

Signal Processing for Line Spectra: New Sensor Arrays, Algorithms, and Theoretical Results

Thesis by
Pranav Dhananjay Kulkarni

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy



CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended May 13th, 2025

© 2025

Pranav Dhananjay Kulkarni
ORCID: 0000-0002-1461-0948

All rights reserved

ACKNOWLEDGEMENTS

My journey at Caltech has been incredibly enriching and transformative. This experience would not have been possible without the unwavering support, encouragement, guidance, and mentorship of the many individuals who contributed to my growth. It is impossible to fully express in words the depth of my gratitude to each person who has shaped this journey. This acknowledgment is only a small reflection of the appreciation I carry in my heart.

First and foremost, I would like to express my heartfelt gratitude to my advisor, Prof. P. P. Vaidyanathan. His guidance, insightful teachings, and constant support have been central to my journey at Caltech. From the very first time I met him, even before I applied to Caltech, he has been incredibly supportive. With each passing year, my appreciation for him has only deepened. I could not have asked for a more exemplary mentor and role model. PP is a truly inspiring researcher, embodying sincerity, intellectual rigor, and an enduring passion for knowledge. He taught me to ask fundamental research questions and to pursue them with persistence, even when the path seemed unclear. He also instilled in me the importance of restraint and precision in scientific communication. Beyond his academic brilliance, PP is a kind, humble, and compassionate person. His values, both as a scholar and as a human being, have offered me lessons that go far beyond the standard advisor-advisee relationship. I am also grateful to him and Usha aunty for kindly caring about me on a personal level, especially when I was sick with COVID. I am deeply honored to have been mentored by him, and I will always strive to carry forward the principles and spirit he exemplifies.

I would also like to thank the members of my candidacy and defense committees: Prof. Yaser Abu-Mostafa, Prof. Victoria Kostina, and Prof. Venkat Chandrasekaran. I am grateful for their valuable feedback and suggestions on my work. In addition to being researchers I greatly admire, all three are exceptional educators. I thoroughly enjoyed their mathematically rigorous courses, which enriched my academic experience at Caltech. I am also thankful to the other professors at Caltech, including Prof. Konstantin Zuev, Prof. Babak Hassibi, Prof. Katie Bouman, Prof. Andrew Stuart, Prof. Yisong Yue, Prof. Antonio Rangel, and others, for imparting both foundational and cutting-edge knowledge with their exciting courses, and for instilling in me a disciplined and curious approach to learning and problem-solving. I have thoroughly enjoyed attending research seminars by numerous professors and

researchers from Caltech, as well as visiting professors and scholars, all of which have been both inspiring and intellectually stimulating.

I would like to sincerely thank the members of my research group—Po-Chih, Oguzhan, Srikanth, Piya, and Chun-Lin—for their inspiring and outstanding research, which set a very high standard for me to aspire to, and for the valuable feedback they offered on my work. The many insightful conversations I had with these academic siblings—both technical and personal—were instrumental in helping me navigate the challenges of the PhD journey and also for what lies beyond.

I am also grateful for the administrative support I received throughout my time at Caltech. I would like to especially thank the dedicated administrative staff in Moore, particularly Tanya, Caroline, and Christine, for their assistance. I also appreciate the support provided by the advisors from Caltech International Student Programs (ISP). Additionally, I want to sincerely thank the exceptionally dedicated team at the Caltech Student Wellness Center for taking care of my well-being. Finally, I am thankful for the generous financial support from the Office of Naval Research (ONR), the National Science Foundation (NSF), and Caltech, which made this research possible.

I would like to take this opportunity to also thank the many academic mentors and teachers who have guided me and helped me reach this stage at Caltech. During my undergraduate years at IIT Bombay, Prof. Vikram Gadre played a pivotal role in developing my research skills and preparing me for graduate studies. I am deeply grateful to him for always believing in me, mentoring me, caring for me on a personal level, and encouraging me to pursue my academic aspirations. His mentorship, along with the support for my graduate school applications, played a key role in helping me reach Caltech. I am also thankful to Prof. Preeti Rao at IIT Bombay and Prof. C.-C. Jay Kuo at USC for giving me the opportunity to conduct research in their groups under their guidance, and for their support in my graduate school applications. I sincerely thank the professors at IIT Bombay—especially, Prof. Ajit Rajwade, Prof. Prasanna Chaporkar, Prof. Saisundararaman Vishwanathan, Prof. Virendra Singh, Prof. Sibi Raj Pillai, Prof. M. B. Patil, Prof. Jaykrishnan Nayar, Prof. Suyash Awate, and Prof. Ganesh Ramakrishnan—for their engaging and intellectually stimulating courses, which played an important role in helping me build a strong foundation during my undergraduate years. I am also deeply grateful to my high school teachers and mentors from Jnana Prabodhini (JP) and Jnana Prabodhini Prashala (JPP), who played a foundational role in nurturing my curiosity in science and mathematics, as

well as supporting my personal development beyond academics. Jnana Prabodhini is home to countless inspiring veterans, and I would like to especially thank Ponkshe Sir, Girish Rao, Mahendra Bhai, Subhash Rao, Vivek Kulkarni Sir, Prashant Sir, and Anagha Tai. Their direct and indirect influence has inspired, motivated, and guided me, and they have instilled in me a strong sense of social responsibility. Aakash Dada has been a mentor to me since my NTSE preparation days, and over the years, he has become a close friend and now a colleague. I sincerely thank him for his unwavering guidance and support, both academically and personally. I would like to thank Gondhalekar Sir, whose engaging teaching made mathematics my favorite subject in school. I am also thankful to Monish Dada and Milind Sir for sparking my interest in physics, and to Neha Tai, Mukulika Tai, and Shantala Tai for inspiring my broader curiosity in science. I am especially grateful to Nachiket Dada and Aditya Dada, who gave me my earliest glimpses into the world of research—I thank them wholeheartedly for opening that door for me. I also extend my gratitude to other teachers at JPP, including Ashwini Tai, Wagh Sir, Hasabnis Sir, Joshi Sir, Ragini Tai, and Sanskruti Tai, for believing in me and supporting my academic growth. I am thankful to Kishore Darak Sir, M. Prakash Sir, Sanjeet Sir, Sahu Sir, and the teachers from Pace for training me in the scientific thinking and mathematical rigor essential for the journey beyond high school and for preparing me for the IIT entrance. I am also grateful to Aau and Sandhya Kaku for guiding me during important phases in my life and helping me gain confidence in my abilities. Finally, I would like to sincerely thank Sutar Sir and my teachers from Abhinava Vidyalaya for the important role they played at the very beginning of my academic journey.

The Philosophy of Yoga club at Caltech has been instrumental in rekindling my interest in spirituality and encouraging deeper inquiry into the purpose of life. I sincerely thank Kunal and Yamini for their warm association and for hosting the weekly *Sanga* gatherings at their home—something I always looked forward to. Their spiritual sincerity, coupled with their professional productivity and ability to balance academic, personal, and social responsibilities, continues to inspire me deeply. I also thank Akshay, Rathin, Saransh, Shreya, Ashesha, Nandini Radha, Guruprasad, Karla, Raju, and Ronak for fostering my sincerity and for exemplifying that academic and spiritual pursuits need not be compartmentalized, but can be meaningfully integrated into a holistic life. I am equally grateful to the Gyaan Ganga group at IIT Bombay for the wonderful association and the regular weekly meditations, which helped me remain calm and composed, especially during the turbulent final years of my undergraduate studies. I am deeply grateful to Gadre

ma'am and Gandhi Sir for their thoughtful teachings and for introducing me to aspects of Indian philosophy and structured meditations, which have had a lasting impact on me. Praveen, Vaishnavi, Nishkala, Ritu, and other members of the IITB Gyaan Ganga group have been a great inspiration, and I thank them for their support and encouragement. These communities—both at Caltech and IIT Bombay—have helped me navigate not only the inevitable highs and lows of academic life but also the broader challenges of life itself, all while reminding me to stay grounded and to continually seek the deeper meaning of existence.

One of my key support systems has been the Jnana Prabodhibni Foundation (JPF), a non-profit organization where I have had the opportunity to volunteer. It is incredibly inspiring to be surrounded by a dedicated group of volunteers who selflessly contribute to social causes and care for one another like family. I want to sincerely thank Pranav K., Vidya Tai, Revati, Venkatesh, Vedavati Tai, Shruti, Kalyani, Niramay, Supriya Tai, Anjali Tai, Nikhil, Gaurav, Priyanka P., Prajakta, Swapneeta, Swanand, Anuj, Bhooshan, Akshay, Shilpa Tai, Chetan, Aalok, Nath, Kunal, Prasad, two more Pranav K.-s, Abhishek S., Vijay, Omkar, and all the other JPF volunteers for their familial friendships and for always looking out for me. Although I have met most of them in person only once or not at all, it has never felt that way. I am also grateful to my relatives and family friends in the U.S., including Samruddhi Tai, Nikhil Dada, Jyoti Atya, Anand Kaka, Nachiket, Vinay Mama, Sharavati Mami, Mandar Dada, Ashish, Anu, Parag, Sameer, and Prasad, for providing a homely feeling while I am away from my home country and for their invaluable support and guidance in both my academic journey and everyday life in the U.S.

I would also like to thank all my friends from various circles who made my journey more enjoyable and also supported me in many different ways. I am especially grateful to Viraj and Yashvi for being close friends and for the wonderful times we shared at Catalina graduate housing, along with the many fun and memorable road trips together. I would like to thank my friends at Caltech, including Shawn, Vikrant, Sameer, Taylan, Danil, Abhimanyu, Robby, Thomas, James, Siming, Garima, Shivam, and Yeo, for their support during my time at Caltech. I will fondly remember the time spent playing tennis with Karteek, Souvik, Sumit, and Jagannath, as well as the fun and competitive spirit of the games with fellow members of the Caltech Chess Club, including Tony, Michael, Kaustav, and Jason. I am very grateful to the seniors at Caltech—Siddharth, Anjul, Deepan, Ayush, Sahin, Abhinav, Saumitra, and Domonic—who shared their experiences and invaluable

advice. I am equally grateful to my ‘academic cousins’ and friends—Pulak, Sina, Kuan-Lin, Robin, Rohan, and Mehmet—for their guidance and help during my job applications. Special thanks to Parthsarathi, Shreyas, John, Pranav, and Roshan for the fun times together during conferences. Many seniors from IITB guided me in my academic journey and continue to support me whenever I reach out. For that, I thank Parth, Samarth, Sudeep, Akhil, Unmesh, Omkar T., Arka, Parthe, Yash, Shounak, Aditya, Abhin, and others.

I am deeply grateful for the lifelong friendships I have forged at IIT Bombay and before that at Pace, particularly with Saurabh, Suyash, Karan, Anmol, Udayan, Rohan, Aditya W., Sanjeev, and Anand. They are always just a phone call away. During my IIT preparation, and while staying at IIT Bombay, they went above and beyond to help me, and their invaluable support at crucial junctures in my life is something I will never forget. I would also like to thank Shivam, Chinmay T., Dhruv, Arpit, Srivatsan, and Aditya T. for their continued friendship beyond our time at IIT Bombay. I’m equally grateful to Pranav P., Shreesh, Prathamesh, Chinmay S., Sudhendu, and Kanika for the enjoyable learning experiences we shared at Pace. I deeply appreciate my classmates from Jnana Prabodhini Prashala, with whom I shared wonderful memories throughout high school and beyond. I am grateful to Abhishek for being a close friend and for always inspiring me with his endless insights into various subjects. I also want to thank Soham, Samvid, Shravan, and Anish for motivating me to excel academically. My gratitude goes to Pranav, Unmesh, Apurv, and Mihir for lasting friendships, even after moving to the US. I also cherish the entire ‘7-13’ group—Hrishikesh, Swapnil, Rhugwed, Mohit, Chinmay, Gaurav, Sushant, Parth, Tushar, and others—for the memorable times we have shared both during and after school. Additionally, I am thankful to the friends from the Yoganand Park group, including Nikita, Sayali, Anagha, Meera, Mandar, Rutvik, and Advait. Lastly, I would like to thank my friends and mentors from my recent internship at Analog Devices. I greatly appreciate Xiao, Michael, Alan, Atulya, Frank, Varun, Ravi, Sefa, Niklas, Srilakshmi, and Berk for their guidance, inspiring research, and their emphasis on the ‘problem-first’ approach to industry research. I also thank Adrien, Sneha, Ajit, Gayatri, Aditya K., Revati D., Shouvik, Mahmoud, Erkan, Melihcan, and Vanessa for the enjoyable times we have shared together in Boston.

Last, but certainly not least, I want to express my deepest gratitude to my family back in India. My parents—Aai and Baba—and my grandparents—Ajji, Nana,

Babujji, and Ada—have nurtured me with unwavering love, values, and support. They have played a crucial role in shaping the person I am today. Their sacrifices, guidance, and boundless support are as the foundation of my life upon which all my aspirations rest. Words can never fully capture the depth of my appreciation for their enduring encouragement and care. I am also deeply thankful to my immediate uncles and aunts—Kaku, Sharumavshi, Tatamavshi, Kaka, and Sonkaka—who always showered me with immense care and affection. My cousins—Shambhavi, Chinmay, Ira, and Neel—are more like siblings to me, and I treasure the bond we share. I also extend heartfelt thanks to all my relatives, both maternal and paternal, whose love and constant support have been a source of strength throughout my journey. Finally, I am also grateful to my fiancée, Priyanka. Her love, support, and kindness mean a lot to me. I am excited to see what the future holds for us.

Ultimately, I am eternally grateful to the supreme absolute divine, whom I call *Śrī Kṛṣṇa*. His teachings from *Bhagavad-Gītā* have always been one of the most inspiring and influential things for me. The verses from the *Gītā* have guided me in some most challenging moments and periods in life, and my faith in Him has given me the strength to overcome the difficulties. I am forever thankful for the wisdom He imparts through the *Gītā* and other sacred texts, and through the teachings of exalted *Acharyas* and *Gurus*, all of which continue to nurture my spiritual growth, encourage me to follow the righteous path, and enrich my existence. Therefore, I dedicate this thesis to Him.

ABSTRACT

Line spectrum signals appear in diverse application areas such as molecular dynamics, power electronics, speech processing, and target localization. They are composed of sums of complex exponentials with distinct frequencies. Identifying the parameters of these constituent complex exponentials has been a prominent research topic in signal processing for over four decades. In this thesis, we focus on two specific applications involving line spectrum signals: direction of arrival (DOA) estimation using sensor arrays, and denoising of discrete-time periodic signals.

The main contribution of this thesis on the topic of DOA estimation is to propose unconventional sensor array geometries and algorithms in the presence of aperture constraints. In the first part, we demonstrate that under an aperture constraint, the traditional integer arrays (defined as arrays with sensors placed at integer multiples of the half-wavelength distance $\lambda/2$) can perform only suboptimally because of the restrictive sensor placement at integer locations. To address this, we propose to use ‘rational arrays’ that can have sensors located at rational multiples of $\lambda/2$. This offers greater flexibility in sensor placement under aperture constraints. In particular, we propose rational coprime arrays that can approach the Cramér-Rao bound (CRB) even at low signal-to-noise ratio (SNR) and with a limited number of snapshots, and can outperform the integer arrays. Numerical simulations show that rational arrays are also better equipped to resolve closely separated DOAs. To enable the derivation of the theoretical results and identifiability guarantees for rational coprime arrays, we extend the number-theoretic concepts such as greatest common divisor and coprimality to the case of rational numbers, and prove several number-theoretic properties. Rational arrays are also demonstrated to have important advantages when the DOAs are known to lie in a sector of the space, and for identifying $O(N^2)$ uncorrelated sources using N sensors under aperture constraint.

In the second part of the thesis, we propose modifications to the traditionally used sparse (integer) array design criteria. These modifications are aimed at mitigating the impact of mutual coupling on DOA estimation and reducing the required aperture. To reduce the impact of mutual coupling, we propose two types of sparse arrays that have either double or triple minimum inter-element spacing compared to the traditionally used $\lambda/2$ spacing. This introduces ‘holes’ at lags 1 and 2 in the difference coarrays (defined as the set of differences in sensor locations). The first type of arrays, called weight-constrained sparse arrays, have $O(N)$ aperture,

making them suitable when the available aperture is constrained and the number of DOAs is small. A general array construction, to further reduce the weights at other coarray lags, is also proposed. The second type of arrays, called weight-constrained nested arrays, have $O(N^2)$ degrees of freedom and are suitable when there are no aperture restrictions. Extensive Monte-Carlo simulations demonstrate that the proposed arrays have significantly smaller DOA estimation errors compared to the well-known sparse arrays from the literature, in the presence of high mutual coupling.

Because of the central holes in the difference coarrays of the weight-constrained arrays, there are two segments of consecutive entries in their coarrays: one on the positive side and the other on the negative side. To leverage these both, we propose to use an augmented coarray covariance matrix for the subspace-based algorithms such as multiple signal classification (MUSIC) and estimation of signal parameters via rotational invariance (ESPRIT). This further reduces the DOA estimation error for the weight-constrained arrays, and the computation time of augmented-MUSIC is significantly less than that of optimization-based methods, such as coarray interpolation and dictionary-based methods. We also develop methods to algorithmically interpolate the missing entries in the coarray at lags 1 and 2, to generate a larger coarray matrix. This approach demonstrates the capability to identify up to twice as many DOAs compared to what can be achieved using only the one-sided segment of consecutive lags in the coarray. This mitigates the main disadvantage of having central holes in the coarrays of weight-constrained arrays, while still benefiting from their advantage in reducing the impact of mutual coupling.

One major drawback of using coarray-MUSIC for DOA estimation is its inefficiency (i.e., the mean squared error (MSE) does not approach CRB, even asymptotically). We conduct several experiments to provide new insights into the complex relationship of coarray-MUSIC MSE on several parameters, such as array geometry, DOA separation, and accuracy of the estimated array output correlations. Furthermore, we demonstrate that an alternative way of constructing the Toeplitz covariance matrix can greatly improve the MSE compared to coarray-MUSIC, and can lead to efficient DOA estimation. This approach is based on solving an optimization problem whose objective is derived using the asymptotic error distribution of the known entries from the covariance matrix. We also propose a modification to the Toeplitz covariance matrix construction approach to account for the presence of mutual coupling and provide simulations with different sparse arrays.

The third part of the thesis is focused on developing a periodicity-aware signal denoising framework using Capon-optimized Ramanujan filter banks and pruned Ramanujan dictionaries. The signal reconstruction (synthesis) is done by solving a regularized optimization problem, based on the outputs of the analysis filter bank. This hybrid analysis-synthesis framework ensures that the denoised output is necessarily composed of discrete-time periodic components. Capon beamforming principles from array signal processing are utilized to optimize the Ramanujan filters to the incoming data. A computationally efficient way of obtaining the inverses of the required autocorrelation matrices is derived using Levinson's recursion. The proposed denoising method is observed to be effective even when the signal length is small and demonstrates a high SNR gain across a wide range of input signal SNRs. Furthermore, we derive several decimation properties of Ramanujan subspace signals, which help in reducing the required computations by appropriately downsampling the filter outputs without any loss of information.

Towards the end of the thesis, we theoretically investigate the locations of zeros of Ramanujan filters. Additionally, we propose an ideal interpolation filter model for Ramanujan subspace signals, which has potential application in developing a synthesis filter bank counterpart to the Ramanujan analysis filter bank for perfect signal reconstruction. We also explore the use of dictionary learning to represent periodic signals, and adapt a convolutional neural network based DOA estimation method to sparse arrays.

PUBLISHED CONTENT AND CONTRIBUTIONS

All the publications below resulted from research carried out by Pranav D. Kulkarni and Professor Palghat P. Vaidyanathan. Out of these, Pranav D. Kulkarni was the primary contributor of all journal papers and conference papers. He proposed the main results, proved theorems and lemmas, conducted experimental validations, wrote papers, and corresponded with editors and reviewers. Professor Palghat P. Vaidyanathan applied for research grants, supervised research projects, discussed research topics with Pranav D. Kulkarni, and reviewed the main results, paper drafts, and the correspondences with reviewers. Professor Palghat P. Vaidyanathan was the primary contributor of the book chapter [9]. He compiled the material and wrote the book chapter. Pranav D. Kulkarni helped with experimental simulations, demonstrations, and the literature review on a few topics.

- [1] P. Kulkarni and P. P. Vaidyanathan, “On the efficiency of a Toeplitz covariance matrix construction approach for DOA estimation,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Oct. 2025, under review, The content of this paper is described in Chapter 6.
- [2] P. Kulkarni and P. P. Vaidyanathan, “Statistically efficient algorithms for DOA estimation with sparse arrays,” in preparation, to be submitted to an IEEE journal, The content of this paper is based primarily on Chapter 6.
- [3] P. Kulkarni and P. P. Vaidyanathan, “Generalized constructions of weight-constrained sparse arrays,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2025, The content of this paper is described in Chapter 5.
- [4] P. Kulkarni and P. P. Vaidyanathan, “Interpolation for weight-constrained nested arrays having non-central ULA segments in the coarray,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2025. DOI: 10.1109/ICASSP49660.2025.10890061, The content of this paper is described in Chapter 5.
- [5] P. Kulkarni and P. P. Vaidyanathan, “Efficient use of non-central ULA segments in the coarrays of weight-constrained sparse arrays,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2024. DOI: 10.1109/IEEECONF60004.2024.10943027, The content of this paper is described in Chapter 5.

- [6] P. Kulkarni and P. P. Vaidyanathan, “Sparse, weight-constrained arrays with $O(N)$ aperture for reduced mutual coupling,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 12 891–12 895. DOI: 10.1109/ICASSP48485.2024.10446770, The content of this paper is described in Chapter 4.
- [7] P. Kulkarni and P. P. Vaidyanathan, “Weight-constrained nested arrays with $w(1) = w(2) = 0$ for reduced mutual coupling,” in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Jul. 2024, pp. 1–5. DOI: 10.1109/SAM60225.2024.10636451, The content of this paper is described in Chapter 4.
- [8] P. Kulkarni and P. P. Vaidyanathan, “Weight-constrained sparse arrays for direction of arrival estimation under high mutual coupling,” *IEEE Transactions on Signal Processing*, Sep. 2024. DOI: 10.1109/TSP.2024.3461720, The content of this paper is described in Chapter 4.
- [9] P. P. Vaidyanathan and P. Kulkarni, “Sparse arrays: Fundamentals,” *Sparse Arrays for Radar, Sonar, and Communications*, pp. 1–40, 2024. DOI: 10.1002/9781394191048.ch1, The content from some parts of this book chapter is described in Chapter 2.
- [10] P. Kulkarni and P. P. Vaidyanathan, “Difference coarrays of rational arrays,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095426, The content of this paper is described in Chapter 3.
- [11] P. Kulkarni and P. P. Vaidyanathan, “Interpolation filter model for Ramanujan subspace signals,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095682, The content of this paper is described in Chapter 8.
- [12] P. Kulkarni and P. P. Vaidyanathan, “On the efficiency of coarray-based direction of arrival estimation,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2023, pp. 1334–1338. DOI: 10.1109/IEEECONF59524.2023.10476980, The content of this paper is described in Chapter 6.
- [13] P. Kulkarni and P. P. Vaidyanathan, “Periodicity-aware signal denoising using Capon-optimized Ramanujan filter banks and pruned Ramanujan dictionaries,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 494–511, Feb. 2023. DOI: 10.1109/TSP.2023.3244109, The content of this paper is described in Chapter 7.
- [14] P. Kulkarni and P. P. Vaidyanathan, “Non-integer arrays for array signal processing,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 5457–

5472, Nov. 2022. doi: 10.1109/TSP.2022.3221862,
The content of this paper is described in Chapter 3.

- [15] P. Kulkarni and P. P. Vaidyanathan, “Rational arrays for DOA estimation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 5008–5012. doi: 10.1109/ICASSP43922.2022.9746954,
The content of this paper is described in Chapter 3.
- [16] P. Kulkarni and P. P. Vaidyanathan, “Rational arrays for DOA estimation: New insights and performance evaluation,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2022, pp. 91–95. doi: 10.1109/IEEECONF56349.2022.10051963,
The content of this paper is described in Chapter 3.
- [17] P. Kulkarni and P. P. Vaidyanathan, “Feature engineering for DOA estimation using a convolutional neural network, for sparse arrays,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2021, pp. 246–250. doi: 10.1109/IEEECONF53345.2021.9723112,
The content of this paper is described in Chapter 9.
- [18] P. Kulkarni and P. P. Vaidyanathan, “Periodic signal denoising: An analysis-synthesis framework based on Ramanujan filter banks and dictionaries,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 5100–5104. doi: 10.1109/ICASSP39728.2021.9413689,
The content of this paper is described in Chapter 7.
- [19] P. Kulkarni and P. P. Vaidyanathan, “K-SVD based periodicity dictionary learning,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2020, pp. 1333–1337. doi: 10.1109/IEEECONF51394.2020.9443567,
The content of this paper is described in Chapter 9.
- [20] P. Kulkarni and P. P. Vaidyanathan, “On the zeros of Ramanujan filters,” *IEEE Signal Processing Letters*, vol. 27, pp. 735–739, Apr. 2020. doi: 10.1109/LSP.2020.2990610,
The content of this paper is described in Chapter 8.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	ix
Published Content and Contributions	xii
Table of Contents	xiv
List of Illustrations	xviii
List of Tables	xxxiii
Chapter I: Introduction	1
1.1 Sensor Arrays	3
1.2 Array Output Equation	4
1.3 Periodic Signals and Ramanujan Sums	7
1.4 Learned Transformation for Signal Representation	9
1.5 Contributions and Outline of This Thesis	11
1.6 Notations	18
1.7 List of Acronyms	20
Chapter II: Review of Sparse Arrays and DOA Estimation Algorithms	22
2.1 Introduction	22
2.2 Subspace-Based DOA Estimation Algorithms	22
2.3 Sparse Arrays and Their Difference Coarrays	26
2.4 The Effect of Mutual Coupling	30
2.5 Coarray-Based DOA Estimation	32
2.6 CRB, Analytical MSE, and Efficiency of DOA Estimation	34
2.7 Concluding Remarks	38
Chapter III: Non-integer Arrays for Direction of Arrival Estimation	39
3.1 Introduction	39
3.2 Rational Number Theory: Some Basic Results	42
3.3 Theory of Rational Arrays	50
3.4 Rational Coprime Arrays	58
3.5 Restricted Spatial Scope	72
3.6 Arrays with Arbitrary (Non-Rational) Sensor Locations	80
3.7 Difference Coarrays of Rational Arrays	82
3.8 Concluding Remarks	91
Chapter IV: Weight-Constrained Arrays For DOA Estimation Under High Mutual Coupling and Aperture Constraints	92
4.1 Introduction	92
4.2 Coarray-MUSIC Using One-Sided ULA Segment in Coarray	98
4.3 Proposed Weight-Constrained Sparse Arrays with $O(N)$ Aperture	101
4.4 Proposed Weight-Constrained Nested Arrays with $O(N^2)$ Aperture	106
4.5 Simulation Results	110
4.6 Concluding Remarks	134

Chapter V: Improved DOA Estimation Algorithms and Generalized Construc-	
tions for Weight-Constrained Arrays	136
5.1 Introduction	136
5.2 Augmented Coarray Root-MUSIC	139
5.3 Covariance Interpolation for WCNA's	141
5.4 Simulations Comparing Proposed Algorithms	144
5.5 Generalized Construction of Weight-Constrained Sparse Arrays . . .	157
5.6 Simulations with Generalized Construction of WCSAs	161
5.7 Concluding Remarks	165
Chapter VI: On the Efficiency of Coarray-Domain DOA Estimation Algorithms	168
6.1 Introduction	168
6.2 Efficiency of Coarray-MUSIC for ULA	171
6.3 Two Variations of Coarray Covariance Matrix	173
6.4 Optimization-Based Construction of Coarray Covariance Matrix . . .	180
6.5 Efficiency of the Optimization-Based Approach for Different Arrays .	182
6.6 Modification of the Optimization-Based Approach in the Presence	
of Mutual Coupling	186
6.7 Concluding Remarks	190
Chapter VII: Periodicity-Aware Signal Denoising Using Capon-Optimized	
Ramanujan Filter Banks and Pruned Ramanujan Dictionaries	193
7.1 Introduction	193
7.2 Ramanujan Filter Banks and Dictionaries	196
7.3 Hybrid Analysis-Synthesis Framework	200
7.4 Optimizing Ramanujan Filters Using Capon Beamforming Principles	209
7.5 Discussions on Computational Complexity	216
7.6 Multirate Properties of Ramanujan Subspace Signals	218
7.7 Comparative Simulations with Decimation, and Other Investigations	226
7.8 Concluding Remarks	236
Chapter VIII: Theoretical Properties of Ideal Ramanujan Filters	239
8.1 Introduction	239
8.2 Interpolation Filter Model for Ramanujan Subspace Signals	241
8.3 Further Results on the Output Space	244
8.4 Locations of Zeros of FIR Ramanujan Filters	249
8.5 Identifying Further Structure in the Locations of Zeros	252
8.6 Efficient Structure for Ramanujan Filter Banks	257
8.7 Concluding Remarks	258
Chapter IX: Learning-Based Methods for Representing Periodic Signals and	
DOA Estimation	259
9.1 Introduction	259
9.2 Review of Periodicity Dictionaries and K-SVD Algorithm	261
9.3 Goals and Experimental Setting	264
9.4 Experimental Results	266
9.5 Review of CNN-Based Method from [149]	270
9.6 Proposed Modifications for Sparse Arrays	272
9.7 Proposed Proxy Spectra	273

9.8 Simulation Results	276
9.9 Concluding Remarks	281
Chapter X: Conclusions and Future Directions	282
Bibliography	285

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 A representative example of the DTFT of a general line spectrum signal from Eq. (1.1).	2
1.2 Examples of line spectra arising in the context of DOA estimation and periodic signals. (a) A MUSIC spectrum (see Sec. 2.2) showing line spectrum in the angular domain in the context of DOA estimation. (b) A DTFT of a noisy periodic signal showing a line spectrum in the frequency domain.	3
1.3 Standard model of a passive linear array of N sensors receiving plane waves from D far-field sources at angles $\theta_1, \theta_2, \dots, \theta_D$ from the array normal.	5
2.1 (a) A nested array consists of a dense ULA of N_1 sensors followed by a sparse ULA of N_2 sensors with interelement spacing $N_1 + 1$. (b) A coprime array is a union of two sparse ULAs with interelement spacings N_1 and N_2 where $(N_1, N_2) = 1$	29
2.2 Coarray-MUSIC plots. (a) Identifying $D = 25$ DOAs using a 10-sensor nested array with $N_1 = N_2 = 5$ when SNR = 0 dB and $K = 300$. (b) Identifying $D = 35$ DOAs with 17-sensor coprime array with $N_1 = 5, N_2 = 8$ when SNR = 10 dB and $K = 500$. The black vertical lines correspond to the true DOAs.	34
2.3 Plots of CRB for sparse arrays identifying (a) $D = 10$ and (b) $D = 30$ DOAs with $N = 16$ sensors. When $D < N$, CRB does not saturate, whereas when $D \geq N$, CRB saturates at large SNR.	36
3.1 When the available array aperture A equals $m - 1$, the standard integer ULA can be used for DOA estimation.	51
3.2 Situation where $A < m - 1$. (a) Integer ULA can only fit $\lfloor A \rfloor + 1$ sensors in the available aperture and can identify only $\lfloor A \rfloor$ DOAs. (b) A shrunk rational ULA with appropriately chosen interelement spacing $r < 1$ that satisfies the aperture constraint can identify $m - 1$ DOAs.	52

3.3	Comparison of (a) analytical MSE in the absence of mutual coupling and (b) numerical (Monte-Carlo) MSE in the presence of mutual coupling (Eq. (3.48)) when one DOA is fixed at 80° and the other DOA is varied. Integer ULA ($r = 1$) has higher MSE when sources impinge from the directions far away from the array normal on both sides.	53
3.4	Situation where $A > m - 1$. (a) An integer ULA of m sensors cannot utilize the full available aperture. (b) An integer coprime array [97], [126] consisting of an $(m - 1)$ -sensor sparse ULA with interelement spacing M and an additional sensor located at N where $(M, N) = 1$ can be suitable for this case. (c) An example with $A = 99$ and $m = 27$, where the integer coprime array cannot utilize the available aperture well. Most of the sensors are concentrated in only half of the aperture. (d) We cannot simply use a sparse rational ULA because it cannot identify the DOAs unambiguously.	55
3.5	Identifying 6 sources from Example 1. MUSIC spectrum obtained with (a) rational coprime array \mathbf{z}_1 , (b) standard ULA \mathbf{z}_2 , and (c) modified ULA \mathbf{z}_3 . True DOAs are shown in black vertical lines. . . .	60
3.6	Identifying 11 closely spaced sources from Example 2. MUSIC spectrum obtained with (a) rational coprime array \mathbf{z}_4 , (b) standard ULA \mathbf{z}_5 , and (c) modified ULA \mathbf{z}_6 . True DOAs are shown in black vertical lines.	62
3.7	Effect of the location of r_2 of array \mathbf{z}_1 from <i>Example 1</i> on the MSE. Choosing $r_2 = 1$, or equivalently $r_2 = A - 1$ leads to the smallest MSE.	63
3.8	Applying root-MUSIC to rational array \mathbf{z}_{10} . Phases of six zeros closest to the unit circle (shown in red lines) correctly estimate the DOAs with (a) $K = 400$ but not with (b) $K = 100$. Green lines correspond to scaled values of true DOAs, i.e. $\{r\omega_i\}_{i=1}^6$, where $r = (r_1, r_2) = 1/3$, and blue dots denote the zeros of $Q(z)$ in the sector $[-\pi r, \pi r)$	66
3.9	Identifying 6 nearby DOAs -12.6° , -7.7° , -2.8° , 2.8° , 7.7° , 12.6° when $m = 8$ and $A = 17$. MUSIC spectrum obtained with (a) integer ULA \mathbf{z}_7 , (b) modified ULA \mathbf{z}_8 , (c) integer coprime array \mathbf{z}_9 , and (d) rational coprime array \mathbf{z}_{10} . True DOAs are shown in black vertical lines.	67

3.10	Experimental (Monte-Carlo) MSE (shown in solid lines), theoretical MSE (shown with dashed lines), and CRB (shown in dotted lines) for the four arrays under consideration as (a) SNR is varied when the number of snapshots $K = 100$, (b) number of snapshots is varied when SNR = 0 dB.	69
3.11	Comparison of (a) theoretical, and (b) Monte-Carlo MSE values when one DOA is fixed at 30° and the other DOA is varied from -90° to 90° . The proposed rational coprime arrays \mathbf{z}_{10} and \mathbf{z}_{11} have the lowest MSE, except possibly when the two DOAs come close to being equivalent modulo $2\pi/r_1$	70
3.12	Comparison of numerical (Monte-Carlo) MSE when two sources are fixed at 0° and 8° and a third in-between DOA is varied. The rational coprime array is found to have the best resolution.	71
3.13	Identifying 8 sources in restricted spatial scope $\Theta = [-45^\circ, +45^\circ)$. MUSIC spectrum obtained with (a) standard integer ULA (b) rational array (3.80) with $\beta = 1.4$. The true DOAs are shown in solid black lines.	74
3.14	Identifying 11 closely spaced sources from Example 2. MUSIC spectrum obtained with (a) arbitrary (non-rational) coprime array \mathbf{z}_{13} , (b) standard ULA \mathbf{z}_5 , and (c) modified ULA \mathbf{z}_6 . True DOAs are shown in black vertical lines.	83
3.15	Identifying 12 sources with the 12-element nested rational array (3.118) having aperture $A = 10$. (a) MUSIC spectrum and (b) root-MUSIC plot. The true DOAs (in MUSIC plot) and corresponding angles $\omega_i/2$ (in root-MUSIC plot) are shown in solid black lines. . . .	86
3.16	Identifying 17 sources with the 12-element rational array (3.125) having aperture $A = 15$ in presence of mutual coupling (3.127). (a) MUSIC spectrum and (b) root-MUSIC plot. The true DOAs (in MUSIC plot) and corresponding angles $3\omega_i/4$ (in root-MUSIC plot) are shown in solid black lines.	88
3.17	Monte-Carlo MSE and CRB for array (3.125) when identifying 17 DOAs equispaced in ω domain. (a) Number of snapshots $K = 1000$, and (b) SNR = 10 dB.	89

- 4.1 Coarray weights $w(l)$, $0 \leq l \leq 40$ for the arrays under consideration. All arrays have $N = 16$. Out of the proposed arrays shown here, \mathbf{z}_1 , $\mathbf{z}_2^{(2)}$, $\mathbf{z}_3^{(2)}$, $\mathbf{w}_1^{(1)}$, and $\mathbf{w}_1^{(2)}$ have $w(1) = 0$, and \mathbf{z}_4 , \mathbf{z}_5 , \mathbf{z}_6 , and \mathbf{w}_2 have $w(1) = w(2) = 0$. The only other arrays that have $w(1) = 0$ are CADiS [83] and (approximate) constrained MRAs (cMRA) [84]. See Sec. 4.5.1 for details. 115
- 4.2 Coarray-MUSIC spectra obtained when identifying $D = 22$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with different 16-sensor arrays from Table 4.4. SNR of 0 dB, mutual coupling with $c = 0.3$ and $B = 10$, and 100 snapshots are considered. Black vertical lines correspond to the true DOA locations. Except for CADiS and cMRA, all other arrays from the literature do not correctly identify all 22 DOAs and miss out on at least one DOA. On the other hand, the proposed arrays are able to identify all DOAs correctly with a small MSE by producing coarray-MUSIC spectra having peak locations close to each true DOA. 116
- 4.3 MSE when estimating $D = 6$ DOAs uniformly spaced in ω from -0.8π to 0.8π with 16-sensor arrays. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. Although many of the proposed arrays perform worse than other arrays from the literature, \mathbf{z}_6 is found to perform the best in most situations. . . . 118
- 4.4 MSE when estimating $D = 20$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. Arrays with $w(1) = 0$ have smaller MSE, whereas arrays with $w(1) \neq 0$ cannot identify all the DOAs correctly, resulting in high MSE. 119
- 4.5 MSE when estimating $D = 6$ DOAs spaced non-uniformly in ω domain with different arrays under aperture constraint $A \leq 100$ and $N \leq 50$. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. The proposed $O(N)$ aperture arrays perform much better than other sparse arrays, because under the aperture constraint, they can fit in more sensors while maintaining small coupling leakage. 122

- 4.6 MSE when estimating $D = 20$ DOAs uniformly spaced in ω from -0.9π to 0.9π with different arrays under aperture constraint $A \leq 100$ and $N \leq 50$. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.3$ and $K = 1000$, (c) $c = 0.3$ and SNR = 5 dB. Despite high coupling leakage \mathcal{L} , ULA performs the best under low SNR or a small number of snapshots. The proposed array \mathbf{z}_1 performs well at higher SNRs or when a large number of snapshots are available. CADiS and ULA are found to perform well under very high mutual coupling. 123
- 4.7 MSE when identifying D DOAs spaced uniformly in the ω domain from -0.9π to 0.9π with 25-sensor arrays. SNR = 5 dB, $K = 500$, and $c = 0.3$. Array \mathbf{w}_2 has the smallest MSE when $35 \leq D \leq 80$. Arrays $\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2 can identify more DOAs than CADiS. . . . 125
- 4.8 MSE when identifying D DOAs spaced uniformly in the ω domain from -0.9π to 0.9π with 30-sensor arrays. SNR = 5 dB, $K = 500$, and $c = 0.4$. Array \mathbf{w}_2 has the smallest MSE when $30 \leq D \leq 90$ 126
- 4.9 Comparing the proposed arrays $\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2 with CADiS and cMRA in terms of the number of identifiable DOAs using coarray-MUSIC (D_m), coupling leakage \mathcal{L} when $c = 0.3$, and required array aperture. For each value of N , the CADiS array configuration having the longest ULA segment in the coarray is considered. cMRA is available only for $N < 20$. The breaks in the plot for CADiS correspond to the values of N for which CADiS is not available. The proposed WCNA can identify more DOAs than CADiS with coarray-MUSIC, and \mathbf{w}_2 has a comparable or smaller coupling leakage than CADiS for most values of N . However, apertures of $\mathbf{w}_1^{(2)}$ and \mathbf{w}_2 are roughly twice that of CADiS and $\mathbf{w}_1^{(1)}$ 127
- 4.10 MSE when estimating $D = 30$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 20-sensor arrays. All possible configurations of CADiS are considered. The parameters of the CADiS arrays are mentioned in the brackets (M , N , and p from [83], respectively). (a) SNR = 5 dB and $K = 500$, (b) $c = 0.3$ and $K = 500$, (c) $c = 0.3$ and SNR = 5 dB. Array \mathbf{w}_2 performs the best in this case when $c \leq 0.5$ for any number of snapshots and SNR. 128

