & + w_{22} y_{2v}(k-1) y_{2v}(k-1),
 \end{aligned} \tag{7.11}$$

where $\mathbf{w} = [w_1, w_2, w_{11}, w_{12}, w_{22}]^T$ is the forward weights of the time domain FIR Volterra filter.

Figure 7.7: Spatial IIR Volterra and Time domain FIR Volterra

7.4.4 Spatial IIR Volterra and Time-domain IIR Volterra

In Figure 7.6, a linear array with three sensors is considered, $\mathbf{x}(k) = [x_1(k) \ x_2(k) \ x_3(k)]^T$ is the snapshot of the array at time k , for illustration, $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ are used to obtain the final output of the FIR Volterra filter in time domain. In spatial domain, we obtain $y_{2v}(k)$, $y_{2v}(k-1)$ and $y_{2v}(k-2)$ by using (7.2).

For simplicity, two delayed outputs $y(k-1)$ and $y(k-2)$ are used to obtain the final

output. The final output $y(k)$ at time k is given by

$$\begin{aligned}
y(k) = & w_1 y_{2v}(k) + w_2 y_{2v}(k-1) \\
& + w_{11} y_{2v}(k) y_{2v}(k) + w_{12} y_{2v}(k) y_{2v}(k-1) \\
& + w_{22} y_{2v}(k-1) y_{2v}(k-1) \\
& + g_1 y(k-1) + g_2 y(k-2) \\
& + g_{11} y(k-1) y(k-1) + g_{12} y(k-1) y(k-2) \\
& + g_{22} y(k-2) y(k-2),
\end{aligned} \tag{7.12}$$

where $\mathbf{w} = [w_1, w_2, w_{11}, w_{12}, w_{22}]^T$ and $\mathbf{g} = [g_1, g_2, g_{11}, g_{12}, g_{22}]^T$ are the forward and backward weights of the time domain IIR Volterra filter.

Figure 7.8: Spatial IIR Volterra and Time domain IIR Volterra

7.5 Conclusion

In this Chapter, we have proposed the possible extensions for IIR and Volterra based array processing, namely,

1. DOA estimation technique using approximate spatial IIR structure,
2. Extending the time domain IIR Volterra model to spatial domain,
3. Joint space and time processing based on the Volterra model.

Chapter 8

Conclusion and Recommendations

8.1 Conclusion

In this thesis, we extend the concept of IIR filtering to array processing using approximate spatial IIR structure. The possibility of replacing an FIR structure with an approximate IIR structure for DOA estimation is analyzed and verified. Furthermore, we propose a primitive IIR array implementation in spatial domain to perform beamforming. One of the drawbacks of the proposed beamformer is that a large uniform linear array is needed to make sure the algorithm converges.

The Volterra signal models have been applied to a variety of applications. However, the Volterra signal model has not been applied in spatial domain. To fill the gaps between Volterra filtering in time domain and processing in spatial domain, we extend the FIR and IIR Volterra model in time domain to spatial domain.

Two DOA estimation algorithms are proposed based on the Volterra model in spatial domain. The new technique basically uses additionally the higher order terms of Volterra series to produce augmented Volterra snapshots. Using the same idea, steering vector is also augmented. With the augmented snapshots and augmented steering vector, the new Volterra based Capon and MUSIC DOA estimators can be implemented. Analytical and

simulation results show that the better performance of the proposed methods are obtained when the sources are non-Gaussian or the channel fading affects are considered. The Volterra based DOA estimation algorithms with spatial smoothing are considered to deal with the correlated or coherent source signals. Analytical and simulation results show that spatial smoothing technique can be applied to the proposed Volterra based method.

At last, the signal models are further extended. The DOA estimator using IIR notch structure is extended to generalized IIR structure. The marriage between IIR and Volterra based processing are obtained by using IIR Volterra model in spatial domain. Joint space and time processing is considered, generalized models consist of an FIR or IIR Volterra structure in spatial domain followed by an FIR or IIR Volterra signal models in time domain are proposed.

8.2 Recommendations for Further Research

8.2.1 Array Processing Using Approximate Spatial IIR Structure

In this thesis, the adaptive algorithm we use is recursive steepest descent algorithm. There are many other adaptive algorithms to adjust filter coefficients of the IIR notch filters. Sign algorithm (SA) [96] is the simplest among them, but it may not perform adequately well according to application requirements. The plain gradient algorithm (PG) [97] is the most popular one, which is very simple but works reasonably well. Because its performance and complexity is well balanced, the PG has become a benchmark algorithm in both theory and practice. Other algorithms can also be used, such as the lattice algorithm, p-power algorithm, memoryless nonlinear gradient algorithm, and so on.

In this thesis, we focus on the IIR notch filter's good resolution property when using the adaptive IIR notch filtering DOA estimation method. Furthermore, the IIR notch filter also has good tracking property. It is useful when the direction of the signal is changing, such as the target is moving fast, the array is moving, or both the target and array are

moving. How well the IIR notch filtering DOA estimation method works still need to be further investigated.

A large array aperture is needed when implement an IIR array, it will increase the cost and computational complexity. Can we use a small aperture to implement an IIR array? Are there any other good properties of IIR arrays comparing with the conventional FIR arrays? We have not studied sufficiently in these topics yet. These are all challenging problems for us to explore.

8.2.2 Volterra based Joint Space and Time Processing

We propose a unified Volterra model, which is obtained by joint space and time processing. The model consists of an FIR or IIR Volterra spatial structure followed by an FIR or IIR filter in time domain. The performance of array processing based on the unified model needs to be verified. Furthermore, the computational complexity of such kind of methods are also important. In real applications, trade off is made between performance and complexity.

Another extension is considering the broadband array processing problem based on the Volterra model. The model is more complex if space, time and frequency information is considered in the same model.

The model we considered here is a fundamental model, many existing array signal processing methods, such as DOA estimation and beamforming can be easily extended to this model.

With the development of both the hardware and the software, the computational power of DSP is increased greatly recently. Many computational intensive methods attract more attentions again. Even though the unified Volterra model is complex, it may become a powerful and widely used tool to model the practical problems in real applications in the near future.