- 4.11 Comparing different arrays under the constraints $A \leq 50$ and $N \leq 30$. Mutual coupling coefficients are of the form $c_l = (c_1/l)\exp(-j(l-1)\pi/M)$. (a) $M = 5$, (b) $M = 7$, and (c) $M = 8$. $D = 4$, $K = 1000$, and $\text{SNR} = 0$ dB. Under different conditions, proposed sparse arrays \mathbf{z}_1 or $\mathbf{z}_2^{(1)}$ can perform better than other sparse arrays with $O(N^2)$ aperture. \mathbf{z}_3 in the figure legends refer to the array $\mathbf{z}_3^{(1)}$ from Table 4.2. 130
- 4.12 MSE when identifying $D = 6$ DOAs spaced uniformly in the ω domain from -0.8π to 0.8π for arrays with $A \leq 100$ and $N \leq 50$. $\text{SNR} = 5$ dB, $K = 500$, and $c = 0.5$. The general trend of the proposed arrays compared to existing arrays remains consistent as the parameter B is changed. 131
- 4.13 MSE when estimating $D = 10$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays when $\text{SNR} = 5$ dB, and $K = 500$. The correlation coefficient between the second and fifth source is taken to be $\rho e^{-j\pi/4}$. (a) $c = 0$, and (b) $c = 0.4$ 132
- 4.14 MSE when estimating $D = 10$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays when $\text{SNR} = 5$ dB, and $K = 500$. The correlation coefficient between the second and fifth source is taken to be $\rho e^{-j\pi/4}$, and the correlation coefficient between the seventh and ninth source is $\rho/2 \cdot e^{j\pi/3}$. (a) $c = 0$, and (b) $c = 0.4$. . . 133
- 5.1 A representative demonstration of the coarrays of the weight-constrained arrays proposed in Chapter 4. Because of the central holes at lags 1 and 2, the coarrays do not contain a *central* ULA segment. There is a one-sided ULA segment on the positive side of the coarray from lags L_1 to L_2 , and a corresponding ULA segment on the negative side of the coarray from lags $-L_2$ to $-L_1$ 137

- 5.2 MUSIC spectra (left) and corresponding root-MUSIC plots (right) when estimating DOAs using WCNA $\mathbf{w}_1^{(1)}$ with $N = 8$. SNR is 0 dB, $K = 200$, and no mutual coupling is present. For root-MUSIC plots, the solid circles show D conjugate pairs of roots closest to the unit circle, and crosses denote other roots of the root-MUSIC polynomial. The black lines denote the angles corresponding to true DOAs $\{\omega_i\}_{i=1}^D$. (a) Estimating $D = 6$ non-uniformly spaced DOAs using one-sided ULA segment in coarray, as in Chapter 4, $\text{MSE} = 0.0990$. (b) Estimating the same six DOAs with interpolation, $\text{MSE} = 1.3785 \times 10^{-4}$. The MUSIC spectrum has much sharper peaks in this case. (c) Estimating $D = 18$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with interpolation, $\text{MSE} = 5.9150 \times 10^{-5}$ 143
- 5.3 Coarray root-MUSIC MSE for estimating $D = 6$ DOAs uniformly separated in ω from -0.9π to 0.9π with 16-sensor arrays. Solid lines show MSE obtained when using a one-sided ULA segment in the coarray (i.e., matrix $\hat{\mathbf{R}}^{(+)}$), and dashed lines show MSE obtained when using both sides of the ULA segments in the coarray (i.e., matrix $\hat{\mathbf{R}}_{aug}$). (a) $c = 0.4$ and $K = 500$, (b) $c = 0.4$ and SNR = 5 dB, (c) SNR = 5 dB and $K = 500$. Augmented root-MUSIC with $\hat{\mathbf{R}}_{aug}$ improves the DOA estimation MSE compared to $\hat{\mathbf{R}}^{(+)}$ for most of the arrays. 146
- 5.4 Simulation settings are the same as in Fig. 5.3, but with $D = 20$ instead of $D = 6$. Augmented root-MUSIC with $\hat{\mathbf{R}}_{aug}$ improves the DOA estimation MSE compared to $\hat{\mathbf{R}}^{(+)}$ for WCSAs and WCNA, as well as CADiS and cMRA over a wide range of SNR, snapshots, and coupling strength. 147
- 5.5 MSE when estimating $D = 6$ DOAs uniformly spaced in ω from -0.8π to 0.8π with 16-sensor arrays using augmented root-MUSIC. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. These plots are analogous to the plots from Fig. 4.3. We observe here that because of the use of both ULA segments present in the coarrays of weight-constrained sparse arrays, now many of the proposed arrays can perform at par or better than other arrays from the literature. 148

- 5.6 Simulation settings are the same as in Fig. 5.4, but using coarray ESPRIT and augmented ESPRIT instead of coarray root-MUSIC and augmented root-MUSIC. Improvement in MSE obtained with augmented ESPRIT is much smaller than the improvement observed with augmented root-MUSIC. 150
- 5.7 Dictionary-based DOA estimation. (a) and (b) show the dictionary-based spectra obtained for array \mathbf{z}_5 with $N = 16$ sensors, $K = 500$, SNR 5 dB, and $c = 0.4$ when $D = 6$ and $D = 20$, respectively. There are several spurious peaks when $D = 20$ and the DOAs are not correctly identified. (c) and (d) show the variation in Monte-Carlo MSE averaged over 200 trials, as the coupling strength c is varied on the X -axis, when $D = 6$ and $D = 20$, respectively. When $D = 20$, most arrays have high MSE when $c > 0.2$ 152
- 5.8 MUSIC spectra when estimating DOAs using WCNA $\mathbf{w}_l^{(1)}$ with $N = 8$, SNR 0 dB, $K = 200$, and $c = 0.2$. (a) Estimating $D = 6$ non-uniformly spaced DOAs located at -50° , 10° , 0° , 40° , 50° , and 80° using one-sided ULA segment in coarray, as in the previous chapter, MSE = 0.0977. (b) Estimating the same six DOAs with interpolation, MSE = 1.00×10^{-3} . The MUSIC spectrum has much sharper peaks in this case. (c) Estimating $D = 18$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with interpolation, MSE = 1.23×10^{-4} . 153
- 5.9 MSE when estimating $D = 6$ DOAs spaced uniformly in ω domain from -0.8π to 0.8π . The solid lines denote the MSE obtained after interpolation, and the dashed lines denote the MSE obtained (without interpolation) using the longest ULA segment in the coarray. (a) SNR = 5 dB and $K = 500$, (b) SNR = 5 dB and $c = 0.2$, (c) $K = 500$ and $c = 0.2$ 154
- 5.10 MSE when estimating $D = 16$ DOAs spaced uniformly in ω domain from -0.9π to 0.9π . Coarray root-MUSIC is used for SNA, whereas interpolation (Eq. (5.14)) followed by root-MUSIC is used for WC-NAs and CADiS. The coprime array with 8 sensors cannot identify 16 DOAs (even after interpolation) and is excluded. (a) SNR = 5 dB and $K = 500$, (b) SNR = 5 dB and $c = 0.2$, (c) $K = 500$ and $c = 0.2$. . 155
- 5.11 The generalized construction of weight-constrained sparse array \mathbf{z}_* from Eq. (5.18) when P is odd. 158

- 5.12 The effect of varying the parameter P on the aperture A , maximum number of identifiable DOAs using coarray MUSIC $D_m = \lfloor L/2 \rfloor$, and coupling leakage \mathcal{L} of the proposed generalized construction of WCSA \mathbf{z}_* . The aperture and D_m peak around $P \approx N/2$, and the coupling leakage is lowest for $5 \leq P \leq 7$ when $25 \leq N \leq 55$ 160
- 5.13 Comparing coarray ESPRIT MSE for estimating $D = 25$ DOAs uniformly separated in ω from -0.95π to 0.95π with 30-sensor arrays. (a) $c = 0.3$, $K = 500$, (b) SNR = 5 dB, $c = 0.3$, and (c) SNR = 5 dB, $K = 500$. The proposed WCSA \mathbf{z}_* with several different values of P has lower MSE than previously proposed WCSAs (\mathbf{z}_1 to \mathbf{z}_6) for a wide range of SNR and snapshots when the mutual coupling strength is lower than 0.5. 162
- 5.14 Comparing coarray ESPRIT MSE for estimating $D = 20$ DOAs uniformly separated in ω from -0.9π to 0.9π with 16-sensor arrays. (a) $c = 0.4$, $K = 500$, (b) SNR = 5 dB, $c = 0.4$, and (c) SNR = 5 dB, $K = 500$. The proposed WCSA \mathbf{z}_* , CADiS, and cMRA are the arrays that have $w(1) = 0$ and they can correctly identify the DOAs with small MSE when the coupling strength is larger than 0.2. . . . 163
- 5.15 Comparing coarray ESPRIT MSE of the arrays with $N = 50$ under the aperture constraint $A \leq 300$ when estimating $D = 25$ DOAs separated uniformly in ω from -0.9π to 0.9π . All possible configurations of CADiS satisfying the constraints are compared with three configurations of the proposed array satisfying the constraints. (a) $c = 0.3$, $K = 500$, (b) SNR = 5dB, $c = 0.3$, (c) SNR = 5dB, $K = 500$. The proposed array \mathbf{z}_* with $P = 5$ is observed to have the lowest MSE. 164
- 6.1 Experimental (Monte-Carlo) MSE, theoretical MSE [87], [122], and CRB [86] for direct-MUSIC (using $\hat{\mathbf{R}}_{\mathbf{xx}}$) and coarray-MUSIC (using $\hat{\mathbf{R}}_{DA}$) when (a) SNR is varied when the number of snapshots $K = 500$, (b) number of snapshots is varied when SNR = 0 dB. Corresponding efficiency (Eq. (2.35)) plots are shown in (c) and (d) respectively. . . 172
- 6.2 Estimating three DOAs -30° , 10° , and 50° with an 11-sensor ULA, 200 snapshots, and SNR 0 dB. (a) Standard coarray-MUSIC (using $\hat{\mathbf{R}}_{DA}$), (b) tall-MUSIC (using $\hat{\mathbf{R}}_{DA}^{tall}$), (c) fat-MUSIC (using $\hat{\mathbf{R}}_{DA}^{fat}$), and (d) direct-MUSIC (using $\hat{\mathbf{R}}_{\mathbf{xx}}$). The phases (angle with X-axis) of the blue and the red lines indicate the true DOAs ω_i and estimated DOAs $\hat{\omega}_i$ respectively. 175

- 6.3 Estimating six DOAs $-60^\circ, -40^\circ, 0^\circ, 15^\circ, 40^\circ$, and 60° with an 11-sensor ULA. Direct-MUSIC, coarray-MUSIC, tall-MUSIC, and fat-MUSIC are considered. (a) SNR is varied when the number of snapshots $K = 200$, (b) the number of snapshots is varied when $\text{SNR} = 10$ dB. Tall-MUSIC performs at par with coarray-MUSIC, and fat-MUSIC performs slightly worse. Fat-MUSIC has the best computational complexity. 176
- 6.4 Estimating 2 DOAs with an 11-sensor ULA, with 400 snapshots, and $\text{SNR} = 10$ dB. One DOA is fixed at 0° , and the other DOA is varied. (a) MSE with direct-MUSIC, coarray-MUSIC, tall-MUSIC, and fat-MUSIC. (b) Efficiency for direct-MUSIC and coarray-MUSIC. (c) Using a 21-sensor ULA with everything else the same as in (b). Coarray-MUSIC efficiency is observed to be much more sensitive to DOA separation than direct-MUSIC. 178
- 6.5 DOA estimation MSE for 15-sensor ULA when array output correlations are perturbed from their ideal value, one at a time. Three different perturbation variances, 0.01, 0.1, and 0.5, are considered. MSE is largest when correlations at lags around $N/2$ are perturbed, and smallest for $l_0 = 1$ and $l_0 = N - 1$ 179
- 6.6 Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 10$ DOAs separated uniformly in the ω -domain from -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. 183
- 6.7 Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 20$ DOAs separated uniformly in the ω -domain from -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. 184

6.8	Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\widehat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 30$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots.	185
6.9	Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\widehat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11), and the CRB when estimating $D = 10$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.	187
6.10	Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\widehat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11), and the CRB when estimating $D = 20$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.	188
6.11	Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\widehat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11), and the CRB when estimating $D = 30$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.	189
7.1	Dictionary coherence values of Ramanujan, Farey, and natural periodic dictionaries compared against the Welch bound as P_{max} is varied, keeping the number of rows constant at 100.	198

7.2	A qualitative plot of the frequency response magnitude of the FIR Ramanujan filter $C_9^{(l)}(e^{j\omega})$, when l is finite [194].	200
7.3	Four stages of the proposed periodicity-aware denoising framework that uses Ramanujan analysis filter bank and adaptive synthesis dictionary. See Sec. 7.3 for details.	203
7.4	Comparison of different denoising methods for a period-12 signal. See Sec. 7.3.4 for details and discussion. (a) Original signal $x(n)$, (b) noisy version $y(n)$, and (c)-(j) denoised versions $\hat{x}(n)$ using various methods.	206
7.5	Comparison of SNR gains offered by different denoising methods averaged over 1000 Monte-Carlo runs consisting of (a) period-12 signals and (b) sum of period-7 and period-12 (effective period 84) signals.	208
7.6	Period vs. time plane plot [63], [64] of a period-18 signal, produced by (a) RFB and (b) Capon-optimized filter bank.	210
7.7	Magnitude response of the Capon-optimized filter \mathbf{h}_{10} closely approximates the Ramanujan filter $C_{10}^{(l)}(z)$ in the passbands and puts zeros very close to the other frequency components present in the signal (denoted with black vertical lines).	212
7.8	Total output energies for each filter in RFB and Capon-optimized banks. The Capon-optimized filter bank does not produce spurious peaks for higher period filters, thus making a good choice of threshold easy.	213
7.9	Period vs. time plane plot after denoising the period-18 signal considered in Fig. 7.6. Denoising and plotting Period vs. time plane plots using (a) RFB and (b) Capon-optimized filter bank.	214
7.10	Comparison of SNR gains obtained by the hybrid denoising system when the analysis bank is (i) RFB and (ii) Capon-optimized. The plots are obtained by averaging the performance over 1000 Monte Carlo runs for period-12 signals.	215
7.11	Frequency domain representation of a signal $x(n) \in S_q$. Decimated signal $x(Mn)$ still belongs to S_q with permuted α_k s if and only if $(M, q) = 1$. See Lemma 8.	218
7.12	A signal $x(n) \in S_q^{(p)}$ has its Fourier transform concentrated around the coprime frequencies $2\pi k/q$ with $(k, q) = 1$	221

7.13	Coprime decimation approximately preserves the average per-sample energy of the filter outputs for both RFB (top) and Capon-optimized filter bank (bottom).	224
7.14	Initial stages of proposed periodicity-aware denoising framework that uses Capon-optimized filters and coprime decimation. The later stages of forming a pruned synthesis dictionary and denoising by solving the optimization problem are the same as in Fig. 7.3.	225
7.15	Comparing the proposed denoising method that uses coprime decimation with other denoising methods. SNR gains averaged over 1000 Monte-Carlo runs. (a) Period-15 signals, decimation ratio M_q^s (Eq. 7.46) (b) period-10 signals, decimation ratio M_q^{10} (Eq. 7.47).	227
7.16	Comparing the effect of decimation values and signal period on denoising performance when denoising with (a) Capon-optimized filter bank and ‘with-divisors’ dictionary (b) RFB and ‘with-divisors’ dictionary. SNR gains averaged over 1000 Monte-Carlo runs.	228
7.17	(a)-(f) Effect of hyperparameters on average SNR gain of 1000 signals having periods from 6 to 15 and SNR 0 dB. The default values of the parameters are as follows: threshold = 0.5, $\lambda = 0.01$, $l = N_q/q = 3$, $P_{max} = 40$, and smallest coprime decimation M_q^s (Eq. (7.46)). One of the parameters is varied in each sub-figure, keeping the others fixed.	232
7.18	Probability of correct (a) period estimation (b) repetition index estimation for 1000 signals with periods from 6 to 15.	233
7.19	Separating period-7 and period-12 components from a zero-mean period-84 signal with the proposed denoising framework that uses the Capon-optimized bank with the smallest coprime decimation M_q^s and ‘with-divisors’ dictionary. (a) Noisy signal, (b)-(d) noiseless signal and its components, (e)-(g) denoised signal and its separated components according to Eq. (7.48).	235
7.20	Denoising an ECG recording of a normal sinus rhythm. (a) Original ECG (resampled), (b) noisy ECG, and (c) denoised ECG.	237
8.1	An interpolation filter. Here the filter has impulse response $c_q(n)$ which is the q th Ramanujan sum.	242
8.2	Ramanujan multirate synthesis filter bank for signals satisfying $y(n) = y(n + q)$. Here q_i are the D divisors of the integer q , and M_i are coprime to q_i , that is, $(M_i, q_i) = 1$	244

8.3	Figure for Lemma 12. When $(M, q) = g$, the output spaces \mathcal{V}_M and \mathcal{V}_g are identical.	246
8.4	Zeros of $C_q(z)$ for some selected values of q plotted on the z -plane.	253
8.5	Zeros of $C_7^{(l)}(z)$ for $1 \leq l \leq 6$. As l increases, only the number of zeros on the unit circle increases, whereas the zeros inside the unit circle stay the same.	256
8.6	(a) Standard implementation of two filters from RFB, (b) Equivalent efficient implementation, by extracting a common factor $P(z)$	257
9.1	Variation of dictionary similarities with SNR and datapoints. (a) Ramanujan Dictionary, and (b) Farey dictionary.	266
9.2	Variation of representation errors with SNR and datapoints for Ramanujan dictionary. (a) Error e_1 , and (b) error e_2	267
9.3	Variation of (a) dictionary similarity and (b) representation error with sparsity and maximum sparsity.	267
9.4	Variation of (a) dictionary similarity and (b) representation error with the allowed number of dictionary columns.	268
9.5	Variation of (a) dictionary similarity and (b) representation error with the maximum period.	268
9.6	Variation of (a) dictionary similarity and (b) representation error with signal length.	269
9.7	Variation of dictionary similarities for different K-SVD versions and initialization. (a) Ramanujan dictionary, and (b) Farey dictionary.	269
9.8	Effect of initialization on representation error e_2 for Ramanujan dictionary.	270
9.9	The CNN network structure used in [149]. The output size of each layer is maintained to be L by appropriate zero padding before performing convolutions, where L is the number of discretized directions considered (see Eq. (9.12)).	272
9.10	Comparison of different proxy spectra for a two-source scenario from training data. The original proxy has a non-zero ‘floor’, whereas pseudoinverse, DFT, and pseudo-DFT proxies have ringing patterns.	274
9.11	Comparison of different proxy spectra for a seven-source scenario used for testing (where the number of sensors is only 6). The peaks in original proxy (from [149]) are much harder to identify in this case.	275
9.12	Training mean squared losses for different input proxy spectra	277
9.13	Test mean squared losses for different input proxy spectra	278

- 9.14 Estimated spectra for a seven-source scenario, with a 6-sensor nested array. True DOAs are denoted in red diamonds. Original proxy (9.26) and Hermitian proxy (9.27) do not identify all the sources, whereas all other proposed proxies correctly identify the sources. 279
- 9.15 Estimated DOAs (shown in dots) with all proxies track the true DOAs (solid lines) well. The two sources present in the six segments are separated by approximately 5.5° , 13.5° , 20.67° , 50° , 60° , and 70° . . . 280
- 9.16 DOA estimation errors with different proxies obtained from the above plot. Some large outliers are not seen due to the restricted Y-axis range. 280

LIST OF TABLES

<i>Number</i>	<i>Page</i>
4.1 Key criteria considered for sparse array design, their drawbacks, and modifications considered in this chapter.	94
4.2 Coarray properties of the weight-constrained sparse arrays with $O(N)$ aperture proposed in Sec. 4.3. Array aperture (A), coarray weights $w(l)$, ULA segment in the coarray, coarray holes, and maximum number of identifiable DOAs with coarray-MUSIC D_m are compared. ‘ A ’ in the second last column refers to the aperture listed in the second column. Array \mathbf{z}_6 has $w(1) = w(2) = 0$ and constant $w(3)$ and $w(4)$ independent of N	102
4.3 Coarray properties of the proposed weight-constrained nested arrays with $O(N^2)$ aperture proposed in Sec. 4.4. For a given number of sensors N , the optimal values of N_1 and N_2 for nested array are $N_1 = \text{round}((N + 1)/2)$, and $N_2 = N - \text{round}((N + 1)/2)$ [34]. We assume $N_1 \geq 5$. The optimal values of N_1 and N_2 for $\mathbf{w}_1^{(1)}$ and $\mathbf{w}_1^{(2)}$ are given by Eq. (4.22), and that for \mathbf{w}_2 are given by Eq. (4.26). The maximum number of identifiable DOAs with coarray-MUSIC is $\lfloor L/2 \rfloor$	107
4.4 Properties of arrays with $N = 16$. The last 9 arrays (\mathbf{z}_1 to \mathbf{w}_2) are the new proposed arrays. Coupling leakage \mathcal{L} is calculated with $c = 0.3$ and $B = 10$. MSE for estimating 22 DOAs is the smallest for arrays with $w(1) = 0$ (see Sec. 4.5.1).	114
4.5 Properties of arrays when $A \leq 100$ and $N \leq 50$. Coupling leakage \mathcal{L} is calculated with $c = 0.3$ and $B = 10$. Many of the proposed $O(N)$ aperture arrays have more sensors but smaller \mathcal{L} compared to the $O(N^2)$ aperture arrays from the literature.	121
5.1 Comparison of computation times of different DOA estimation methods applied to array \mathbf{z}_5 with $N = 16$, averaged over 400 Monte-Carlo runs. The runtime of coarray root-MUSIC using a one-sided ULA segment in the coarray is taken as the reference value (x).	149
7.1 Values of the hyperparameters (filter bank threshold and regularization level λ) selected by joint search over the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ respectively for 1000 random signals with periods from 6 to 15 when using the RFB in the first stage of the proposed denoising framework.	231

7.2	Values of the hyperparameters (filter bank threshold and regularization level λ) selected by joint search over the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ respectively for 1000 random signals with periods from 6 to 15 when using the Capon-optimized filter bank in the first stage of the proposed denoising framework.	231
9.1	Average absolute errors and mean squared errors obtained with different input proxies	279

Chapter 1

INTRODUCTION

A key aspect of many signal processing systems is transforming signals from their natural domain (typically the time domain or space domain) into an alternate representation domain. Representing signals in such an alternate domain can be helpful in enhancing intuitive understanding and also simplifying their processing. The ‘transforms’ that convert signals from one domain into another typically do so by representing the signals using a new set of basis functions. One of the most widely used tools for such signal transformations is the Fourier transform. It converts the time-domain or space-domain signals into the frequency domain by representing them as a linear combination of a set of complex exponentials as basis functions. The complex exponential basis functions are parametrized by the ‘frequency’ variable ω . The frequency-domain representation of signals enables several important tasks in signal processing, such as filtering, noise reduction, compression, and parameter extraction.

When a signal is composed of a finite number of discrete frequency components, its frequency spectrum is characterized by distinct ‘lines’ at specific frequencies, and is called a **Line Spectrum**. Signals having line spectra are also commonly referred to as sum-of-sinusoids because they can be represented as the sum of finitely many sinusoids (or, complex exponentials) with distinct frequencies. Mathematically, a discrete-time line spectrum signal can be expressed as

$$x(n) = \sum_{i=1}^M A_i e^{j\omega_i n}, \quad (1.1)$$

where ω_i are discrete-time frequencies, n is discrete-time index, and A_i are the complex amplitudes. The discrete-time Fourier transform (DTFT) of such a signal is shown in Fig. 1.1. Line spectrum signals frequently arise in various applications, including direction of arrival (DOA) estimation using sensor arrays [1], [2], bearing and range estimation in synthetic aperture radar (SAR) [3], channel estimation in wireless communications [4], pitch estimation in speech processing [5], [6], detecting periodicity structure in repeating signals [7], power electronics [8], and simulation of atomic systems in molecular dynamics [9]. These applications rely on accurately estimating the discrete frequencies that make up the signal, making line

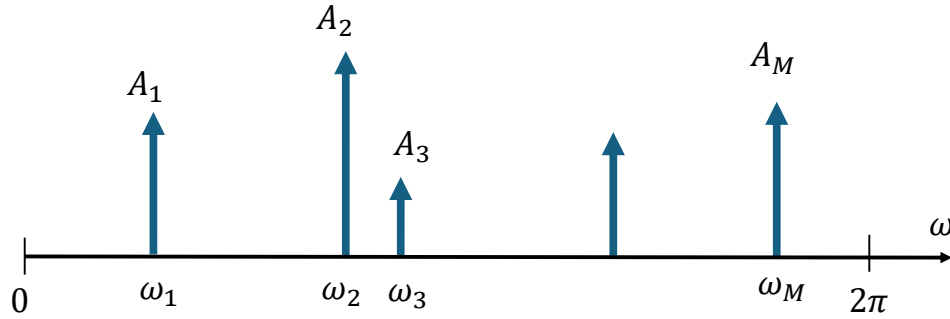


Figure 1.1: A representative example of the DTFT of a general line spectrum signal from Eq. (1.1).

spectrum estimation an essential tool in signal processing. Some of the techniques proposed in the literature for line spectrum estimation include classical subspace-based algorithms such as multiple signal classification (MUSIC) [10] and estimation of signal parameters via rotational invariance techniques (ESPRIT) [11], maximum likelihood based method [12], sparsity-based and dictionary-based algorithms [13], [14], compressed sensing [15], and atomic-norm based methods [16], [17].

This thesis focuses on two specific scenarios involving line spectrum estimation: i) DOA estimation using sensor arrays, and ii) denoising of discrete-time periodic signals. Both scenarios share a common characteristic: the line spectral structure of the underlying signals, but they arise in distinctly different contexts with different objectives. DOA estimation is typically used in applications where detecting the location of a target is essential, and the data is collected using sensor arrays that spatially sample incoming waves. In contrast, periodic signal denoising is primarily concerned with time-domain signals, where the goal can be to remove noise from periodic structures (such as those found in electrocardiograms (ECG) [18] or gravitational waves [19]) to gain a clearer understanding of the physical phenomena that generate the signals. Examples of the line spectra arising in the context of DOA estimation and periodic signals are shown in Fig. 1.2. For DOA estimation, the line spectrum is in the angular domain, whereas a periodic signal exhibits a line spectrum in the frequency domain.

In the rest of this chapter, we provide a brief introduction to sensor arrays, DOA estimation, and denoising of periodic signals. We then outline the new contributions of the thesis. The outline of the chapter is as follows. In Sec. 1.1, we discuss the important applications of sensor arrays. Sec. 1.2 reviews the passive array output equation. Sec 1.3 discusses periodic signals and the role of Ramanujan sums in

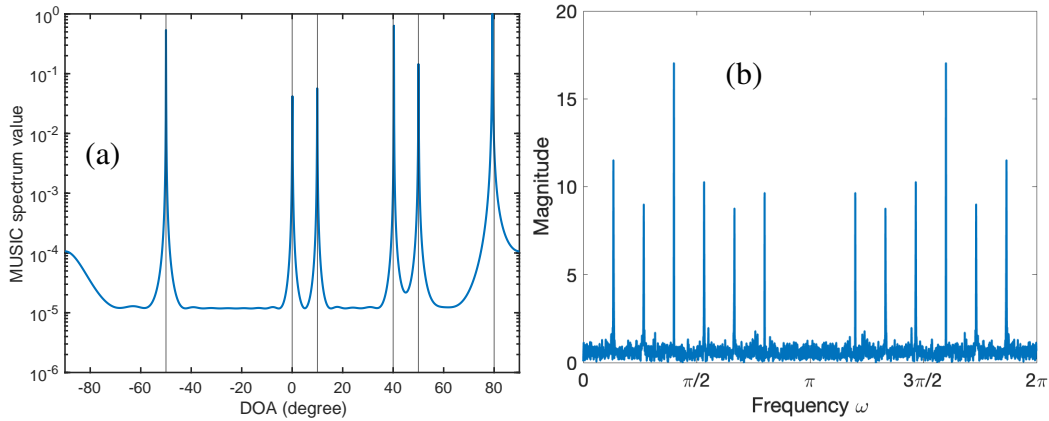


Figure 1.2: Examples of line spectra arising in the context of DOA estimation and periodic signals. (a) A MUSIC spectrum (see Sec. 2.2) showing line spectrum in the angular domain in the context of DOA estimation. (b) A DTFT of a noisy periodic signal showing a line spectrum in the frequency domain.

analyzing periodicity. Sec. 1.4 motivates the use of learned transformations for signal representation. Sec 1.5 provides a chapter-wise overview and contributions of this thesis. Lastly, Sec. 1.6 and Sec. 1.7 define the notations and acronyms used in this thesis.

1.1 Sensor Arrays

Sensor arrays are composed of multiple sensors that sample incoming waves from various directions. The signals received at these sensors are then processed collectively. This joint processing enables the extraction of information that a single sensor alone would not be able to provide. An important application of sensor arrays is to determine the directions of arrivals (DOAs) of sources impinging on the arrays, to localize the sources, or even to track moving sources over time. Another important application of sensor arrays is to achieve direction-dependent beamforming gain to improve the signal-to-noise ratio (SNR). A beamformer can amplify signals from specific directions and attenuate signals from certain others. The field of array signal processing has evolved over several decades to develop efficient signal processing techniques for DOA estimation and beamforming [1], [2].

DOA estimation and beamforming play crucial roles in numerous application areas. In wireless communication and cellular networks, MIMO (multiple input multiple output) systems employ transmitting and receiving arrays to enhance channel capacity, reduce interference, and facilitate efficient communication [20], [21]. In radar and sonar systems, DOA estimation is essential for target detection, tracking,

and localization [22], [23]. In radio astronomy, antenna arrays gather signals from distant celestial objects and improve image resolution by creating large synthetic apertures [24]. In seismology, array signal processing is used to detect seismic waves, locate earthquakes, and study geological structures [25], [26]. Microphone arrays are used in acoustical array signal processing to enhance speech signals and reduce noise [27], [28]. Furthermore, array signal processing finds applications in medical imaging techniques, such as ultrasound and electroencephalogram (EEG) [21], [29]. More recently, in self-driving cars, sensor arrays and automotive radar are used for mapping and autonomous navigation, supplementing other detection modalities such as cameras and LIDAR [30].

1.2 Array Output Equation

We consider a linear array \mathbf{z} consisting of N sensors placed at locations $z_1, z_2, \dots, z_{N-1}, z_N$ that sample the electromagnetic waves impinging on them. This array is represented as

$$\mathbf{z} = \begin{bmatrix} z_1 & z_2 & \cdots & z_{N-1} & z_N \end{bmatrix}. \quad (1.2)$$

In some situations, it is mathematically convenient to assume that the leftmost sensor is placed at the origin. In this case, the sensors are assumed to be placed at locations $0, z_1, \dots, z_{N-2}, z_{N-1}$, and the array is represented as

$$\mathbf{z} = \begin{bmatrix} 0 & z_1 & \cdots & z_{N-2} & z_{N-1} \end{bmatrix}. \quad (1.3)$$

The convention for the array representation that we use will be clear from the context. We assume that the sensors are omnidirectional and arranged in a linear configuration, as illustrated in Fig. 1.3. The aperture A of the array is defined as the distance between the leftmost and rightmost sensors:

$$A = z_N - z_1. \quad (1.4)$$

There are D electromagnetic waves impinging on the array, which are emitted by D far-field sources. Because of the far-field assumption, the impinging waves can be well approximated as plane waves. The sources are assumed to be monochromatic with wavelength λ . All sources are assumed to lie in a plane. The angles of the incoming plane waves measured from the array normal are called directions of arrivals (DOAs) and are denoted by $\theta_1, \theta_2, \dots, \theta_D$. DOAs are assumed to be in the range $[-90^\circ, 90^\circ]$.

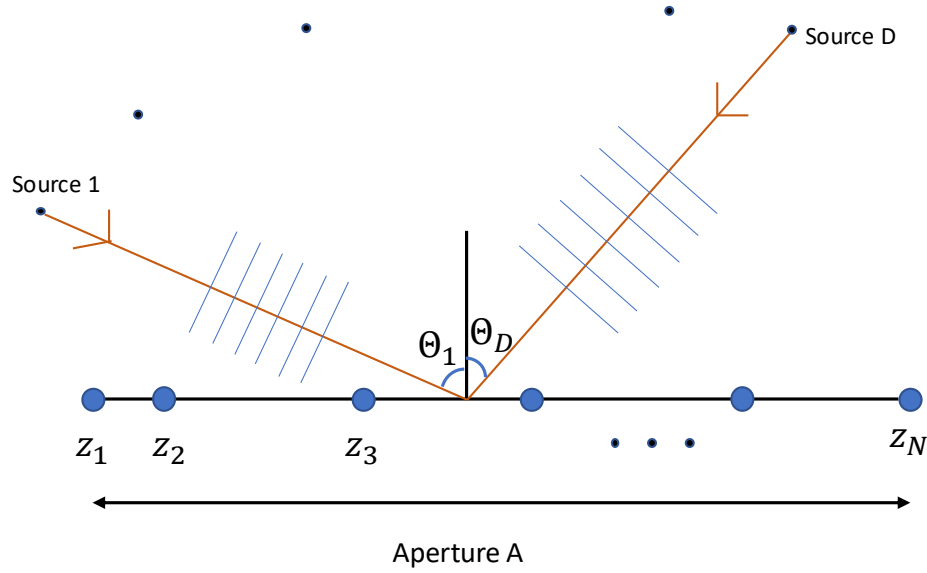


Figure 1.3: Standard model of a passive linear array of N sensors receiving plane waves from D far-field sources at angles $\theta_1, \theta_2, \dots, \theta_D$ from the array normal.

Traditionally, the sensors are placed at locations that correspond to integer multiples of a unit separation distance d :

$$z_i = n_i \cdot d, \quad (1.5)$$

where n_i are integers. The uniform linear array (ULA) is most commonly used in array signal processing, and has sensors placed at locations $n_i = i - 1$. Now, if the amplitude from the i -th source is s_i , the output produced at N sensors by this single source is proportional to [1]

$$s_i \cdot \underbrace{\begin{bmatrix} e^{j\omega_i n_1} \\ e^{j\omega_i n_2} \\ \vdots \\ e^{j\omega_i n_N} \end{bmatrix}}_{\mathbf{a}(\omega_i)}, \quad (1.6)$$

where $\omega_i = 2\pi d \sin \theta_i / \lambda$, and

$$\mathbf{a}(\omega_i) = \begin{bmatrix} e^{j\omega_i n_1} & e^{j\omega_i n_2} & \dots & e^{j\omega_i n_N} \end{bmatrix}^T \quad (1.7)$$

is called the steering vector corresponding to direction ω_i . When there are D sources, the array output is given by

$$\mathbf{x} = \sum_{i=1}^D s_i \mathbf{a}(\omega_i) + \mathbf{n} = \mathbf{A}\mathbf{s} + \mathbf{n}. \quad (1.8)$$

Here, the vector $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_D]^T \in \mathbb{C}^D$ contains the source amplitudes and $\mathbf{n} \in \mathbb{C}^N$ is the additive random noise. The array manifold matrix \mathbf{A} contains steering vectors corresponding to D source directions:

$$\mathbf{A} = [\mathbf{a}(\omega_1) \ \mathbf{a}(\omega_2) \ \dots \ \mathbf{a}(\omega_D)]. \quad (1.9)$$

If the unit distance d is larger than the half-wavelength distance $\lambda/2$, there is spatial aliasing [1]. So typically, d is taken to be $\lambda/2$, and the relationship between the physical DOA θ_i and the corresponding ω_i becomes

$$\omega_i = \pi \sin \theta_i. \quad (1.10)$$

As the DOAs θ_i lie in the range $[-90^\circ, 90^\circ]$, ω_i belong to the range $[-\pi, \pi]$. Note that $\omega = \pi \sin \theta$ is an invertible map when θ is in the range $[-90^\circ, 90^\circ]$, and hence there is a one-to-one correspondence between the physical DOA domain θ and the parametric domain ω . Thus, we interchangeably refer to both θ_i and ω_i as the DOAs. Given the array output vector \mathbf{x} , the information about sources, such as DOAs, signal powers, and the number of sources D , can be estimated.

Throughout this thesis, we make a normalizing assumption $d = \lambda/2 = 1$, without loss of generality [1]. Arrays that have sensors placed on a grid with unit separation $d = 1$ are called **integer arrays**. Integer arrays that are not ULAs are called sparse integer arrays, or simply **sparse arrays**. Most of the array signal processing literature focuses on integer arrays. However, we will point out in Chapter 3 that the integer constraint on sensor locations can be too restrictive, especially when a given number of sensors N are to be placed in an available aperture A . This can prevent optimal placement of the sensors, leading to a higher DOA estimation error. To overcome this drawback, we propose to use rational arrays in Chapter 3.

1.2.1 Difference Coarray

For an integer array $\mathbf{z} = [n_1 \ n_2 \ \dots \ n_{N-1} \ n_N]$, the difference coarray is defined as the set of all pairwise differences in the sensor locations:

$$\mathbb{D}_{\mathbf{z}} = \{n_i - n_j\}_{i,j=1}^N. \quad (1.11)$$

The difference coarray is symmetric, and $\mathbb{D}_{\mathbf{z}}^+$ is used to represent the set of non-negative entries in $\mathbb{D}_{\mathbf{z}}$. The coarray weights are defined as

$$w(l) = |\{(n_i, n_j) \text{ s.t. } n_i - n_j = l\}|, \quad (1.12)$$

where $|\cdot|$ denotes the cardinality of the set. The l -th coarray weight $w(l)$ denotes the number of sensor pairs with distance l between them. In discrete mathematics, a related concept of difference basis has been studied widely [31], [32]. The set of integers $\{n_1, n_2, \dots, n_N\}$ is a difference basis with respect to an integer L if the difference coarray contains all integers from $-L$ to L . We will see in Chapter 2 that such a uniform segment in coarray is of importance for DOA estimation with sparse arrays. Many well-known sparse arrays, such as minimum redundancy arrays (MRAs) [33], nested arrays [34], and coprime arrays [35], have a large segment of $O(N^2)$ consecutive lags present in the difference coarray, and such arrays are well-suited to estimate the DOAs using coarray-domain processing, as we will see.

1.2.2 Effect of Mutual Coupling Between Sensors

When sensors have electromagnetic coupling between them, the array output does not follow the standard model in Eq. (1.8). This is called the mutual coupling effect, and it is challenging to precisely model mathematically [36]. In practice, it is usually unknown to the user. In numerical simulations, the effect of coupling on the array output is typically modeled as [37]–[39]:

$$\mathbf{x} = \mathbf{C}\mathbf{A}\mathbf{s} + \mathbf{n}, \quad (1.13)$$

where \mathbf{C} is the coupling matrix. The structure of the coupling matrix depends on several factors, as discussed later in Sec. 2.4. Mutual coupling can severely degrade the accuracy of DOA estimation [1], [39]. Sparse arrays are better at reducing the impact of mutual coupling because the sensors are farther apart from each other on average, compared to a ULA. However, many of the existing sparse arrays from the literature may still not be robust to the presence of strong mutual coupling, as we will demonstrate in Chapter 4. Thus, we propose unconventional sparse arrays that are better at mitigating the effect of mutual coupling compared to the known sparse arrays. For such unconventional array geometries, we also propose improved DOA estimation algorithms in Chapter 5.

1.3 Periodic Signals and Ramanujan Sums

Periodicity is one of the most fundamental and pervasive patterns that occur in nature, manifesting across different phenomena ranging from the oscillating pendulum to the rhythmic cycles of biological processes. In the context of signal processing, periodic signals are particularly important as they serve as the foundation for analyzing many naturally occurring signals. Many real-world signals, such as electromagnetic waves, electrical signals, and even biological signals such

as ECG, exhibit periodic characteristics, though often with added noise or irregularities. The periodic component of a signal often carries important information about the underlying phenomenon, such as the frequency of a heartbeat in an ECG or the pitch in a segment of tonal music. Understanding the structural properties of periodic signals, such as their frequency, amplitude, and phase, can provide insights into the system being observed.

Periodic signals also occur naturally as discrete-time sequences. For example, certain biological sequences, like tandem repeats in DNA, exhibit periodic segments which can be indicative of genetic disorders or mutations [40]–[42]. Similarly, protein repeats in amino acid sequences follow periodic patterns that are crucial in determining the structural and binding properties of proteins [43], [44], which are fundamental to cellular processes. The study of these periodic patterns in biological systems has implications for fields such as genetics, bioinformatics, and molecular biology, where understanding the timing and repetition of sequences can reveal insights into genetic codes and molecular structures.

Mathematically, a discrete-time signal $x(n)$ is periodic if there exists an integer P such that

$$x(n + P) = x(n) \quad \forall n \in \mathbb{Z}. \quad (1.14)$$

Any such P is called a repetition index, and the smallest positive repetition index is called the period of $x(n)$. Most traditional methods for estimating component periods in periodic signals rely on the Fourier transform to analyze the non-zero frequency components in the signal [45]–[48]. The drawbacks of these methods for discrete-time signals are highlighted in Chapter 1 of [49], especially when the signal length is short and the period is not a divisor of the signal length.

1.3.1 Ramanujan Sums

Several authors have shown that Ramanujan sums can be used for signal processing applications arising in the context of periodic signals [50]–[59]. Ramanujan sums [60] were introduced by mathematician Srinivasa Ramanujan in 1918, and they exhibit several beautiful number-theoretic properties. For every integer $q > 0$, the q^{th} Ramanujan sum is defined as

$$c_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^q e^{j2\pi kn/q} = \sum_{\substack{k=1 \\ (k,q)=1}}^q W_q^{-kn} = \sum_{\substack{k=1 \\ (k,q)=1}}^q W_q^{kn}. \quad (1.15)$$

Here,

$$W_q = e^{-j2\pi/q} \quad (1.16)$$

is a q -th complex root of unity, and $(k, q) = 1$ means that k and q are coprime integers. Ramanujan showed that these sequences have several interesting properties [60]:

1. The q -th Ramanujan sum $c_q(n)$ is periodic with period exactly equal to q (and not a proper divisor of q).
2. Although defined in terms of the powers of the complex root of unity, $c_q(n)$ is always real and integer-valued. The fact that $c_q(n)$ is real can be easily deduced, but $c_q(n)$ being integer-valued is a surprising property.
3. For any distinct integers p and q , the corresponding Ramanujan sums $c_q(n)$ and $c_p(n)$ are orthogonal over a length that is a common multiple of p and q .
4. Many arithmetic functions, such as the divisor function and the Euler totient function [61], can be expressed as functions of Ramanujan sums.

A comprehensive analysis of Ramanujan sums in the context of signal processing for detecting periods in a discrete-time signal was developed in [52] and [62]. Ramanujan sums were generalized to Ramanujan subspaces \mathcal{S}_q , and dictionary approaches were also developed for detecting periods. A generalized framework of nested periodic dictionaries is presented in [53]. Ramanujan periodicity transform (RPT) is shown to be useful in applications such as robust detection of brain stimuli for brain-computer interfaces [54]–[56] and removal of interference from ECG signals [57]–[59]. Ramanujan filters and Ramanujan filter banks (RFB) were first proposed in [63] and further developed in [64]. An overview of developments related to Ramanujan subspaces, filters, and dictionaries can be found in [65].

In Chapter 7, we will introduce a denoising method for periodic signals based on Ramanujan dictionaries and Ramanujan filter banks, and prove several multirate properties of Ramanujan sums. We will also propose an interpolation filter model for Ramanujan subspace signals and provide a characterization of zeros of Ramanujan filters in Chapter 8.

1.4 Learned Transformation for Signal Representation

The Fourier transform and the frequency-domain representation may not always provide the most suitable representation for every type of signal. Depending on the specific characteristics of the signal and the intended application, several alternative transformations are frequently used in signal processing. For example, the discrete

cosine transform (DCT) is particularly effective for natural images, especially in image compression tasks [66]. Similarly, the Haar transform and more general wavelet transforms are useful for multi-resolution analysis of signals [67].

In certain applications, it can be beneficial to use a transformation specifically designed for the signals and tasks at hand, rather than relying on standard transformations like the Fourier transform or DCT. For example, principal component analysis (PCA) [68] is a widely used method for efficiently representing high-dimensional data. PCA is closely related to the Karhunen-Loève transform (KLT) [69], which learns to represent signals using a few orthogonal principal components that capture the most variability in the data. This learned transformation enables efficient data representation and is commonly applied in dimensionality reduction and other related applications.

Unlike the Fourier transform, DCT, and PCA, which all provide invertible transformations, it is also possible to learn an overcomplete set for representing a class of signals. The key advantage of this approach is that signals can be expressed as *sparse linear combinations* of the components in the learned representation set. A widely used method for learning such overcomplete representation is dictionary learning [70]–[73]. There are several different dictionary learning techniques presented in the literature, and K-SVD [74] is one of the most popular algorithms. It has demonstrated effectiveness across a range of applications [75], [76]. In the first half of Chapter 9, we will explore the role of dictionary learning methods for the representation of periodic signals.

In the last decade, the use of machine learning and deep learning has become widespread across various fields [77] for representing and processing signals and data. Deep neural networks and other network architectures can be trained for specific tasks, allowing them to inherently learn useful features from the data that are difficult to capture analytically. While the universal function approximation capabilities of neural networks have been known for decades [78]–[81], with the recent advancements in hardware and algorithms for fast computations, the use of deep networks has become practical. Instead of using DCT coefficients as features for the image compression task, neural networks such as autoencoders [82] can learn to compress and reconstruct images effectively, and can also be tailored to a specific class of images. In the second half of Chapter 9, we adapt a convolutional neural network (CNN) based DOA estimation method for sparse arrays by proposing various alternative inputs to the network.

1.5 Contributions and Outline of This Thesis

With the aforementioned background, this thesis primarily focuses on developing novel sensor array geometries, algorithms, and theoretical results for processing line spectrum signals, with applications in DOA estimation and periodic signal denoising. Chapter 2 provides a brief review of sparse arrays and DOA estimation algorithms. In Chapter 3, we introduce non-integer arrays for DOA estimation and demonstrate their advantages over traditional integer arrays. Chapter 4 presents new weight-constrained sparse arrays, which are particularly useful for DOA estimation under high mutual coupling and aperture constraints. We also develop new algorithms to enhance the performance of these weight-constrained arrays in Chapter 5. In Chapter 6, we elaborate on certain lesser-known defects of coarray-domain DOA estimation, and also explore a few ways to combat them. Following this, Chapter 7 proposes a framework for periodicity-aware signal denoising using Ramanujan filter banks and dictionaries. In Chapter 8, we develop some further theoretical properties of Ramanujan filters. In Chapter 9, we examine two widely used learning-based methods for DOA estimation and periodic signal representation. Finally, Chapter 10 concludes the thesis and provides directions for future research.

1.5.1 Review of Sparse Arrays and DOA Estimation Algorithms (Chapter 2)

In this chapter, we provide a brief summary of sparse sensor arrays and the subspace-based DOA estimation algorithms that will be utilized throughout the thesis. We also explain the importance of the difference coarray and coarray-domain DOA estimation algorithms. In later chapters (Chapters 3, 4, and 5), we will adapt these algorithms appropriately for the new array configurations that we propose. We also discuss the practicality of the mutual coupling model used in array signal processing. Next, we explain that the coarray-MUSIC is observed to be asymptotically inefficient for sparse arrays. This means that the coarray-MUSIC mean squared error (MSE) is larger than the Cramér-Rao bound (CRB) even for a large SNR and a large number of snapshots. Later in Chapter 6, we explore different ways of constructing the coarray-domain covariance matrices and experimentally analyze their effect on the efficiency of DOA estimation.

1.5.2 Non-Integer Arrays for Direction of Arrival Estimation (Chapter 3)

Linear arrays with sensors positioned at integer locations (in multiples of the half-wavelength distance $d = \lambda/2$) are widely used in array signal processing. However, the requirement for sensor locations to be integers can be overly restrictive in appli-

cations where a limited number of sensors are to be placed within a fixed aperture. To address this limitation, this chapter explores arrays where the sensor locations can be rational numbers. It is shown that rational arrays possess several important advantages over integer arrays, primarily by offering greater design flexibility. In particular, we introduce sparse and coprime rational arrays, and provide general array constructions that are applicable for any given number of sensors N and available aperture A . To demonstrate the theoretical identifiability properties of these proposed arrays, we first extend integer number-theoretic concepts, such as the greatest common divisor and coprimality, to rational numbers. Using these number-theoretic extensions, we provide results regarding the steering vector invertibility of general rational arrays and the unique identifiability of rational coprime arrays. Some of the theoretical results can be further extended to arbitrary arrays where sensor locations can even be irrational.

By adapting the standard DOA estimation algorithm, such as root-MUSIC, for proposed rational arrays, we demonstrate their advantages over traditional integer arrays when limited sensors are to be placed in a given aperture. Extensive Monte-Carlo simulations reveal that rational arrays can approach the CRB even at low SNR and with a limited number of snapshots, outperforming integer arrays. Furthermore, rational arrays are observed to be better equipped for resolving closely spaced sources, and they also have smaller MSE for most two-DOA configurations. Additionally, we show that when signals are coming from a confined angular sector, rational arrays can leverage this information more effectively, allowing for sensor placement over a larger aperture while retaining unique identifiability properties.

Lastly, we highlight another advantage of rational arrays by examining their difference coarrays. While integer arrays, including nested and coprime arrays, are well-known for their capability to identify $O(N^2)$ uncorrelated sources using N sensors in the difference coarray domain, they typically require a sufficiently large aperture to do this. In contrast, rational arrays can identify $O(N^2)$ uncorrelated sources even with a limited aperture. We demonstrate that by adding a few sensors at non-integer locations within an otherwise integer array, it is possible to introduce a significant number of fractional lags at which autocorrelation can be estimated. Furthermore, carefully designed sparse integer arrays can be scaled to create rational arrays that fit within the available aperture and have a large ULA segment in coarray at rational locations. To illustrate these benefits, we provide Monte-Carlo simulations and address practical challenges associated with reducing the aperture

size, such as the increased mutual coupling.

1.5.3 Weight-Constrained Arrays For DOA Estimation Under High Mutual Coupling and Aperture Constraints (Chapter 4)

Following the development of nested arrays and coprime arrays, several improved sparse array constructions have been proposed in the literature to identify $O(N^2)$ directions with N sensors. These constructions typically aim to maximize the degrees of freedom (DOFs) of the arrays and reduce the impact of mutual coupling on DOA estimation. In this chapter, we propose modifications to the traditionally followed sparse array design criteria to better mitigate the effect of mutual coupling and reduce the required aperture for such sparse arrays. We argue that having $O(N^2)$ degrees of freedom may not be of interest, especially for large N . Furthermore, a large aperture for these arrays may not be practical when space for sensor placement is limited. With the modified array design criteria, we present two types of sparse array designs that can effectively address high mutual coupling by ensuring that the coarray weights satisfy either $w(1) = 0$ or $w(1) = w(2) = 0$. In addition, several other coarray weights are small constants that do not increase with the number of sensors N . The first type of array has an aperture of $O(N)$, making it suitable when the available aperture is constrained and the number of DOAs does not exceed $O(N)$. These arrays are constructed by appropriately dilating a ULA and adding a few additional sensors. Despite having an aperture of $O(N)$ length, these arrays can still identify more than N DOAs. The second type of array possesses $O(N^2)$ degrees of freedom and is suitable when there are no aperture restrictions. These arrays are created by properly dilating a nested array and augmenting it with several additional sensors. We compare the proposed arrays with existing ones in the literature by analyzing their coarray properties and conducting several Monte-Carlo simulations. In the presence of high mutual coupling, the proposed arrays can estimate DOAs with significantly smaller errors than other arrays from the literature, demonstrating the effectiveness of reducing coarray weights at critical small-valued lags.

1.5.4 Improved DOA Estimation Algorithms and Generalized Constructions for Weight-Constrained Arrays (Chapter 5)

The weight-constrained arrays discussed in Chapter 4 have ‘central holes’ in their difference coarrays at locations 1 and 2. The DOA estimation simulations presented in that chapter only utilize the one-sided ULA segment within the coarrays of weight-constrained arrays. To leverage the ULA segments on the positive and negative sides

of the coarray, we propose using two different techniques: i) augmented root-MUSIC and augmented ESPRIT, and ii) covariance interpolation. These techniques can also be applied to other arrays from the literature that have central holes in the coarray, such as the CADiS [83] and cMRA [84].

The proposed augmented root-MUSIC and augmented ESPRIT methods rely on the singular value decomposition (SVD) of an augmented coarray matrix. This augmented coarray matrix is created by stacking the Toeplitz matrices formed by arranging the estimated correlations at lags corresponding to the ULA segments on positive and negative sides in the coarray. The proposed method can still accurately identify up to $\lfloor L/2 \rfloor$ DOAs, where L is the length of a one-sided ULA segment in the coarray. Our numerical simulations show that the augmented root-MUSIC achieves a smaller MSE compared to the standard root-MUSIC that uses only the one-sided ULA segment from the difference coarray. Additionally, we compare the proposed algorithms to other DOA estimation techniques that can utilize the non-central ULA segments in the coarray, including dictionary-based DOA estimation and coarray interpolation. The computation time for the proposed augmented root-MUSIC and augmented ESPRIT is significantly less than that of other optimization-based methods.

The second method of covariance interpolation results in the generation of a larger Toeplitz matrix, which is then used to estimate DOAs using the root-MUSIC or ESPRIT. This approach demonstrates the capability to accurately identify up to twice as many DOAs compared to what can be achieved using only the one-sided ULA segment in the coarray. Additionally, even when the number of DOAs is limited, the DOA estimation error after interpolation is significantly smaller than that obtained using just the one-sided ULA segment in the coarray. Consequently, we can mitigate the disadvantage of having central holes in the coarrays of weight-constrained arrays while still benefiting from their advantages in reducing the impact of mutual coupling on DOA estimation.

In the last part of this chapter, we also propose a generalized construction of weight-constrained sparse arrays with $w(1) = 0$ that additionally satisfy the property that, for a given integer P , the coarray weights $w(l)$ for $2 \leq l \leq P$ are constants that do not increase with the number of sensors N . We prove that the difference coarrays of the proposed arrays contain a large one-sided ULA segment of length $\mathcal{O}(N)$, starting from lag 2. We analyze the effect of the parameter P on coarray properties and coupling leakage values. Numerical simulations indicate that the proposed arrays

exhibit greater robustness to mutual coupling compared to the weight-constrained arrays proposed in the previous chapter. Additionally, the newly proposed arrays can also have smaller DOA estimation errors when compared to well-known sparse arrays from the literature, especially under conditions of high mutual coupling and aperture constraints. Finally, we discuss the limitations encountered when attempting to further generalize weight-constrained arrays (for example, to make $w(1) = w(2) = w(3) = 0$) due to the currently adopted array generation procedure.

1.5.5 On the Efficiency of Coarray-Domain DOA Estimation Algorithms (Chapter 6)

Sparse arrays are widely known for their ability to identify $O(N^2)$ directions of arrival (DOAs) using N sensors through the difference coarray domain. The coarray-MUSIC is based on the eigendecomposition of a Toeplitz matrix formed by ‘direct augmentation’, where the estimated correlations at consecutive lags from the difference coarray are arranged to form the coarray covariance matrix. However, this approach has been found to be inefficient, even under high SNR and with a large number of snapshots. In the first part of this chapter, to understand when and how the coarray-MUSIC loses its efficiency, we consider the simplest case of a ULA when the number of sources D is less than the number of sensors N . Standard (element space) MUSIC already achieves CRB in this case, but we observe that coarray-MUSIC suffers significant performance loss, similar to what has been observed for sparse arrays. We also consider two variations of the coarray-MUSIC algorithm by using tall and fat variations of the coarray covariance matrix. Additionally, we experimentally find that coarray-MUSIC MSE for two-source cases varies in surprising ways with DOA separation. We also demonstrate that perturbing different array output correlations has different effects on coarray-MUSIC MSE. These observations provide valuable insights into the (in)efficiency of coarray-MUSIC and provide directions for further investigations.

In the second part of this chapter, we demonstrate that an alternative way of constructing the coarray covariance matrix can greatly improve the MSE compared to coarray-MUSIC and can lead to efficient DOA estimation. This approach constructs a Toeplitz covariance matrix by solving an optimization problem whose objective is derived using the asymptotic error distribution of the known entries from the covariance matrix. When the number of sources D is less than the number of sensors N , the DOA estimation MSE with this approach does not saturate at high SNR, unlike that of coarray-MUSIC. We experimentally demonstrate that this leads

to efficient DOA estimation for many different sparse arrays. For the other case, when $D > N$ and the CRB saturates at high SNR, the MSE with this Toeplitz matrix approach is found to be close to the CRB. We identify that whitening the error vector using its asymptotic error distribution is a key feature of this method that is crucial for the observed improvement in MSE. Although this Toeplitz matrix construction approach acts as an interpolation algorithm for arrays that have holes in their difference coarrays (such as coprime arrays), this approach is also observed to greatly improve the DOA estimation MSE for arrays that do not have holes in their difference coarrays (such as nested arrays). These observations establish a relatively simpler alternative to some of the recently proposed methods in the literature for efficient DOA estimation using sparse arrays. We also propose a modification to the Toeplitz covariance matrix construction approach to account for the presence of mutual coupling and provide simulations with different sparse arrays.

1.5.6 Periodicity-Aware Signal Denoising Using Capon-Optimized Ramanujan Filter Banks and Pruned Ramanujan Dictionaries (Chapter 7)

In this chapter, we propose a ‘periodicity-aware’ hybrid analysis-synthesis framework for denoising discrete-time periodic signals. Our method utilizes Ramanujan filter banks (RFB) for analysis and dictionaries for synthesis. The synthesis dictionary retains appropriate subspaces for signal reconstruction by pruning the Ramanujan dictionary based on the outputs of the RFB. Unlike existing denoising methods, our proposed approach ensures that the denoised output consists of integer-periodic components with periods smaller than a pre-selected value. The proposed denoising method is observed to be effective even when the signal length is small and demonstrates a high SNR gain across a wide range of input signal SNRs.

Furthermore, we adapt each filter in the analysis bank to the incoming data by optimizing the filter coefficients through a multi-band Capon formulation. Capon beamforming is a popular beamforming technique in array signal processing, and we utilize it here for optimizing Ramanujan filters. This adaptation helps suppress spurious energy peaks generated by higher period filters in the analysis bank, thus enhancing denoising performance. Implementing multiband Capon filters requires the inversion of several autocorrelation matrices. To reduce the required computations, we introduce a method to recursively compute these inverses using Levinson’s recursion from the linear prediction theory.

Next, we prove several multirate properties of Ramanujan subspace signals. An

important property among these is that, after decimation, a q -th Ramanujan subspace signal remains in the q -th Ramanujan space if and only if the decimation rate M is coprime to q . This insight helps reduce the computations required in the analysis phase of the denoising framework by downsampling the filter outputs. We also discuss the applicability of the decimation property to the non-zero bandwidth outputs of Ramanujan filters and Capon-optimized filters. We provide extensive Monte-Carlo simulations that compare different variants of the proposed method with several existing denoising techniques.

1.5.7 Theoretical Properties of Ideal Ramanujan Filters (Chapter 8)

The Ramanujan filter bank used in the first stage of the denoising framework developed in Chapter 7 does not have a known synthesis filter bank counterpart for perfect signal reconstruction. To make progress towards developing such a synthesis filter bank, we theoretically investigate an ideal interpolation filter model for Ramanujan subspace signals in the first part of this chapter. The interpolation filter consists of an expander $\uparrow M$ followed by the ideal q -th Ramanujan filter $C_q(e^{j\omega})$. The output space of this interpolation filter is, in general, only a proper subspace of the q -th Ramanujan subspace S_q . For the special case when M and q are coprime, we prove that the output space is the entire Ramanujan subspace S_q . We also discuss a more general form of this model for the representation of periodic signals, which may have a potential application in denoising periodic signals. When M and q are not coprime, we provide a bound on the dimension of the output space of the interpolation filter. For this general case, we also conjecture that the provided bound in fact equals the dimension of the output space.

In the second part of the chapter, we analyze the locations of zeros of FIR Ramanujan filters that have the first few periods of Ramanujan sums as their filter coefficients. All the zeros of FIR Ramanujan filters are shown to lie on or inside the unit circle in the z -plane by appealing to Lucas' theorem. A convenient factorization appears as a corollary of this result, which is useful to identify common factors between different Ramanujan filters in a filter bank. For certain families of Ramanujan filters, further structure is identified in the locations of zeros of those filters. It is shown that increasing the number of periods of Ramanujan sums in the filter definition only increases zeros on the unit circle in the z -plane. A potential application of these results is that by identifying common factors between different Ramanujan filters, one can obtain efficient implementations of Ramanujan filter banks, as we demonstrate.

1.5.8 Learning-Based Methods for Representing Periodic Signals and DOA Estimation (Chapter 9)

In this chapter, we explore the role of dictionary learning for periodic signal representation and machine learning for DOA estimation. In the first part, we demonstrate that a well-known dictionary learning algorithm, K-SVD [74], can effectively learn Ramanujan and Farey periodicity dictionaries from noisy, sparse coefficient data generated from these dictionaries, without explicitly enforcing any periodicity structure during the learning process. The similarity between the learned dictionary and the original periodicity dictionary highlights the effectiveness of K-SVD in learning an appropriate dictionary from the data without specific application constraints. Additionally, we explore how different parameter selections impact the similarity of the learned dictionary to the underlying Ramanujan or Farey dictionaries. Two variations of K-SVD, along with different initialization methods, are analyzed for their effects on representation quality and denoising performance.

In the second part, we experiment with the role of machine learning for DOA estimation. In the last decade, deep neural networks have increasingly been utilized to enhance DOA estimation performance. We begin by demonstrating how these methods can be applied to sparse arrays, such as nested arrays, by adapting a recent approach based on convolutional neural networks (CNNs). We propose various alternative inputs (proxy spectra) for the network, and our experiments indicate that even small modifications to the input can improve DOA estimation performance without altering the network structure. Furthermore, these experiments show that with the modified input proxy spectra, it is possible to identify more sources than the number of physical sensors, as one would expect with nested arrays. This finding opens up new possibilities for combining coarray principles with machine learning methods for DOA estimation.

1.6 Notations

The notations used in this thesis are defined in this section. Vectors are represented using boldface lowercase letters such as \mathbf{a} , and matrices are represented by boldface uppercase letters such as \mathbf{A} . The notations \mathbf{A}^T , \mathbf{A}^* , and \mathbf{A}^H respectively denote the transpose, complex conjugate, and Hermitian (or transpose conjugate) of the matrix \mathbf{A} . The k -th column of matrix \mathbf{A} is denoted by $\mathbf{A}(:, k)$, and $\mathbf{A}_{i,j}$ or $[\mathbf{A}]_{i,j}$ is the (i, j) -th entry of the matrix \mathbf{A} . The $n \times n$ identity matrix is denoted by \mathbf{I}_n . We use the notations $\mathbf{1}_n$ to denote the $n \times 1$ column vector containing all ones and $\mathbf{0}_n$ to denote the $n \times 1$ column vector containing all zeros. The subscripts in \mathbf{I}_n , $\mathbf{1}_n$, and

$\mathbf{0}_n$ may be omitted when the sizes are clear from the context. The vectorization of matrix $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_n]$ having columns $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ is defined as

$$\text{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}. \quad (1.17)$$

The Kronecker product of an $m \times n$ matrix \mathbf{A} and a $p \times q$ matrix \mathbf{B} is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} [\mathbf{A}]_{1,1}\mathbf{B} & [\mathbf{A}]_{1,2}\mathbf{B} & \dots & [\mathbf{A}]_{1,n}\mathbf{B} \\ [\mathbf{A}]_{2,1}\mathbf{B} & [\mathbf{A}]_{2,2}\mathbf{B} & \dots & [\mathbf{A}]_{2,n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ [\mathbf{A}]_{m,1}\mathbf{B} & [\mathbf{A}]_{m,2}\mathbf{B} & \dots & [\mathbf{A}]_{m,n}\mathbf{B} \end{bmatrix}. \quad (1.18)$$

The resulting matrix is of size $mp \times nq$. The Khatri-Rao product is the columnwise Kronecker product of two matrices. That is, for matrices $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_n]$ and $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_n]$, their Khatri-Rao product is

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ \mathbf{a}_2 \otimes \mathbf{b}_2 \ \dots \ \mathbf{a}_n \otimes \mathbf{b}_n]. \quad (1.19)$$

For a vector \mathbf{v} , $\|\mathbf{v}\|_1$ and $\|\mathbf{v}\|_2$ denote the L_1 and L_2 norms respectively. If the subscript is omitted, the L_2 norm is assumed. $\|\mathbf{v}\|_0$ represented the L_0 pseudonorm. For a Hermitian symmetric matrix \mathbf{A} , $\mathbf{A} \succeq 0$ means that it is a positive semidefinite matrix. The rank of the matrix is denoted by $\text{rank}(\mathbf{A})$. Trace of a matrix is the sum of the entries of the matrix on its principal diagonal, and is denoted by $\text{tr}(\mathbf{A})$. The operation $\text{diag}(\mathbf{A})$ extracts the diagonal of the matrix \mathbf{A} and returns it as a column vector. A diagonal matrix with entries corresponding to those in a vector \mathbf{a} is represented as $\text{diag}(\mathbf{a})$.

Sets are represented in blackboard or calligraphic fonts such as \mathbb{P} or \mathcal{P} . The sets of natural numbers (i.e., positive integers), integers, real numbers, and complex numbers are denoted by \mathbb{N} , \mathbb{Z} , \mathbb{R} , and \mathbb{C} , respectively. \mathbb{R}^N and \mathbb{C}^N denote the N -dimensional real and complex Euclidean vector spaces, respectively. The dimension of a vector space X is denoted by $\dim X$. We use the notation $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ to denote the space formed by the span (i.e., all linear combinations) of the vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$. We also use $\text{span}\{f_1, f_2, \dots, f_n\}$ to denote the space formed by the span of functions or sequences f_1, f_2, \dots, f_n .

The union of two sets \mathbb{P}_1 and \mathbb{P}_2 is denoted by $\mathbb{P}_1 \cup \mathbb{P}_2$. For a set of numbers \mathbb{P} , $\alpha\mathbb{P}$ is the set formed by multiplying each element in \mathbb{P} by α , whereas $\mathbb{P} + \alpha$ is the set formed by adding α to each element in \mathbb{P} . The cardinality of a set \mathbb{P} is denoted by $|\mathbb{P}|$. The notation $[[n_1, n_2]]$ is used to denote the set of integers n s.t. $n_1 \leq n \leq n_2$.

The greatest common divisor (gcd) of two integers a and b is denoted as $\gcd(a, b)$ or simply (a, b) . So $(a, b) = 1$ means that the integers a and b are *coprime*, and have no common factor other than 1. For any positive integer a , $(a, 0) = a$ by definition. $\text{lcm}(a, b)$ denotes the least common multiple of the two integers. The notation $a|b$ means that integer a divides integer b . The notation $(a \bmod b)$ is used to denote the remainder when integer a is divided by integer b . The number of integers b in $1 \leq b \leq a$ satisfying $(a, b) = 1$ is called the Euler totient function [61] and is denoted as $\phi(a)$. The sum of Euler's totient function from 1 to a is denoted by $\Phi(a)$:

$$\Phi(a) = \sum_{i=1}^a \phi(i). \quad (1.20)$$

The largest integer not greater than a real number x is denoted by $\lfloor x \rfloor$, and the smallest integer not smaller than x is denoted by $\lceil x \rceil$. These are called the floor and the ceiling, respectively. Rounding operation $\text{round}(x)$ denotes the integer closest to x .

The imaginary unit is $j = \sqrt{-1}$. The q -th complex root of unity is $W_q = e^{-j2\pi/q}$. The term ‘coprime frequencies’ with respect to an integer q is used to refer to the frequencies $2\pi k/q$, where $0 \leq k < q$, and $(k, q) = 1$. The 2-dimensional complex plane is often referred to as the z -plane in the context of digital signal processing.

The notation $\mathbb{E}(\cdot)$ is reserved for the statistical expectation operation. A random vector \mathbf{p} that is asymptotically normally distributed with zero mean and covariance matrix \mathbf{W} , is denoted as

$$\mathbf{p} \sim \text{AsN}(\mathbf{0}, \mathbf{W}). \quad (1.21)$$

The standard big-O notation of complexity is denoted by $O(\cdot)$. For example, $O(N)$ denotes the asymptotically linear growth and $O(N^2)$ denotes the asymptotically quadratic growth in terms of the parameter N .

1.7 List of Acronyms

The following acronyms are commonly used in this thesis:

- CRB: Cramér–Rao bound

- DFT: discrete Fourier transform
- DOA: direction of arrival
- ESPRIT: estimation of signal parameters via rotational invariance techniques
- EVD: eigenvalue decomposition
- FIR: finite impulse response
- IIR: infinite impulse response
- MRA: minimum redundancy array
- MSE: mean square error
- MUSIC: multiple signal classification
- RFB: Ramanujan filter bank
- SNR: signal-to-noise ratio
- SVD: singular value decomposition
- ULA: uniform linear array
- WCNA: weight-constrained nested array (see Chapter 4)
- WCSA: weight-constrained sparse array (see Chapter 4)

Chapter 2

REVIEW OF SPARSE ARRAYS AND DOA ESTIMATION ALGORITHMS

2.1 Introduction

DOA estimation has been an active area of research for the past several decades [2], and subspace-based DOA estimation algorithms such as MUSIC [10], root-MUSIC [85], and ESPRIT [11] that are based on the eigendecomposition of the covariance matrix of the array output have been widely popular. In recent years, DOA estimation using sparse arrays has gained significant attention [34], [35], [83], [86]–[90]. Well-designed sparse arrays, such as minimum redundancy arrays (MRAs) [33], nested arrays [34], coprime arrays [35], and their numerous generalizations proposed in the literature [39], [83], [91], [92], can identify $O(N^2)$ sources using N sensors through the so-called ‘difference coarray domain’. Coarray domain DOA estimation uses the ‘coarray covariance matrix’ formed either by spatial smoothing of the array output covariance matrix [34] or by the direct augmentation approach [93]. Then, subspace-based DOA estimation algorithms are applied to the coarray covariance matrix. This way of estimating DOAs through the difference coarray domain is called coarray-based DOA estimation. In this chapter, we provide a brief review of the subspace-based DOA estimation algorithms, the fundamentals of sparse arrays, and coarray-based DOA estimation. The known results regarding the CRB, analytical MSE, and efficiency (defined in Sec. 2.6.3) are also summarized.

Chapter Outline: In Sec. 2.2, we provide a brief overview of the popular subspace-based DOA estimation algorithms such as MUSIC and root-MUSIC. Then we introduce well-known sparse arrays from the literature and the concept of difference coarray in Sec. 2.3. Sec. 2.4 discusses the effect of mutual coupling. Next, Sec. 2.5 reviews the coarray-MUSIC algorithm and the two commonly used ways of forming the coarray covariance matrix. Sec. 2.6 summarizes the known results on the efficiency of direct-MUSIC and coarray-MUSIC. The content of this chapter is based on some parts of [94].

2.2 Subspace-Based DOA Estimation Algorithms

As explained in Sec. 1.2, the array output (Eq. (1.8)) is expressed in terms of the array manifold matrix \mathbf{A} , source amplitude vector \mathbf{s} , and the random noise vector

\mathbf{n} . Both the source amplitudes and the noise are considered to be random variables, while the DOAs $\theta_1, \theta_2, \dots, \theta_D$ are unknown constants. The array output is recorded over a period of time, and the sampled array outputs, indexed by $k = 1, 2, \dots, K$, are referred to as the ‘snapshots’ of the array output:

$$\mathbf{x}[k] = \mathbf{A}\mathbf{s}[k] + \mathbf{n}[k]. \quad (2.1)$$

Throughout this thesis, we assume that the number of sources D is known and fixed. If D is not known, there are methods available in the literature to estimate D , based either on information-theoretic criteria (such as Akaike information criterion or Bayesian information criterion) or based on the eigenvalues of the estimated signal covariance matrix, such as SORTS [95]. We make the following standard statistical assumptions on the random source amplitudes and noise [1]:

$$\mathbb{E}(\mathbf{s}) = \mathbb{E}(\mathbf{n}) = \mathbf{0}, \quad (2.2)$$

$$\mathbb{E}(\mathbf{n}\mathbf{n}^H) = \sigma_n^2 \mathbf{I}, \quad \mathbb{E}(\mathbf{s}\mathbf{s}^H) = \mathbf{R}_{\mathbf{ss}}, \quad (2.3)$$

$$\mathbb{E}(\mathbf{s}\mathbf{n}^H) = \mathbf{0}. \quad (2.4)$$

These assumptions imply that the source amplitudes and noise are zero-mean random vectors uncorrelated with each other. Furthermore, noise is white with variance σ_n^2 , and can typically be assumed to be Gaussian. The source covariance matrix $\mathbf{R}_{\mathbf{ss}}$ is assumed to have full rank D . While source amplitudes can be correlated in general, we will see in Sec. 2.5 that the coarray-domain formulation requires an additional assumption: that the source amplitudes are uncorrelated (i.e. $\mathbf{R}_{\mathbf{ss}}$ is diagonal). In this thesis, the SNR is computed as

$$\text{SNR} = \sum_{i=1}^D p_i / (D\sigma_n^2), \quad (2.5)$$

where $p_i = [\mathbf{R}_{\mathbf{ss}}]_{ii}$ is the power of the i -th source. The source powers are assumed to be equal in most simulations, except where it is explicitly mentioned otherwise.

Under these statistical assumptions, the array output covariance matrix can be readily shown to be

$$\mathbb{E}(\mathbf{x}\mathbf{x}^H) = \mathbf{R}_{\mathbf{xx}} = \mathbf{A}\mathbf{R}_{\mathbf{ss}}\mathbf{A}^H + \sigma_n^2 \mathbf{I}. \quad (2.6)$$

Note that when there are fewer sources than the number of sensors (i.e., $D < N$), the first term $\mathbf{A}\mathbf{R}_{\mathbf{ss}}\mathbf{A}^H$ has rank D , and its column span is the same as the column span of \mathbf{A} . This D -dimensional space is called the ‘signal subspace’, and its $N - D$ dimensional orthogonal complement in \mathbb{C}^N is called the ‘noise subspace’. The

subspace-based DOA estimation algorithms aim at estimating these subspaces based on the eigendecomposition of \mathbf{R}_{xx} .

Let \mathbf{R}_{xx} have eigendecomposition

$$\mathbf{R}_{xx} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H, \text{ where } \mathbf{E} = \begin{bmatrix} \mathbf{E}_s & \mathbf{E}_n \end{bmatrix}. \quad (2.7)$$

Here, the diagonal matrix $\mathbf{\Lambda}$ has eigenvalues in non-increasing order, and \mathbf{E} is a unitary matrix containing the eigenvectors. Then it can be shown that the matrix $\mathbf{E}_s \in \mathbb{C}^{N \times D}$ containing the top- D eigenvectors spans the signal subspace, whereas $\mathbf{E}_n \in \mathbb{C}^{N \times (N-D)}$ containing the rest of the eigenvectors spans the noise subspace.

In practice, the ideal output covariance matrix \mathbf{R}_{xx} is not known and the sample covariance matrix $\widehat{\mathbf{R}}_{xx}$ obtained from K snapshots of \mathbf{x} is used as an estimate of the unknown \mathbf{R}_{xx} :

$$\widehat{\mathbf{R}}_{xx} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}[k] \mathbf{x}^H[k]. \quad (2.8)$$

Based on the eigendecomposition of $\widehat{\mathbf{R}}_{xx}$, estimates of signal and noise subspace are obtained as follows:

$$\widehat{\mathbf{R}}_{xx} = \widehat{\mathbf{E}}\widehat{\mathbf{\Lambda}}\widehat{\mathbf{E}}^H, \text{ where } \widehat{\mathbf{E}} = \begin{bmatrix} \widehat{\mathbf{E}}_s & \widehat{\mathbf{E}}_n \end{bmatrix}. \quad (2.9)$$

The estimated signal and noise subspace matrices $\widehat{\mathbf{E}}_s \in \mathbb{C}^{N \times D}$ and $\widehat{\mathbf{E}}_n \in \mathbb{C}^{N \times (N-D)}$ are used to estimate the DOAs using algorithms such as MUSIC [10], root-MUSIC [85], and ESPRIT [11].

2.2.1 MUSIC Algorithm

Based on Eqs. (2.6) and (2.7), it can be shown that

$$\mathbf{a}(\omega_i)^H \mathbf{E}_n = \mathbf{0} \quad (2.10)$$

for all true DOAs $\omega_1, \omega_2, \dots, \omega_D$. Using this fact, the MUSIC (Multiple Signal Classification) algorithm first obtains a pseudospectrum based on the estimated noise subspace $\widehat{\mathbf{E}}_n$ as follows [10]:

$$P(\omega) = \frac{1}{\mathbf{a}^H(\omega) \widehat{\mathbf{E}}_n \widehat{\mathbf{E}}_n^H \mathbf{a}(\omega)}. \quad (2.11)$$

Because $\mathbf{a}(\omega_i)^H \widehat{\mathbf{E}}_n \approx \mathbf{0}$ for true DOAs $\omega_1, \omega_2, \dots, \omega_D$, the MUSIC spectrum $P(\omega)$ will produce sharp peaks when ω is close to the true DOAs. Furthermore, if the augmented manifold matrix defined as

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{a}(\omega_1) & \mathbf{a}(\omega_2) & \dots & \mathbf{a}(\omega_D) & | & \mathbf{a}(\omega_{D+1}) \end{bmatrix} \quad (2.12)$$

has full column rank for all ω_{D+1} in the range $[-\pi, \pi)$ that are distinct to any of $\omega_1, \omega_2, \dots, \omega_D$, it can be shown [94], [96], [97] that there is no other $\omega \in [-\pi, \pi)$ that satisfies Eq. (2.10), thus ensuring that there are no false peaks in $P(\omega)$. A ULA satisfies these conditions for any set of D distinct DOAs when $D < N$, because of the Vandermonde structure of the augmented manifold matrix.

Under these conditions, the locations of the peaks of $P(\omega)$ provide an estimate of the DOAs in the ω domain. The heights of the peaks in the MUSIC spectrum simply indicate how close the quantity $\mathbf{a}^H(\omega)\hat{\mathbf{E}}_{\mathbf{n}}$ is to 0, and are not related to the source powers. The MUSIC algorithm based on the eigendecomposition of $\hat{\mathbf{R}}_{\mathbf{xx}}$ is sometimes called ‘direct-MUSIC’ or ‘element space MUSIC’ to contrast it with ‘coarray-MUSIC’ explained in Sec. 2.5. Note that to obtain the locations of peaks in the MUSIC spectrum, one needs to perform a dense grid search. A denser grid can lead to a more accurate DOA estimate, but it requires more computations. Thus, search-free DOA estimation algorithms such as root-MUSIC and ESPRIT are preferred for some array geometries, such as ULA.

2.2.2 Root-MUSIC Algorithm

Root-MUSIC [85] offers a search-free alternative to the MUSIC algorithm, and is usually used when the array is a ULA. The N -sensor ULA

$$\mathbf{z} = \begin{bmatrix} 0 & 1 & 2 & \dots & N-1 \end{bmatrix} \quad (2.13)$$

has a Vandermonde steering vector given by

$$\mathbf{a}(\omega) = \begin{bmatrix} 1 & e^{j\omega} & e^{j2\omega} & \dots & e^{j(N-1)\omega} \end{bmatrix}^T. \quad (2.14)$$

Based on the estimated noise subspace matrix $\hat{\mathbf{E}}_{\mathbf{n}}$, the root-MUSIC algorithm first forms a degree $2N - 2$ polynomial

$$Q(z) = \mathbf{v}^T(z^{-1})\hat{\mathbf{E}}_{\mathbf{n}}\hat{\mathbf{E}}_{\mathbf{n}}^H\mathbf{v}(z), \quad (2.15)$$

where

$$\mathbf{v}(z) = \begin{bmatrix} 1 & z & z^2 & \dots & z^{N-1} \end{bmatrix}^T \quad (2.16)$$

is obtained by substituting $z = e^{j\omega}$ in $\mathbf{a}(\omega)$. The roots of $Q(z)$ appear in reciprocal conjugate pairs. The arguments (angles from X -axis) of the D roots of $Q(z)$ that are inside the unit circle and closest to the unit circle in the complex z -plane provide estimates of the DOAs. Note that the MUSIC spectrum $P(\omega)$ is the reciprocal of $Q(z)$ evaluated on the unit circle $|z| = 1$. Root-MUSIC usually has a smaller

estimation error than MUSIC, as the magnitude errors in the roots of $Q(z)$ do not affect the DOA estimates obtained from root-MUSIC [98]. As originally proposed, the Root-MUSIC is applicable only to the ULA.