Appendix A

Mean Squared Error of DOA Estimate

The mean squared error (MSE) of estimating the DOA is defined as

$$\begin{aligned}\text{MSE}(\hat{\theta}_k) &= \text{E} \left\{ \left(\hat{\theta}_k - \theta_k \right)^2 \right\} \\ &= \text{E} \left\{ \left[\left(\hat{\theta}_k - \text{E}(\hat{\theta}_k) \right) + \left(\text{E}(\hat{\theta}_k) - \theta_k \right) \right]^2 \right\} \\ &= \text{E} \left\{ \left[\hat{\theta}_k - \text{E}(\hat{\theta}_k) \right]^2 \right\} + \left[\text{E} \left\{ \hat{\theta}_k - \theta_k \right\} \right]^2.\end{aligned}\tag{A.1}$$

Let $\bar{\theta}_k = \text{E}(\hat{\theta}_k)$ and exchange the position of the first and second term of (A.1), $\text{MSE}(\hat{\theta}_k)$ can be rewritten as

$$\begin{aligned}\text{MSE}(\hat{\theta}_k) &= \left[\text{E} \left\{ \hat{\theta}_k - \theta_k \right\} \right]^2 + \text{E} \left\{ \left[\hat{\theta}_k - \bar{\theta}_k \right]^2 \right\} \\ &= \left[\text{E} \left\{ \hat{\theta}_k - \bar{\theta}_k \right\} + (\bar{\theta}_k - \theta_k) \right]^2 + \text{E} \left\{ \left[\hat{\theta}_k - \bar{\theta}_k \right]^2 \right\}.\end{aligned}\tag{A.2}$$

Let $\Delta\bar{\theta}_k = \hat{\theta}_k - \bar{\theta}_k$ and $\Delta\theta_k = \bar{\theta}_k - \theta_k$, then

$$\text{MSE}(\hat{\theta}_k) = \left[\text{E} \left\{ \Delta\bar{\theta}_k \right\} + \Delta\theta_k \right]^2 + \text{E} \left\{ (\Delta\bar{\theta}_k)^2 \right\}.\tag{A.3}$$

Appendix B

Performance Analysis of the Proposed Method

Using the Taylor series expansion on the IIR notch filter $H_N(z)$ and the gradient filter $H_S(z)$ in the vicinity of a_0 , $H_N(z)$ and $H_S(z)$ defined in (6.21) and (6.24) respectively, can now be expressed as

$$H_N(z) \approx \frac{\partial H_N(z)}{\partial a} \Big|_{a=a_0} (\hat{a} - a_0) + \frac{\partial^2 H_N(z)}{\partial a^2} \Big|_{a=a_0} (\hat{a} - a_0)^2, \quad (\text{B.1})$$

$$H_S(z) \approx \frac{\partial H_S(z)}{\partial a} \Big|_{a=a_0} (\hat{a} - a_0) + \frac{\partial^2 H_S(z)}{\partial a^2} \Big|_{a=a_0} (\hat{a} - a_0)^2, \quad (\text{B.2})$$

where the first order and second order gradient of $H_N(z)$ and $H_S(z)$ are given by

$$\frac{\partial H_N(z)}{\partial a} = \frac{(1-r)z^{-1}}{(1+raz^{-1})^2}, \quad (\text{B.3})$$

$$\frac{\partial^2 H_N(z)}{\partial a^2} = \frac{r(1-r)z^{-2}}{(1+raz^{-1})^3}, \quad (\text{B.4})$$

$$\frac{\partial H_S(z)}{\partial a} = z^{-1} - \frac{r(1-r)z^{-2}}{(1+raz^{-1})^2}, \quad (\text{B.5})$$

$$\frac{\partial^2 H_S(z)}{\partial a^2} = \frac{r^2(1-r)z^{-3}}{(1+raz^{-1})^3}. \quad (\text{B.6})$$

If the expansion terms higher than the second order of $H_N(z)$ are neglected, we have

$$\begin{aligned} H_N(z) &\approx \frac{(1-r)z^{-1}}{(1+raz^{-1})^2}(\hat{a} - a_0) \\ &\quad + \frac{r(1-r)z^{-2}}{(1+raz^{-1})^3}(\hat{a} - a_0)^2, \end{aligned} \quad (\text{B.7})$$

$$\begin{aligned} H_N(e^{j\omega_0}) &\approx \frac{(1-r)e^{-j\omega_0}}{(1+rae^{-j\omega_0})^2}(\hat{a} - a_0) \\ &\quad + \frac{r(1-r)e^{-j2\omega_0}}{(1+rae^{-j\omega_0})^3}(\hat{a} - a_0)^2 \\ &= H_{11}e^{j\omega_{11}}\delta_a + H_{12}e^{j\omega_{12}}\delta_a^2, \end{aligned} \quad (\text{B.8})$$

where H_{11} , ω_{11} , H_{12} and ω_{12} are defined as

$$H_{11} = \left| \frac{(1-r)e^{-j\omega_0}}{(1+ra_0e^{-j\omega_0})^2} \right|, \quad (\text{B.9})$$

$$\omega_{11} = \arg \left(\frac{(1-r)e^{-j\omega_0}}{(1+ra_0e^{-j\omega_0})^2} \right), \quad (\text{B.10})$$

$$H_{12} = \left| \frac{r(1-r)e^{-j2\omega_0}}{(1+ra_0e^{-j\omega_0})^3} \right|, \quad (\text{B.11})$$

$$\omega_{12} = \arg \left(\frac{r(1-r)\exp^{-j2\omega_0}}{(1+ra_0e^{-j\omega_0})^3} \right). \quad (\text{B.12})$$