2.3 Sparse Arrays and Their Difference Coarrays

An N -sensor ULA can identify up to $N - 1$ DOAs using the above-mentioned subspace-based DOA estimation algorithms [1], and is widely used in several array signal processing tasks. However, there is another important subclass of linear integer arrays called **sparse arrays**. Sparse arrays have their sensors separated by non-uniform distances, and have $n_{i+1} - n_i > 1$ for at least one pair of consecutive sensors. There are three main advantages of sparse arrays over ULA [94]:

1. Well-designed sparse arrays with N sensors can identify up to $O(N^2)$ DOAs unambiguously, if the source amplitudes are uncorrelated and the number of snapshots K is large enough. This is significantly larger than ULA, which can identify at most $N - 1$ DOAs.
2. Sparse arrays have sensors farther apart from each other on average compared to a ULA. This reduces the impact of mutual coupling between sensors, as discussed in Sec. 2.4.
3. For a fixed number of sensors, sparse arrays with N sensors have a larger aperture than an N -sensor ULA. The larger aperture typically implies a smaller DOA estimation MSE, and the array can resolve closely spaced DOAs [99].

However, unlike the ULA, the manifold matrix of sparse arrays is not a Vandermonde matrix, and thus, the sparse arrays can suffer from ambiguity in DOA estimation. However, well-designed sparse arrays can employ coarray-based DOA estimation algorithms and can unambiguously identify up to $O(N^2)$ DOAs. The key to identifying $O(N^2)$ DOAs with sparse arrays is difference coarray, which we review next.

2.3.1 Difference Coarray

Consider a sparse integer array

$$\mathbf{z} = \begin{bmatrix} n_1 & n_2 & \cdots & n_{N-1} & n_N \end{bmatrix}. \quad (2.17)$$

Its difference coarray is the set formed by pairwise differences between sensor locations:

$$\mathbb{D}_{\mathbf{z}} = \{n_i - n_j\}_{i,j=1}^N. \quad (2.18)$$

The difference coarray is symmetric, and $\mathbb{D}_{\mathbf{z}}^+$ is used to represent the non-negative entries in $\mathbb{D}_{\mathbf{z}}$. Note that $\mathbb{D}_{\mathbf{z}}$ can contain up to $N^2 - N + 1$ distinct elements. The coarray weights are defined as

$$w(l) = |\{(n_i, n_j) \text{ s.t. } n_i - n_j = l\}|. \quad (2.19)$$

The l -th coarray weight $w(l)$ denotes the number of sensor pairs with distance l between them. The missing coarray elements have $w(l) = 0$, which implies that no two sensors are separated by a distance l . These are known as the ‘holes’ in the difference coarray.

The importance of the difference coarray becomes clear when, in addition to the standard statistical assumption from Eq. (2.2), the source amplitudes are assumed to be uncorrelated:

$$\mathbf{R}_{\text{ss}} = \begin{bmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & p_D \end{bmatrix}. \quad (2.20)$$

Here, $p_i > 0$ is the power of the i -th source. Under this uncorrelated assumption, it can be verified that

$$\mathbb{E}(\mathbf{x}_k \mathbf{x}_m^*) = \sum_{i=1}^D p_i e^{j\omega_i l} + \sigma_n^2 \delta(l), \text{ where } l = n_k - n_m, \quad (2.21)$$

and $\delta(l) = 1$ when $l = 0$, and 0 otherwise. Because these pairwise correlations between sensor outputs depend only on the distance l between sensors, we denote them as $R(l)$.

In practice, we can estimate these correlations by averaging the corresponding entries from the sample covariance matrix of the array output:

$$\hat{R}(l) = \frac{1}{w(l)} \sum_{n_i - n_j = l} \frac{1}{K} \sum_{k=1}^K x_{n_i}[k] x_{n_j}^*[k] = \frac{1}{w(l)} \sum_{n_i - n_j = l} [\hat{\mathbf{R}}_{\mathbf{xx}}]_{i,j}. \quad (2.22)$$

Note that $\hat{R}(l)$ can thus be calculated for all ‘lags’ l that appear in the difference coarray $\mathbb{D}_{\mathbf{z}}$. This explains the central role of difference coarray and associated

coarray weights for sparse arrays. DOA estimation performed using the estimated correlations $\widehat{R}(l)$, $l \in \mathbb{D}_{\mathbf{z}}$ is called coarray-based DOA estimation. Sparse arrays that can estimate correlations at a large number of **consecutive lags** are particularly suitable to identify $D > N$ DOAs. For most sparse arrays from the literature, the consecutive lags in the difference coarray are of the form $-L \leq l \leq L$, and such a segment of consecutive lags is called a ‘central ULA segment’ in the coarray. Well-designed sparse arrays can have $L = O(N^2)$, and can identify $O(N^2)$ DOAs using coarray domain processing, as we will see in Sec. 2.5.

2.3.2 Commonly Encountered Sparse Arrays from the Literature

In this section, we review the most commonly encountered sparse arrays from the literature.

Minimum Redundancy Arrays

Minimum redundancy arrays (MRAs) are one of the earliest known sparse arrays [33]. The sensor positions of MRAs are chosen such that the coarray size is maximized, under the constraint that the coarray is a ULA (i.e., the coarray contains only consecutive lags). However, finding such an optimal MRA requires an exhaustive search because no closed-form expression for sensor positions is available. Such an exhaustive search quickly becomes computationally intractable even for moderately large N . The tabulated sensor positions are available for only a small number of sensors [33], [84], [100]. The introduction of nested arrays [34] and coprime arrays [35], which have closed-form expressions for sensor locations, has opened up a new direction for research in the design of sparse arrays. Although nested arrays and coprime arrays can identify fewer DOAs than an MRA with the same number of sensors, their closed-form expression for sensor positions for any N makes them more practical and attractive.

Nested Arrays

Nested arrays [34] are formed by the union of a standard ULA with N_1 sensors and a sparse ULA with N_2 sensors as shown in Fig. 2.1(a):

$$\mathbf{z}_{NA} = \{1, 2, \dots, N_1\} \cup \{N_1 + 1, 2(N_1 + 1), \dots, N_2(N_1 + 1)\}. \quad (2.23)$$

The difference coarray of the nested arrays is the following ULA:

$$\mathbb{D}_{\mathbf{z}_{NA}} = \{-N_2(N_1 + 1) + 1, \dots, -1, 0, 1, \dots, N_2(N_1 + 1) - 1\}. \quad (2.24)$$

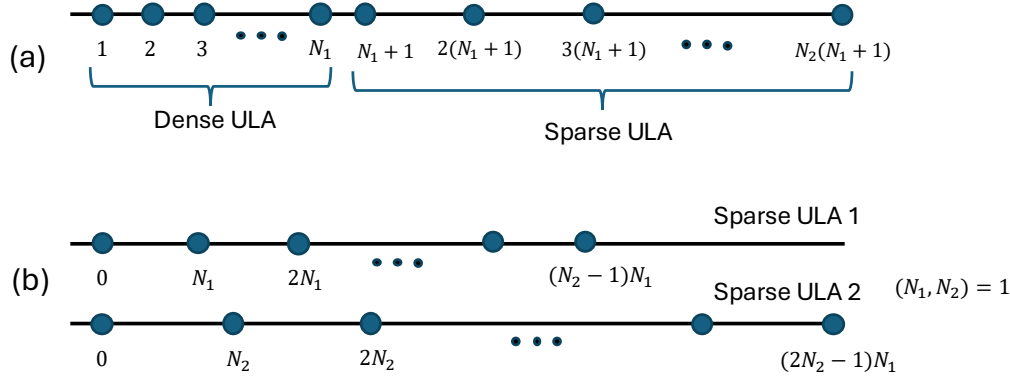


Figure 2.1: (a) A nested array consists of a dense ULA of N_1 sensors followed by a sparse ULA of N_2 sensors with interelement spacing $N_1 + 1$. (b) A coprime array is a union of two sparse ULAs with interelement spacings N_1 and N_2 where $(N_1, N_2) = 1$.

For a given total number of sensors $N = N_1 + N_2$, the optimal values of N_1 and N_2 that maximize the length of the coarray are [34]

$$N_1 = \lfloor N/2 \rfloor, \text{ and } N_2 = N - \lfloor N/2 \rfloor. \quad (2.25)$$

Although the length of the ULA segment in the coarray of a nested array is less than that of an MRA with the same number of sensors, it is easy to generate the nested array for any given number of sensors N , unlike an MRA. However, one of the drawbacks of the nested array is that it contains several sensors that are closely placed, because of the dense ULA segment of N_1 sensors, and this can lead to increased mutual coupling [39], [101].

Coprime Arrays

Coprime arrays [35] are formed by the union of two sparse ULAs as shown in Fig. 2.1(b). The first sparse ULA has elements spaced by the integer N_1 , and has N_2 sensors. The second array has adjacent elements spaced by the integer N_2 , and has $2N_1$ sensors. The integers N_1 and N_2 are chosen to be coprime, that is, $(N_1, N_2) = 1$.

$$\mathbf{z}_{CPA} = \{0, N_1, 2N_1, \dots, (N_2 - 1)N_1\} \cup \{0, N_2, 2N_2, \dots, (2N_1 - 1)N_2\}. \quad (2.26)$$

The total number of sensors is $N = 2N_1 + N_2 - 1$. It has been shown [35], [83] that the difference coarray of the coprime array contains a central ULA segment from $-N_1(N_2 + 1) + 1$ to $N_1(N_2 + 1) - 1$. For a fixed number of sensors N , the ULA segment in the coarray of the coprime array is smaller than that of the nested arrays. However,

coprime arrays are better at mitigating the effect of mutual coupling because they have fewer sensor pairs separated by a small distance [91].

The introduction of nested arrays and coprime arrays provided a new direction to sparse array design research. Numerous generalizations of nested arrays and coprime arrays along with new improved sparse array designs have been proposed in the literature [39], [83], [91], [92], [101]–[105]. Designing sparse arrays that have long central ULA segments in the coarray along with other desirable properties has since been an active and growing area of research.

2.4 The Effect of Mutual Coupling

When sensors have electromagnetic coupling between them, the array output does not follow the standard model in Eq. (2.1). This is called the mutual coupling effect, and it is challenging to precisely model mathematically [36]. In practice, it is usually unknown to the user. In numerical simulations, the effect of coupling is typically modeled as [37]–[39]

$$\mathbf{x}[k] = \mathbf{C}\mathbf{A}\mathbf{s}[k] + \mathbf{n}[k], \quad (2.27)$$

where \mathbf{C} is the coupling matrix with entries

$$\mathbf{C}_{i,j} = c_{|n_i - n_j|}. \quad (2.28)$$

The coupling coefficients c_l are assumed to be non-zero for $l \leq B$. Here, $c_0 = 1$, and the rest of the non-zero coupling coefficients c_l for $1 \leq l \leq B$ have their magnitudes inversely proportional to l , i.e.,

$$|c_m/c_n| = n/m. \quad (2.29)$$

For a ULA, \mathbf{C} is a banded Toeplitz matrix [37]. The coupling strength is characterized by $|c_1| = c$. Another metric to characterize the amount of mutual coupling is the coupling leakage defined as

$$\mathcal{L} = \frac{\|\mathbf{C} - \mathbf{C}_{diag}\|_F}{\|\mathbf{C}\|_F}. \quad (2.30)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and \mathbf{C}_{diag} is a diagonal matrix with the same diagonal entries as that of \mathbf{C} . Because the coupling coefficients are inversely proportional to the distance between sensors, the sensor pairs with small separations contribute the most to the effect of mutual coupling. Thus, many sparse arrays from

the literature aim at reducing the coarray weights $w(1)$, $w(2)$, and $w(3)$ to reduce the effect of mutual coupling on DOA estimation.

To mitigate the impact of mutual coupling on DOA estimation, some papers [106]–[108] estimate the coupling matrix along with the DOAs. In this thesis, we take a different approach of simply using well-designed sparse arrays for DOA estimation, without explicitly estimating the coupling matrix. In this ‘coupling-agnostic’ approach, we do not explicitly account for mutual coupling algorithmically but rather design sparse arrays that can effectively handle the presence of mutual coupling. Reducing the effect of mutual coupling is a motivation for the modified sparse array design criteria that we propose later in Chapter 4.

2.4.1 A Note on the Practicality of the Mutual Coupling Model

While this basic phenomenon of coupling is a direct consequence of Faraday’s law of induction, the actual application in an antenna array is quite complicated (for example, see p. 475 of [109], and [110]). Furthermore, the geometry of the antennas used in the array (i.e., whether they are dipole antennas, sleeve dipole antennas, spiral antennas, etc.) plays a role in determining the mutual coupling behavior [111], [112], [113]. The coupling matrix \mathbf{C} is actually the inverse of a so-called normalized impedance matrix \mathbf{Z}_0 that has been used to model the self- and mutual-coupling in the antenna elements [114], [115], [116], [106]. As seen from these references, there are multiple ways to formulate the matrix \mathbf{Z}_0 , and they yield quite different results. A summary of different methods and their pros and cons can be found in [110]. While these papers are insightful, they do not offer any way to estimate the ratio $c = |c_1|$, which actually comes from the inverse \mathbf{Z}_0^{-1} .

The element $[\mathbf{Z}_0]_{k,m}$ is proportional to z_{km} which is the mutual impedance between the k th and m th sensor. These impedances are approximately proportional to $1/(n_k - n_m)$. This is consistent with the fact that the radiation term for the electric field from an accelerated charge (as in the case of a sinusoidal current in an antenna wire) varies as $1/r$ rather than as $1/r^2$ (Chapter 28 of [117]). The $1/r$ assumption has been used in the past, and it can be verified in the estimates of impedances reported in Table 1 of [115] and Table 2 of [116]. The fact that the dependence is not exactly of the form $1/r$ is well known even from basic examples such as can be found on page 189 of [118] (e.g., set $d \gg a, b$ in Eqs. (10) and (11) therein).

Now, it is difficult to see how the above assumption on $[\mathbf{Z}_0]_{k,m}$ reflects in the matrix \mathbf{C} . As explained earlier, it is often assumed that $\mathbf{C}_{i,j} = c_{|n_i - n_j|}$ and that

$|c_m/c_n| = n/m$. While this is similar to the properties of $[\mathbf{Z}_0]_{k,m}$, it is hard to give a theoretical justification that the properties of \mathbf{Z}_0 will be inherited like this by the elements of the inverse $\mathbf{C} = \mathbf{Z}_0^{-1}$. A more detailed description of the difficulties involved in modeling mutual coupling can be found in [119]. However, this model of mutual coupling has been widely used [38], [39], [106] in the signal processing community, and the same is used in this thesis as well.

2.5 Coarray-Based DOA Estimation

The sample covariance matrix $\hat{\mathbf{R}}_{\mathbf{xx}}$ (Eq. (2.8)) of the array output has size $N \times N$. So, based on its eigendecomposition, one can identify no more than $N - 1$ DOAs. However, as discussed in Sec. 2.3.1, when the source amplitudes are uncorrelated, correlations $\hat{R}(l)$ can be estimated for all lags l in the difference coarray. The sparse arrays can estimate $D > N$ DOAs through coarray-domain DOA estimation performed using these correlations, as explained next.

Well-designed sparse arrays such as MRAs, nested arrays, coprime array, and their generalizations contain a large central ULA segment in their difference coarrays. Suppose the difference coarray $\mathbb{D}_{\mathbf{z}}$ of the array \mathbf{z} contains central ULA segment from $l = -(L - 1)$ to $l = L - 1$. This means that $w(l) \neq 0$ for $l = -L + 1, -L + 2, \dots, L - 2, L - 1$. The total length of the ULA segment in the coarray is $2L - 1$. Based on the correlations estimated at these consecutive lags, the spatial smoothing based algorithm computes the following matrix [34]:

$$\hat{\mathbf{R}}_{SS} = \frac{1}{L} \sum_{l=-L+1}^0 \hat{r}_l \cdot \hat{r}_l^H, \quad (2.31)$$

where $\hat{r}_l \in \mathbb{C}^L$ is a vector containing L consecutive correlation entries starting from lag l :

$$\hat{r}_l = \begin{bmatrix} \hat{R}(l) & \hat{R}(l+1) & \dots & \hat{R}(l+L-1) \end{bmatrix}^T. \quad (2.32)$$

Note that the matrix $\hat{\mathbf{R}}_{SS}$ acts like a proxy for the output covariance matrix that a ULA with L physical sensors would have produced. The spatial smoothing MUSIC algorithm (SS-MUSIC) simply uses $\hat{\mathbf{R}}_{SS}$ in place of $\hat{\mathbf{R}}_{\mathbf{xx}}$ to estimate the DOAs. As the central ULA segments in the coarrays of well-designed arrays can have lengths as large as $O(N^2)$, the matrix $\hat{\mathbf{R}}_{SS}$ has size $O(N^2) \times O(N^2)$, and can thus identify up to $O(N^2)$ DOAs.

In [93], it was noted that there is an equivalent alternative to SS-MUSIC, called direct augmentation MUSIC (DA-MUSIC). It proposes to construct a matrix by

arranging the estimated correlations at consecutive lags $\widehat{R}(-L+1), \dots, \widehat{R}(L-1)$ in a Toeplitz matrix $\widehat{\mathbf{R}}_{DA}$ as follows:

$$[\widehat{\mathbf{R}}_{DA}]_{i,j} = \widehat{R}(i-j). \quad (2.33)$$

It is shown in [93] that the matrices $\widehat{\mathbf{R}}_{SS}$ and $\widehat{\mathbf{R}}_{DA}$ are related as follows:

$$\widehat{\mathbf{R}}_{SS} = \frac{1}{K} \widehat{\mathbf{R}}_{DA} \widehat{\mathbf{R}}_{DA}^H. \quad (2.34)$$

Note that this relation is valid even for a finite number of snapshots K . Both $\widehat{\mathbf{R}}_{DA}$ and $\widehat{\mathbf{R}}_{SS}$ are Hermitian Toeplitz matrices. Furthermore, $\widehat{\mathbf{R}}_{SS}$ is positive semidefinite, but $\widehat{\mathbf{R}}_{DA}$ may not be. Considering the above relation between $\widehat{\mathbf{R}}_{DA}$ and $\widehat{\mathbf{R}}_{SS}$, DA-MUSIC proposed to use matrix $\widehat{\mathbf{R}}_{DA}$ instead of $\widehat{\mathbf{R}}_{SS}$ to estimate the DOAs. Since $\widehat{\mathbf{R}}_{DA}$ may have non-negative eigenvalues, the eigenvectors corresponding to the D eigenvalues with largest *magnitude* form the signal subspace.

It is equivalent to use either one of these matrices for subspace-based DOA estimation when a large number of snapshots are available. With limited snapshots, the direct-augmentation method can yield better performance as shown in [120]. Performing MUSIC using either of the coarray matrices $\widehat{\mathbf{R}}_{DA}$ and $\widehat{\mathbf{R}}_{SS}$ is called ‘coarray-MUSIC’. An example of coarray-MUSIC spectra obtained with a nested array and a coprime array for estimating $D > N$ DOAs is shown in Fig. 2.2. Fig. 2.2(a) estimates $D = 25$ DOAs using a 10-sensor nested array with $N_1 = N_2 = 5$ when $\text{SNR} = 0$ dB and $K = 300$ snapshots. Fig. 2.2(b) estimates $D = 25$ DOAs using a 17-sensor coprime array with $N_1 = 5$ and $N_2 = 8$ when $\text{SNR} = 10$ dB and $K = 500$. One can also use adaptations of other subspace-based DOA estimation algorithms, such as root-MUSIC and ESPRIT, with the coarray matrices ($\widehat{\mathbf{R}}_{DA}$ and $\widehat{\mathbf{R}}_{SS}$) instead of $\widehat{\mathbf{R}}_{xx}$. These coarray-domain algorithms are called coarray root-MUSIC and coarray-ESPRIT.

It is important to note that the fundamental assumption in using sparse arrays and coarray-based estimation of more DOAs than the number of sensors is that the sources are uncorrelated. Without this assumption, Eq. (2.21) is not valid, and the coarray domain DOA estimation is not well-motivated. However, such an assumption is difficult to verify practically, and it may not hold true, especially in situations where multipath propagation is possible. Developing algorithms and sparse arrays that are robust to the presence of correlations is of practical interest. One of the recent attempts in this direction is the MESA algorithm [121] based on a stochastic maximum likelihood formulation. It is shown in [121] that coarray-based

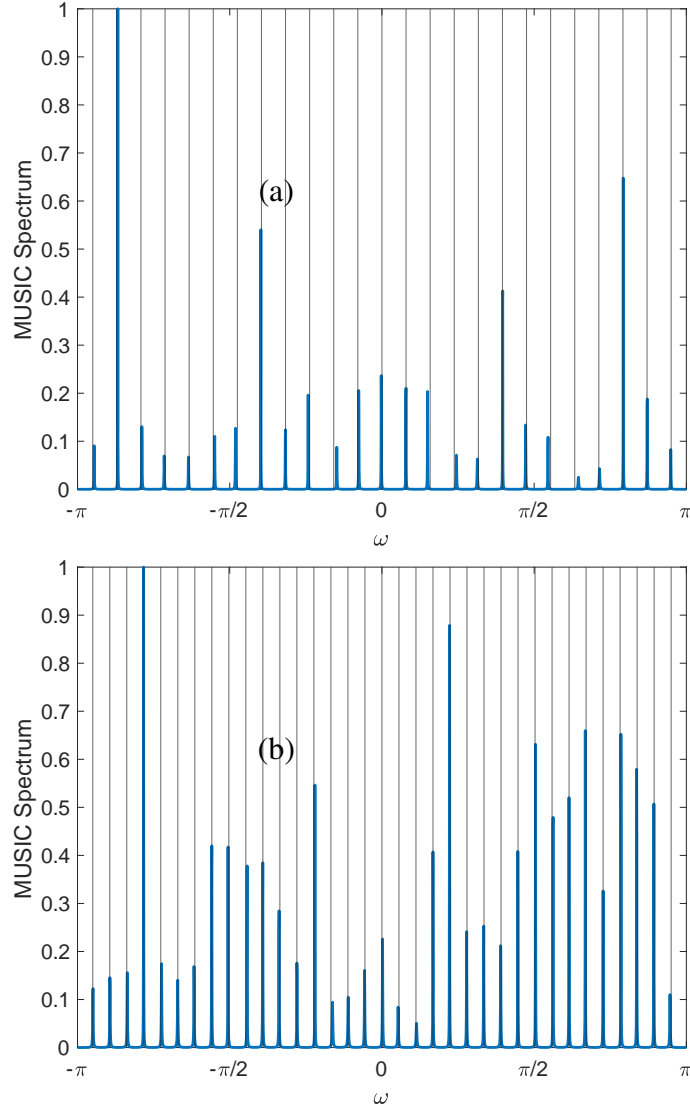


Figure 2.2: Coarray-MUSIC plots. (a) Identifying $D = 25$ DOAs using a 10-sensor nested array with $N_1 = N_2 = 5$ when $\text{SNR} = 0$ dB and $K = 300$. (b) Identifying $D = 35$ DOAs with 17-sensor coprime array with $N_1 = 5, N_2 = 8$ when $\text{SNR} = 10$ dB and $K = 500$. The black vertical lines correspond to the true DOAs.

methods can be sensitive to source correlations due to the destruction of the Toeplitz covariance structure, and the MESA algorithm can be robust to source correlations, especially when $D < N$. Further research in this direction is required to enhance the practical utility of sparse arrays.

2.6 CRB, Analytical MSE, and Efficiency of DOA Estimation

The Cramér-Rao bound (CRB) is used to obtain a lower bound on the variance of an unbiased estimator. For DOA estimation using MUSIC, CRB and analytical MSE

expressions were derived many decades ago for the case when there are fewer sources than the number of sensors in [122]. However, the CRB for estimating more sources than the number of sensors was derived only recently [86], [87]. Furthermore, [87] also provides an analytical MSE expression for coarray-MUSIC. It has been noted [87] that the DOA estimation MSE for coarray-MUSIC using sparse arrays does not equal CRB even asymptotically when more than one source is present. This is in contrast with the ‘direct-MUSIC’ which achieves CRB asymptotically [122]. The discrepancy between the MSE and the CRB is quantified by the efficiency (defined formally in Sec. 2.6.3). In this section, we summarize the details regarding the CRB, MSE, and efficiency for direct-MUSIC and coarray-MUSIC.

2.6.1 Cramér-Rao Bound for DOA Estimation

The expression for CRB for DOA estimation derived in [96], [122] provides a lower bound on the variances of unbiased DOA estimates obtained using any algorithm, including direct-MUSIC and coarray-MUSIC. However, this CRB expression is valid only when $D < N$, and not valid when $D \geq N$. Even for $D < N$, the above CRB expression does not use the a-priori assumption that the sources are uncorrelated, which is a common assumption in coarray-based methods. A general expression for CRB under the uncorrelated assumption was derived recently in [86], [87]. This CRB expression is valid for both situations $D < N$ and $D \geq N$ as long as the Fisher information matrix (FIM) is nonsingular, and the precise condition for this nonsingularity is also provided. There is an important difference in the behavior of CRB as a function of SNR in the two situations.

For a fixed number of snapshots, the CRB decreases monotonically with SNR when $D < N$, but in most situations when $D \geq N$, the CRB does not keep decreasing as SNR increases but rather saturates for large SNR. This behavior is illustrated in Fig. 2.3 by plotting the CRB values for MRA, nested array, and coprime arrays with $N = 16$ sensors when the DOAs are uniformly spaced in the ω -domain from -0.98π to 0.98π , and $K = 500$ snapshots are used. The values plotted are the average CRB values of the ω -domain DOAs. When $D = 10$, the CRB does not saturate, whereas when $D = 30$, the CRB saturates for all three arrays. The CRB for MRA is the lowest and saturates around 6.37×10^{-7} . The CRBs for nested array and coprime array saturate around 2.45×10^{-6} and 3.76×10^{-6} respectively. As one would expect based on the sizes of the difference coarrays, the MRA has the lowest CRB, followed by the nested array and the coprime array. To the best of our knowledge, a CRB for DOA estimation in the presence of mutual coupling is not known in the literature.

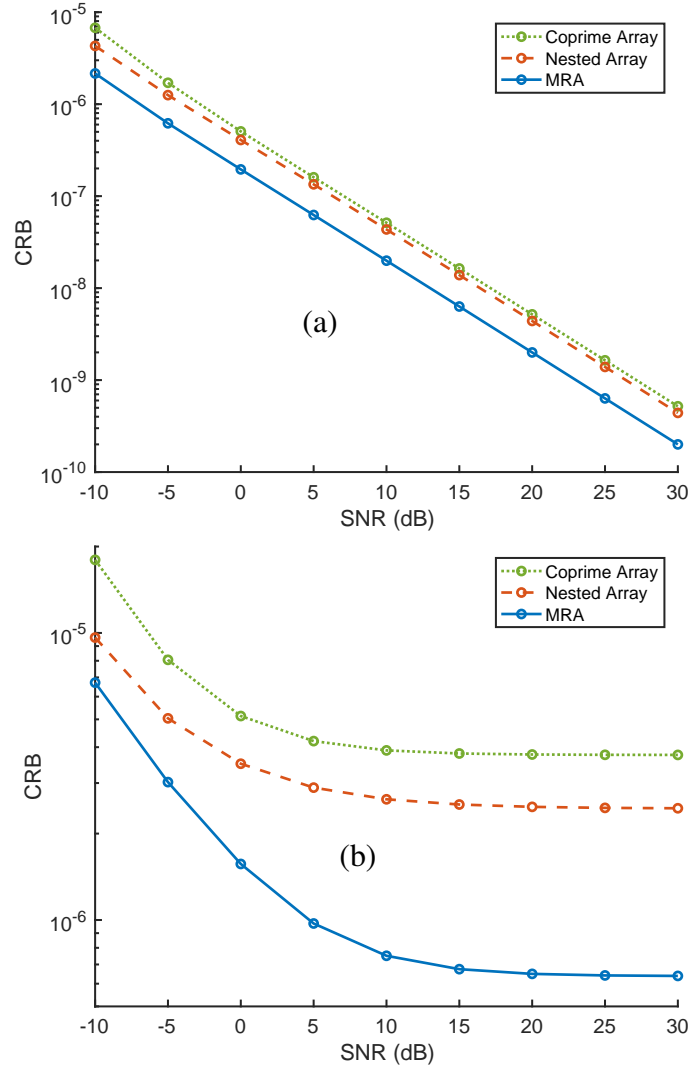


Figure 2.3: Plots of CRB for sparse arrays identifying (a) $D = 10$ and (b) $D = 30$ DOAs with $N = 16$ sensors. When $D < N$, CRB does not saturate, whereas when $D \geq N$, CRB saturates at large SNR.

2.6.2 Analytical MSE of Direct-MUSIC and Coarray-MUSIC

In addition to the CRB, another quantity of interest is the analytical MSE of DOA estimation algorithms like MUSIC and root-MUSIC. The analytical MSE is usually derived based on first-order perturbation analysis and is also referred to as the theoretical MSE. Note that this is different from the experimental or Monte-Carlo MSE obtained by performing numerical simulations. For element space MUSIC, the analytical MSE expressions are derived in [122], and dependence on the number of snapshots K , number of sensors N , and SNR is analyzed. The error analysis for element space root-MUSIC for ULA is derived in [98]. Asymptotically, with a large

number of snapshots, both element space MUSIC and element space root-MUSIC have the same analytical MSE. In spite of this, root-MUSIC is preferred over MUSIC because the radial errors in the locations of the roots of the root-MUSIC polynomial do not change the DOAs estimated with root MUSIC, but can change the locations of peaks in the MUSIC spectrum [98]. An additional advantage of root-MUSIC is that it is a search-free algorithm, unlike MUSIC. The error analysis of ESPRIT for ULA is performed in [123], and two variations of ESPRIT, namely LS-ESPRIT and TLS-ESPRIT, are shown to have the same asymptotic MSE.

The analytical expression for the MSE of coarray-MUSIC was derived recently in [87]. It was found that the MSE for SS-MUSIC and DA-MUSIC is identical due to the equivalence of the two methods. However, coarray-domain DOA estimation is distinctly different from element-space domain estimation in terms of efficiency, as we will see next.

2.6.3 Efficiency of Direct-MUSIC and Coarray-MUSIC

The efficiency of an estimate is defined as the ratio of the CRB to the MSE. The average efficiency over D DOAs is defined as [87]

$$\kappa = \frac{\sum_{i=1}^D CRB(\omega_i)}{\sum_{i=1}^D MSE(\omega_i)}. \quad (2.35)$$

The MSE can either be obtained based on a Monte-Carlo simulation, or an analytical expression can be used if available. For an unbiased DOA estimator, the efficiency is always between 0 and 1, i.e. $0 \leq \kappa \leq 1$. DOA estimation is said to be efficient when $\kappa = 1$, and inefficient when $\kappa < 1$. It has been noted in [96], [122] that asymptotically, with either a large number of snapshots K , or at large SNR, the analytical MSE of direct-MUSIC is the same as the CRB. That is, direct-MUSIC is known to be asymptotically efficient. However, the coarray-MUSIC is observed to be inefficient even asymptotically [87].

First, consider the variation of CRB and coarray-MUSIC MSE with the number of snapshots K . Asymptotically, for a large number of snapshots, the CRB and coarray-MUSIC MSE expressions have $1/K$ dependence on the number of snapshots K , and hence as K is increased, CRB and MSE decrease without a bound irrespective of whether $D < N$ or $D \geq N$. Although both quantities go to zero asymptotically, it has been observed that when $D > 1$, their ratio κ remains strictly smaller than 1 [87]. Thus, coarray-MUSIC is not efficient asymptotically with a large number of snapshots.

On the other hand, the dependence of efficiency κ on SNR is more complicated. When $1 < D < N$, the CRB decreases monotonically with SNR without a bound, whereas the MSE saturates to a non-zero value at large SNR [87]. This implies that the coarray-MUSIC based estimation of DOAs is highly inefficient when $1 < D < N$, i.e., κ is nearly 0 when the SNR is large [87]. For the other case of $D \geq N$, both CRB and MSE saturate to non-zero values as SNR increases and $0 < \kappa < 1$. Although κ does not go to zero in this case, it remains smaller than 1 even at large SNR. Thus, coarray-MUSIC based DOA estimation is asymptotically inefficient with SNR when $D > 1$.

For nested arrays in particular, it has been proved in [124] that even for $D = 2$ and in the absence of noise ($\sigma_n = 0$), coarray-MUSIC cannot precisely identify the true signal and noise subspaces, unless the source signals are temporally orthogonal. This means that even asymptotically at large SNR, coarray-MUSIC can only find a close approximation to the true signal and noise subspaces, and cannot identify them precisely. Thus, the DOA estimation MSE with coarray-MUSIC cannot go to zero at large SNR, leading to the loss of efficiency $\kappa < 1$. In another recent paper [125], a non-asymptotic (finite snapshot) analysis of coarray-ESPRIT is presented.

To alleviate these drawbacks of coarray-MUSIC, recently some alternate algorithms such as weighted least squares [90] method, augmented Toeplitz matrix reconstruction approach [89], and weighted proxy covariance matrix construction approach [124] have been proposed in the literature to improve the efficiency of coarray-based DOA estimation. Out of these, the weighted least squares approach provably achieves asymptotic efficiency for both $D \geq N$ and $D < N$ cases, and the gap between CRB and MSE for coarray DOA estimation becomes negligible as the number of snapshots increases.

2.7 Concluding Remarks

In this chapter, we briefly reviewed the important concepts related to DOA estimation that will be used throughout this thesis. In particular, we reviewed subspace-based DOA estimation algorithms, the fundamentals of sparse arrays, and coarray-based DOA estimation. This serves as the foundation for Chapter 3 to Chapter 6 of this thesis. In Chapters 3, 4, and 5, we will develop new sparse array geometries and algorithms for DOA estimation under aperture constraints and mutual coupling. Chapter 6 will focus on further investigations regarding the efficiency of coarray-domain DOA estimation.

Chapter 3

NON-INTEGER ARRAYS FOR DIRECTION OF ARRIVAL ESTIMATION

3.1 Introduction

The majority of the array signal processing literature has been focused on the use of linear integer arrays for DOA estimation and beamforming. The linear integer arrays have m sensors¹ placed at locations $z_i = n_i \cdot \lambda/2$, where n_i are integers for $i = 1, 2, \dots, m$, and λ is the wavelength of impinging monochromatic sources. As discussed in the previous chapter, the ULA with $n_i = i - 1$ is widely used due to its simplicity and Vandermonde manifold structure [1], [10]. Sparse integer arrays such as MRAs, nested arrays, and coprime arrays can even identify more sources than the number of sensors through the difference coarray domain. Some non-integer linear arrays have been mentioned in the literature for beamforming and to some extent for DOA estimation (see below), but they have not been studied in detail.

In this chapter, we consider non-integer arrays for the DOA estimation task. First, we systematically introduce rational arrays that have sensors placed at locations $r_i \cdot \lambda/2$, where r_i are rational numbers, i.e., $r_i = P_i/Q_i$ where P_i and Q_i are integers. These are, in general, non-integer arrays and include integer arrays ($r_i = \text{integers}$) as special cases. It will be demonstrated that this broader class of rational arrays can offer some advantages compared to integer arrays, in addition to increased flexibility.

We show that rational arrays are able to better utilize the aperture when a limited number of sensors are to be distributed in a fixed aperture. In particular, we propose rational coprime arrays for this and show that they have better performance (reduced DOA estimation error) than integer array alternatives. Furthermore, if the impinging signals are known to span only a portion of the entire spatial scope of $[-90^\circ, +90^\circ]$, sensors can be better placed at rational locations to ensure maximum aperture, while still maintaining unique identifiability property. Rational sensor positions also allow scaling of already developed integer arrays. An appropriately scaled array preserves the desirable properties of the original integer array, like symmetry or a large ULA segment in the coarray, and at the same time satisfies the aperture constraint. Later

¹In this chapter, we use m to denote the number of sensors, to be consistent with the notation used in [88]. In the rest of the thesis, N is used to denote the number of sensors.

in this chapter, we also briefly consider arbitrary arrays, where sensors can be placed at any real numbers instead of rational numbers.

For such non-integer arrays, it is important to consider steering vector invertibility and unique identifiability of sources without ambiguity, using subspace-based algorithms such as MUSIC. Papers [97], [126] provide a necessary and sufficient condition for steering vector invertibility for integer arrays. We extend it here for the case of rational arrays. This requires extensions of some common arithmetic notions, such as gcd (greatest common divisor), lcm (least common multiple), and coprimality, to the case of rational numbers. Some online sources, such as [127], [128], present some discussions about gcd and lcm for rational numbers. However, these notions are introduced in this chapter systematically with proofs for completeness. We also propose a unifying notion of coprimality for rational numbers, which plays a central role in many theoretical developments considered in this chapter.

The well-known conditions for the unique unambiguous identifiability of sources with MUSIC in terms of the rank of the augmented array manifold matrix [96] are applicable to non-integer arrays as well. For m -sensor integer ULAs, the Vandermonde structure ensures that they can identify $m - 1$ sources unambiguously [1]. However, even for general integer arrays, it is not easy to characterize the conditions for unambiguous identifiability with MUSIC explicitly in terms of sensor positions. Some insights on detecting and resolving manifold ambiguities for general linear arrays can also be found in [129] and [130], respectively. For the case of rational coprime arrays that are proposed in this chapter, we will present generalizations of identifiability results given in [97], [126] to the non-integer case. All identifiability discussions in this chapter are based on the second-order statistics of the received signal only. It is possible to identify more sources with higher order statistics [131], [132], but that is not the focus of this chapter.

The above developments on rational arrays can be further generalized in two directions. First, for the case of restricted spatial scope (i.e., when the sources are known to be in only a sector of the space), we present a relaxed coprimality condition that still ensures unambiguous identifiability. Then we further extend the results to arrays with sensors placed at arbitrary (real number) locations based on the generalization of gcd and coprimality to the case of real numbers.

Lastly, we will demonstrate that rational arrays have another important advantage over integer arrays, through their difference coarrays. Rational arrays can identify $O(m^2)$ uncorrelated sources using m sensors through the difference coarray domain,

even in the presence of aperture constraint. Integer arrays such as minimum redundancy arrays (MRAs), nested arrays, and coprime arrays can also identify $O(m^2)$ uncorrelated sources using m sensors. However, they can do so only when there is no aperture constraint, that is, a large enough aperture is available. The unique advantage of rational arrays is that they can do the same even when only a short aperture is available.

Note that for the case of rational arrays, it is possible to rewrite the rational sensor locations r_i as dn_i where n_i are integers and d is a rational ‘scaling factor’. While the ‘scaled integer representation’ for rational arrays will be useful at times, we will see that the rational framework considered in this chapter has some advantages. When a fixed number of sensors are to be placed in a given aperture, it is often more convenient and straightforward to directly toy with rational placements of the sensors. The ‘scaling factor’ d and equivalent integers n_i vary haphazardly even with small changes in sensor locations r_i , and the equivalent scaled integer representation may not be useful to understand the properties and DOA estimation performance of the array. Thus the general rational framework presented in this chapter is indeed useful. Furthermore, note that it may not be possible to express arbitrary (non-rational) sensor position z_i as dn_i where n_i are integers as we explain in Sec. 3.6. For such arbitrary (real-valued) arrays, the theoretical results follow easily based on the further generalization of the rational framework, in particular by extending coprimality to the case of real numbers.

3.1.1 Past Work on Non-Integer Arrays

Non-integer arrays do arise in practice when the sensor positions are optimized to achieve a certain objective. Such arrays have been proposed for many beamforming applications. For example, some papers propose to optimize the sensor locations in order to reduce the sidelobe level in beamforming [133], [134]. Similarly, some methods optimize sensor location to suppress interferences [135], or achieve a desired beampattern by choosing sensor locations appropriately [136]. The resulting optimal arrays in these cases are not integer arrays.

Apart from beamforming, there are some methods in the literature that perform DOA estimation with such arbitrary non-integer arrays. Usually, such techniques approximate array manifolds of arbitrary non-uniform arrays with array manifolds of virtual ULAs, in order to easily apply root-MUSIC like techniques for DOA estimation. This includes techniques such as array interpolation [137] and manifold

separation [138]. Some papers have developed a method to interpolate the covariance matrix of the data obtained from non-integer arrays to detect more sources than the number of sensors [139], [140]. There are also Fourier domain MUSIC methods [141] that can be applied to arbitrary two-dimensional non-integer arrays to perform DOA estimation. Another paper has proposed irregular Vandermonde decomposition and atomic norm minimization based gridless DOA estimation for non-uniform linear arrays [142]. Although these methods are appealing and work well in practice, there has been no clear account of when such methods produce theoretically unambiguous DOA estimates.