If the expansion terms higher than second order of $H_S(z)$ are neglected, then

$$\begin{aligned} H_S(z) &\approx z^{-1} - \frac{r(1-r)z^{-2}}{(1+raz^{-1})^2}(\hat{a} - a_0) \\ &\quad + \frac{r^2(1-r)z^{-3}}{(1+raz^{-1})^3}(\hat{a} - a_0)^2, \end{aligned} \quad (\text{B.13})$$

$$\begin{aligned} H_S(e^{j\omega_0}) &\approx e^{-j\omega_0} - \frac{r(1-r)e^{-j2\omega_0}}{(1+rae^{-j\omega_0})^2}(\hat{a} - a_0) \\ &\quad + \frac{r^2(1-r)e^{-j3\omega_0}}{(1+rae^{-j\omega_0})^3}(\hat{a} - a_0)^2 \\ &= e^{-j\omega_0} - H_{21}e^{j\omega_{21}}\delta_a + H_{22}e^{j\omega_{22}}\delta_a^2, \end{aligned} \quad (\text{B.14})$$

where H_{21} , ω_{21} , H_{22} and ω_{22} are defined as

$$H_{21} = \left| \frac{r(1-r)e^{-j2\omega_0}}{(1+ra_0e^{-j2\omega_0})^2} \right|, \quad (\text{B.15})$$

$$\omega_{21} = \arg \left(\frac{r(1-r)e^{-j2\omega_0}}{(1+rae^{-j\omega_0})^2} \right), \quad (\text{B.16})$$

$$H_{22} = \left| \frac{r^2(1-r)e^{-j3\omega_0}}{(1+ra_0e^{-j\omega_0})^3} \right|, \quad (\text{B.17})$$

$$\omega_{22} = \arg \left(\frac{r^2(1-r)e^{-j3\omega_0}}{(1+rae^{-j\omega_0})^3} \right). \quad (\text{B.18})$$

As a result, the output signal $y(t)$ of the notch filter $H_N(z)$ with input signal $\mathbf{x}(t)$ defined in (6.22) is given by

$$\begin{aligned} y(t) &= H_{11}\delta_a(t)A(\omega_0 - \omega_{11})s(t) \\ &\quad + H_{12}\delta_a^2(t)A(\omega_0 - \omega_{12})s(t) + v_1(t), \end{aligned} \quad (\text{B.19})$$

where $v_1(t)$ is the output of $v(t)$ defined in (6.22) through the notch filter $H_N(z)$. The

variance of noise $v_1(t)$ of the notch filter is calculated by

$$\begin{aligned} E[v_1^2(t)] &= \frac{\delta_v^2}{2\pi j} \oint_C H_N(z) H_N(z^{-1}) z^{-1} dz \\ &= \frac{\delta_v^2}{2\pi j} \oint_C \frac{z + a_0}{z + ra_0} \frac{1 + a_0 z}{1 + ra_0 z} \frac{1}{z} dz, \end{aligned} \quad (\text{B.20})$$

where δ_v^2 is the variance of noise $v(t)$ which is defined in (6.22). From (B.20), we can see that the three poles of $E[v_1^2(t)]$ are $z_0 = 0$, $z_1 = -ra_0$ and $z_2 = -\frac{1}{ra_0}$.

Applying the residue theorem to the above equation, one has

$$\text{Res}(z_0) = \frac{1}{r}, \quad (\text{B.21})$$

$$\text{Res}(z_1) = \frac{ra_0^2 - 1}{ra_0(1 - r^2a_0^2)}, \quad (\text{B.22})$$

$$\text{Res}(z_2) = \frac{ra_0^2 - 1}{ra_0(1 + r^2a_0^2)}, \quad (\text{B.23})$$

where $\text{Res}(z_0)$, $\text{Res}(z_1)$ and $\text{Res}(z_2)$ are the residues of the three poles. We then have

$$\begin{aligned} \delta_{v1}^2 &= \delta_v^2 \{ \text{Res}(0) + \text{Res}(z_1) + \text{Res}(z_2) \} \\ &= \frac{\delta_v^2}{r} + \frac{2(ra_0^2 - 1)}{ra_0(1 - r^2a_0^2)(1 + r^2a_0^2)} \delta_v^2. \end{aligned} \quad (\text{B.24})$$

The output signal of $\mathbf{x}(t)$ through the gradient filter $H_S(z)$ at steady state is given by

$$\begin{aligned} \varphi(t) &= s(t) - H_{21}\delta_a(t)A(\omega_0 - \omega_{21})s(t) \\ &\quad + H_{22}\delta_a^2(t)A(\omega_0 - \omega_{22})s(t) + v_2(t), \end{aligned} \quad (\text{B.25})$$

where $v_2(t)$ is the output of $v(t)$ defined in (6.22) through the gradient filter $H_S(z)$. The

variance of $v_2(t)$ can be calculated by the method used in (B.20)

$$\begin{aligned} E[v_2^2(t)] &= \frac{\delta_v^2}{2\pi j} \oint_C H_S(z) H_S(z^{-1}) z^{-1} dz \\ &= \frac{\delta_v^2}{2\pi j} \oint_C \frac{(1-r)^2}{(z+ra_0)(1+ra_0z)} dz, \end{aligned} \quad (\text{B.26})$$

where $z_1 = -ra_0$, $z_2 = -\frac{1}{ra_0}$. Again applying the theory of residues to the above equation, one has

$$Res(z_1) = \frac{1-r^2}{1-r^2a_0^2}, \quad (\text{B.27})$$

$$Res(z_2) = \frac{-ra_0(1-r^2)}{1-r^2a_0^2}, \quad (\text{B.28})$$

then

$$\delta_{v2}^2 = \delta_v^2 \{Res(z_1) + Res(z_2)\} = \frac{(1-r^2)}{1+ra_0}. \quad (\text{B.29})$$

It is seen that the variance δ_{v2}^2 will be very small as long as the poles of the notch filter are varying close to the unit circle (r is close to unit), implying that the additive noise at the filter input passes the notch filter with little change in energy.

The correlation between $v_1(t)$ and $v_2(t)$ can be expressed by

$$\begin{aligned} E[v_1(t)v_2(t)] &= \frac{\delta_v^2}{2\pi j} \oint_C H_N(z) H_S(z^{-1}) z^{-1} dz \\ &= \frac{\delta_v^2}{2\pi j} \oint_C \frac{(z+a_0)}{(z+ra_0)} \frac{(1-r)}{(1+ra_0z)} dz. \end{aligned} \quad (\text{B.30})$$

Similarly, using the theory of residues, the new residues are

$$Res(z_1) = \frac{a_0(1-r)^2}{1-r^2a_0^2}, \quad (\text{B.31})$$

$$Res(z_2) = \frac{(1-r)(1-ra_0^2)}{1-r^2a_0^2}, \quad (\text{B.32})$$

which leads to

$$\begin{aligned}
 R_{12} &= E[v_1(t)v_2(t)] \\
 &= \delta_v^2 \{ \text{Res}(z_1) + \text{Res}(z_2) \} \\
 &= \frac{[1 + a_0(1-r) - ra_0^2](1-r)}{1 - r^2a_0^2} \delta_v^2.
 \end{aligned} \tag{B.33}$$