3.1.2 Scope and Outline of This Chapter

This chapter systematically introduces non-integer arrays and demonstrates the usefulness of such arrays. In Sec. 3.2, we review some basic properties of rational numbers and extend the notions of gcd, lcm, and coprimality to the case of rational numbers. In Sec. 3.3, we introduce rational arrays and discuss when such arrays can be better suited. Depending on the aperture and number of sensors available, we discuss the possible advantages provided by rational arrays. We also provide a necessary and sufficient condition for steering vector invertibility in this section. Sec. 3.4 proposes rational coprime arrays and provides a unique identifiability result for such arrays when used with MUSIC. We then provide Monte-Carlo simulations comparing MSE and CRB of rational arrays and integer arrays. In Sec. 3.5, we extend these results for the case of restricted spatial scope. We then prove the theoretical results on steering vector invertibility and identifiability. In Sec. 3.6, we further generalize number theoretic concepts and related DOA estimation results to arrays whose sensors can be arbitrary real numbers. Sec. 3.7 demonstrates that adding a few rational location sensors to an otherwise integer array increases the number of resolvable sources through the difference coarray domain. This leads to rational arrays that satisfy the aperture constraint and at the same time identify $O(m^2)$ sources with m sensors. We also discuss the effect of mutual coupling in such situations and provide Monte-Carlo performance evaluations. Finally, Sec. 3.8 concludes the chapter and provides future directions. The content of this chapter is based on papers [88], [143]–[145].

3.2 Rational Number Theory: Some Basic Results

In this section, we explain how some basic definitions, properties, and results from integer number theory can be extended to the case of rational numbers. A real

number r is said to be rational if it can be expressed in the form $r = P/Q$, where $Q \neq 0$, and $P, Q \in \mathbb{Z}$. Furthermore, if $(P, Q) = 1$, P/Q is called the ‘reduced form’ or ‘lowest form’ of the rational number r .

When dealing with integers, the notions of gcd, lcm, and coprimality are commonly encountered. They also play an important role in determining some properties of the integer arrays in terms of sensor positions. For example, the necessary and sufficient condition for the invertibility of the steering vector [97], [126] of an integer array with sensors placed at locations $0, n_1, n_2, \dots, n_{m-1} \in \mathbb{N}$ is given by

$$\gcd(n_1, n_2, \dots, n_{m-1}) = 1. \quad (3.1)$$

Similarly, Euclid’s theorem of coprime numbers was used to obtain results pertaining to large ULA segments in the coarray of sparse coprime arrays (for instance, properties 1-5 from sec. VI of [35]). To introduce rational coprime arrays (Sec. 3.4) and develop some similar identifiability results for such arrays, we first extend these arithmetic notions of gcd and coprimality to rationals. Although some online sources like [127], [128] discuss gcd and lcm for rational numbers, here we establish more specific results that are useful in the context of array signal processing.

3.2.1 GCD and LCM of Rational Numbers

First, let us formally define the notion of ‘common divisors’ for a set of rational numbers.

Definition 1 (Common divisor of rationals). A rational number r is called a common divisor of rational numbers r_1, r_2, \dots, r_m if

$$r_i = K_i \cdot r, \quad \text{where } K_i \in \mathbb{Z}, \quad i = 1, 2, \dots, m. \quad (3.2)$$

A common rational divisor always exists for a set of rational numbers. For example, if $r_i = P_i/Q_i$ then $r = 1/\prod_{i=1}^m Q_i$ is always a common divisor. Now the gcd and lcm for rationals can be defined analogously to integers:

Definition 2 (gcd of rationals). The greatest common divisor (gcd) of m rational numbers r_1, r_2, \dots, r_m is defined as the largest possible positive rational number r such that

$$r_i = K_i \cdot r, \quad \text{where } K_i \in \mathbb{Z}, \quad i = 1, 2, \dots, m. \quad (3.3)$$

The gcd is denoted as $\gcd(r_1, r_2, \dots, r_m)$, or simply (r_1, r_2, \dots, r_m) .

Note that for r to be the gcd in the above definition, $(K_1, K_2, \dots, K_m) = 1$. If not, let $K' = (K_1, K_2, \dots, K_m) > 1$. Then $r' = rK' > r$ also satisfies the above gcd definition, which contradicts that r is the *greatest* common divisor.

Definition 3 (lcm of rationals). The least common multiple (lcm) of m rational numbers r_1, r_2, \dots, r_m is defined as the smallest possible positive rational number l such that

$$l = L_i \cdot r_i, \quad \text{where } L_i \in \mathbb{Z}, \quad i = 1, 2, \dots, m. \quad (3.4)$$

The lcm is denoted as $\text{lcm}(r_1, r_2, \dots, r_m)$. For l to be the lcm in the above definition, $(L_1, L_2, \dots, L_m) = 1$. The gcd and lcm of any m rational numbers always exist, and they are positive rational numbers themselves. They can be expressed in terms of numerators and denominators of the individual rational numbers as follows [127], [128]:

Fact 1 (Formula for gcd and lcm of rationals). Let $r_i = P_i/Q_i$, $i = 1, 2, \dots, m$ be positive rational numbers in their reduced form (i.e., $P_i, Q_i \in \mathbb{N}$, $(P_i, Q_i) = 1 \forall i$). Then,

$$\text{gcd}(r_1, r_2, \dots, r_m) = \frac{\text{gcd}(P_1, P_2, \dots, P_m)}{\text{lcm}(Q_1, Q_2, \dots, Q_m)}, \quad (3.5)$$

and

$$\text{lcm}(r_1, r_2, \dots, r_m) = \frac{\text{lcm}(P_1, P_2, \dots, P_m)}{\text{gcd}(Q_1, Q_2, \dots, Q_m)}. \quad (3.6)$$

Proof. Let $r = P/Q$ (with $P, Q \in \mathbb{N}$ and $(P, Q) = 1$) be a common rational divisor of r_1, r_2, \dots, r_m , i.e.,

$$r_i = \frac{P_i}{Q_i} = K_i \cdot \frac{P}{Q} \quad \text{where, } K_i \in \mathbb{N} \quad \forall i. \quad (3.7)$$

Now let $K_{i1}, K_{i2} \in \mathbb{N}$ be such that

$$K_{i2} = (K_i, Q), \text{ and } K_{i1} = K_i/K_{i2}, \quad (3.8)$$

so that $K_{i2}|Q$ and $(K_{i1}, Q) = 1$. Thus, we get

$$\frac{P_i}{Q_i} = \frac{PK_{i1}}{Q/K_{i2}} \quad \forall i. \quad (3.9)$$

Since $(P, Q) = 1$, $K_{i2}|Q$, and $(K_{i1}, Q) = 1$, we have $(PK_{i1}, Q/K_{i2}) = 1$. Since both LHS and RHS of Eq. (3.9) are in reduced form, we get

$$P_i = PK_{i1}, Q_i = Q/K_{i2} \text{ for some } K_{i1}, K_{i2} \in \mathbb{N}. \quad (3.10)$$

Since this is true for every i , we have that

$$P|P_i \quad \forall i, \quad \text{and} \quad Q_i|Q \quad \forall i. \quad (3.11)$$

For r to be the largest of all such possible divisors, one must choose the largest numerator P and the smallest denominator Q satisfying the above constraints. Thus, if $r = P/Q$ is the gcd, we must have

$$P = \gcd(P_1, P_2, \dots, P_m), \quad \text{and} \quad Q = \text{lcm}(Q_1, Q_2, \dots, Q_m). \quad (3.12)$$

This completes the proof for the formula in Eq. (3.5). One can similarly prove the formula (3.6) for the lcm of rational numbers. \square

It can further be shown that $(P, Q) = 1$ in the above proof so that the gcd $r = P/Q$ is indeed in the reduced form.

Example. Consider $r_1 = 4/5$, $r_2 = 2/7$, and $r_3 = 10/3$. We can calculate the gcd of these rational numbers using the formula (3.5) as follows:

$$\gcd(r_1, r_2, r_3) = \gcd(4, 2, 10) / \text{lcm}(5, 7, 3) = 2/105. \quad (3.13)$$

Indeed, $r_1 = 42 \cdot (2/105)$, $r_2 = 15 \cdot (2/105)$, and $r_3 = 175 \cdot (2/105)$. Note that the ‘quotients’ 42, 15, and 175 are coprime. Similarly, by (3.6), the lcm of the numbers is

$$\text{lcm}(r_1, r_2, r_3) = \text{lcm}(4, 2, 10) / \gcd(5, 7, 3) = 20/1. \quad (3.14)$$

Indeed, $25 \cdot r_1 = 20$, $70 \cdot r_2 = 20$, and $6 \cdot r_3 = 20$ with the ‘multipliers’ 25, 70, and 6 being coprime.

Now, consider a special case when only two numbers are under consideration. It is well known that for two integers a and b , $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$. A similar relation holds true for rational numbers as well. For two positive rational number $r_1 = P_1/Q_1$, $r_2 = P_2/Q_2$ with $(P_1, Q_1) = 1$ and $(P_2, Q_2) = 1$,

$$\gcd(r_1, r_2) \cdot \text{lcm}(r_1, r_2) = \frac{\gcd(P_1, P_2)}{\text{lcm}(Q_1, Q_2)} \cdot \frac{\text{lcm}(P_1, P_2)}{\gcd(Q_1, Q_2)} \quad (3.15)$$

$$= \frac{P_1 P_2}{Q_1 Q_2} = r_1 r_2. \quad (3.16)$$

Another interesting way to motivate the gcd and lcm of rational numbers is by extending the well-known prime factorization theorem [61], [146] to rational numbers by allowing negative powers in the exponents of primes. This extension of prime factorization and connection to the gcd definition is explained next.

3.2.2 Rational Numbers: Prime Factorization and GCD

According to the fundamental theorem of arithmetic, every integer greater than unity can be represented uniquely as a product of prime numbers (up to the ordering of factors). Let $\{a_k\}, k = 1, 2, \dots$ be the list of primes in increasing order. With this, the unique canonical representation of every positive integer n can be written as [61], [146]

$$n = \prod_{k=1}^{\infty} a_k^{m_k}, \quad m_k \in \mathbb{Z}, \quad m_k \geq 0, \quad (3.17)$$

where a finite number of the exponents m_k are non-zero. With this, it can be seen that for N positive integers n_1, n_2, \dots, n_N having prime factorization

$$n_i = \prod_{k=1}^{\infty} a_k^{m_{ik}}, \quad m_{ik} \in \mathbb{Z}, \quad m_{ik} \geq 0 \quad i = 1, 2, \dots, N, \quad (3.18)$$

the gcd and lcm can be expressed as

$$\gcd(n_1, n_2, \dots, n_N) = \prod_{k=1}^{\infty} a_k^{\min(m_{1k}, m_{2k}, \dots, m_{Nk})}, \quad (3.19)$$

$$\text{lcm}(n_1, n_2, \dots, n_N) = \prod_{k=1}^{\infty} a_k^{\max(m_{1k}, m_{2k}, \dots, m_{Nk})}. \quad (3.20)$$

Note. Eq. (3.19) and (3.20) for two integers n_1 and n_2 also provide a quick proof of the property that the product of gcd and lcm of two positive integers is the same as the product of the integers:

$$\gcd(n_1, n_2) \text{lcm}(n_1, n_2) = \prod_{k=1}^{\infty} a_k^{\min(m_{1k}, m_{2k})} \prod_{k=1}^{\infty} a_k^{\max(m_{1k}, m_{2k})} \quad (3.21)$$

$$= \prod_{k=1}^{\infty} a_k^{m_{1k}} a_k^{m_{2k}} = n_1 n_2. \quad (3.22)$$

Extension for rational numbers. The integer canonical factorization (3.17) can be extended to represent positive rational numbers r by allowing the exponents m_i to be negative as follows [146]:

$$r = \prod_{k=1}^{\infty} a_k^{m_i}, \quad m_i \in \mathbb{Z}. \quad (3.23)$$

The primes with positive exponents contribute to the numerator of the rational number, and the primes with negative exponents contribute to the denominator of

the rational number. Thus, every positive rational number other than unity has a unique prime factorization of the form (3.23). For example, $6/7 = 2^1 \cdot 3^1 \cdot 7^{-1}$, and $100/21 = 2^2 \cdot 3^{-1} \cdot 5^2 \cdot 7^{-1}$.

Now, for N rational numbers given by

$$r_i = \prod_{k=1}^{\infty} a_k^{m_{ik}}, \quad m_{ik} \in \mathbb{Z}, \quad i = 1, 2, \dots, N, \quad (3.24)$$

we can define the gcd and lcm similar to that in (3.19) and (3.20) as follows:

$$\gcd(r_1, r_2, \dots, r_N) = \prod_{k=1}^{\infty} a_k^{(\min_i m_{ik})}, \quad (3.25)$$

$$\text{lcm}(r_1, r_2, \dots, r_N) = \prod_{k=1}^{\infty} a_k^{(\max_i m_{ik})}. \quad (3.26)$$

It can be verified that these canonical definitions are equivalent to Definition 2 and Definition 3. Here we provide a proof for the equivalence of the gcd formulae (3.5) and (3.25):

Proof. Let the prime factorization of N rational numbers be

$$r_i = P_i/Q_i = \prod_{k=1}^{\infty} a_k^{m_{ik}}, \quad m_{ik} \in \mathbb{Z}, \quad i = 1, 2, \dots, N. \quad (3.27)$$

Thus, the numerators and denominators can be written as

$$P_i = \prod_{k:m_{ik} \geq 0} a_k^{m_{ik}} = \prod_{k=1}^{\infty} a_k^{\max(0, m_{ik})}, \quad (3.28)$$

and

$$Q_i = \prod_{k:m_{ik} \leq 0} a_k^{-m_{ik}} = \prod_{k=1}^{\infty} a_k^{-\min(0, m_{ik})}. \quad (3.29)$$

From the canonical formula for gcd of rationals (Eq. 3.25), we get

$$\gcd(r_1, r_2, \dots, r_N) = \prod_{k=1}^{\infty} a_k^{(\min_i m_{ik})} \quad (3.30)$$

$$= \prod_{k:\min_i m_{ik} \geq 0} a_k^{(\min_i m_{ik})} \cdot \left[\prod_{k:\min_i m_{ik} \leq 0} a_k^{(-\min_i m_{ik})} \right]^{-1} \quad (3.31)$$

$$= \prod_{k=1}^{\infty} a_k^{\min_i \max(0, m_{ik})} \cdot \left[\prod_{k=1}^{\infty} a_k^{\max_i -\min(0, m_{ik})} \right]^{-1} \quad (3.32)$$

$$= \gcd(P_1, \dots, P_N) \cdot [\text{lcm}(Q_1, \dots, Q_N)]^{-1}. \quad (3.33)$$

□

The equivalence of lcm formulae (3.6) and (3.26) can be proved similarly.

3.2.3 Coprimality of Rational Numbers

We know two integers a and b are coprime if $(a, b) = 1$. Note that for rationals r_1 and r_2 , $(r_1, r_2) = 1$ can only happen if $r_1, r_2 \in \mathbb{Z}$ in view of Definition 2. So, instead of using $(r_1, r_2) = 1$, we define two rational numbers to be coprime if $(r_1, r_2) \leq 1$. More generally:

Definition 4 (Coprime rationals). A set of m positive rational numbers r_1, r_2, \dots, r_m is said to be coprime if $(r_1, r_2, \dots, r_m) \leq 1$.

Let $r_i = P_i/Q_i$ (with $P_i, Q_i \in \mathbb{N}$ and $(P_i, Q_i) = 1 \forall i$). In view of the formula (3.5), r_i are coprime if and only if

$$\gcd(P_1, P_2, \dots, P_m) \leq \text{lcm}(Q_1, Q_2, \dots, Q_m). \quad (3.34)$$

i.e., the gcd of numerators is smaller than or equal to the lcm of denominators. Note that if $r_i \in \mathbb{N} \forall i$, $\text{lcm}(Q_1, Q_2, \dots, Q_m) = 1$, as $Q_i = 1 \forall i$. Thus the condition in Definition 4 becomes

$$\gcd(P_1, P_2, \dots, P_m) \leq 1, \quad (3.35)$$

which reduces to the integer coprime definition

$$\gcd(P_1, P_2, \dots, P_m) = 1 \quad (3.36)$$

because P_i -s are integers.

To the best of our knowledge, the notion of coprimality has not been extended to rational numbers before. The above definition unifies the coprimality for integers and rationals in the following sense: two or more numbers, either integers or rationals, are coprime if and only if they *cannot* be expressed as integer multiples of a number *larger* than unity. This unification serves as the motivation for the above definition. Furthermore, this coprimality definition is useful in characterizing steering vector invertibility (Theorem 1) and DOA identifiability with rational coprime arrays (Theorem 2).

Examples. With the above definition of coprimality, $7/3$ and $41/5$ are coprime because $(7/3, 41/5) = 1/15 < 1$. On the other hand, $64/3$ and $80/5$ are not, because $(64/3, 80/5) = 16/15 > 1$.

Now we enumerate some properties related to the coprimality of rationals:

1. If $r_i \leq 1$ for some i , then $P_i \leq Q_i$. Thus we have $\gcd(P_1, \dots, P_m) \leq P_i \leq Q_i \leq \text{lcm}(Q_1, \dots, Q_m)$, and hence $\gcd(r_1, \dots, r_m) \leq 1$, as expected.
2. Note that the gcd cannot be larger than $r_i - r_j$ for any $i \neq j$, as it has to ‘divide’ both the numbers r_i and r_j (in view of Definition 1). Thus, if $r_i - r_j \leq 1$ for any $1 \leq j < i \leq m$, then $\gcd(r_1, r_2, \dots, r_m) \leq 1$.
3. For the case of two rational numbers, it can be verified that a generalization of Euclid’s lemma (or Bézout’s identity) [61] holds. See Lemma 1 below.
4. Furthermore, a generalized form of Bézout’s identity for more than two numbers holds for the case of rational numbers as well. See Lemma 2 below.

Lemma 1 (Generalized Euclid’s Lemma). Two positive rational numbers r_1, r_2 are coprime if and only if

$$\exists a_1, a_2 \in \mathbb{Z} \text{ such that } 0 < a_1 r_1 + a_2 r_2 \leq 1. \quad (3.37)$$

Proof. Let $(r_1, r_2) = r$. Then,

$$r_i = K_i \cdot r, \quad i = 1, 2 \text{ and } (K_1, K_2) = 1. \quad (3.38)$$

Now, consider the expression

$$a_1 r_1 + a_2 r_2 = r(a_1 K_1 + a_2 K_2). \quad (3.39)$$

Only if part: Assume $0 < r \leq 1$. Since $(K_1, K_2) = 1$, by Euclid’s lemma (Bézout’s identity) for integers, we can find $b_1, b_2 \in \mathbb{Z}$ such that $b_1 K_1 + b_2 K_2 = 1$. Choosing $a_1 = b_1, a_2 = b_2$ in (3.39) gives

$$0 < a_1 r_1 + a_2 r_2 = r \leq 1. \quad (3.40)$$

If part: Let there be $a_1, a_2 \in \mathbb{Z}$ such that $0 < a_1 r_1 + a_2 r_2 = r' \leq 1$. Thus we get

$$r(a_1 K_1 + a_2 K_2) = r'. \quad (3.41)$$

Since $a_1 K_1 + a_2 K_2 \in \mathbb{N}$, $r \leq r' \leq 1$. This completes the proof of Lemma 1. \square

Lemma 2. Let r_1, r_2, \dots, r_m be positive rational numbers with the greatest common divisor r . Then there exist integers K_1, K_2, \dots, K_m such that

$$K_1 r_1 + K_2 r_2 + \dots + K_m r_m = r. \quad (3.42)$$

More generally, the set of all numbers of the form $K_1 r_1 + K_2 r_2 + \dots + K_m r_m$, where $K_i \in \mathbb{Z}$ is the set of all numbers that are integer multiples of r .

Lemma 2 can be proved in a similar fashion to Lemma 1, by using the generalization of Bézout's identity for more than two integers.

3.3 Theory of Rational Arrays

In this section, we formally introduce rational arrays and present some results. Rational arrays have the form

$$\mathbf{z} = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-1} \end{bmatrix}, \quad (3.43)$$

where

$$r_0 = 0 < r_1 < r_2 < \cdots < r_{m-1}, \quad (3.44)$$

and

$$r_i = P_i/Q_i, \quad P_i, Q_i \in \mathbb{N}, \quad (P_i, Q_i) = 1 \quad \forall i > 0. \quad (3.45)$$

Here, the first sensor r_0 is assumed to be located at the origin, without loss of generality. These arrays will be called *rational arrays* to contrast them with *integer arrays*, where the sensor positions are all integers.

Note that it is possible to rewrite the array (3.43) as

$$\mathbf{z} = \begin{bmatrix} n_0 d & n_1 d & n_2 d & \cdots & n_{m-1} d \end{bmatrix} \quad (3.46)$$

for some rational number d and integers n_i . Although this ‘scaled integer representation’ can be useful in some situations, our rational representation has some advantages. As we discuss next, we often operate under the setting where the aperture A and the number of sensors m are fixed (or given). Under these constraints, the task is to design an array that can provide a theoretical guarantee for identifiability and improve DOA estimation performance compared to standard integer arrays, including ULA. For this task, it is often convenient and straightforward to directly toy with rational placements of the sensors and come up with suitable arrays rather than to construct an integer array with possibly a much larger aperture and scale it down with some d . Interpreting rational arrays as larger integer arrays with some small d is possible, but such a presentation doesn't shed much light on why these arrays would perform better than integer arrays (with $d = \lambda/2 = 1$). For example, if aperture $A = 3$ and number of available sensors $m = 4$, the array $[0, 0.95, 2.05, 3]$ can be considered to be $0.05 * [0, 19, 41, 60]$. However, artificially constructing such integer locations does not help in understanding, for example, how this array would perform compared to the integer array $[0, 1, 2, 3]$. Furthermore, the rational

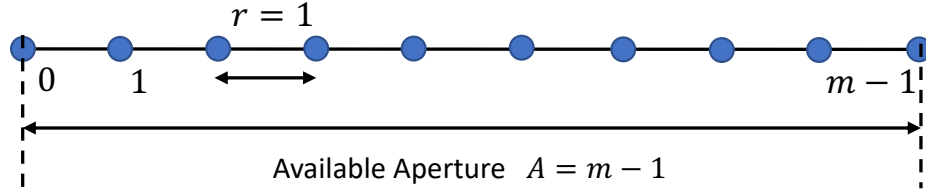


Figure 3.1: When the available array aperture A equals $m - 1$, the standard integer ULA can be used for DOA estimation.

framework used here generalizes naturally to arrays with non-rational sensor positions. As discussed later in Sec. 3.6, arbitrary (non-rational) arrays cannot, in general, be expressed as in (Eq. (3.46)) with integers n_i . But the results presented in Sec. 3.3 to 3.5 considering rational sensor positions with a constant $d = \lambda/2 = 1$ are easily generalizable (as Theorems 6 and 7) to arrays with arbitrary (real-valued) sensor positions.

We now discuss how rational arrays can have some benefit over integer arrays when m sensors are to be placed in a limited aperture of length A . The standard m -sensor integer ULA ($r_i = i$) can be used when $A = m - 1$ as shown in Fig. 3.1. ULAs have Vandermonde array manifold matrix (Eq. (1.9)), which ensures unambiguous identifiability of $m - 1$ sources. Furthermore, the Vandermonde structure also facilitates the application of Root-MUSIC and ESPRIT algorithms to the ULA output. When $A \neq m - 1$, two cases are possible, which are discussed next.

3.3.1 Situations where $A < m - 1$

In this case, the standard m -sensor integer ULA cannot be used. One possibility is to use a shorter ULA $\mathbf{z} = [0 \ 1 \ 2 \ \cdots \ \lfloor A \rfloor]$, that fits the available aperture as shown in Fig. 3.2(a). However, this does not use all the available sensors and restricts the number of resolvable sources to $\lfloor A \rfloor$. If we relax the integer constraint on sensor locations, we can use the following array:

$$\mathbf{z} = [0 \ r \ 2r \ \cdots \ (m-1)r], \quad r < 1. \quad (3.47)$$

This array is a scaled version of the standard integer ULA, as shown in Fig. 3.2(b). The shrinking factor $r < 1$ can be appropriately chosen to satisfy the aperture constraint. This shrunk rational ULA still has a Vandermonde array manifold matrix and has no ambiguity for sources in $[-90^\circ, 90^\circ)$. This allows the array to be used to identify $m - 1$ sources. Disadvantages of such shrinking include increased DOA estimation error and CRB due to reduced aperture [96] compared to the standard

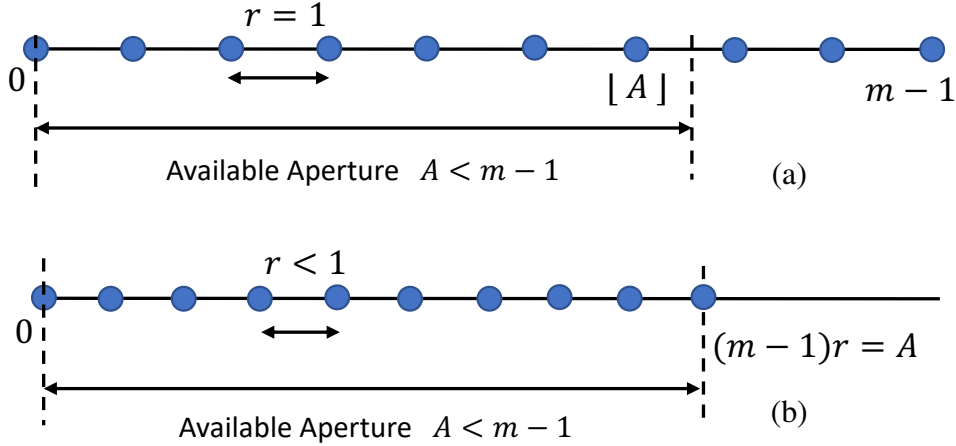


Figure 3.2: Situation where $A < m - 1$. (a) Integer ULA can only fit $\lfloor A \rfloor + 1$ sensors in the available aperture and can identify only $\lfloor A \rfloor$ DOAs. (b) A shrunk rational ULA with appropriately chosen interelement spacing $r < 1$ that satisfies the aperture constraint can identify $m - 1$ DOAs.

m -sensor integer ULA. One way to counterbalance this effect is to increase the number of snapshots.

However, contrary to what one may intuitively expect, when sources impinge on the array from directions away from the array normal, shrunk rational ULAs can provide more accurate DOA estimates than a standard integer ULA. In Fig. 3.3(a), one source is fixed at 80° , and we plot the analytical MSE in the frequency variables ω corresponding to the DOAs [122] as the second source is varied. When the second source is between -90° and -60° , the MSE for the integer ULA ($r = 1$) is worse than many rational ULAs. Note that MSE at 90° and -90° is the same for ULAs as $\mathbf{a}(\omega)$ is periodic in ω for ULAs. So sources that are far away from the array normal on both sides (like 85° and -85°) are difficult to differentiate based on their steering vectors $\mathbf{a}(\omega)$. At the extreme, the steering vectors corresponding to 90° and -90° are exactly the same for any integer array. This results in higher MSE when sources from both extreme ends of the interval $[-90^\circ, -90^\circ)$ are present. However, for rational arrays, $\mathbf{a}(\omega)$ may not be periodic with period 2π , and thus MSE does not increase even when extreme sources are present. This point about the periodicity of ω for rational arrays is further discussed in Sec. 3.3.3.

In addition to the theoretical increase in MSE due to reduced aperture, another effect to consider in practice is the increased mutual coupling between sensors due to array shrinking. For a demonstrative simulation, here we consider the coupling model as

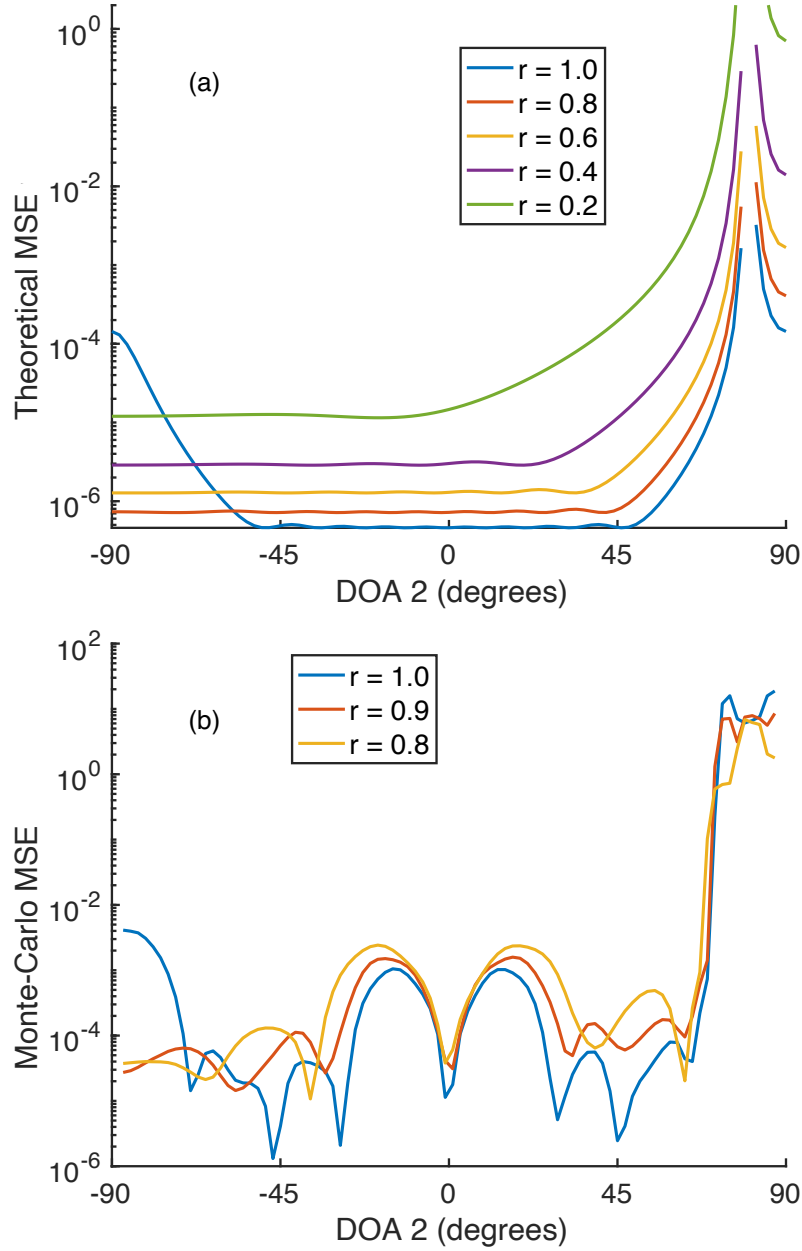


Figure 3.3: Comparison of (a) analytical MSE in the absence of mutual coupling and (b) numerical (Monte-Carlo) MSE in the presence of mutual coupling (Eq. (3.48)) when one DOA is fixed at 80° and the other DOA is varied. Integer ULA ($r = 1$) has higher MSE when sources impinge from the directions far away from the array normal on both sides.

in Eq. (2.27), and coupling coefficients of the form

$$c_d = \begin{cases} 0.3e^{j\pi/4}/d & \text{if } 0 < d \leq 1, \\ 0.3e^{j\pi/7}/d & \text{if } 1 < d \leq 3. \end{cases} \quad (3.48)$$

Fig. 3.3(b) shows the MSE of the DOA frequency variables ω averaged over 500 Monte-Carlo simulations for shrunk ULAs with different values of r in Eq. (3.47), in the presence of mutual coupling. We see that shrunk rational ULAs still have better MSE than the integer ULA when the second source is between -90° to -70° . It should be noted that the MSE for shrunk ULAs increases significantly if r is made smaller than 0.8 and very strong mutual coupling between sensors is considered.

In summary, compared to integer ULA, a shrunk rational ULA with $r = 0.9$ or $r = 0.95$ worsens the estimation error for DOAs close to the array normal only marginally but can improve the estimation error for DOAs away from the normal significantly, even in the presence of moderate mutual coupling.

3.3.2 Situations where $A > m - 1$

For this complementary scenario, we propose rational coprime arrays in Sec. 3.4. But first, consider two integer arrays that can be used in this case. We will show that these integer arrays have some limitations. These limitations motivate the rational coprime array proposed in the next section.

One integer array that can be used in this case is the standard m -sensor ULA, as its aperture $m - 1$ is smaller than the available aperture A as shown in Fig. 3.4(a). This ULA can identify up to $m - 1$ sources unambiguously. However, it utilizes only a part of the available aperture. It does not take advantage of the benefits of the larger aperture [96], such as a smaller mean squared error in DOA estimation and reduced CRB.

Another integer array that can be considered is the integer coprime array proposed in [97], [126]. These arrays have the form

$$\mathbf{z} = \begin{bmatrix} 0 & M & 2M & \cdots & (m-2)M & | & N \end{bmatrix}, \quad (3.49)$$

where M and N are coprime integers. The array consists of a sparse integer ULA with an extra sensor at the location N as shown in Fig. 3.4(b). Theorem 2 of [97], [126] shows that such an array can identify any $m - 2$ sources with MUSIC if no two sources are equivalent modulo $2\pi/M$ in the ω domain. Although this array cannot identify $m - 1$ sources like the standard ULA, M and N satisfying $(M, N) = 1$ can be chosen so as to span a larger aperture than the standard ULA. One possible choice is to take M to be $\lfloor A/(m-2) \rfloor$ and N to be the largest integer coprime to M that is smaller than A . This choice of M and N utilizes the available aperture almost entirely.

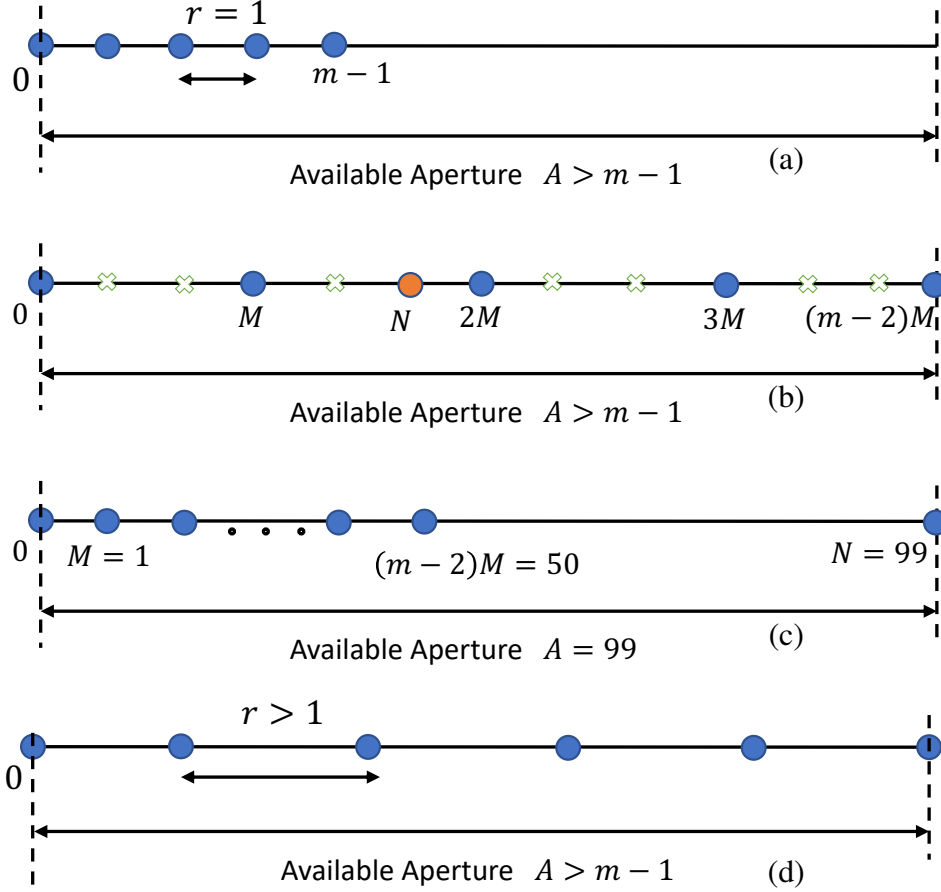


Figure 3.4: Situation where $A > m - 1$. (a) An integer ULA of m sensors cannot utilize the full available aperture. (b) An integer coprime array [97], [126] consisting of an $(m - 1)$ -sensor sparse ULA with interelement spacing M and an additional sensor located at N where $(M, N) = 1$ can be suitable for this case. (c) An example with $A = 99$ and $m = 27$, where the integer coprime array cannot utilize the available aperture well. Most of the sensors are concentrated in only half of the aperture. (d) We cannot simply use a sparse rational ULA because it cannot identify the DOAs unambiguously.

However, in addition to maximizing the array aperture, there are other factors that help reduce the MSE and CRB. [99], [147] show that increasing second-order and higher moments of sensor positions helps achieve this. One way of increasing second-order and higher moments of sensor positions is to distribute the sensors in a ‘spread-out’ fashion (while still satisfying identifiability conditions). Now, for the integer coprime array (3.49), some values of m and A might not allow a well ‘spread-out’ positioning of sensors because of the integer constraint on M and N . As an example, consider $m = 52$ and the available aperture $A = 99$. In this case,

only $M = 1$ is possible, and thus the ULA part of (3.49) is concentrated in only half of the aperture from 0 to 50 as shown in Fig. 3.4(c). We can still place the additional sensor at $N = 99$ so as to make use of the entire aperture. Instead of this, the rational array proposed in Sec. 3.4 spreads out the ULA part of the array to span the entire aperture, instead of just moving one sensor to the extreme. Thus, the rational counterpart (3.56) provides better DOA estimates than the integer coprime array (3.49,) as we shall see. It also offers some identifiability guarantees as we shall explain in Sec. 3.4.

Note that one may also consider expanding the m -sensor integer ULA by a factor $r > 1$ to span the available aperture as shown in Fig. 3.4(d). However, this creates ambiguity. The steering vector of such a sparse ULA is given by

$$\mathbf{a}(\omega) = \begin{bmatrix} 1 & e^{j\omega r} & e^{j\omega 2r} & \dots & e^{j\omega(m-1)r} \end{bmatrix}, \quad r > 1. \quad (3.50)$$

Note that for two sources ω_1 and ω_2 such that $\omega_1 - \omega_2 = 2\pi/r$, $\mathbf{a}(\omega_1) = \mathbf{a}(\omega_2)$. Hence, these sources cannot be distinguished from the array output. However, such sparse ULAs will be revisited later in Sec. 3.5 for their use in the case of restricted spatial scope.

3.3.3 Remark on Frequency Interpretation of ω

It is important to note that the quantity $e^{j\omega z}$ does not in general have a repetition interval of 2π in ω , i.e. $e^{j(\omega+2\pi)z} \neq e^{j\omega z}$, in general. Instead, $e^{j\omega z}$ is periodic with period $2\pi/z$ in ω . Thus, only when z is an integer, $e^{j\omega z}$ has a repetition interval of 2π with respect to the variable ω . Hence, when dealing with non-integer arrays (i.e. z_i are not integers), we cannot consider other shifted intervals of ω with length 2π , like $[0, 2\pi)$, as a substitute for the range $[-\pi, \pi)$. Furthermore, the naturally used concept in DSP about the equivalence of digital frequencies ω and $\omega \pm 2k\pi$ is no longer true. Thus, in this chapter, we plot the MUSIC spectrum with physical DOA variable θ on the X -axis instead of ω . Note that the MUSIC spectrum thus plotted for rational arrays does not always have the same value at angles -90° and 90° , unlike integer arrays.

3.3.4 Invertibility of Steering Vector

For any algorithm to be able to unambiguously identify the DOAs $\theta_1, \dots, \theta_D$ from the array output \mathbf{x} (Eq. (2.1)), it is necessary that

$$\mathbf{a}(\omega_1) \neq \mathbf{a}(\omega_2), \text{ for } -\pi \leq \omega_1 < \omega_2 < \pi. \quad (3.51)$$

In other words, the mapping $\omega \mapsto \mathbf{a}(\omega)$ should be invertible for $\omega \in [-\pi, \pi)$. It is well-known that invertibility of the manifold, while necessary, is not sufficient for unambiguous DOA identifiability. For integer arrays, the setwise coprimality of the sensor locations (Eq. (3.1)) was shown to be the necessary and sufficient condition for the invertibility of the steering vector [97], [126]. With the extended definition of coprimality to rational numbers, we have the following generalization.

Theorem 1 (Invertibility of steering vector). For $-\pi \leq \omega < \pi$, the steering vector $\mathbf{a}(\omega) = \begin{bmatrix} 1 & e^{j\omega r_1} & e^{j\omega r_2} & \dots & e^{j\omega r_{m-1}} \end{bmatrix}^T$ of the rational array in Eq. (3.43) is invertible if and only if the sensor locations r_i are coprime, i.e.

$$\gcd(r_1, r_2, \dots, r_{m-1}) \leq 1. \quad (3.52)$$

Or equivalently, when $r_i = P_i/Q_i$ are in their reduced forms,

$$\gcd(P_1, P_2, \dots, P_{m-1}) \leq \text{lcm}(Q_1, Q_2, \dots, Q_{m-1}). \quad (3.53)$$

The proof follows along similar lines to the integer case [97], [126], with appropriate modifications for rational sensor positions and corresponding use of gcd and coprimality definitions for rational numbers. In Sec. 3.5 we generalize this result further to the case when signals are impinging from a restricted spatial scope on the array (Theorem 4) and provide the proof for this further generalized version of the result.

Note that for the special case of integer array, $r_i = n_i \in \mathbb{Z} \forall i$, the above condition reduces to (3.1). If $r_i \leq 1$ for any i or $r_i - r_j \leq 1$ for some $1 \leq j < i \leq m-1$, the condition (3.52) is readily satisfied, given the gcd properties from Sec. 3.2.

Examples. The steering vector corresponding to the array

$$\mathbf{z} = \begin{bmatrix} 0 & 10/3 & 25/3 & 10 & 45/4 \end{bmatrix} \quad (3.54)$$

is invertible, because $\gcd(10/3, 25/3, 10, 45/4) = 5/12 < 1$. On the other hand, the steering vector corresponding to the array

$$\mathbf{z} = \begin{bmatrix} 0 & 9/4 & 45/8 & 81/8 & 45/4 & 18 \end{bmatrix} \quad (3.55)$$

is not invertible because $\gcd(9/4, 45/8, 81/8, 45/4, 18) = 9/8 > 1$.

3.4 Rational Coprime Arrays

For the case considered in Sec. 3.3.2 where aperture $A > m - 1$, we propose to use the following m -sensor rational array:

$$\mathbf{z} = \begin{bmatrix} 0 & r_1 & 2r_1 & \cdots & (m-2)r_1 & | & r_2 \end{bmatrix}. \quad (3.56)$$

Here, r_1 and r_2 are rational coprime numbers, i.e., $(r_1, r_2) \leq 1$. The array is a sparse rational ULA with $m - 1$ elements, plus an extra element appended at r_2 . Note that the steering vector of this array is invertible (see Theorem 1) as

$$\gcd(r_1, 2r_1, \dots, (m-2)r_1, r_2) = \gcd(r_1, r_2) \leq 1. \quad (3.57)$$

While this array ‘spreads-out’ the ULA part compared to its integer counterpart (3.49), we must still ensure that it can identify the DOAs unambiguously. It is well-known [96] that the MUSIC algorithm can identify D sources corresponding to $\omega_1, \dots, \omega_D$ unambiguously, i.e., without producing any false peaks, if and only if the augmented array manifold matrix of the array given by

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{a}(\omega_1) & \mathbf{a}(\omega_2) & \dots & \mathbf{a}(\omega_D) & | & \mathbf{a}(\omega_{D+1}) \end{bmatrix} \quad (3.58)$$

is full column rank for all ω_{D+1} in the range $[-\pi, \pi)$ that are distinct to any of $\omega_1, \omega_2, \dots, \omega_D$. This condition ensures that no other steering vector $\mathbf{a}(\omega)$ lies in the column span of the steering vectors corresponding to the DOAs, and thus no false peaks are produced. It can be verified that the result holds valid for *any* linear array, and in particular for the rational arrays under consideration. Using the rank condition for \mathbf{A}_{aug} , we can arrive at the following identifiability result similar to Theorem 2 of [97], [126].

Theorem 2 (Rational coprime array: Identifiability with MUSIC). Consider an m -sensor rational array as in Eq. (3.56), with $(r_1, r_2) \leq 1$. Let there be $D \leq m - 2$ sources with distinct DOAs $\theta_1, \theta_2, \dots, \theta_D$. If the frequency variables $\omega_1, \omega_2, \dots, \omega_D$, where $\omega_i = \pi \sin \theta_i$, are distinct modulo $2\pi/r_1$, then the MUSIC algorithm will identify the DOAs without producing any false peaks.

Note that $\omega_1, \dots, \omega_D$ are said to be distinct modulo $2\pi/r_1$ if

$$\omega_i - \omega_l \neq 2\pi k/r_1, \quad i \neq l \quad (3.59)$$

for any integer k . Note that (3.59) is equivalent to the condition

$$e^{j\omega_i r_1} \neq e^{j\omega_l r_1}, \quad i \neq l. \quad (3.60)$$

In practice, such distinctness is satisfied with probability one when the spatial distribution of sources is assumed to be uniform in $[-90^\circ, 90^\circ]$. However, if the two sources are approximately equivalent modulo $2\pi/r_1$, the MSE can increase substantially as shown in Fig. 3.11 later. The severity with which MSE increases is experimentally found to be very sensitive to the position of the extra sensor at r_2 relative to the sparse ULA part of the array. In Sec. 3.4.2, we will further discuss the effect of sensor location r_2 on DOA estimation performance and provide two ways of constructing rational coprime arrays that mitigate the worsening of DOA estimation error.

Similar to Theorem 1, we generalize Theorem 2 further to the case when signals are impinging from a restricted spatial scope in Sec. 3.5, and also provide a detailed proof parallel to [97], [126].

3.4.1 DOA Estimation Examples with the Proposed Rational Coprime Arrays

In this section, we demonstrate that it is indeed possible to unambiguously identify sources with the proposed rational coprime arrays by presenting simulation examples. We also illustrate the scenario where rational arrays are better suited.

Example 1. Consider $m = 8$ sensors and an available aperture $A = 11$. We consider the following three arrays under this setting:

$$\mathbf{z}_1 = \left[0 \quad 11/6 \quad 22/6 \quad 33/6 \quad 44/6 \quad 55/6 \quad 11 \quad | \quad 26/5 \right], \quad (3.61)$$

$$\mathbf{z}_2 = \left[0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \right], \quad (3.62)$$

$$\mathbf{z}_3 = \left[0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad | \quad 11 \right]. \quad (3.63)$$

Of the three arrays, \mathbf{z}_1 is a rational coprime array of the form (3.56), with $r_1 = 11/6$ and $r_2 = 26/5$ satisfying $(r_1, r_2) = 1/30 < 1$. Hence according to Theorem 2, it can identify any $6(= m - 2)$ sources that satisfy condition (3.59). Arrays \mathbf{z}_2 and \mathbf{z}_3 are two possible integer alternatives considered in Sec. 3.3.2. Array \mathbf{z}_2 is a standard 8-sensor integer ULA that can identify up to 7 sources unambiguously, but it only utilizes a part of the entire available aperture. Array \mathbf{z}_3 has a 7-sensor ULA segment and an extra sensor at location 11 in order to utilize the entire available aperture. The ULA part with 7 sensors still ensures that it can unambiguously identify any 6 sources. Also note that \mathbf{z}_3 can be viewed as an array of the form (3.49) with $M = 1$ and $N = 11$. It is not possible to use $M \geq 2$ here, as that would violate the aperture constraint.

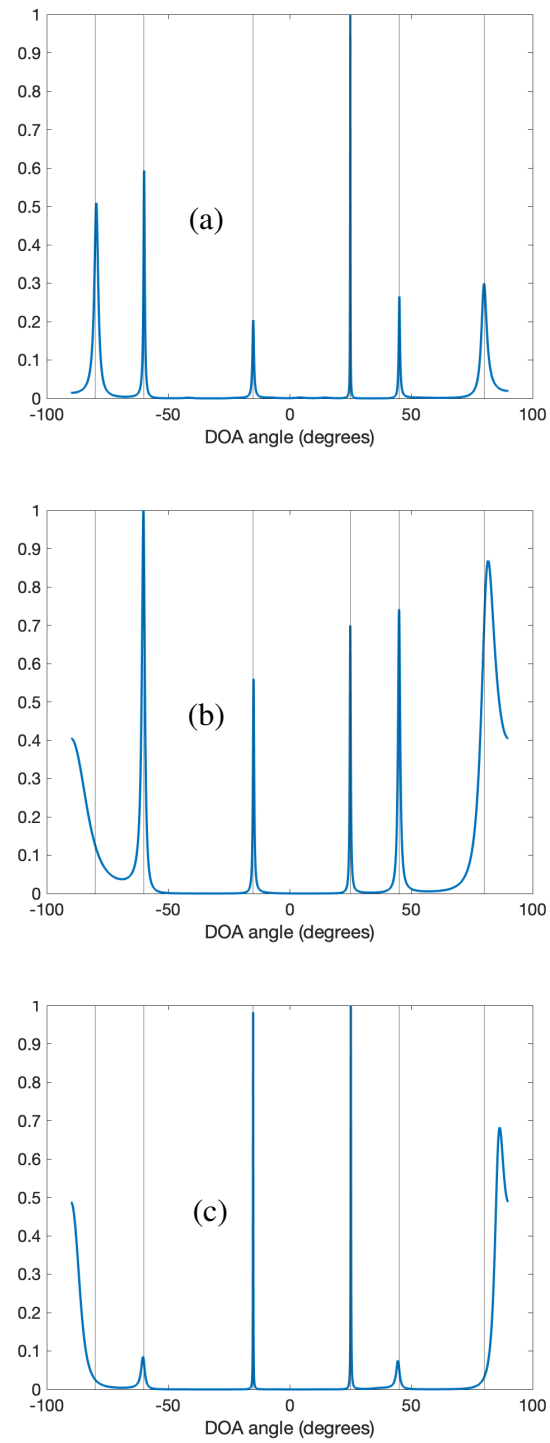


Figure 3.5: Identifying 6 sources from Example 1. MUSIC spectrum obtained with (a) rational coprime array \mathbf{z}_1 , (b) standard ULA \mathbf{z}_2 , and (c) modified ULA \mathbf{z}_3 . True DOAs are shown in black vertical lines.

Now consider 6 uncorrelated sources with DOAs $-80^\circ, -60^\circ, -15^\circ, 25^\circ, 45^\circ$, and 80° impinging on these arrays. The sources have SNR of 0 dB, and $K = 600$ snapshots are used. Fig. 3.5 shows the MUSIC spectrum produced by these arrays. It can be seen that the integer arrays \mathbf{z}_2 and \mathbf{z}_3 do not correctly identify the source at -80° , and have significant DOA estimation error for the source at 80° . On the other hand, our proposed rational coprime array \mathbf{z}_1 produces six clear peaks in the spectrum that are close to the true DOAs and thus provides accurate DOA estimates.

Example 2. For this example, we consider $m = 13$ sensors and an available aperture of 20. We compare three arrays, similar to those considered in the previous example:

$$\mathbf{z}_4 = \begin{bmatrix} 0 & 20/11 & 2 \cdot 20/11 & \cdots & 20 & | & 37/5 \end{bmatrix}, \quad (3.64)$$

$$\mathbf{z}_5 = \begin{bmatrix} 0 & 1 & 2 & 3 & \cdots & 12 \end{bmatrix}, \quad (3.65)$$

$$\mathbf{z}_6 = \begin{bmatrix} 0 & 1 & 2 & 3 & \cdots & 11 & | & 20 \end{bmatrix}. \quad (3.66)$$

Now consider 11 uncorrelated sources with nearby DOAs $-25^\circ, -20^\circ, \dots, 20^\circ, 25^\circ$ and SNR of 20 dB impinging on the arrays. $K = 1000$ snapshots are used. Fig. 3.6 shows the MUSIC spectrum produced by the three arrays. It can be seen that the proposed rational array \mathbf{z}_4 clearly estimates 11 sources that are close to the true DOAs, whereas the integer arrays \mathbf{z}_5 and \mathbf{z}_6 do not identify all the sources correctly, and produce an incorrect number of peaks in the spectrum.

These examples clearly demonstrate the advantages of rational coprime arrays over other integer arrays. Sparse distribution of sensors in a given aperture helps rational coprime arrays to identify the correct number of sources (example 2), along with smaller DOA estimation error (example 1) with MUSIC compared to the integer-array alternatives.

3.4.2 Which Rational Coprime Array is Better?

The rational coprime array in Eq. (3.56) has two design parameters, namely, the rational numbers r_1 and r_2 . For simulations in the previous subsection, we considered the available aperture A and the available number of sensors m satisfying $A > m - 1$. To design the rational array, we chose $r_1 = A/(m - 2)$, and r_2 to be any number coprime to r_1 . However, we found that the CRB and the DOA estimation error of the array vary considerably depending on the value of r_2 . Especially when the sources become approximately equivalent modulo $2\pi/r_1$, the increase in the DOA estimation error was found to be very sensitive to the location of the extra

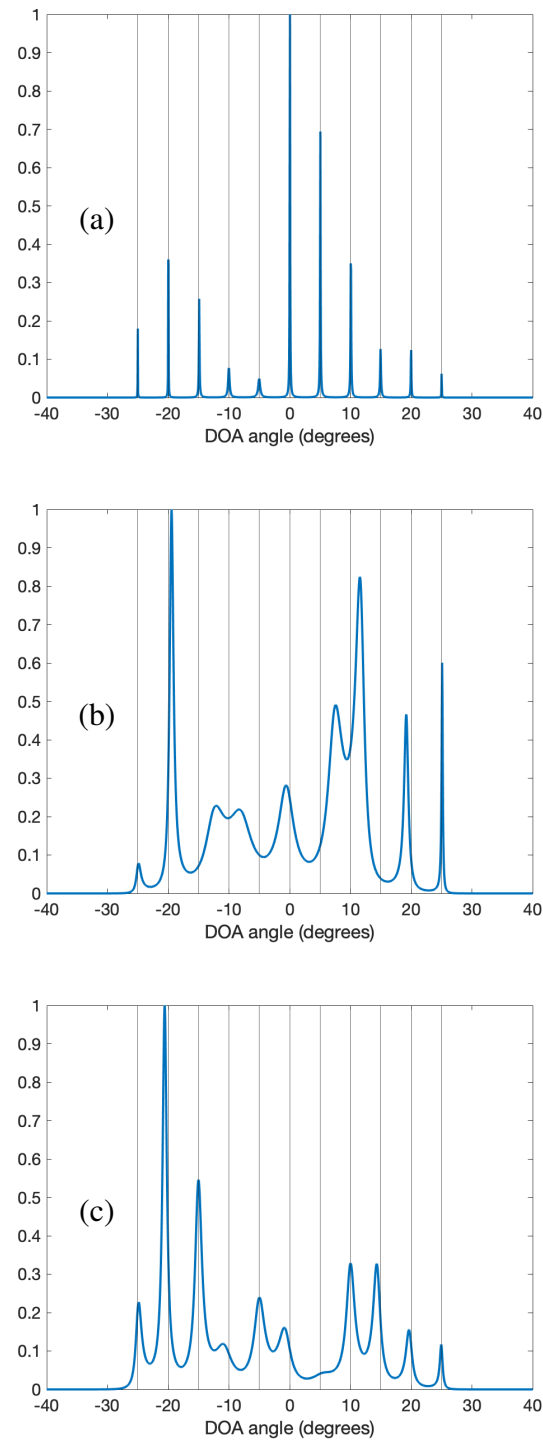


Figure 3.6: Identifying 11 closely spaced sources from Example 2. MUSIC spectrum obtained with (a) rational coprime array \mathbf{z}_4 , (b) standard ULA \mathbf{z}_5 , and (c) modified ULA \mathbf{z}_6 . True DOAs are shown in black vertical lines.

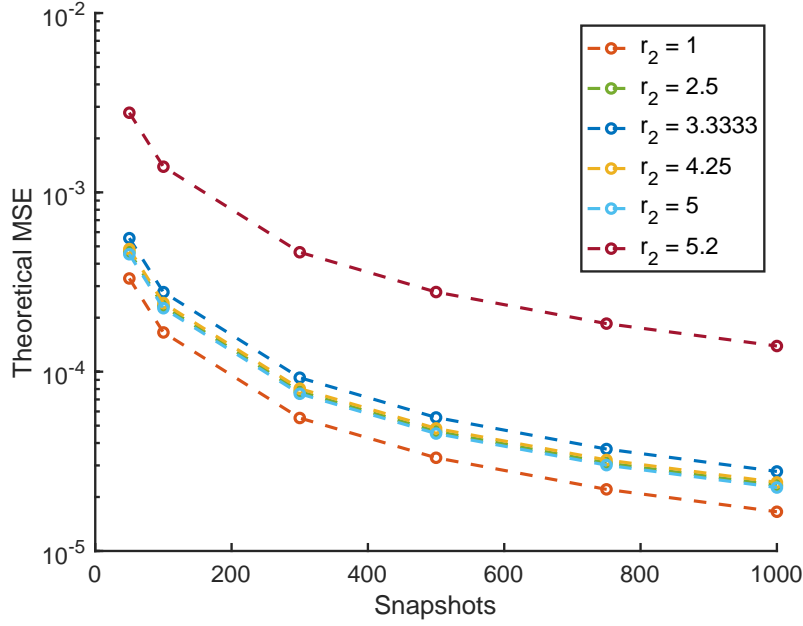


Figure 3.7: Effect of the location of r_2 of array \mathbf{z}_1 from *Example 1* on the MSE. Choosing $r_2 = 1$, or equivalently $r_2 = A - 1$ leads to the smallest MSE.

sensor at r_2 . In this section, we demonstrate this effect of the location r_2 on DOA estimation errors and discuss two ways of choosing r_1 and r_2 to mitigate this effect.

Consider the situation in *Example 1*. In Fig. 3.7, we keep fixed $r_1 = 11/6$, and vary the location of r_2 to see its effect on the analytical MSE [122] of the array. Notice that the array \mathbf{z}_1 with $r_2 = 26/5$ is in fact not an optimal array. The array with $r_2 = 1$ (or equivalently $r_2 = A - 1 = 10$, due to symmetry) was found to have the smallest MSE. We experimentally found this to be the case for many other settings of A , m , and DOAs configurations. Based on this observation, we suggest two specific rational array designs for a given aperture A and number of sensors m , both of which are found to have similar DOA estimation performance:

Design 1: $r_1 = A/(m - 2)$, and $r_2 = A - 1$, so that

$$\mathbf{z} = \left[0 \quad \frac{A}{m-2} \quad \frac{2A}{m-2} \quad \cdots \quad A \quad | \quad A - 1 \right]. \quad (3.67)$$

Design 2: $r_1 = (A - 1)/(m - 2)$, and $r_2 = A$, so that

$$\mathbf{z} = \left[0 \quad \frac{A-1}{m-2} \quad \frac{2(A-1)}{m-2} \quad \cdots \quad A - 1 \quad | \quad A \right]. \quad (3.68)$$

It can be verified that r_1 and r_2 thus chosen are always guaranteed to be rationally coprime, i.e., $(r_1, r_2) \leq 1$. For array (3.68), if $(A - 1)/(m - 2)$ is the irreducible

form of r_1 (i.e. $A - 1$ and $m - 2$ are coprime) then by (3.5), we have

$$\gcd(r_1, r_2) = \frac{\gcd(A - 1, A)}{\text{lcm}(m - 2, 1)} = \frac{1}{m - 2} \leq 1. \quad (3.69)$$

If $A - 1$ and $m - 2$ are not coprime, let $r_1 = a/b$ be the irreducible form of r_1 , with $(a, b) = 1$. Then a is a divisor of $A - 1$, so we have $(a, A) \leq (A - 1, A) = 1$, thus implying $(a, A) = 1$. With this,

$$\gcd(r_1, r_2) = \frac{\gcd(a, A)}{\text{lcm}(b, 1)} = \frac{1}{b} \leq 1. \quad (3.70)$$

Similar arguments also show that r_1 and r_2 are coprime for *Design 1* as well. We will also show in Sec. 3.4.4 that the rational coprime arrays thus designed have better performance than most rational coprime arrays with $r_1 = A/(m - 2)$ and randomly chosen r_2 , especially around the ambiguity regions (i.e. when sources are approximately equivalent modulo $2\pi/r_1$). Another point to note is that these array designs do not have a large mutual coupling between adjacent sensors. *Design 2* has no sensor pairs with a distance smaller than 1. *Design 1* has no sensor pair with a distance smaller than 1 if $A \geq 2(m - 2)$, and has exactly one sensor pair with a distance smaller than 1 if $A < 2(m - 2)$.

Note that here it was much more convenient to directly toy with the rational sensor locations to design an array. It was not possible to artificially consider our rational array as a scaled version of some integer array, as in (3.46,) to design a value of r_2 that guarantees coprimality of r_1 and r_2 . As r_2 is changed, the equivalent d and corresponding integer sensor locations n_i change very haphazardly and are not useful to find an optimal r_2 for our rational coprime array.

3.4.3 Root-MUSIC for Rational Arrays

Even when the MUSIC spectrum of rational coprime array provably produces peaks close to the true DOA values (Theorem 2), the accuracy of the DOAs estimated through the MUSIC peak search depends on the grid size on which $P(\omega)$ from Eq. (2.11) is evaluated. Thus, for fair Monte-Carlo MSE comparisons, using a gridless DOA estimation method for rational arrays is desired. Two gridless DOA estimation methods that avoid grid-based DOA search are popular in the literature, namely, root-MUSIC [85] and ESPRIT [11] (see Sec. 2.2). Non-uniform rational arrays, and in particular rational coprime arrays, do not have shift invariance in the sensor positions, which is required for applying ESPRIT. Root-MUSIC (Eq. (2.15)), on the other hand, is usually used for ULAs. The arguments of the D roots of $Q(z)$ in

Eq. (2.15) that are inside the unit circle and closest to the unit circle in the complex z -plane provide estimates of the DOAs.

For integer arrays ($r_i = n_i \in \mathbb{N}$) that are not ULAs, root-MUSIC can in principle still be applied by substituting $\mathbf{v}(z) = \begin{bmatrix} 1 & z^{n_1} & z^{n_2} & \dots & z^{n_{m-1}} \end{bmatrix}^T$ in Eq. (2.15), as $Q(z)$ is still a polynomial (with integer powers of z). Although the polynomial degree can be large now, if the augmented manifold matrix has full column rank, the roots of $Q(z)$ closest to the unit circle in the z -plane will still correspond to the true DOAs when SNR is not too low. However, for rational arrays, r_i are not integers, and $Q(z)$ is no longer a polynomial. To handle this, we substitute $z = e^{j\omega r}$ instead of $z = e^{j\omega}$ in $\mathbf{a}(\omega)$, where $r = (r_1, \dots, r_{m-1}) \leq 1$ so that

$$\mathbf{v}(z) = \begin{bmatrix} 1 & z^{r_1/r} & z^{r_2/r} & \dots & z^{r_{m-1}/r} \end{bmatrix}^T. \quad (3.71)$$

Substituting this $\mathbf{v}(z)$ in Eq. (2.15) produces a polynomial $Q(z)$ with integer powers, as r_i/r are integers (see Definition 2). This substitution may create a polynomial of a very large degree, especially if r is small, which increases the computations required to find all the zeros of $Q(z)$. The closer r is to 1, the lower is the degree of the polynomial $Q(z)$, and the lower is the computational complexity. Note that since we substituted $z = e^{j\omega r}$, we only consider the zeros of $Q(z)$ having phase values in the range $[-\pi r, \pi r]$. But since $Q(z)$ can typically have a large degree, we get many more than D zeros having phase values in the range $[-\pi r, \pi r]$. It is possible that some of the spurious zeros are closer to the unit circle than the zeros corresponding to the true DOAs, especially when SNR or the number of snapshots is low. To handle this, we first find $L \geq D$ zeros of $Q(z)$ inside the unit circle and closest to the unit circle having phase values in the range $[-\pi r, \pi r]$. We then multiply the phase values of the L zeros by $1/r$ to get DOA candidates $\{\hat{\omega}_l\}_{l=1}^L \in [-\pi, \pi]$. Now we evaluate MUSIC spectrum $P(\omega)$ from Eq. (2.11) at each candidate DOA $\{\hat{\omega}_l\}_{l=1}^L$ corresponding to each of the $L \geq D$ zeros closest to the unit circle. The $\hat{\omega}_l$ -s that produce D largest values of $P(\hat{\omega}_l)$ are taken to be the final DOA estimates. In practice, we found that using $L = D$ is enough for most combinations of DOAs when SNR is not too low, and the number of snapshots K is large enough.

As an example, Fig. 3.8 shows that six DOAs -85° , -40° , -20° , 25° , 50° , and 75° with SNR 3 dB are correctly estimated by root-MUSIC with $L = D$ using rational coprime array \mathbf{z}_{10} (explained in the next section) when $K = 400$ snapshots are used but not when $K = 100$ snapshots are used. In the next section, all the numerical (Monte-Carlo) simulations to calculate MSE for rational arrays are based on this adaptation of root-MUSIC with $L = D$.

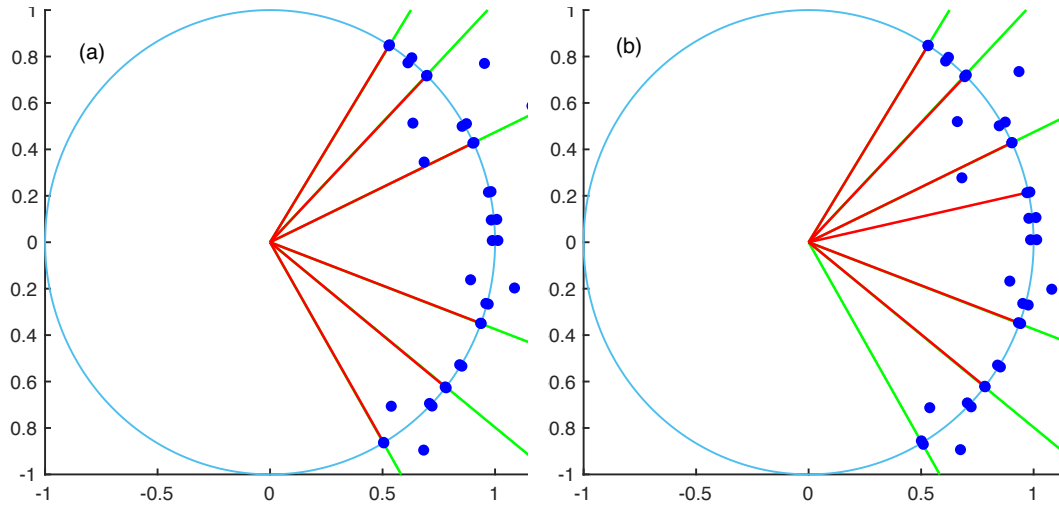


Figure 3.8: Applying root-MUSIC to rational array \mathbf{z}_{10} . Phases of six zeros closest to the unit circle (shown in red lines) correctly estimate the DOAs with (a) $K = 400$ but not with (b) $K = 100$. Green lines correspond to scaled values of true DOAs, i.e. $\{r\omega_i\}_{i=1}^6$, where $r = (r_1, r_2) = 1/3$, and blue dots denote the zeros of $Q(z)$ in the sector $[-\pi r, \pi r]$.

3.4.4 Performance Evaluation of Rational Coprime Arrays

In this section, we quantitatively compare the DOA estimation performance of rational coprime arrays with integer array alternatives. Consider an example where aperture $A = 17$, and $m = 8$ sensors are available. Consider the following arrays:

$$\mathbf{z}_7 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix} \quad (3.72)$$

$$\mathbf{z}_8 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & | & 17 \end{bmatrix} \quad (3.73)$$

$$\mathbf{z}_9 = \begin{bmatrix} 0 & 2 & 4 & 6 & 8 & 10 & 12 & | & 17 \end{bmatrix} \quad (3.74)$$

$$\mathbf{z}_{10} = \begin{bmatrix} 0 & 8/3 & 16/3 & 8 & 32/3 & 40/3 & 16 & | & 17 \end{bmatrix} \quad (3.75)$$

Array \mathbf{z}_7 is the standard integer ULA, and array \mathbf{z}_8 is the modified ULA, like the previous examples. Array \mathbf{z}_9 is an integer coprime array (3.49) with $M = 2$ and $N = 17$. It can identify any 6 ($= m - 2$) DOAs that are distinct modulo $2\pi/M$ [97], [126]. Note that it is not possible to use $M \geq 3$ here, as that would violate the aperture constraint. Array \mathbf{z}_{10} is a rational coprime array of the form (3.68), as proposed in *Design 2* of sec. 3.4.2, and can identify any 6 ($= m - 2$) that are distinct modulo $2\pi/r_1$ (Theorem 2). Now consider 6 nearby uncorrelated sources with DOAs in the range -15° to 15° impinging on these arrays. The sources have an SNR of 5 dB, and $K = 400$ snapshots are used. Fig. 3.9 shows the MUSIC

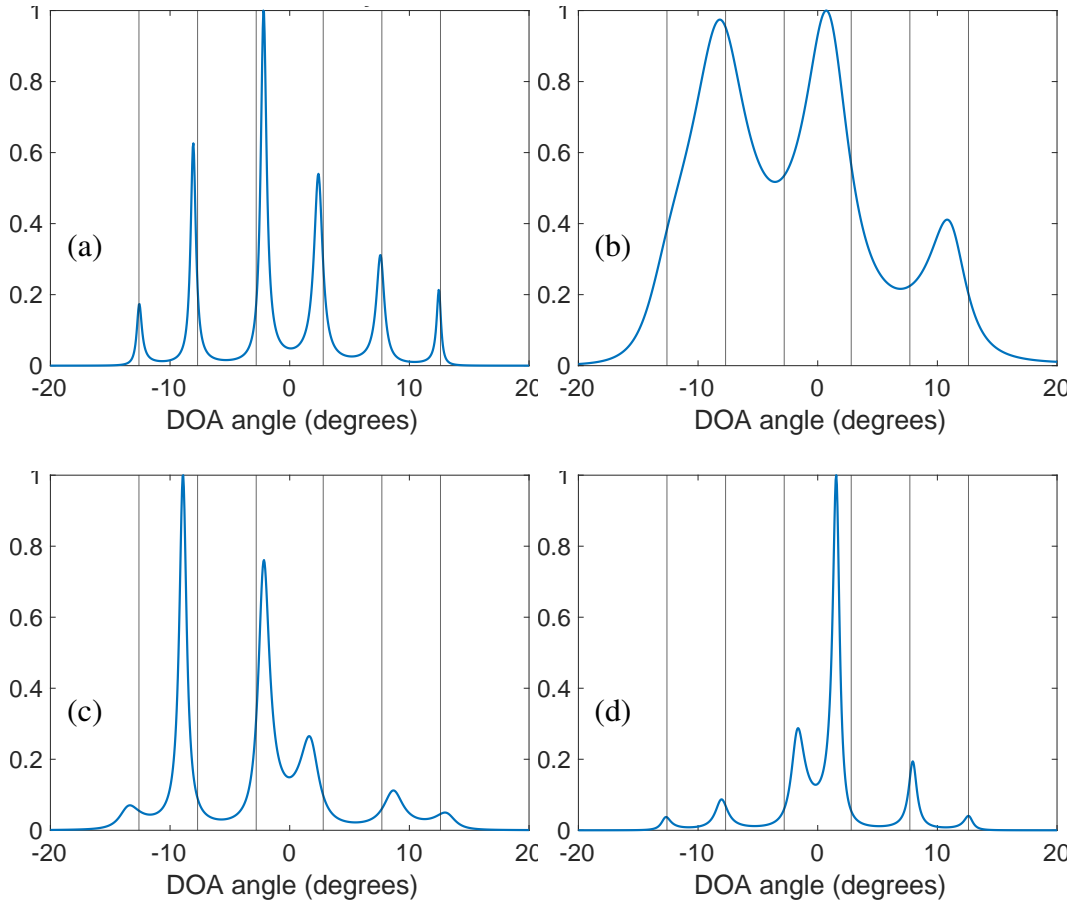


Figure 3.9: Identifying 6 nearby DOAs -12.6° , -7.7° , -2.8° , 2.8° , 7.7° , 12.6° when $m = 8$ and $A = 17$. MUSIC spectrum obtained with (a) integer ULA \mathbf{z}_7 , (b) modified ULA \mathbf{z}_8 , (c) integer coprime array \mathbf{z}_9 , and (d) rational coprime array \mathbf{z}_{10} . True DOAs are shown in black vertical lines.

spectrum produced by these arrays. We see that the rational coprime array provides six clear peaks that are easy to identify, unlike the other three integer arrays.

For numerical comparison of the arrays, we consider CRB, analytical MSE, and experimental (Monte-Carlo) MSE of the estimated frequency variables ω . The CRB and analytical MSE expressions from [122] are used as they are applicable to general non-integer arrays as well. To calculate experimental MSE, we perform 1000 Monte-Carlo simulations for each setting, with randomly generated white Gaussian noise and source amplitude realizations to generate received signal snapshots $\mathbf{x}[k]$. We then apply root-MUSIC with $L = D$ (see Sec. 3.4.3) using each of the four arrays \mathbf{z}_7 to \mathbf{z}_{10} to estimate the DOAs.

First, consider six uncorrelated signals with DOAs -18° , -11° , -4° , 4° , 11° , and

18° . In Fig. 3.10(a), the number of snapshots is fixed at $K = 100$ and the SNR is varied. The rational coprime array \mathbf{z}_{10} has the lowest CRB and analytical MSE, followed next by the integer coprime array \mathbf{z}_9 . The experimental MSE values for the rational coprime array \mathbf{z}_{10} are very close to CRB even at lower SNRs, unlike any of the other arrays. For this challenging scenario of six closely spaced DOAs, the theoretical MSE and CRB for ULA \mathbf{z}_7 are very large for most SNR values. However, as estimated $\hat{\omega}$ always lies in $[-\pi, \pi)$, the experimental MSE saturates and does not grow indefinitely. In Fig. 3.10(b), SNR is fixed at 0 dB, and the number of snapshots is varied. We see that the rational coprime array \mathbf{z}_{10} is clearly superior to all other arrays, providing about two orders of magnitude improvement in the experimental (Monte-Carlo) MSE over the other arrays. The experimental MSE for rational coprime array \mathbf{z}_{10} is close to CRB even with $K = 50$ snapshots, whereas other arrays require a much higher number of snapshots to achieve this.

Next, in Fig. 3.11, we compare theoretical MSE for two source scenarios when one source is fixed at $\theta = 30^\circ$ and the other source is varied from -90° to 90° . SNR of 0 dB and $K = 600$ is assumed. Here, for comparison, we also consider two other rational coprime arrays. We consider an array designed based on *Design 1* with $r_1 = A/(m - 2) = 17/6$ and $r_2 = A - 1$:

$$\mathbf{z}_{11} = \begin{bmatrix} 0 & 17/6 & 17/3 & 17/2 & 34/3 & 85/6 & 17 & | & 16 \end{bmatrix}, \quad (3.76)$$

and a sub-optimal rational coprime array with $r_1 = A/(m - 2) = 17/6$ and a randomly chosen $r_2 = 25/3$:

$$\mathbf{z}_{12} = \begin{bmatrix} 0 & 17/6 & 17/3 & 17/2 & 34/3 & 85/6 & 17 & | & 25/3 \end{bmatrix}. \quad (3.77)$$

The theoretical MSE plots shown in Fig. 3.11(a) reveal many interesting features. As expected, the MSE for all arrays becomes very large when the two DOAs get close. However, the MSE does not decrease monotonically as the separation between two DOAs increases. In addition to small oscillations seen for all arrays, note that the rational coprime arrays \mathbf{z}_{10} to \mathbf{z}_{12} show ‘bumps’ in their MSE when the second DOA is around -70° and -10° . These regions correspond to the ω variables of two sources becoming approximately equivalent modulo $2\pi/r_1$ of the respective arrays (see Theorem 2). As mentioned in the remarks following Theorem 2, we can clearly see that the severity of these ‘bumps’ depends on the rational array design, and in particular, the value of r_2 . As can be seen here, the MSE in these ‘bump’ regions can become significantly large for some sub-optimal values of r_2 . However, the rational arrays \mathbf{z}_{10} and \mathbf{z}_{11} with our proposed designs from the previous subsection

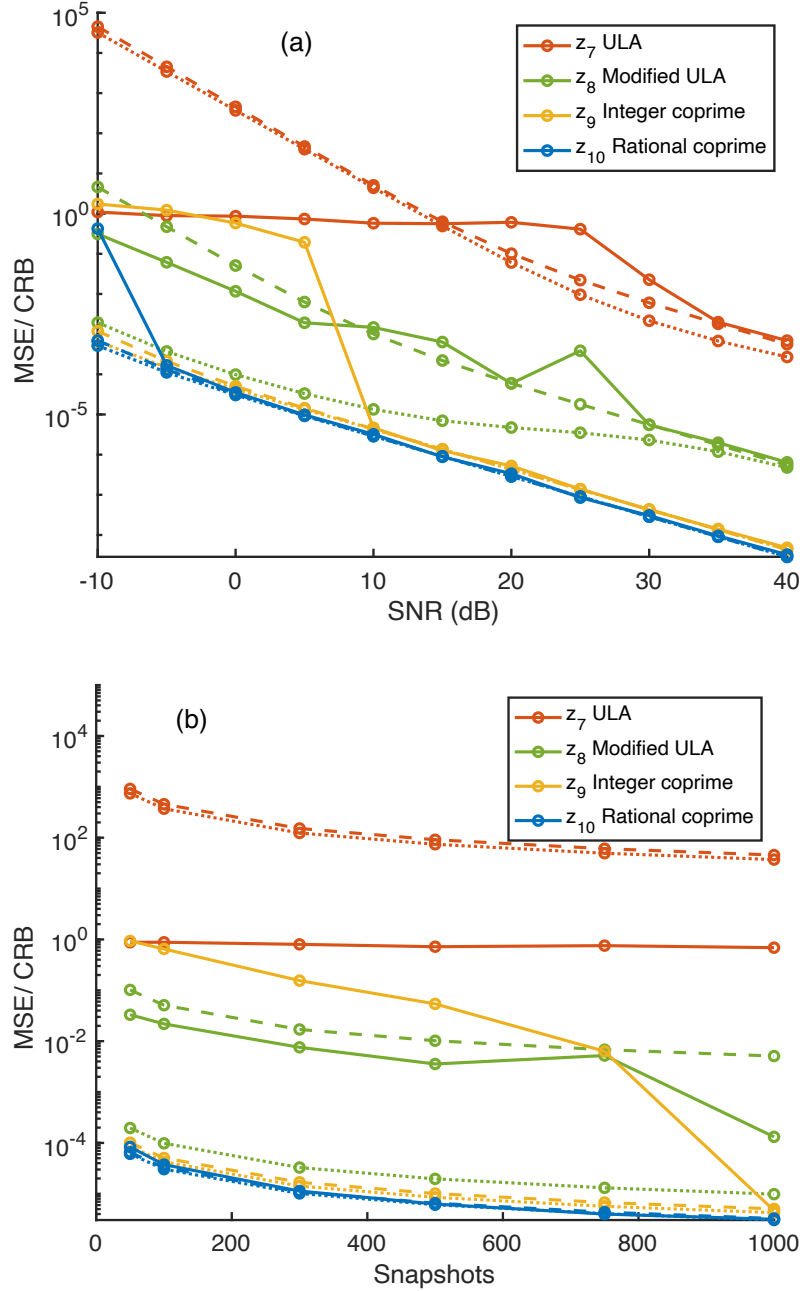


Figure 3.10: Experimental (Monte-Carlo) MSE (shown in solid lines), theoretical MSE (shown with dashed lines), and CRB (shown in dotted lines) for the four arrays under consideration as (a) SNR is varied when the number of snapshots $K = 100$, (b) number of snapshots is varied when $\text{SNR} = 0$ dB.

mitigate this effect quite well. The averaged numerical MSE obtained over 1000 Monte-Carlo simulations for each direction setting is shown in Fig. 3.11(b) and matches quite well with the theoretical MSE. Overall, the two proposed designs of rational arrays provide the smallest MSE for almost all two-DOA scenarios, except

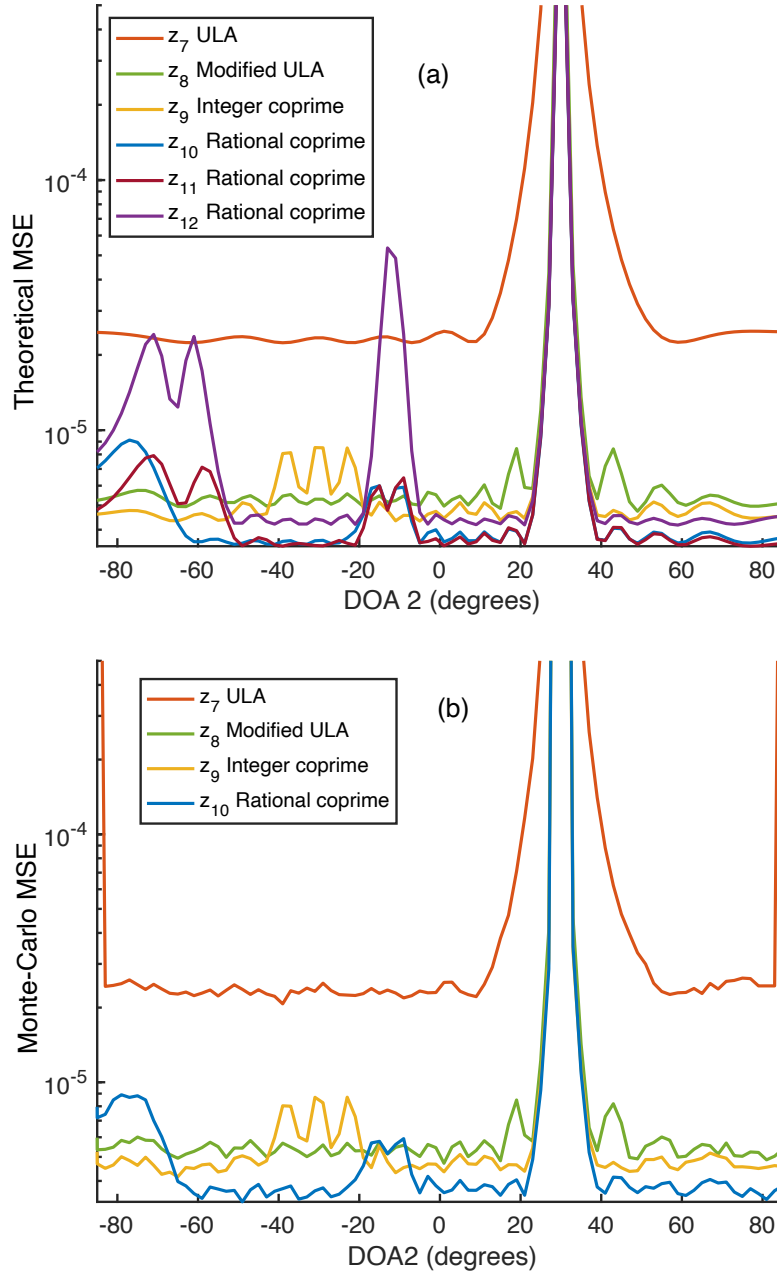


Figure 3.11: Comparison of (a) theoretical, and (b) Monte-Carlo MSE values when one DOA is fixed at 30° and the other DOA is varied from -90° to 90° . The proposed rational coprime arrays \mathbf{z}_{10} and \mathbf{z}_{11} have the lowest MSE, except possibly when the two DOAs come close to being equivalent modulo $2\pi/r_1$.

for the slight increase in MSE around the ‘bump’ regions. The two designs have very similar performance compared to each other.