B.1 Bias Analysis of the Proposed DOA Estimator

We first define $\delta_a = \hat{a} - a_0$ as the estimation error of the filter coefficient. The instantaneous value can be estimated by (23). Taking expectation on both side of (23), we get

$$\begin{aligned}
 E[\delta_a(t+1)] &= E[\delta_a(t)] - \mu E[y(t)\varphi(t)] \\
 &\simeq \left\{ 1 - \mu H_{11}A(\omega_0 - \omega_{11})s^2(t) \right\} E[\delta_a(t)] \\
 &\quad + \mu \left\{ [H_{11}H_{21}A(2\omega_0 - \omega_{11} - \omega_{21}) \right. \\
 &\quad \left. - H_{12}A(\omega_0 - \omega_{12})]s^2(t) \right\} E[\delta_a^2(t)] - \mu R_{12} \\
 &= (1 - \mu\eta_{11})E[\delta_a(t)] + \mu\eta_{12}E[\delta_a^2(t)] \\
 &\quad - \mu R_{12},
 \end{aligned} \tag{B.34}$$

where $y(t)$ and $\varphi(t)$ are given in (B.19) and (B.25) and

$$\eta_{11} = H_{11}A(\omega_0 - \omega_{11})s^2(t), \tag{B.35}$$

$$\begin{aligned}
 \eta_{12} &= [H_{11}H_{21}A(2\omega_0 - \omega_{11} - \omega_{21}) \\
 &\quad - H_{12}A(\omega_0 - \omega_{12})]s^2(t),
 \end{aligned} \tag{B.36}$$

where H_{11} is given in (B.9), H_{12} and H_{21} are given in (B.11) and (B.15), respectively. Terms of $\{\delta_a^k(t) \mid k \geq 3\}$ are ignored for analytical simplicity as they are very small in comparison

with $E[\delta_a(t)]$ and $E[\delta_a^2(t)]$. At the steady state, using

$$E[\delta_a(t+1)]|_{t \rightarrow \infty} = E[\delta_a(t)]|_{t \rightarrow \infty} = E[\delta_a(\infty)], \quad (\text{B.37})$$

and

$$E[\delta_a^2(t)]|_{t \rightarrow \infty} = E[\delta_a^2(\infty)], \quad (\text{B.38})$$

we have

$$E[\delta_a(\infty)] = \frac{\eta_{12}E[\delta_a^2(\infty)]}{\eta_{11}} - \frac{R_{12}}{\eta_{11}}, \quad (\text{B.39})$$

where R_{12} is given in (B.33), η_{11} and η_{12} are given in (B.35) and (B.36), respectively. The expression for $E[\delta_a^2(\infty)]$ is derived in (B.49) in the following section.

B.2 MSE Analysis of the Proposed DOA Estimator

The MSE is defined as

$$\begin{aligned} E[\delta_a^2(t+1)] &= E[(\delta_a(t) - \mu y(t)s(t))^2] \\ &= E[\delta_a^2(t)] - 2\mu E[\delta_a(t)y(t)s(t)] \\ &\quad + \mu^2 E[y^2(t)s^2(t)] \\ &= E[\delta_a^2(t)] - 2\mu I_1 + \mu^2 I_2, \end{aligned} \quad (\text{B.40})$$

where μ is the step size, and I_1 and I_2 are derived as follows:

$$\begin{aligned} I_1 &\simeq H_{11}s^2(t)A(\omega_0 - \omega_{11})E[\delta_a^2(t)] + R_{12}E[\delta_a(t)] \\ &= \eta_{21}E[\delta_a^2(t)] + R_{12}E[\delta_a(t)], \end{aligned} \quad (\text{B.41})$$

where

$$\eta_{21} = H_{11}s^2(t)A(\omega_0 - \omega_{11}), \quad (\text{B.42})$$

$$\begin{aligned} I_2 &\simeq s^2(t)\delta_{v_1}^2 + \delta_{v_1}^2\delta_{v_2}^2 + 2R_{12}^2 \\ &\quad + 4s^2(t)R_{12}H_{11}A(\omega_0 - \omega_{11})E[\delta_a(t)] \\ &\quad - 2s^2(t)\delta_{v_1}^2H_{21}A(\omega_0 - \omega_{21})E[\delta_a(t)] \\ &\quad + s^2(t)\delta_{v_1}^2H_{21}^2A(2\omega_0 - 2\omega_{21})E[\delta_a^2(t)] \\ &\quad + 2s^2(t)\delta_{v_1}^2H_{22}A(\omega_0 - \omega_{22})E[\delta_a^2(t)] \\ &\quad - 4s^2(t)R_{12}H_{11}H_{21}A(2\omega_0 - \omega_{11} - \omega_{21})E[\delta_a^2(t)] \\ &\quad + 4s^2(t)R_{12}H_{12}A(\omega_0 - \omega_{12})E[\delta_a^2(t)] \\ &\quad + s^4(t)H_{11}^2A(2\omega_0 - 2\omega_{11})E[\delta_a^2(t)] \\ &\quad + s^2(t)\delta_{v_2}^2H_{11}^2A(2\omega_0 - 2\omega_{11})E[\delta_a^2(t)] \\ &= \eta_{22}E[\delta_a^2(t)] + \eta_{23}E[\delta_a(t)] + \eta_{24}, \end{aligned} \quad (\text{B.43})$$

where

$$\begin{aligned} \eta_{22} &= s^2(t)\delta_{v_1}^2H_{21}^2A(2\omega_0 - 2\omega_{21}) + 2s^2(t)\delta_{v_1}^2H_{22}A(\omega_0 - \omega_{22}) \\ &\quad - 4s^2(t)R_{12}H_{11}H_{21}A(2\omega_0 - \omega_{11} - \omega_{21}) \\ &\quad + 4s^2(t)R_{12}H_{12}A(\omega_0 - \omega_{12}) + s^4(t)H_{11}^2A(2\omega_0 - 2\omega_{11}) \\ &\quad + s^2(t)\delta_{v_2}^2H_{11}^2A(2\omega_0 - 2\omega_{11}), \end{aligned} \quad (\text{B.44})$$

$$\begin{aligned} \eta_{23} &= 4s^2(t)R_{12}H_{11}A(\omega_0 - \omega_{11}) \\ &\quad - 2s^2(t)\delta_{v_1}^2H_{21}A(\omega_0 - \omega_{21}), \end{aligned} \quad (\text{B.45})$$

$$\eta_{24} = s^2(t)\delta_{v_1}^2 + \delta_{v_1}^2\delta_{v_2}^2 + 2R_{12}^2. \quad (\text{B.46})$$