Next, to compare the ability of the proposed rational arrays to resolve nearby DOAs, we consider the scenario where two sources are fixed at 0° and 8° , and a third

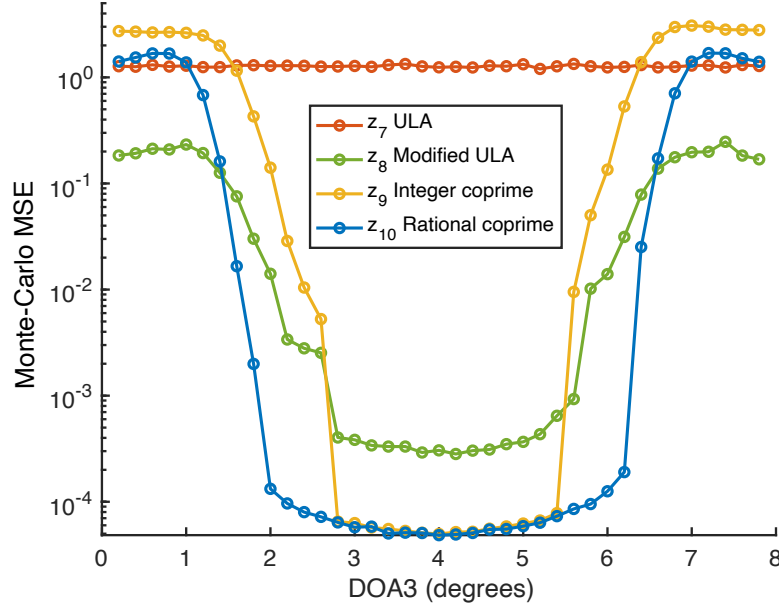


Figure 3.12: Comparison of numerical (Monte-Carlo) MSE when two sources are fixed at 0° and 8° and a third in-between DOA is varied. The rational coprime array is found to have the best resolution.

in-between DOA is varied from 0° to 8° . We consider SNR of 0 dB and $K = 600$ snapshots and perform 1000 Monte-Carlo simulations. Fig. 3.12 shows that the ULA \mathbf{z}_7 cannot resolve any three DOAs in the span of 8° . The other arrays have relatively large MSE when the third DOA is close to the two fixed DOAs, and it becomes lowest around 4° as expected. Notice that the transition region from high MSE to low MSE is the sharpest for the proposed rational coprime array \mathbf{z}_{10} , indicating they have higher DOA resolution. The MSE values are also the smallest for the rational coprime array compared to the other integer arrays.

In summary, the extensive numerical evaluations provided in this section show that the proposed rational coprime array has better MSE and CRB than integer array alternatives. The rational coprime arrays perform close to CRB at a considerably lower number of snapshots and lower SNR compared to the integer arrays, indicating their robustness to adverse SNR and snapshot conditions. The two proposed rational array designs are also found to have the lowest MSE for almost all two-DOA scenarios. Lastly, the rational coprime array also provided the best resolution for nearby sources.

3.5 Restricted Spatial Scope

In many applications, the incoming sources do not necessarily span the entire spatial scope of $[-90^\circ, +90^\circ]$. It is possible that there are applications where the potential range of directions could be known or estimated a-priori based on application specific information. Many recent papers, especially the machine learning based methods for DOA estimation [148], [149], assume that the signals are impinging on the array from a restricted spatial scope, such as, for example, $[-60^\circ, +60^\circ]$. In such a scenario of restricted spatial scope, one may still use the arrays that are designed to work for the entire spatial scope. However, there are certain advantages in incorporating the a-priori spatial scope information in array design, as we will elaborate next.

3.5.1 Sparse ULAs for Restricted Spatial Scope

Consider $D \leq m - 1$ DOAs that are known a-priori to be restricted in spatial scope $\Theta = [\theta_{min}, \theta_{max}]$, where $-90^\circ \leq \theta_{min} < \theta_{max} \leq +90^\circ$. If $\Theta = [-30^\circ, +30^\circ]$, it can be shown (as a consequence of Theorem 3 to be proved next) that sparse integer ULA

$$\mathbf{z} = \begin{bmatrix} 0 & 2 & 4 & \dots & 2(m-1) \end{bmatrix} \quad (3.78)$$

is sufficient to unambiguously identify any $m - 1$ sources in Θ with MUSIC. Even if the standard ULA can be used in this case, the array in (3.78) has a larger aperture than that of the standard m -sensor integer ULA. As a consequence, (3.78) can resolve more closely separated sources, and has lower estimation error than the standard ULA. It can also be shown (Theorem 3) that the array in (3.78) can even identify $m - 1$ sources unambiguously with MUSIC if they are restricted in the spatial scope $\Theta = [0^\circ, +90^\circ]$.

However, when the sources are restricted to some other arbitrary spatial scope $\Theta = [\theta_{min}, \theta_{max}]$, the integer constraint on the sensor location might prevent taking full advantage of the restricted spatial scope. For example, if $\Theta = [-45^\circ, +45^\circ]$, the only integer ULA with unique identifiability property is the standard m -sensor ULA (Theorem 3). If we relax the integer constraint and consider sparse rational ULAs, it is possible to span a larger aperture without losing the unique identifiability property with MUSIC.

Theorem 3 (Sparse ULAs for restricted spatial scope: Identifiability with MUSIC). Consider the task of estimating $D \leq m - 1$ DOAs that are known a-priori to be restricted in spatial scope $\Theta = [\theta_{min}, \theta_{max}]$, where $-90^\circ \leq \theta_{min} < \theta_{max} \leq +90^\circ$.

Let $\omega_{min} = \pi \sin \theta_{min}$ and $\omega_{max} = \pi \sin \theta_{max}$ be such that

$$\omega_{max} - \omega_{min} = 2\pi/\eta \quad (3.79)$$

for some $\eta \geq 1$. Then, the uniform array

$$\mathbf{z} = \begin{bmatrix} 0 & \beta & 2\beta & \dots & (m-1)\beta \end{bmatrix} \quad (3.80)$$

can identify any $D \leq m-1$ distinct DOAs in Θ with MUSIC without producing any false peaks if and only if $\beta \leq \eta$.

Note that just like the standard DOA practice of assuming $\omega \in [-\pi, \pi)$, or equivalently $\theta \in [-90^\circ, +90^\circ)$, the interval $\Theta = [\theta_{min}, \theta_{max})$ is assumed to be open from one end and closed from the other so as to have a sharp theorem statement.

Proof. The steering vector corresponding to array (3.80) is

$$\mathbf{a}(\omega) = \begin{bmatrix} 1 & e^{j\beta\omega} & e^{j2\beta\omega} & \dots & e^{j(m-1)\beta\omega} \end{bmatrix}^T \quad (3.81)$$

To unambiguously identify any D distinct DOAs in Θ using MUSIC, the augmented array manifold matrix of the array

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{a}(\omega_1) & \mathbf{a}(\omega_2) & \dots & \mathbf{a}(\omega_D) & | & \mathbf{a}(\omega_{D+1}) \end{bmatrix} \quad (3.82)$$

should have full column rank for any set of distinct $\omega_1, \dots, \omega_{D+1}$ in $[\omega_{min}, \omega_{max})$. Note that since the array is uniform, \mathbf{A}_{aug} is a Vandermonde matrix. Thus, \mathbf{A}_{aug} is full column rank if and only if the generators $e^{j\beta\omega_1}, e^{j\beta\omega_2}, \dots, e^{j\beta\omega_D}, e^{j\beta\omega_{D+1}}$ are all distinct.

If part. Suppose $\beta \leq \eta$. Since $\omega_1, \dots, \omega_{D+1}$ are distinct and lie in $[\omega_{min}, \omega_{max})$ satisfying (3.79), the generators must be all distinct. If not, let $e^{j\beta\omega_k} = e^{j\beta\omega_l}$ for some $k \neq l$. This implies that $|\omega_k - \omega_l| = 2k\pi/\beta$ for some $k \in \mathbb{N}$. This contradicts that $\omega_k, \omega_l \in [\omega_{min}, \omega_{max})$, because $|\omega_k - \omega_l| = 2k\pi/\beta \geq 2\pi/\eta = \omega_{max} - \omega_{min}$. Thus \mathbf{A}_{aug} is full column rank for any set of distinct $\omega_1, \dots, \omega_{D+1}$ in $[\omega_{min}, \omega_{max})$ if $\beta \leq \eta$.

Only if part. Suppose $\beta > \eta$. Then there exists $\omega_1, \omega_2 \in [\omega_{min}, \omega_{max})$ such that $\omega_2 - \omega_1 = 2\pi/\beta$. Since the generators $e^{j\beta\omega_1}$ and $e^{j\beta\omega_2}$ are the same in this case, \mathbf{A}_{aug} cannot be full column rank. This completes the proof of Theorem 3. \square

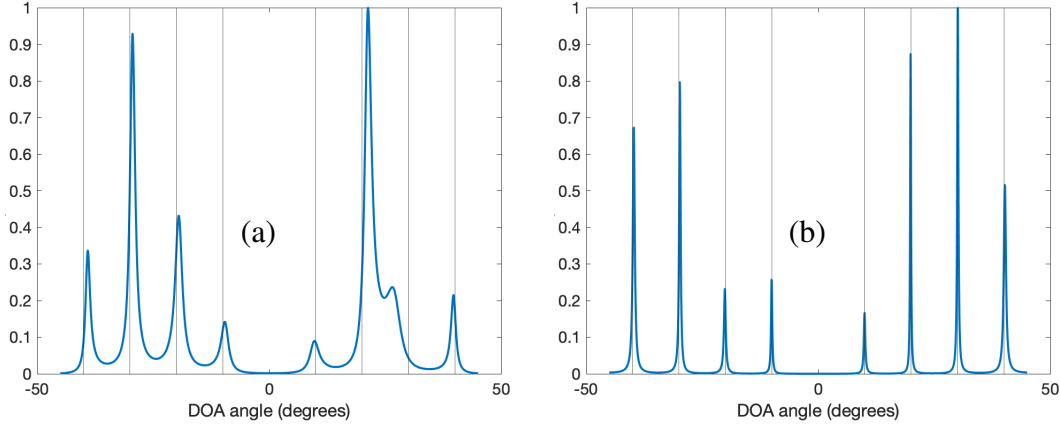


Figure 3.13: Identifying 8 sources in restricted spatial scope $\Theta = [-45^\circ, +45^\circ]$. MUSIC spectrum obtained with (a) standard integer ULA (b) rational array (3.80) with $\beta = 1.4$. The true DOAs are shown in solid black lines.

Examples. For $\beta = \eta = 2$, $\theta_{min} = -30^\circ$ and $\theta_{max} = +30^\circ$ satisfies (3.79), and so does $\theta_{min} = 0^\circ$ and $\theta_{max} = +90^\circ$. Thus, array (3.78) can unambiguously identify up to $m - 1$ DOAs that lie in the corresponding restricted spatial scope, as claimed before.

Now, for $\Theta = [-45^\circ, +45^\circ]$ we have $\eta = \sqrt{2}$. The only integer value of β satisfying $\beta \leq \eta$ is 1 in this case. So among integer ULAs, only the standard m -sensor integer ULA can be used, as claimed before. Now, without the integer array constraint, choosing the largest possible β to maximize the aperture of the corresponding array (3.80) is desirable. In practice, one has to choose β such that array (3.80) is easy to realize as well. One such option could be $\beta = 1.4$. To illustrate the advantage offered by such sparse rational arrays, consider a standard 10-sensor integer ULA and a 10-sensor sparse rational array as in (3.80) with $\beta = 1.4$. Both arrays have no ambiguity for up to 9 sources in $\Theta = [-45^\circ, +45^\circ]$. Consider eight sources with DOAs $-40^\circ, -30^\circ, -20^\circ, -10^\circ, 10^\circ, 20^\circ, 30^\circ$, and 40° with SNR 0 dB impinging on the array. Fig. 3.13 shows the MUSIC spectrum obtained with 300 snapshots from both arrays. It is clear that the sparse rational ULA provides accurate DOA estimates, whereas the integer ULA has higher estimation errors.

3.5.2 Generalizing the Results for Rational Arrays

The results for rational arrays given in Theorems 1 and 2 can be generalized for the case of restricted spatial scope. In this section, we state and prove these generaliza-

tions. We first prove a lemma that is useful in these generalizations.

Lemma 3. Let r_1, r_2 be positive rational numbers and $-\pi \leq \omega_{\min} < \omega_{\max} \leq \pi$ be such that $\omega_{\max} - \omega_{\min} = 2\pi/\eta$ where $\eta \geq 1$. Then the following statements are equivalent:

1. There exist distinct ω_1 and ω_2 in the range $[\omega_{\min}, \omega_{\max})$ such that

$$e^{j\omega_1 r_1} = e^{j\omega_2 r_1} \text{ and } e^{j\omega_1 r_2} = e^{j\omega_2 r_2}; \quad (3.83)$$

2. $\gcd(r_1, r_2) > \eta$.

Proof. Assume the first statement is true. Without loss of generality, let $\omega_2 > \omega_1$. Now, $e^{j\omega_1 r_1} = e^{j\omega_2 r_1}$ implies that

$$(\omega_2 - \omega_1)r_1 = 2\pi l_1, \quad l_1 \in \mathbb{N}. \quad (3.84)$$

Since $\omega_2 - \omega_1 < \omega_{\max} - \omega_{\min} = 2\pi/\eta$, l_1 is constrained as: $1 \leq l_1 < r_1/\eta$. Similarly, $e^{j\omega_1 r_2} = e^{j\omega_2 r_2}$ implies that

$$(\omega_2 - \omega_1)r_2 = 2\pi l_2, \quad l_2 \in \mathbb{N}, \quad 1 \leq l_2 < r_2/\eta. \quad (3.85)$$

From the above two equations, we get

$$t = \frac{2\pi}{\omega_2 - \omega_1} = \frac{r_1}{l_1} = \frac{r_2}{l_2} > \eta. \quad (3.86)$$

Thus,

$$r_1 = tl_1, r_2 = tl_2, \quad l_1, l_2 \in \mathbb{N}. \quad (3.87)$$

This means r_1 and r_2 have a common rational divisor $t > \eta$ which implies $\gcd(r_1, r_2) > \eta$.

Conversely, the second statement from the lemma implies that $\exists t > \eta$ such that

$$r_1 = tl_1, r_2 = tl_2, \quad l_1, l_2 \in \mathbb{N}. \quad (3.88)$$

Now choose $\omega_{\min} \leq \omega_1 < \omega_2 < \omega_{\max}$ such that

$$\omega_2 - \omega_1 = 2\pi l_1/r_1 = 2\pi l_2/r_2 = 2\pi/t. \quad (3.89)$$

It is indeed possible to choose such ω_1 and ω_2 in $[\omega_{\min}, \omega_{\max})$ because $t > \eta$. Now, (3.89) implies that

$$e^{j\omega_1 r_1} = e^{j\omega_2 r_1} \text{ and } e^{j\omega_1 r_2} = e^{j\omega_2 r_2}. \quad (3.90)$$

This completes the proof of the lemma. \square

Interpretation of the Lemma. This lemma establishes a necessary and sufficient condition for the unambiguous identification of the frequency of a complex sinusoid in terms of the sampling locations. Consider a complex sinusoid $e^{j\omega z}$, and we want to identify the frequency $\omega \in [\omega_{\min}, \omega_{\max})$ based on samples of this sinusoid at two locations of $z = r_1$ and $z = r_2$. Eq. (3.83) means that there are two frequencies ω_1 and ω_2 in the given range such that the complex exponentials $e^{j\omega_1 z}$ and $e^{j\omega_2 z}$ are indistinguishable based on their sampled values at $z = r_1$ and $z = r_2$. The Lemma claims that this is possible if and only if $(r_1, r_2) > \eta$. For the special case of $\eta = 1$, the range $[\omega_{\min}, \omega_{\max})$ covers the entire range of discrete-time frequencies $[-\pi, \pi)$ and the condition $(r_1, r_2) > 1$ implies that r_1 and r_2 are *not* rationally coprime.

Theorem 4 (Invertibility of steering vector). Let $\mathbf{a}(\omega) = [1, e^{j\omega r_1}, e^{j\omega r_2}, \dots, e^{j\omega r_{m-1}}]^T$ be the steering vector for rational array (3.43). Let ω_{\min} and ω_{\max} , where $-\pi \leq \omega_{\min} < \omega_{\max} \leq \pi$, be such that $\omega_{\max} - \omega_{\min} = 2\pi/\eta$ for some $\eta \geq 1$. Then $\omega \in [\omega_{\min}, \omega_{\max})$ can be found uniquely from $\mathbf{a}(\omega)$ if and only if the sensor locations $r_i = P_i/Q_i$ (where $(P_i, Q_i) = 1, P_i, Q_i \in \mathbb{N} \forall i$) are such that

$$\gcd(r_1, r_2, \dots, r_{m-1}) \leq \eta. \quad (3.91)$$

Or equivalently,

$$\gcd(P_1, P_2, \dots, P_{m-1}) \leq \eta \cdot \text{lcm}(Q_1, Q_2, \dots, Q_{m-1}). \quad (3.92)$$

Note that for the special case of $\eta = 1$, we get the statement of Theorem 1.

Proof. Only if part. If $\gcd(r_1, r_2, \dots, r_{m-1}) > \eta$, then

$$\underbrace{\gcd(P_1, P_2, \dots, P_{m-1})}_g > \eta \cdot \underbrace{\text{lcm}(Q_1, Q_2, \dots, Q_{m-1})}_h. \quad (3.93)$$

Taking out the common factor g from numerators P_i , we can write

$$P_i = gx_i, \quad x_i \in \mathbb{N}, \quad (x_i, Q_i) = 1 \quad \forall i. \quad (3.94)$$

Thus, the steering vector can be written as

$$\mathbf{a}(\omega) = [1, e^{j\omega g \frac{x_1}{Q_1}}, e^{j\omega g \frac{x_2}{Q_2}}, \dots, e^{j\omega g \frac{x_{m-1}}{Q_{m-1}}}]^T. \quad (3.95)$$

Now consider $\omega' = \omega + 2\pi h/g$. Since $g > \eta h$ from the assumption, there exists such ω, ω' in the range $[\omega_{\min}, \omega_{\max})$. We now show that $\mathbf{a}(\omega') = \mathbf{a}(\omega)$. For this, consider the $(i+1)$ -th entry of $\mathbf{a}(\omega')$:

$$e^{j\omega' g \frac{x_i}{Q_i}} = e^{j\omega g \frac{x_i}{Q_i}} \cdot e^{j2\pi h g \frac{x_i}{Q_i}} = e^{j\omega g \frac{x_i}{Q_i}} \cdot e^{j2\pi x_i \frac{h}{Q_i}} = e^{j\omega g \frac{x_i}{Q_i}}. \quad (3.96)$$

which is nothing but the $(i + 1)$ -th entry of $\mathbf{a}(\omega)$. The last equality in (3.96) follows because Q_i divides h for every i . Since (3.96) holds for every $i = 1, 2, \dots, m - 1$, we have found $\omega, \omega' \in [\omega_{\min}, \omega_{\max})$ having identical steering vectors:

$$\mathbf{a}(\omega') = \mathbf{a}(\omega).$$

This proves that condition $\gcd(r_1, r_2, \dots, r_{m-1}) \leq \eta$ is necessary for invertibility of steering vector in the range $[\omega_{\min}, \omega_{\max})$.

If part. Here we take a constructive approach. We want to be able to uniquely determine $\omega \in [\omega_{\min}, \omega_{\max})$ from $\mathbf{a}(\omega)$ given that (3.91) or equivalently (3.92) is true. From each individual elements $[\mathbf{a}(\omega)]_{i+1} = e^{j\omega r_i}$ of the steering vector, we can uniquely identify ω only within a modulo $2\pi/r_i$ ambiguity. That is, we can write

$$\omega = \Omega_i + 2\pi \frac{l_i}{r_i}, \quad l_i \in \mathbb{Z} \quad (3.97)$$

with a unique $\omega_{\min} \leq \Omega_i < \omega_{\min} + 2\pi/r_i$ satisfying $e^{j\Omega_i r_i} = [\mathbf{a}(\omega)]_{i+1}$. The integer l_i can possibly take a value in the range $0 \leq l_i < r_i/\eta$. To uniquely identify ω , we shall prove that there exists a unique set of l_i s for $i = 1, 2, \dots, m - 1$ such that

$$\Omega_1 + 2\pi \frac{l_1}{r_1} = \Omega_2 + 2\pi \frac{l_2}{r_2} = \dots = \Omega_{m-1} + 2\pi \frac{l_{m-1}}{r_{m-1}}. \quad (3.98)$$

If possible, let there be another possible set of \hat{l}_i s for $i = 1, 2, \dots, m - 1$ such that

$$\Omega_1 + 2\pi \frac{\hat{l}_1}{r_1} = \Omega_2 + 2\pi \frac{\hat{l}_2}{r_2} = \dots = \Omega_{m-1} + 2\pi \frac{\hat{l}_{m-1}}{r_{m-1}}. \quad (3.99)$$

Subtracting (3.99) from (3.98) gives

$$\frac{l_1 - \hat{l}_1}{r_1} = \frac{l_2 - \hat{l}_2}{r_2} = \dots = \frac{l_{m-1} - \hat{l}_{m-1}}{r_{m-1}}. \quad (3.100)$$

Note that $l_i \neq \hat{l}_i$ for any i , because otherwise the entire sets of l_i s and \hat{l}_i s become identical. Also, without loss of generality, let $l_1 > \hat{l}_1$. Since $r_i > 0 \forall i$, this means $l_i > \hat{l}_i \forall i$ as well. Further, note that since $l_i, \hat{l}_i \in \mathbb{Z}$ and $0 \leq l_i, \hat{l}_i < r_i/\eta$, we have that $l_i - \hat{l}_i < r_i/\eta$ for every i . Thus taking reciprocal of Eq. (3.100) gives us

$$t = \frac{r_1}{l_1 - \hat{l}_1} = \frac{r_2}{l_2 - \hat{l}_2} = \dots = \frac{r_{m-1}}{l_{m-1} - \hat{l}_{m-1}} > \eta. \quad (3.101)$$

Thus,

$$r_i = t(l_i - \hat{l}_i) \quad \forall i. \quad (3.102)$$

Hence we have found $t > \eta$ that is a common rational divisor of all r_i -s, which contradicts that $\gcd(r_1, r_2, \dots, r_{m-1}) \leq \eta$. This proves that condition $\gcd(r_1, r_2, \dots, r_{m-1}) \leq \eta$ is sufficient. \square

Theorem 5 (Rational coprime array: Identifiability with MUSIC). Consider the task of estimating DOAs that are known a-priori to be restricted in spatial scope $\Theta = [\theta_{\min}, \theta_{\max})$, where $-90^\circ \leq \theta_{\min} < \theta_{\max} \leq +90^\circ$. Let $\omega_{\min} = \pi \sin \theta_{\min}$ and $\omega_{\max} = \pi \sin \theta_{\max}$ be such that $\omega_{\max} - \omega_{\min} = 2\pi/\eta$ for some $\eta \geq 1$. Let $D \leq m - 2$ sources with DOAs $\theta_1, \theta_2, \dots, \theta_D$ be such that the frequency variables $\omega_1, \omega_2, \dots, \omega_D$, where $\omega_i = \pi \sin \theta_i$, are distinct modulo $2\pi/r_1$. Then the m -sensor rational array as in Eq. (3.56), with $(r_1, r_2) \leq \eta$ can identify these with MUSIC without producing any false peaks.

Note that for the special case of $\eta = 1$, we get the statement of Theorem 2.

Proof. The augmented array manifold matrix for this array is given by

$$\mathbf{A}_{\text{aug}} = \left[\begin{array}{cccc|c} \mathbf{a}_{r_1}(\omega_1) & \mathbf{a}_{r_1}(\omega_2) & \dots & \mathbf{a}_{r_1}(\omega_D) & \mathbf{a}_{r_1}(\omega_{D+1}) \\ e^{j\omega_1 r_2} & e^{j\omega_2 r_2} & \dots & e^{j\omega_D r_2} & e^{j\omega_{D+1} r_2} \end{array} \right], \quad (3.103)$$

where $\mathbf{a}_{r_1}(\omega) = [1, e^{j\omega r_1}, e^{j2\omega r_1}, \dots, e^{j(m-2)\omega r_1}]^T$. We need to show that \mathbf{A}_{aug} is full column rank for any sources satisfying (3.59) and any $\omega_{D+1} \in [\omega_{\min}, \omega_{\max})$ that is distinct from $\omega_1, \dots, \omega_D$. Note that the $(m-1) \times D$ submatrix

$$\mathbf{A}_1 = \left[\mathbf{a}_{r_1}(\omega_1) \quad \mathbf{a}_{r_1}(\omega_2) \quad \dots \quad \mathbf{a}_{r_1}(\omega_D) \right] \in \mathbb{C}^{(m-1) \times D} \quad (3.104)$$

is Vandermonde. Since the sources $\omega_1, \omega_2, \dots, \omega_D$ are such that (3.60) holds, the generators $e^{j\omega_i r_1}$ of the Vandermonde matrix \mathbf{A}_1 are all distinct. Furthermore, since $D \leq m - 2$, the matrix is tall. This ensures that \mathbf{A}_1 has full column rank D . Now consider the augmented version of \mathbf{A}_1 :

$$\mathbf{A}_2 = \left[\mathbf{a}_{r_1}(\omega_1) \quad \mathbf{a}_{r_1}(\omega_2) \quad \dots \quad \mathbf{a}_{r_1}(\omega_D) \quad | \quad \mathbf{a}_{r_1}(\omega_{D+1}) \right], \quad (3.105)$$

where ω_{D+1} is distinct from any of sources. Now consider the following two cases.

Case 1. Let ω_{D+1} be distinct modulo $2\pi/r_1$ from all the D sources frequencies. In this case, \mathbf{A}_2 has full column rank $D + 1$ as the generator elements are all distinct and $D + 1 \leq m - 1$ (i.e. \mathbf{A}_2 is either square or tall). This also ensures that the augmented manifold matrix (3.103) is full rank.

Case 2. Let $\omega_{D+1} - \omega_l = 2\pi k/r_1$ for some source ω_l . Since (3.59) holds, there can be only one such source ω_l . Without loss of generality let $\omega_{D+1} - \omega_1 = 2\pi k/r_1$, or equivalently,

$$e^{j\omega_1 r_1} = e^{j\omega_{D+1} r_1}. \quad (3.106)$$

Thus we have $\mathbf{a}_{r_1}(\omega_1) = \mathbf{a}_{r_1}(\omega_{D+1})$, so \mathbf{A}_2 is no longer full column rank. Here, the additional sensor placed at r_2 comes to help. If \mathbf{A}_{aug} does not have full column rank, there should be a non-zero vector $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_D & | & c_{D+1} \end{bmatrix}^T$ such that

$$\mathbf{A}_{aug} \mathbf{c} = 0. \quad (3.107)$$

This, in particular, means that

$$\mathbf{A}_2 \mathbf{c} = 0. \quad (3.108)$$

Since $\mathbf{a}_{r_1}(\omega_1) = \mathbf{a}_{r_1}(\omega_{D+1})$, we can rewrite (3.108) as

$$\mathbf{A}_1 \begin{bmatrix} c_1 + c_{D+1} & c_2 & \dots & c_D \end{bmatrix}^T = 0. \quad (3.109)$$

As \mathbf{A}_1 is full column rank, (3.109) implies that

$$c_1 + c_{D+1} = 0, \text{ and } c_2 = c_3 = \dots = c_D = 0. \quad (3.110)$$

Note that c_1 and c_{D+1} cannot be zero as $\mathbf{c} \neq \mathbf{0}$. With this, the last row of the matrix equation (3.107) implies that

$$e^{j\omega_1 r_2} = e^{j\omega_{D+1} r_2}. \quad (3.111)$$

But from Lemma 3, we have that (3.106) and (3.111) are not possible together as $(r_1, r_2) \leq \eta$. Thus, we have shown that \mathbf{A}_{aug} must have rank $D + 1$ for any choice of $\omega_{D+1} \in [\omega_{min}, \omega_{max})$ that is distinct from $\omega_1, \dots, \omega_D$ as long as the source frequencies are distinct modulo $2\pi/r_1$. Thus, MUSIC will work without producing false peaks. This proves Theorem 5. \square

The results presented in this section can be further generalized to the case where sensor positions are possibly irrational. It requires the gcd and coprimality concepts of rational numbers to be further extended to real numbers. The next section explains these generalizations.

3.6 Arrays with Arbitrary (Non-Rational) Sensor Locations

In this section, we consider linear arrays with arbitrary sensor locations that may be any (possibly non-rational) real numbers. This discussion is motivated more by academic interest than by potential practical impact. Due to limited precision in practice, the sensor positions of any array are likely to be slightly perturbed from their intended location, making it a non-rational array with probability one. Also, the non-integer optimum sensor positions obtained for beamforming [133], [134], or interference suppression [135] are often irrational. While irrational quantities are eventually approximated by rational quantities in practice, it is still of academic interest to explore the irrational case, as it offers a further generalization of the rational case.

In Sec. 3.2 we extended some number-theoretic properties of integers to rational numbers. However, there are significant differences when it comes to the case of real numbers. For instance, prime factorization could be extended to rational numbers by allowing exponents of primes to be negative (see Sec. 3.2.2). However, for arbitrary irrational numbers, a further extension of prime factorization does not seem possible. Nonetheless, there is a way to define gcd and coprimality for real numbers, as we show next. We also demonstrate the usefulness of these definitions for sensor arrays and DOA identifiability.

Definition 5 (gcd of real numbers). The greatest common divisor (gcd) of m real numbers z_1, z_2, \dots, z_m is defined as the largest possible positive real number z such that

$$z_i = K_i \cdot z \quad \text{where,} \quad K_i \in \mathbb{Z}, \quad i = 1, 2, \dots, m, \quad (3.112)$$

if it exists.

It is denoted as $z = (z_1, z_2, \dots, z_m)$ or $\gcd(z_1, z_2, \dots, z_m)$. We explained in Sec. 3.2 that for rationals, a common rational divisor (Definition 1) always exists. However, for real numbers that are not necessarily rationals, the gcd may or may not exist! This is because, for two real numbers z_1 and z_2 , there may or may not exist a number z such that

$$z_1 = k_1 z, \quad z_2 = k_2 z \quad \text{for some } k_1, k_2 \in \mathbb{Z}. \quad (3.113)$$

For example, let $z_1 = \pi$ and $z_2 = \pi^2$. If a common divisor z exists, then

$$\frac{z_2}{z_1} = \pi = \frac{k_2}{k_1}, \quad (3.114)$$

which leads to a contradiction because π is irrational whereas k_2/k_1 is a rational number. But consider another example with $z_1 = \sqrt{2}$ and $z_2 = 5/\sqrt{2}$. In this case, $z = 1/\sqrt{2}$ is a common factor and in fact the gcd of z_1 and z_2 . If the ratio of two real numbers is not a rational number, then there does not exist a common divisor of the two real numbers.

Next, we extend the coprime definition to real numbers:

Definition 6 (Coprime real numbers). A set of positive real numbers z_1, z_2, \dots, z_m is said to be coprime if $\gcd(z_1, z_2, \dots, z_m)$ either does not exist or is less than or equal to 1.

Just like the case of rationals, the above definition unifies the coprimality for any type of numbers (including rationals and integers) in the following sense: two (or more) positive real numbers are coprime if and only if they *cannot* be expressed as integer multiples of a number *larger* than unity. Note that this includes two cases: i) gcd does not exist, and ii) gcd exists but is less than or equal to one.

With this notion of coprimality, the steering vector invertibility (Theorem 4) can be further extended to arrays with arbitrary (real-valued) sensor locations as follows.

Theorem 6 (Invertibility of steering vector). Let $\mathbf{a}(\omega) = [1, e^{j\omega z_1}, e^{j\omega z_2}, \dots, e^{j\omega z_{m-1}}]^T$ be the steering vector of an arbitrary (possibly non-rational) array

$$\begin{bmatrix} 0 & z_1 & z_2 & \dots & z_{m-1} \end{bmatrix}. \quad (3.115)$$

Let $-\pi \leq \omega_{\min} < \omega_{\max} < \pi$ be such that $\omega_{\max} - \omega_{\min} = 2\pi/\eta$ for some $\eta \geq 1$. Then $\omega \in [\omega_{\min}, \omega_{\max})$ can be found uniquely from $\mathbf{a}(\omega)$ if and only if the sensor locations z_i satisfy either of the following two conditions:

1. $\gcd(z_1, z_2, \dots, z_{m-1})$ does not exist;
2. $\gcd(z_1, z_2, \dots, z_{m-1}) \leq \eta$.

Next, similar to rational coprime array (3.56), we propose the following m -sensor coprime array:

$$\mathbf{z} = \begin{bmatrix} 0 & z_1 & 2z_1 & \dots & (m-2)z_1 & | & z_2 \end{bmatrix}, \quad (3.116)$$

where z_1 and z_2 are coprime real numbers according to Definition 6. Similar to Theorem 5, we can state the following identifiability result for such arrays.

Theorem 7 (Arbitrary coprime array: Identifiability with MUSIC). Consider the task of estimating DOAs that are known a-priori to be restricted in spatial scope $\Theta = [\theta_{\min}, \theta_{\max})$, where $-90^\circ \leq \theta_{\min} < \theta_{\max} \leq +90^\circ$. Assume $\omega_{\min} = \pi \sin \theta_{\min}$ and $\omega_{\max} = \pi \sin \theta_{\max}$ are such that $\omega_{\max} - \omega_{\min} = 2\pi/\eta$ for some $\eta \geq 1$. Let $D \leq m - 2$ sources with DOAs $\theta_1, \theta_2, \dots, \theta_D$ be such that the frequency variables $\omega_1, \omega_2, \dots, \omega_D$, where $\omega_i = \pi \sin \theta_i$, are distinct modulo $2\pi/z_1$. Then the m -sensor array (3.116) where (z_1, z_2) either does not exist, or is $\leq \eta$, can identify these sources with MUSIC without producing any false peaks.

Theorem 6 and Theorem 7 can be proved in similar fashions as Theorem 4 and Theorem 5 respectively, by considering the gcd and coprimality notions for real numbers and appropriately modifying the arguments in the proofs. For example, the *only if part* of the proof for Theorem 6 follows by showing that $\mathbf{a}(\omega + 2\pi/z) = \mathbf{a}(\omega)$, when $(z_1, z_2) = z > \eta$. The *if part* can be proved just like the *if part* of Theorem 4 by replacing rational sensor locations r_i with real-valued sensor locations z_i . The same argument holds for the proof of Theorem 7. The detailed proofs are omitted here to avoid repetition.

As a demonstrative example, we consider the situation in *Example 2* from Sec. 3.4.1, with $A = 20$ and $m = 13$. We compare the integer arrays \mathbf{z}_5 and \mathbf{z}_6 with a real-valued coprime array, with $z_1 = \sqrt{3}$ and $z_2 = 20$ in (3.116):

$$\mathbf{z}_{13} = \begin{bmatrix} 0 & \sqrt{3} & 2\sqrt{3} & \cdots & 11\sqrt{3} & | & 20 \end{bmatrix}. \quad (3.117)$$

The MUSIC spectra obtained with the three arrays² are shown in Fig. 3.14. We see that just like the rational coprime array \mathbf{z}_4 from Fig. 3.6, arbitrary (non-rational) coprime array \mathbf{z}_{13} also produces very sharp peaks near the true DOAs in contrast to the integer arrays \mathbf{z}_5 and \mathbf{z}_6 .

3.7 Difference Coarrays of Rational Arrays

The difference coarray of the general rational array from Eq. (3.43) consists of all possible unique differences in sensor positions: $\{r_i - r_j\}_{i,j=0}^{m-1}$. We have seen that well-designed sparse integer arrays ($r_i \in \mathbb{Z}$) such as MRAs, nested arrays, and coprime arrays contain a large central segment of $2m' - 1$ consecutive integer differences in their difference coarray, where m' is $O(m^2)$. Array output correlations can be estimated at these lags, and an $m' \times m'$ Toeplitz coarray covariance matrix is formed

²The implementation on a digital computer typically constrains the irrational numbers to be approximated with a 64-bit or 128-bit representation.