The terms of $\{\delta_a^k(t) \mid k \geq 3\}$ are all ignored for analytical simplicity. The mean square error is given by

$$\begin{aligned} E[\delta_a^2(t+1)] &= (1 - \mu\eta_{21} + \mu^2\eta_{22})E(\delta_a^2(t)) \\ &\quad - (\mu R_{12} - \mu^2\eta_{23})E(\delta_a(t)) + \mu^2\eta_{24}, \end{aligned} \quad (\text{B.47})$$

where η_{21} is given in (B.42), R_{12} is given in (B.33), η_{22} , η_{23} and η_{24} are given in (B.44), (B.45) and (B.46), respectively.

At steady state,

$$\begin{aligned} (\eta_{21} - \eta_{22})E[\delta_a^2(\infty)] &= -(R_{12} - \mu\eta_{23})E[\delta_a(\infty)] \\ &\quad + \mu\eta_{24}. \end{aligned} \quad (\text{B.48})$$

Substitute $E[\delta_a(\infty)]$ into the above equation, we get the estimation MSE of the algorithm as follows:

$$E[\delta_a^2(\infty)] = \frac{\frac{R_{12}^2}{\eta_{11}} - \mu(\frac{\eta_{23}}{\eta_{11}}R_{12} - \eta_{24})}{\eta_{21} + \frac{\eta_{12}R_{12}}{\eta_{11}} - \mu(\eta_{22} + \frac{\eta_{12}\eta_{23}}{\eta_{11}})}, \quad (\text{B.49})$$

where R_{12} is given in (B.33), η_{11} and η_{12} are given in (B.35) and (B.36), respectively. η_{21} , η_{22} , η_{23} and η_{24} are given in (B.42), (B.44), (B.45) and (B.46), respectively. Substitute $E[\delta_a^2(\infty)]$ back to (B.39), the bias can be finally evaluated.

Appendix C

Dimension of the Signal Subspace

N_s

The dimension of the signal subspace N_s is equivalent to the Rank of $\mathbf{R}_v$ for the noiseless case. In order to analyze the Rank of $\mathbf{R}_v$, let us look at some of the properties of Rank:

1. For a matrix $\mathbf{B} \in \mathbb{C}^{m \times n}$, $\text{Rank}(\mathbf{B}) \leq \min(m, n)$.
2. For two matrices $\mathbf{B} \in \mathbb{C}^{m \times n}$ and $\mathbf{C} \in \mathbb{C}^{n \times t}$, $\text{Rank}(\mathbf{BC}) \leq \min(\text{Rank}(\mathbf{B}), \text{Rank}(\mathbf{C}))$.
3. For two matrices $\mathbf{B} \in \mathbb{C}^{m \times n}$ and $\mathbf{D} \in \mathbb{C}^{t \times n}$, $\text{Rank}\left(\begin{bmatrix} \mathbf{B} \\ \mathbf{D} \end{bmatrix}\right) \leq \min(m + t, n)$.

For the noiseless case, the Volterra snapshots $\mathbf{x}_v$ is given by

$$\mathbf{x}_v = \begin{bmatrix} \mathbf{A}(\theta)\mathbf{S} \\ (\mathbf{A}(\theta) \otimes \mathbf{A}^*(\theta)) \cdot (\mathbf{S} \odot \mathbf{S}^*) \end{bmatrix}, \quad (\text{C.1})$$

where $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \dots, \mathbf{s}(k), \dots, \mathbf{s}(K)]$ and $\odot$ denotes Khatri-Rao product, which is a column-wise Kronecker product.

The estimated noiseless covariance matrix is obtained by

$$\mathbf{R}_v = \frac{1}{K} (\mathbf{X}_v \mathbf{X}_v^H). \quad (\text{C.2})$$

Based on the properties (1) and (2) of Rank, $\text{Rank}(\mathbf{A}(\theta)) \leq L$ and $\text{Rank}(\mathbf{S}) \leq L$, then $\text{Rank}(\mathbf{A}(\theta)\mathbf{S}) \leq L$.

The Kronecker product can be rewritten as

$$\mathbf{s}(k) \otimes \mathbf{s}^*(k) = \text{vec}(\mathbf{s}(k)\mathbf{s}^*(k)^T), \quad (\text{C.3})$$

where $\text{vec}(X)$ denotes the vectorization of the matrix $\mathbf{X}$ formed by stacking the columns of $\mathbf{X}$ into a single column vector.

Since

$$\begin{aligned} \mathbf{s}(k)\mathbf{s}^*(k)^T &= (\mathbf{s}(k) \cdot \mathbf{s}^*(k)^T)^T \\ &= \begin{bmatrix} s_1(k)s_1^*(k) & s_1(k)s_2^*(k) & \cdots & s_1(k)s_L^*(k) \\ s_2(k)s_1^*(k) & s_2(k)s_2^*(k) & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_L(k)s_1^*(k) & \cdots & \cdots & s_L(k)s_L^*(k) \end{bmatrix}, \end{aligned} \quad (\text{C.4})$$

then the rank of $\mathbf{S} \odot \mathbf{S}^*$ is

$$\text{Rank}(\mathbf{S} \odot \mathbf{S}^*) \leq L^2. \quad (\text{C.5})$$

and

$$\text{Rank}(\mathbf{A}(\theta) \otimes \mathbf{A}^*(\theta)) \leq L^2, \quad (\text{C.6})$$

and

$$\text{Rank}((\mathbf{A}(\theta) \otimes \mathbf{A}^*(\theta)) \cdot (\mathbf{S} \odot \mathbf{S}^*)) \leq L^2. \quad (\text{C.7})$$

Table C.1: The dimension of the signal subspace and the number of noise eigenvectors.

M sensors	L sources	Dimension of $\mathbf{x}_v$	N_s	N_e noise eigenvectors
M=2	L=1	5×1	2	3
M=3	L=1	9×1	2	7
	L=2	9×1	6	3
M=4	L=1	14×1	2	12
	L=2	14×1	6	8
	L=3	14×1	12	2

Based on the above property (3), we obtain

$$\text{Rank} \begin{bmatrix} \mathbf{A}(\theta)\mathbf{S} \\ (\mathbf{A}(\theta) \otimes \mathbf{A}^*(\theta)) \cdot (\mathbf{S} \odot \mathbf{S}^*) \end{bmatrix} \leq L + L^2. \quad (\text{C.8})$$

The dimension of the signal subspace

$$N_s = L + L^2. \quad (\text{C.9})$$

The number of noise eigenvectors N_e used to perform DOA estimation is given by

$$N_e = N - N_s = \frac{M(M+3)}{2} - L(L+1). \quad (\text{C.10})$$

Table C.1 shows the dimension of the signal subspace N_s and the number of noise eigenvectors N_e used to perform DOA estimation, for different number of sensors M and different number of sources L .

Author's Publications

- [1] Fuxi Wen, Boon Poh Ng and Vinod Veera Reddy, "Extending the Concept of IIR Filtering to Array Processing Using Approximate Spatial IIR Structure ", *Multidimensional Systems and Signal Processing*, vol. 24, pp. 157–179, 2013.
- [2] Fuxi Wen and Boon Poh Ng, "A New DOA Estimation Approach Using Volterra Signal Model", submitted to *Multidimensional Systems and Signal Processing*, minor revision.
- [3] Fuxi Wen, "Numerical Study of Likelihood Ratio Test Based Single Change Point Estimation", *Electronics Letters*, vol. 48, no. 24, pp. 1532–1533, 2012.
- [4] Nijasure Yogesh, Wee Peng Tay, Erry Gunawan, Fuxi Wen, Zhang Yang, Yong Liang Guan and Ai Ping Chua, "An Impulse Radio Ultra Wideband System for Contactless Non-invasive Respiratory Monitoring", *IEEE Transactions on Biomedical Engineering*, Jan 2013, DOI: 10.1109/TBME.2012.2237401.
- [5] Fuxi Wen and Wee Peng Tay, "Localization for Mixed Near-Field and Far-Field Sources Using Data Supported Optimization ", Proceedings of the 15th International Conference on Information Fusion (Fusion 2012), Singapore, 9 - 12 July, 2012.
- [6] Fuxi Wen and Wee Peng Tay, "Tensor Decomposition Based R-Dimensional Matrix Pencil Method ", Proceedings of the 15th International Conference on Information Fusion (Fusion 2012), Singapore, 9 - 12 July, 2012.

Bibliography

- [1] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing*. John Wiley & Sons, 2002.
- [2] L. C. Godara, “Application of antenna arrays to mobile communications. II. beam-forming and direction-of-arrival considerations,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, 1997.
- [3] C. Balanis, *Introduction to Smart Antennas*. Morgan and Claypool, 2007.
- [4] H.-T. Liu, S. Gao, and T.-H. Loh, “Electrically small and low cost smart antenna for wireless communication,” *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 3, pp. 1540–1549, 2012.
- [5] M. A. Syed and V. J. Mathews, “Lattice algorithms for recursive least squares adaptive second-order volterra filtering,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 3, pp. 202–214, 1994.
- [6] J. Lee and V. J. Mathews, “A fast recursive least squares adaptive second order Volterra filter and its performance analysis,” *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1087–1102, 1993.
- [7] M. A. Syed and V. J. Mathews, “QR-decomposition based algorithms for adaptive Volterra filtering,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 6, pp. 372–382, 1993.
- [8] V. J. Mathews, “Adaptive Volterra filters using orthogonal structures,” *IEEE Signal Processing Letters*, vol. 3, no. 12, pp. 307–309, 1996.
- [9] H. Krim and M. Viberg, “Two decades of array signal processing research: the parametric approach,” *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.
- [10] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [11] A. Barabell, “Improving the resolution performance of eigenstructure-based direction-finding algorithms,” in *Proc. IEEE Int Acoustics, Speech, and Signal Processing Conf. ICASSP '83*, vol. 8, 1983, pp. 336–339.

- [12] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1110–1121, 1991.
- [13] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [14] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–824, 1990.
- [15] Q. T. Zhang, "A statistical resolution theory of the beamformer-based spatial spectrum for determining the directions of signals in white noise," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 1867–1873, 1995.
- [16] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Prentice Hall, 1993.
- [17] J. P. Burg, "Maximum entropy spectral analysis," in *37th Meeting of Society of Exploration Geophysicists*, Oklahoma City, OK, 1967.
- [18] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.
- [19] A. Hassanien, S. Shahbazpanahi, and A. B. Gershman, "A generalized Capon estimator for localization of multiple spread sources," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 280–283, 2004.
- [20] J. Benesty, J. Chen, and Y. Huang, "A generalized MVDR spectrum," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 827–830, 2005.
- [21] M. Pesavento, A. B. Gershman, and M. Haardt, "Unitary root-MUSIC with a real-valued eigendecomposition: a theoretical and experimental performance study," *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1306–1314, 2000.
- [22] B. Porat and B. Friedlander, "Direction finding algorithms based on high-order statistics," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 2016–2024, 1991.
- [23] P. Chevalier, L. Albera, A. Ferreol, and P. Comon, "On the virtual array concept for higher order array processing," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1254–1271, 2005.
- [24] P. Chevalier, A. Ferreol, and L. Albera, "High-resolution direction finding from higher order statistics: The 2q-MUSIC algorithm," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2986–2997, 2006.
- [25] P. Chevalier, A. Ferreol, L. Albera, and G. Birot, "Higher order direction finding from arrays with diversely polarized antennas: The PD-2q-MUSIC algorithms," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5337–5350, 2007.

- [26] W. K. Ma, T. H. Hsieh, and C. Y. Chi, "DOA estimation of quasi-stationary signals with less sensors than sources and unknown spatial noise covariance: A khatri-rao subspace approach," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2168–2180, 2010.
- [27] M. Haardt, "Structured least squares to improve the performance of ESPRIT-type algorithms," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 792–799, 1997.
- [28] A. L. Swindlehurst, B. Ottersten, R. Roy, and T. Kailath, "Multiple invariance ESPRIT," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 867–881, 1992.
- [29] G. Xu, S. D. Silverstein, R. H. Roy, and T. Kailath, "Beamspace ESPRIT," *IEEE Transactions on Signal Processing*, vol. 42, no. 2, pp. 349–356, 1994.
- [30] M. Haardt and J. A. Nossek, "Unitary ESPRIT: how to obtain increased estimation accuracy with a reduced computational burden," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1232–1242, 1995.
- [31] M. Haardt, M. D. Zoltowski, C. P. Mathews, and J. Nossek, "2D unitary ESPRIT for efficient 2D parameter estimation," in *Proc. Int Acoustics, Speech, and Signal Processing ICASSP-95. Conf.*, vol. 3, 1995, pp. 2096–2099.
- [32] K. T. Wong and M. D. Zoltowski, "Closed-form multi-dimensional multi-invariance ESPRIT," in *Proc. IEEE Int Acoustics, Speech, and Signal Processing ICASSP-97. Conf.*, vol. 5, 1997, pp. 3489–3492.
- [33] M. D. Zoltowski, M. Haardt, and C. P. Mathews, "Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT," *IEEE Transactions on Signal Processing*, vol. 44, pp. 316–328, 1996.
- [34] F. Roemer and M. Haardt, "Tensor-structure structured least squares (TS-SLS) to improve the performance of multi-dimensional ESPRIT-type algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP 2007*, vol. 2, 2007.
- [35] —, "Multidimensional unitary tensor-ESPRIT for non-circular sources," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP 2009*, 2009, pp. 3577–3580.
- [36] A. Thakre, M. Haardt, and K. Giridhar, "Single snapshot R-D unitary tensor-ESPRIT using an augmentation of the tensor order," in *Proc. 3rd IEEE Int Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) Workshop*, 2009, pp. 81–84.
- [37] A. Thakre, M. Haardt, F. Roemer, and K. Giridhar, "Tensor-based spatial smoothing (TB-SS) using multiple snapshots," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2715–2728, 2010.
- [38] F. Roemer, H. Becker, and M. Haardt, "Analytical performance assessment for multi-dimensional tensor-ESPRIT-type parameter estimation algorithms," in *Proc. IEEE Int Acoustics Speech and Signal Processing (ICASSP) Conf.*, 2010, pp. 2598–2601.

- [39] T. K. Sarkar and O. Pereira, "Using the matrix pencil method to estimate the parameters of a sum of complex exponentials," *IEEE Antennas and Propagation Magazine*, vol. 37, no. 1, pp. 48–55, 1995.
- [40] N. Yilmazer, S. Ari, and T. K. Sarkar, "Multiple snapshot direct data domain approach and ESPRIT method for direction of arrival estimation," *Digit. Signal Process.*, vol. 18, pp. 561–567, July 2008.
- [41] J. E. F. del Ro and M. F. Ctedra-Prez, "A comparison between matrix pencil and root-MUSIC for direction-of-arrival estimation making use of uniform linear arrays," *Digital Signal Processing*, vol. 7, pp. 153–162, 1997.
- [42] Q. Cheng, "Optimally weighted matrix pencil approach for 2-D harmonic retrieval," in *Proc. TENCON 2003. Conf. Convergent Technologies for Asia-Pacific Region*, vol. 1, 2003, pp. 109–112.
- [43] N. Yilmazer, J. Koh, and T. K. Sarkar, "Utilization of a unitary transform for efficient computation in the matrix pencil method to find the direction of arrival," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 1, pp. 175–181, 2006.
- [44] M. F. Khan and M. Tufail, "Beamspace matrix pencil method for direction of arrival estimation," *IEICE Electronics Express*, vol. 6, no. 16, pp. 1168–1173, 2009.
- [45] Z. Gaoyi and T. Bin, "Decoupled estimation of 2D DOA for coherently distributed sources using 3D matrix pencil method," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 151:1–151:7, January 2008.
- [46] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 10, pp. 1553–1560, 1988.
- [47] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 477–489, 1988.
- [48] M. I. Miller and D. R. Fuhrmann, "Maximum-likelihood narrow-band direction finding and the EM algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 9, pp. 1560–1577, 1990.
- [49] Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1081–1089, 1986.
- [50] P. Stoica, Z. Wang, and J. Li, "Robust Capon beamforming," *IEEE Signal Processing Letters*, vol. 10, no. 6, pp. 172–175, 2003.
- [51] J. Li, P. Stoica, and Z. Wang, "On robust Capon beamforming and diagonal loading," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1702–1715, 2003.

- [52] ———, “Doubly constrained robust Capon beamformer,” *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2407–2423, 2004.
- [53] R. G. Lorenz and S. P. Boyd, “Robust minimum variance beamforming,” *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1684–1696, 2005.
- [54] A. Beck and Y. C. Eldar, “Doubly constrained robust Capon beamformer with ellipsoidal uncertainty sets,” *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 753–758, 2007.
- [55] R. Abrahamsson, A. Jacobsson, and P. Stoica, “A Capon-like spatial spectrum estimator for correlated sources,” in *12th European Signal Processing Conference : EUSIPCO 2004*, 2004.
- [56] C. D. Richmond, “Capon algorithm mean-squared error threshold SNR prediction and probability of resolution,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2748–2764, 2005.
- [57] J. X. Wu, T. Wang, Z. Y. Suo, and Z. Bao, “DOA estimation for ULA by spectral Capon rooting method,” *Electronics Letters*, vol. 45, no. 1, pp. 84–85, 2009.
- [58] P. Stoica, P. Hndel, and T. Sderstrm, “Study of Capon method for array signal processing,” *Circuits, Systems, and Signal Processing*, vol. 14, pp. 749–770, 1995, 10.1007/BF01204683.
- [59] C. Vaidyanathan and K. M. Buckley, “Performance analysis of the MVDR spatial spectrum estimator,” *IEEE Transactions on Signal Processing*, vol. 43, no. 6, pp. 1427–1437, 1995.
- [60] L. Ahlin, J. Zander, and S. Ben Slimane, *Principles of Wireless Communications*. Studentlitteratur, 2006.
- [61] T. Rappaport, *Wireless Communications: Principles and practice*, 2nd ed. Prentice Hall, 2002.
- [62] P. Stoica and N. Arye, “MUSIC, maximum likelihood, and Cramer-Rao bound,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, 1989.
- [63] X.-L. Xu and K. M. Buckley, “Bias analysis of the MUSIC location estimator,” *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2559–2569, 1992.
- [64] M. Kaveh and A. Barabell, “The statistical performance of the MUSIC and the minimum-norm algorithms in resolving plane waves in noise,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 2, pp. 331–341, 1986.
- [65] Q. T. Zhang, “Probability of resolution of the MUSIC algorithm,” *IEEE Transactions on Signal Processing*, vol. 43, no. 4, pp. 978–987, 1995.

- [66] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound: further results and comparisons," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2140–2150, 1990.
- [67] S. Haykin, *Array Signal Processing*. Prentice Hall, 1985.
- [68] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985.
- [69] W. Gabriel, "Spectral analysis and adaptive array superresolution techniques," *Proceedings of the IEEE*, vol. 68, no. 6, pp. 654 – 666, june 1980.
- [70] F. Haber and M. Zoltowski, "Spatial spectrum estimation in a coherent signal environment using an array in motion," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 301–310, 1986.
- [71] J. Cadzow, "High performance spectral estimation—a new ARMA method," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 5, pp. 524 – 529, oct 1980.
- [72] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989.
- [73] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.
- [74] Y. Liu and Z. Lin, "Active array beamforming using the frequency-response masking technique," in *Proc. Int. Symp. Circuits Syst. ISCAS*, 2004, pp. 197–200.
- [75] ———, "On the applications of the frequency-response masking technique in array beamforming," *Circuits, Systems and Signal Processing*, vol. 25, no. 2, pp. 201–224, 2006.
- [76] P. P. Vaidyanathan and C. Weng, "Active beamforming with interpolated FIR filtering," in *Proc. Int. Symp. Circuits Syst. ISCAS*, 2010, pp. 173–176.
- [77] Y. Xiao, L. Ma, and K. Khorasani, "A novel wideband DOA estimation technique based on harmonic source model for a uniform linear array," in *Proc. Int. Symp. Circuits and Systems ISCAS '04*, vol. 3, 2004.
- [78] S. Pei and C. Tseng, "Complex adaptive IIR notch filter algorithm and its applications," *IEEE Transactions on Circuits and Systems Part II: Analog and Digital Signal Processing*, vol. 41, no. 2, pp. 158–163, 1994.
- [79] J. Shynk, "A complex adaptive algorithm for IIR filtering," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 34, no. 5, pp. 1342–1344, 1986.

- [80] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 33, no. 4, pp. 983–996, 1985.
- [81] J. Zhou and G. Li, "Plain gradient based direct frequency estimation using second-order constrained adaptive IIR notch filter," *Electronics Letters*, vol. 40, no. 5, pp. 351–352, 2004.
- [82] Y. Xiao, Y. Takeshita, and K. Shida, "Steady-state analysis of a plain gradient algorithm for a second-order adaptive IIR notch filter with constrained poles and zeros," *IEEE Transactions on Circuits and Systems Part II: Analog and Digital Signal Processing*, vol. 48, no. 7, pp. 733–740, 2001.
- [83] C. Tseng and S. Pei, "Stable IIR notch filter design with optimal pole placement," *IEEE Transactions on Signal Processing*, vol. 49, no. 11, pp. 2673–2681, 2001.
- [84] Y. Xiao, L. Ma, K. Khorasani, and A. Ikuta, "Statistical performance of the memoryless nonlinear gradient algorithm for the constrained adaptive IIR notch filter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 8, pp. 1691–1702, 2005.
- [85] P. Stoica and A. Nehorai, "Performance analysis of an adaptive notch filter with constrained poles and zeros," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 36, no. 6, pp. 911–919, 1988.
- [86] C. Ko and C. Li, "An adaptive IIR structure for the separation, enhancement, and tracking of multiple sinusoids," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2832–2834, 1994.
- [87] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [88] P. S. R. D. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd ed. Boston, MA: Springer, 2008.
- [89] G. L. Sicuranza, "Quadratic filters for signal processing: principles and applications," in *Proc. IEEE Int Systems Engineering Conf*, 1992, pp. 5–8.
- [90] A. Monin and G. Salut, "IIR Volterra filtering with application to bilinear systems," *IEEE Transactions on Signal Processing*, vol. 44, no. 9, pp. 2209–2221, 1996.
- [91] G. Salut and A. Monin, "IIR Volterra filtering with application to bilinear systems," *IEEE Transactions on Signal Processing*, vol. 44, no. 9, 1996.
- [92] E. Roy, R. W. Stewart, and T. S. Durrani, "Theory and applications of adaptive second order IIR volterra filters," in *Proc. Conf. IEEE Int Acoustics, Speech, and Signal Processing ICASSP-96*, vol. 3, 1996, pp. 1597–1600.

- [93] A. Monin, G. Salut, and V. Teuliere, "Multisource discrimination using IIR volterra filtering," *IEEE Transactions on Communications*, vol. 49, no. 5, pp. 922–931, 2001.
- [94] Z. Chen and A. Manikas, "Joint space-time transmitter-receiver beamforming," in *Proc. IEEE Global Telecommunications Conf. GLOBECOM 2010*, 2010, pp. 1–5.
- [95] C. M. Panazio and J. M. T. Romano, "Performance of joint space-time equalization and decoding techniques for wireless systems," in *International Telecommunications Symposium - ITS*, 2002.
- [96] R. Punchalard, J. Koseeyaporn, and P. Wardkein, "Modified sign algorithm for a constrained adaptive IIR notch filter," in *Proc. 14th Asia-Pacific Conf. Communications APCC 2008*, 2008, pp. 1–5.
- [97] Y. Xiao, Y. Tadokoro, and Y. Kobayashi, "A new memoryless nonlinear gradient algorithm for a second-order adaptive IIR notch filter and its performance analysis," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 4, pp. 462–472, 1998.