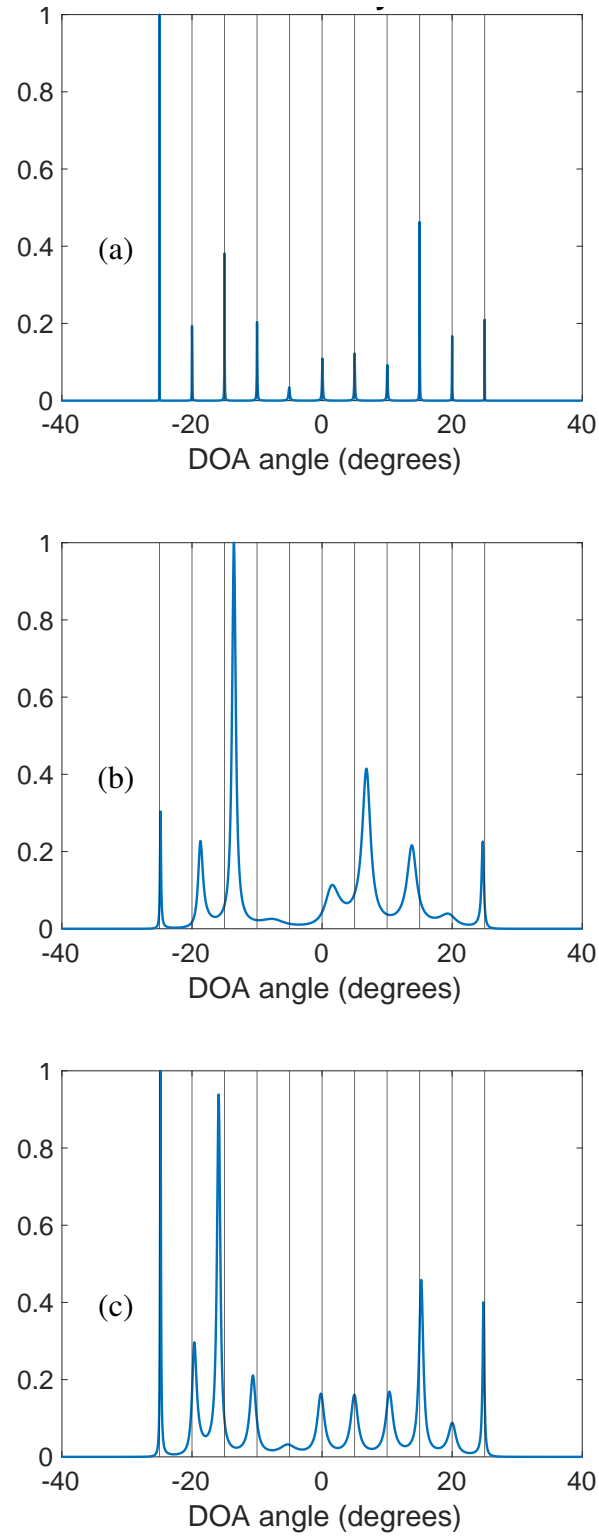


Figure 3.14: Identifying 11 closely spaced sources from Example 2. MUSIC spectrum obtained with (a) arbitrary (non-rational) coprime array \mathbf{z}_{13} , (b) standard ULA \mathbf{z}_5 , and (c) modified ULA \mathbf{z}_6 . True DOAs are shown in black vertical lines.

by using the method described in Sec. 2.5 to identify DOAs using coarray-domain DOA estimation algorithms.

However, when there is an aperture constraint, the situation changes. If the available aperture is A , clearly one cannot place $m > \lfloor A \rfloor$ sensors at integer locations. Furthermore, the farthest distance between the sensors placed at integer locations in an aperture of length A also cannot exceed $\lfloor A \rfloor$. This limits the number of sources that can be identified via sparse integer arrays under the aperture constraint. Without the aperture constraint, well-designed m -sensor sparse arrays could identify $O(m^2)$ sources. But under the aperture constraint, since the largest coarray lag that the array can produce is limited to $\lfloor A \rfloor$, the number of resolvable sources cannot exceed $\lfloor A \rfloor$, even when the DOA estimation is done through the coarray domain.

To summarize, if $D \geq m$ and $D \geq A$, no m -sensor integer array with aperture A can identify all DOAs. In this scenario, placing some sensors at rational locations allows us to satisfy the aperture constraint as well as estimate more sources than the number of sensors, as we demonstrate next.

3.7.1 Motivating Examples

Consider a standard integer ULA with $m = 11$ sensors. This can identify up to 10 ($= m - 1$) sources unambiguously using MUSIC [10]. Suppose now we have an extra sensor that can be used without increasing the array aperture and without moving the already placed sensors. If we place the extra sensor at a non-integer location 9.5, the array becomes

$$\mathbf{z} = [0 \quad 1 \quad 2 \quad \dots \quad 7 \quad 8 \quad 9 \quad 9.5 \quad 10]. \quad (3.118)$$

With this extra sensor, the difference coarray now consists of all half-integer lags from -9.5 to 9.5 in addition to integer lags from -10 to 10 . The one-sided coarray steering vector is given by

$$\mathbf{a}_c(\omega) = [1 \quad e^{j\omega/2} \quad e^{j\omega} \quad e^{j3\omega/2} \quad \dots \quad e^{j10\omega}]. \quad (3.119)$$

Thus, with the addition of an extra sensor, we can potentially identify up to 20 sources with the coarray methods. To do this, first a 21×21 Toeplitz coarray covariance matrix $\hat{\mathbf{R}}$ is constructed in a similar manner to the direct-augmentation approach described in Sec. 2.5. Based on the estimate of the noise subspace $\hat{\mathbf{E}}_n$ obtained by eigendecomposition of $\hat{\mathbf{E}}_n$, The MUSIC spectrum is then obtained as [10]

$$P(\omega) = \left(\mathbf{a}_c^H(\omega) \hat{\mathbf{E}}_n \hat{\mathbf{E}}_n^H \mathbf{a}_c(\omega) \right)^{-1}, \quad (3.120)$$

where $\mathbf{a}_c(\omega)$ is the coarray steering vector (3.119).

Since the coarray is uniform, a suitably modified root-MUSIC [85] can also be used to get DOA estimates. First, a degree 40 polynomial

$$Q(z) = \mathbf{v}^T(z^{-1})\hat{\mathbf{E}}_n\hat{\mathbf{E}}_n^H\mathbf{v}(z), \quad (3.121)$$

where $\mathbf{v}(z) = [1 \ z \ z^2 \ \dots \ z^{20}]^T$, is formed by substituting $z = e^{j\omega/2}$ in $\mathbf{a}_c(\omega)$. Note that this is different from the standard root-MUSIC (Chapter 2) for ULA, where $z = e^{j\omega}$ is used. For ULA, the arguments of the D roots of $Q(z)$ that are inside the unit circle and closest to the unit circle in the complex z -plane provide estimates of the DOAs. But for array (3.118), since we used $z = e^{j\omega/2}$ instead of $z = e^{j\omega}$, only the roots lying in the **half-plane** corresponding to the positive real axis are considered.

Fig. 3.15 shows an example. There are 12 DOAs corresponding to angles -80° , -55° , -40° , -25° , -15° , -5° , 5° , 15° , 25° , 40° , 55° , and 80° with SNR -5 dB. 800 snapshots are used. The MUSIC spectrum shows 12 clear peaks close to the true DOA angles. The root-MUSIC plot has 12 pairs of reciprocal conjugate roots lying close to the unit circle in the right half-plane. The arguments of these roots correspond to $\omega_i/2$ for sources $\omega_1, \dots, \omega_{12}$. Note that any integer array with aperture 10 could have identified only up to 10 sources, but placing a sensor at a rational location allows us to estimate more sources. We will see in the next subsection that even more sources can be correctly identified if the array aperture can be increased.

We would like to remark that in the above example if we are allowed to change the sensor positions while keeping the aperture the same, we can also consider the following array:

$$\left[0 \ 1/3 \ 4/3 \ \dots \ 22/3 \ 25/3 \ 28/3 \ 10\right]. \quad (3.122)$$

This array is just a shifted version of the array

$$\left[-1/3 \ 0 \ 1 \ \dots \ 8 \ 9 \ 9+2/3\right]. \quad (3.123)$$

Array (3.123) has a standard 10-sensor ULA with two additional sensors: one on the left side at $-1/3$ and the other at $9 + 2/3$ on the right side. Note that the array aperture is still 10, so the aperture constraint is satisfied. The addition of two rational location sensors to the 10-sensor integer ULA ensures that the difference coarray of

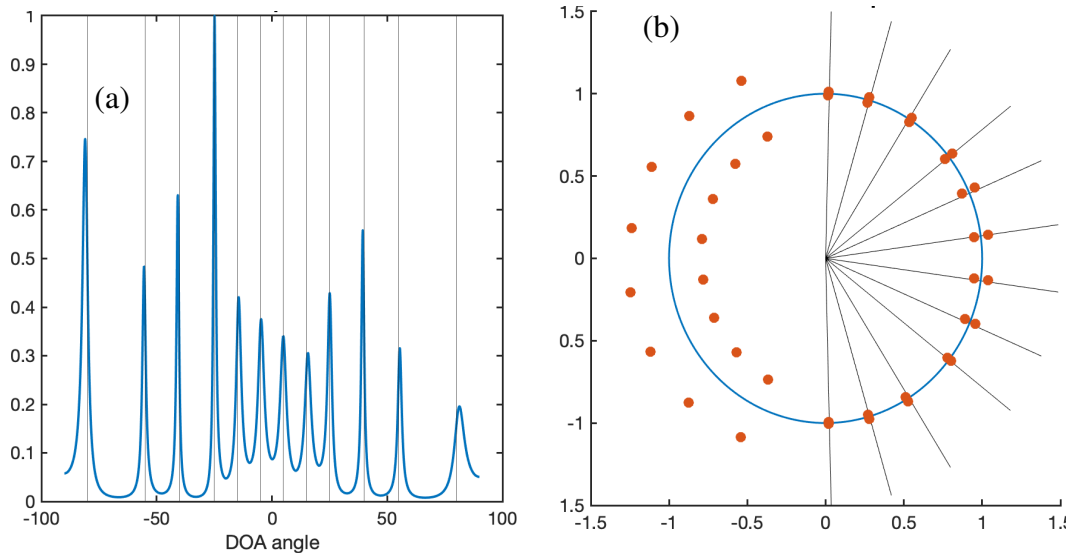


Figure 3.15: Identifying 12 sources with the 12-element nested rational array (3.118) having aperture $A = 10$. (a) MUSIC spectrum and (b) root-MUSIC plot. The true DOAs (in MUSIC plot) and corresponding angles $\omega_i/2$ (in root-MUSIC plot) are shown in solid black lines.

this array has equispaced lags of the form

$$k/3, \text{ where } k = -30, -29, \dots, 29, 30. \quad (3.124)$$

Thus, this array can ideally identify up to 30 sources. The coarray of the array is a rational ULA and is equivalent to an integer ULA scaled by a factor of $1/3$. A disadvantage of such shrinking is the increase in the DOA estimation mean squared error and reduction of resolution (ability to resolve two nearby sources) [86], [96]. To counter this effect and still identify a large number of sources with such a coarray, a very high number of snapshots may be required, which may not be feasible in practice. We observed in our simulations that as long as the effective coarray scaling factor is not 0.5 or lower, a large number of well-separated DOAs could be identified with a reasonable (up to 1000) number of snapshots.

3.7.2 Mutual Coupling and Monte-Carlo Simulations

In addition to the increase in MSE due to reduced aperture, another effect to consider in practice is the increased mutual coupling between sensors due to array shrinking. For a demonstrative simulation with mutual coupling, consider a scenario with $m = 12$ sensors and aperture $A = 15$. The goal is to identify more than 15 sources. No integer array (even sparse arrays) in the given aperture can do this. One way to achieve this is to scale the array (3.118) by a factor of 1.5. Note that the resulting

12-element array

$$\mathbf{z} = \begin{bmatrix} 0 & 1.5 & 3 & 4.5 & \dots & 12 & 13.5 & 14.25 & 15 \end{bmatrix} \quad (3.125)$$

is still a rational array. It has all coarray lags of the form

$$3k/4, \text{ where } k = -20, 29, \dots, 19, 20. \quad (3.126)$$

Note that the effective scaling factor for the coarray is 0.75. When the coarray has a uniform segment (possibly rational) with $2m' - 1$ distinct differences, and the effective scaling factor (compared to the integer ULA) is smaller than unity, the array can ideally identify up to $m' - 1$ sources with coarray MUSIC. Thus, similar to (3.118), the 12-element rational array (3.125) can identify up to 20 sources even though the aperture is 15. The increased aperture (compared to (3.118)) helps identify a relatively larger number of sources with a smaller number of snapshots. To identify DOAs with coarray root-MUSIC using this array, we will have to form a degree 40 polynomial by substituting $z = e^{3j\omega/4}$. The roots corresponding to the DOAs lie in the sector of the complex plane that has phase values in the range $[-135^\circ, 135^\circ]$.

Now consider 17 sources with SNR 10 dB impinging on the 12-sensor array (3.125). The sources are well separated, but *not* uniformly spaced in the ω domain. 1000 snapshots are used. We also consider mutual coupling of the form (2.27) with $B = 5$ and coupling coefficients of values $c_0 = 1$, and

$$c_d = 0.1e^{j\pi/4}/d \quad \text{if} \quad 0 < d \leq 5. \quad (3.127)$$

Fig. 3.16 shows the resulting MUSIC spectrum and root-MUSIC plot. Root-MUSIC clearly identifies all 17 sources even in the presence of the mutual coupling considered here.

Notice that due to the sparse nature of such arrays, not many sensor pairs have a distance smaller than 1 between them if the effective scaling factor is not too small. The array (3.125) has only two physical sensor pairs with a distance smaller than 1. Even though the coarray has many ‘virtual sensors’ with a distance smaller than 1, mutual coupling does not depend on the coarray spacing. This is why we saw in Fig. 3.16 that the array (3.125) could still identify all the 17 sources, even in the presence of small mutual coupling as in (3.127).

Fig. 3.17 shows the variation in MSE of root-MUSIC DOA estimates averaged over 2500 Monte-Carlo runs for the array (3.125). Seventeen DOAs uniformly

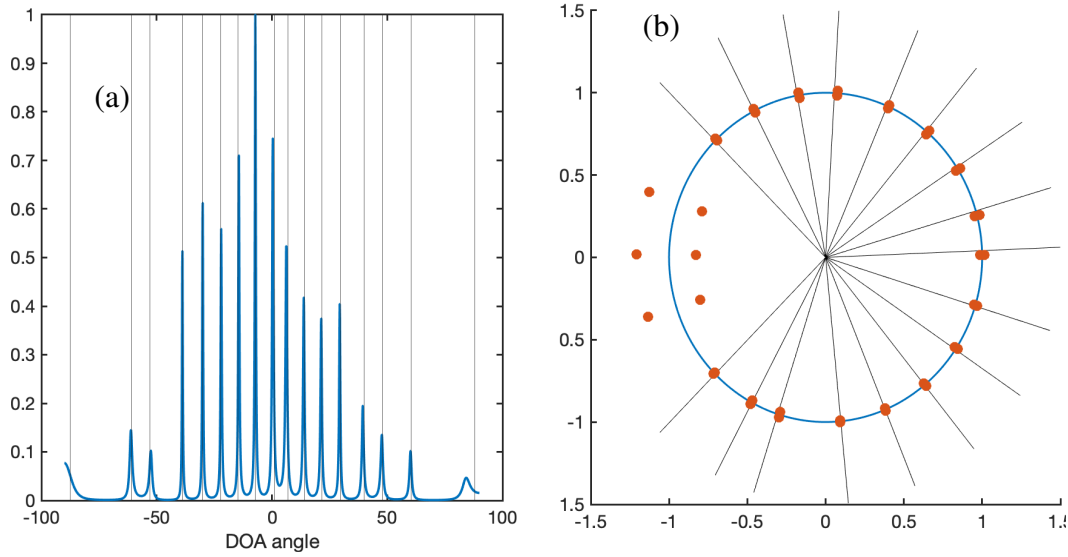


Figure 3.16: Identifying 17 sources with the 12-element rational array (3.125) having aperture $A = 15$ in presence of mutual coupling (3.127). (a) MUSIC spectrum and (b) root-MUSIC plot. The true DOAs (in MUSIC plot) and corresponding angles $3\omega_i/4$ (in root-MUSIC plot) are shown in solid black lines.

spaced in the ω domain are considered. The Cramér-Rao bound (CRB) evaluated according to the expression in [86] is also plotted as a reference. As SNR is varied, keeping the number of snapshots fixed at 1000, the MSE saturates (Fig. 3.17(a)). Whereas, when the number of snapshots is varied keeping the SNR fixed at 10 dB, the MSE reduces monotonically (Fig. 3.17(b)). Furthermore, there is a gap between MSE and CRB in both cases, indicating that the DOA estimates are not statistically efficient. These behaviors are known to arise whenever there are more sources than sensors, even for integer sparse arrays [86], [87]. The MSE in presence of mutual coupling of the form $c_d = 0.1e^{j\pi/4}/d$, $0 < d \leq 5$ is also included in Fig. 3.17(b) for comparison.

3.7.3 General Construction by Scaling Integer Arrays

Note that the arrays (3.118) and (3.125) are in fact scaled versions of a two-level nested array [34]

$$\mathbf{z} = \begin{bmatrix} 0 & 2 & 4 & 6 & \dots & 16 & 18 & 19 & 20 \end{bmatrix}, \quad (3.128)$$

where the sparse subarray is on the left and the dense subarray is on the right. Generalizing this, we can take a general rational nested array to be

$$r \begin{bmatrix} 1 & 2 & \dots & N_1 & (N_1 + 1) & 2(N_1 + 1) & \dots & N_2(N_1 + 1) \end{bmatrix} \quad (3.129)$$

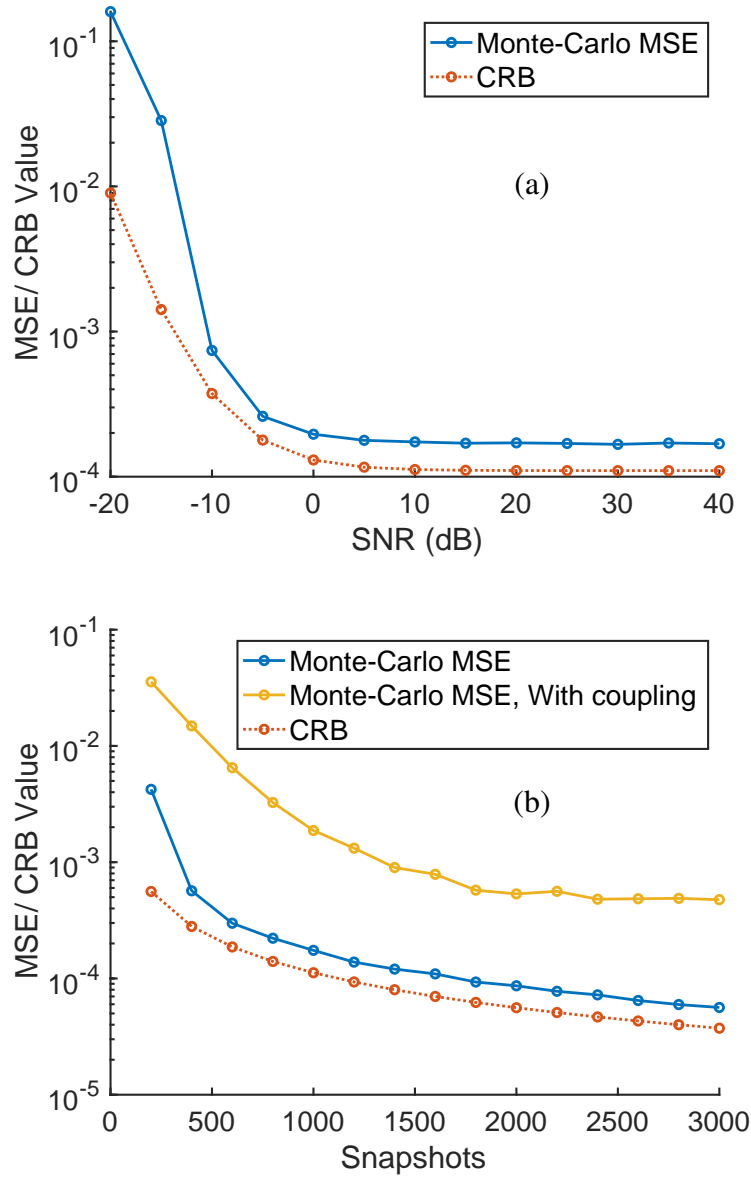


Figure 3.17: Monte-Carlo MSE and CRB for array (3.125) when identifying 17 DOAs equispaced in ω domain. (a) Number of snapshots $K = 1000$, and (b) SNR = 10 dB.

for any rational number r . It contains a dense ULA of N_1 sensors with inter-element distance r , and a sparse ULA of N_2 sensors with inter-element distance $(N_1 + 1)r$. This scaled version of the standard integer nested array has a similarly scaled ULA coarray having $2N_2(N_1 + 1) - 1$ elements [34]. If $r < 1$, the identifiability properties of the coarray will continue to be valid, and we can indeed estimate up to $N_2(N_1 + 1) - 1$ sources with this rational array using coarray MUSIC.

Similarly, we can consider a full version of the coprime rational array (3.56) as follows:

$$\begin{bmatrix} 0 & r_1 & \dots & (m_1 - 1)r_1 & | & r_2 & 2r_2 & \dots & (m_2 - 1)r_2 \end{bmatrix}. \quad (3.130)$$

Here, r_1 and r_2 are coprime rational numbers with $(r_1, r_2) = r < 1$, and m_1 and m_2 are the numbers of elements in the two sparse subarrays. The difference coarray of the array (3.130) has all the elements of the form

$$kr_1 - lr_2, \text{ where } k = 0, 1, \dots, m_1 - 1, l = 0, 1, \dots, m_2 - 1. \quad (3.131)$$

Now, suppose we choose

$$m_1 = n_2 = r_2/r \text{ and } m_2 = 2n_1 = 2r_1/r. \quad (3.132)$$

Note that $n_1, n_2 \in \mathbb{Z}$ and $(n_1, n_2) = 1$ as r is the gcd of r_1 and r_2 . With this particular choice of m_1 and m_2 , the difference coarray elements (3.131) can be written as

$$r[kn_1 - ln_2], \text{ where } k = 0, 1, \dots, n_2 - 1, l = 0, 1, \dots, 2n_1 - 1. \quad (3.133)$$

The coarray elements in (3.133) are nothing but the coarray elements of the integer coprime array [35]

$$\begin{bmatrix} 0 & n_1 & \dots & (n_2 - 1)n_1 & | & n_2 & \dots & (2n_1 - 1)n_2 \end{bmatrix}, \quad (3.134)$$

scaled by a rational factor $r = (r_1, r_2)$. Integer coprime array (3.134) is known to have a ULA segment from $-n_1(n_2 + 1) + 1$ to $n_1(n_2 + 1) - 1$ in its coarray [83]. Thus, the rational coprime array (3.130) has all coarray lags of the form

$$r \cdot k, \text{ where } k = -n_1(n_2 + 1) + 1, \dots, n_1(n_2 + 1) - 1. \quad (3.135)$$

Hence, the particular choice (3.132) simplifies the design procedure for rational coprime arrays. We can start with any integer coprime array (3.134) with $(n_1, n_2) = 1$ and scale it by a factor $r < 1$ to get a rational coprime array (3.130), where $m_1 = n_2$, $m_2 = 2n_1$, $r_1 = rn_1$, and $r_2 = rn_2$. The rational coprime array designed this way can identify $n_1(n_2 + 1) - 1$ sources through coarray MUSIC or root-MUSIC, just like the integer coprime array (3.134). To apply root-MUSIC in the coarray domain with arrays (3.129) and (3.130), we have to substitute $z = e^{j\omega r}$ to form the root-MUSIC polynomial. MRAs can also be scaled similarly when the aperture is constrained.

3.8 Concluding Remarks

In this chapter, we considered non-integer arrays for DOA estimation. In particular, the proposed rational arrays offer more flexibility in array design and have some advantages over integer arrays when a limited number of sensors are to be distributed in a fixed aperture. Along with qualitatively better MUSIC plots with sharp peaks, the rational arrays provided the lowest MSE in DOA estimation that is close to the CRB, even with a low number of snapshots and low SNR. Also, rational arrays resolved nearby sources well and performed better than integer arrays in most two-DOA scenarios. They were also shown to be useful when the spatial scope of impinging signals is restricted. In order to systematically analyze such arrays, we extended number-theoretic concepts such as gcd, lcm, and coprimality to rational numbers. This allowed us to characterize the condition for steering vector invertibility, propose rational coprime arrays, and provide a result on unique identifiability with MUSIC. These results are further extended for the case when signals are impinging from a restricted spatial scope. The notion of coprimality is further extended to real numbers, thus allowing us to propose arbitrary (non-rational) coprime arrays. The theoretical results for arbitrary (non-rational) arrays follow easily because of the proposed generalization of the integer array framework, first to the rational case and then to arbitrary (non-rational) sensor location arrays. Lastly, we presented a unique advantage of rational arrays through their difference coarrays. The rational arrays can identify $O(m^2)$ uncorrelated sources using m sensors even in the presence of aperture constraint. Sparse integer arrays can do this only if a large aperture is available.

Extending gcd and coprimality concepts to rational matrices is an appealing future work that may motivate 2D rational arrays. It would also be interesting to explore other non-subspace-based DOA estimation algorithms in the context of rational arrays as future work. Designing rational arrays that satisfy some rational coprimality condition for the difference coarray positions would be useful to explore in the context of coarray-MUSIC. The effect of mutual coupling on coarray shrinking can also be explored further. The general construction of arrays provided in this chapter for identifying $O(m^2)$ DOAs under aperture constraint was based on scaling on already existing sparse integer arrays with large central ULA segments. This is because we still required the coarray to be uniform, albeit rational. This leads to the following questions: i) how to estimate DOAs using non-uniform rational difference coarrays, and ii) how to choose rational sensor locations to get a particular desired non-uniform difference coarray. These questions can be explored in the future.

Chapter 4

WEIGHT-CONSTRAINED ARRAYS FOR DOA ESTIMATION UNDER HIGH MUTUAL COUPLING AND APERTURE CONSTRAINTS

4.1 Introduction

In the last chapter, we developed rational arrays for DOA estimation, where the sensors are placed at rational locations (in multiples of half-wavelength distance $\lambda/2$). This was primarily motivated by the better flexibility that rational arrays offer under aperture constraints. In this chapter, we consider another commonly encountered challenge in practice: mutual coupling between sensors (Sec. 2.4). Sparse integer arrays are useful to reduce the impact of mutual coupling because the sensors are farther apart from each other compared to a ULA. The commonly encountered sparse integer arrays in the literature, such as MRAs, nested arrays, and coprime arrays, were discussed in Sec. 2.3. Such well-designed sparse arrays can identify up to $O(N^2)$ DOAs through the difference coarray domain when the source amplitudes are uncorrelated. Although nested arrays and coprime arrays can identify fewer DOAs than an MRA with the same number of sensors, their closed-form expression for sensor positions for any N makes them more practical and attractive. These arrays with closed-form expressions of sensor positions that can identify $O(N^2)$ DOAs through the coarray domain have revolutionized the research on sparse array design.

4.1.1 Traditional Array Design Criteria and Their Drawbacks

Following the development of nested arrays and coprime arrays, several generalizations of these arrays, such as generalized coprime arrays [83] and improved nested arrays [102], have been proposed. Several extensions of these arrays including super nested arrays [39], [101], augmented nested arrays [103], dilated nested arrays [92], MISC arrays [104], thinned coprime arrays [91], and padded coprime arrays [105] have also been advanced. Most of these sparse array designs are motivated by the following three key principles:

1. Sensor locations should be easily describable using a closed-form expression or simple generation rules.

2. The difference coarray of the array should have a long central ULA segment, typically of $O(N^2)$ length.
3. The number of sensor pairs with small separations should be kept to a minimum.

The closed-form expression for sensor locations (criterion 1) is necessary to create arrays easily for any given number of sensors N . The large central ULA segment of $O(N^2)$ length in the coarray (criterion 2) is enforced to ensure that the arrays can identify a large number of DOAs using coarray-based DOA estimation algorithms such as coarray-MUSIC and coarray-ESPRIT. However, note that a large difference coarray with $O(N^2)$ unique lags also means that the array aperture is of length $O(N^2)$. Such a large aperture may not be practical when a large number of sensors N is used. It is becoming more common to use almost a hundred sensors for large antenna arrays and massive MIMO in 6G technology [150], [151]. With such a large number of sensors, it may no longer be necessary for the arrays to have the capability to identify $O(N^2)$ sources because the number of DOAs is typically not that large. In practical situations where the arrays are deployed, such as in smartphones and self-driving cars, there may be physical constraints on the array aperture. In such cases with aperture constraints, an $O(N^2)$ aperture may not be of interest or desirable.

The third criterion above is to reduce the impact of mutual coupling between sensors [36], [37], [39] on DOA estimation. We discussed in Sec. 2.4 that the exact impact of mutual coupling on the array output is unknown to the user and can be difficult to model precisely. However, the sensor pairs with small separations contribute the most to the mutual coupling. Thus, to reduce the effect of mutual coupling on DOA estimation error, the coarray weights for small lags such as $w(1)$, $w(2)$, and $w(3)$ that contribute the most to the effect of mutual coupling are kept to a minimum (criterion 3). However, in order to ensure that the difference coarray of the array has a large *central* ULA segment, these coarray weights are ensured to be non-zero. Some extensions of coprime arrays, such as extended padded coprime array [105] and thinned coprime array [91], can already achieve the smallest possible non-zero coarray weights at lags 1, 2, and 3 (i.e., $w(1) = w(2) = w(3) = 1$). However, as we will see later in Sec. 4.5, in the presence of high mutual coupling, these arrays may still result in poorer DOA estimation compared to proposed arrays because of non-zero $w(1)$, $w(2)$, and $w(3)$.

Table 4.1: Key criteria considered for sparse array design, their drawbacks, and modifications considered in this chapter.

Traditionally used array design criteria	Drawback of the criterion	Proposed Modification to the criterion	Drawback for the modification
Closed-form expression or simple rule for sensor positions	Can identify fewer DOAs than MRA with coarray-MUSIC	None	-
Central ULA segment of length $O(N^2)$ in coarray	$O(N^2)$ aperture may not be suitable, and $O(N^2)$ DOFs may not be needed for large N	$O(N)$ aperture sparse arrays having ULA segments of $O(N)$ length in their coarrays	Can identify only up to $O(N)$ DOAs
Coarray weights $w(l)$ small but non-zero for $l = 1, 2, 3$	May not be robust to high mutual coupling	$w(1) = 0$ or $w(1) = w(2) = 0$	Can identify fewer DOAs due to one-sided ULA segment in coarray

To summarize, when dealing with arrays that have a large number of sensors N , we may no longer require $O(N^2)$ degrees of freedom (DOFs). Instead, other practical considerations, such as aperture constraint and robustness to high mutual coupling, become a priority. To address these requirements, we consider modifications to the second and third array design criteria in this chapter and accordingly design sparse arrays. Instead of aiming for a ULA segment of $O(N^2)$ length in the coarray, we propose to develop arrays with $O(N)$ aperture that have a large ULA segment of $O(N)$ length in the coarray. While such sparse arrays that are suitable under aperture constraint can no longer identify $O(N^2)$ DOAs, they can still identify $D > N$ DOAs, as we will see. Furthermore, instead of enforcing coarray weights $w(1)$ and $w(2)$ to be non-zero, we propose arrays that have either $w(1) = 0$ or $w(1) = w(2) = 0$, to further reduce the impact of mutual coupling. Although such arrays do not have a *central* ULA segment in the coarray, we still ensure that there exists a large *one-sided* ULA segment in their difference coarrays. Table 4.1 summarizes the proposed modifications to the array design criteria and their advantages and disadvantages. Sparse arrays with a closed form for sensor positions, such as nested arrays and coprime arrays, can identify fewer DOAs compared to MRAs. However, they became popular because of the practical benefit of having closed-form sensor locations (criterion 1) for any number of sensors. By making modifications to the other two design criteria as described in Table 4.1, we further sacrifice the maximum number of identifiable DOAs, but in turn, achieve practical benefits such as improved robustness to high mutual coupling and suitability under aperture constraint.

4.1.2 Our Contributions and Connections to Literature

Adhering to the modified array design criteria, we first develop several sparse and weight-constrained arrays that have $O(N)$ aperture and either $w(1) = 0$ or $w(1) = w(2) = 0$. These constructions are based on appropriately dilating a uniform linear array (ULA) and augmenting it with a few additional sensors. The *one-sided* ULA segments in the difference coarrays of these arrays start either from lag $l = 2$ or $l = 3$, and the coarrays have only a few holes. These arrays have a small aperture of $O(N)$ length, compared to the traditional sparse arrays having $O(N^2)$ aperture. Thus, when there is a fixed aperture constraint, these arrays can place more sensors in the same aperture compared to traditional sparse arrays. These arrays are better suited when the aperture is constrained, the number of DOAs D is $O(N)$, and high mutual coupling is present. One may wonder why we cannot use an N -sensor uniform linear array (ULA) instead of the proposed $O(N)$ aperture arrays.

This is because DOA estimation using ULA can suffer significantly in the presence of mutual coupling due to the densely packed sensors. Furthermore, an N -sensor ULA can identify only up to $N - 1$ DOAs, whereas the proposed $O(N)$ aperture arrays can identify $D > N$ DOAs (such as $D = 2N$) through the difference coarray domain, although these arrays no longer have $O(N^2)$ degrees of freedom.

Next, for the situation where robustness to high mutual coupling is needed but the array aperture is not constrained, we propose another type of sparse and weight-constrained arrays. These arrays have either $w(1) = 0$ or $w(1) = w(2) = 0$ and $O(N^2)$ degrees of freedom. We achieve this by suitably dilating a nested array and appending it with several extra sensors. The one-sided ULA segment in the difference coarray of these arrays starts either from $l = 2$ or $l = 3$ and has $O(N^2)$ length. Through Monte-Carlo simulations, we demonstrate that in certain situations, our proposed arrays, having $O(N^2)$ aperture and $O(N^2)$ degrees of freedom, can outperform arrays proposed in the literature. Please note that the proposed arrays do not have a *central* ULA segment in their difference coarrays (because $w(1) = 0$), unlike many other well-known sparse arrays. However, we can still apply a modified coarray root-MUSIC on the one-sided ULA segment of the coarray. It is important to note that using a one-sided ULA segment from the coarray reduces the number of identifiable DOAs using coarray-MUSIC compared to the case when there exists a central ULA segment in the coarray.

We now explain the novelty of our contributions in light of what is already known about sparse arrays. To the best of our knowledge, $O(N)$ aperture arrays that can identify more DOAs than the number of sensors have not been systematically developed, or even mentioned in the literature before. Furthermore, to the best of our knowledge, all simulations in the literature considered a fixed number of sensors N . Instead of only comparing different array geometries for a fixed number of sensors N , we also propose comparing different array geometries under a fixed aperture constraint. This can be called ‘aperture-constrained’ evaluation of sensor arrays. To achieve good performance under such aperture-constrained settings, an ‘aperture-aware’ design of sparse arrays is required. One way to do this is to explicitly impose a strict constraint on the array aperture in the design stage. In this chapter, we take a different approach and propose a class of array geometries with a smaller $O(N)$ aperture instead of a large $O(N^2)$ aperture. Because of this, the proposed $O(N)$ aperture arrays can be more suitable than many other arrays from the literature under aperture constraints. In summary, motivated by practical aperture constraints that

certain applications may require, we propose an ‘aperture-constrained’ evaluation of sensor arrays and an ‘aperture-aware’ design approach, through this chapter. Such designs can be more practically useful than the large $O(N^2)$ aperture arrays for certain applications.

Similarly, arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ are not very common in the literature. The CADiS array proposed in [83] is one such sparse array construction with a closed form for sensor locations and $w(l) = 0$ for small values of l such as 1, 2, 3, etc. However, CADiS arrays are not available for certain numbers of sensors (such as 17, 23, and 29) [91]. Furthermore, under aperture constraint, it may not be easy to come up with the design parameters (integers M , N , and p from [83] such that M and N are coprime and p is a divisor of M). Additionally, some of our proposed arrays with $O(N^2)$ aperture can identify more DOAs than CADiS with coarray-MUSIC, as we will see. This is because the one-sided ULA segment in the difference coarray of CADiS does not start immediately after a few initial holes, unlike the proposed arrays. Another array construction called generalized nested array (GNA) [152] includes CADiS arrays as its special case and shares the property that $w(l) = 0$ for small values of l . Similar to the CADiS array, it may not be easy to decide design parameters (coprime integers α and β from [152]), especially under aperture constraint. Some arrays tabulated in [84] also have $w(1) = 0$, but these are obtained through an elaborate search and do not have closed-form expressions for the sensor positions. The tabulation is available only for $N \leq 20$.

The contributions and outline of this chapter are as follows:

1. We discuss the drawbacks of the sparse array design criteria and motivate modifications proposed in Table 4.1 (Sec. 4.1.1).
2. We describe and justify the application of coarray-MUSIC to a one-sided ULA segment in coarray (Sec. 4.2).
3. We introduce new $O(N)$ aperture arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ that can perform better than many sparse arrays from literature (Sec. 4.3).
4. We propose $O(N^2)$ aperture arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ that can identify a large number of DOAs under high mutual coupling (Sec. 4.4).
5. We compare the coarray properties and coarray-MUSIC spectra of the proposed arrays with several arrays from the literature (Sec. 4.5.1). We present

thorough Monte-Carlo simulations under two simulation settings: fixed N (Sec. 4.5.2), and fixed aperture A (Sec. 4.5.3). We present several additional simulations to demonstrate the effect of coupling parameters and source correlations (Secs. 4.5.4, 4.5.5), and discuss when we can expect the proposed arrays to perform well (Sec. 4.5.6).

The content of this chapter is based on papers [153]–[155].

4.2 Coarray-MUSIC Using One-Sided ULA Segment in Coarray

The new arrays proposed in this chapter do not have a *central* ULA segment in the coarray, as they have $w(1) = 0$. These arrays have one-sided ULA segments in the coarray from lag L_1 to L_2 where $0 < L_1 < L_2$. In this section, we describe how to estimate DOAs using such one-sided ULA segment of length $L = L_2 - L_1 + 1$ in the coarray, and show that up to $\lfloor L/2 \rfloor$ DOAs can be identified using the procedure described. This procedure is similar to the direct-augmentation approach with matrix $\hat{\mathbf{R}}_{DA}$ described in Sec. 2.5. However, there are some subtleties involved as we explain next.

Similar to $\hat{\mathbf{R}}_{DA}$, based on the estimated correlations $\hat{R}(l)$, $L_1 \leq l \leq L_2$, we can create a Toeplitz matrix $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{R}(L_2 - r_1 + 1) & \hat{R}(L_2 - r_1) & \dots & \hat{R}(L_1 + 1) & \hat{R}(L_1) \\ \hat{R}(L_2 - r_1 + 2) & \hat{R}(L_2 - r_1 + 1) & \dots & \hat{R}(L_1 + 2) & \hat{R}(L_1 + 1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{R}(L_2 - 1) & \hat{R}(L_2 - 2) & \dots & \hat{R}(L_1 + r_1 - 1) & \hat{R}(L_1 + r_1 - 2) \\ \hat{R}(L_2) & \hat{R}(L_2 - 1) & \dots & \hat{R}(L_1 + r_1) & \hat{R}(L_1 + r_1 - 1) \end{bmatrix}. \quad (4.1)$$

The matrix $\hat{\mathbf{R}}$ has size $r_1 \times r_2$, where $r_1 = \lfloor L/2 \rfloor + 1$ and $r_2 = \lceil L/2 \rceil$, and its entries are $\hat{\mathbf{R}}_{m,n} = \hat{R}(L_2 - r_1 + 1 + m - n)$. The k -th column of $\hat{\mathbf{R}}$ contains r_1 consecutive correlations starting from $\hat{R}(L_1 + r_2 - k)$. Note that when $L_1 = -L_2$, $\hat{\mathbf{R}} = \hat{\mathbf{R}}_{DA}$. $\hat{\mathbf{R}}$ is a square matrix when L is odd, whereas $\hat{\mathbf{R}}$ is a tall matrix when L is even. Even when $\hat{\mathbf{R}}$ is square, it is not a Hermitian matrix in general, unlike $\hat{\mathbf{R}}_{DA}$ and $\hat{\mathbf{R}}_{SS}$. We would like to note that later in Sec. 6.3, two variations of the coarray-covariance matrix (namely, ‘tall’ and ‘fat’ variations) are proposed. Here, $\hat{\mathbf{R}}$ is formed in a similar way, but using the correlation entries from a one-sided ULA segment in the coarray. Furthermore, unlike that in Sec. 6.3, the size of $\hat{\mathbf{R}}$ is fixed and does not depend on the number of DOAs D . The dimensions $(\lfloor L/2 \rfloor + 1) \times \lceil L/2 \rceil$ of the matrix $\hat{\mathbf{R}}$ are chosen such that the number of DOAs we can identify using a

subspace-based algorithm is maximized. As explained next, up to $\lfloor L/2 \rfloor$ DOAs can be identified by computing the SVD of $\widehat{\mathbf{R}}$.

To see this, first consider an ideal version of the matrix $\widehat{\mathbf{R}}$, with entries $R(l)$ instead of $\widehat{R}(l)$. With a large number of snapshots, the matrix $\widehat{\mathbf{R}}$ converges to the ideal $r_1 \times r_2$ matrix \mathbf{R} with entries

$$[\mathbf{R}]_{m,n} = R(L_2 - r_1 + 1 + m - n). \quad (4.2)$$

As the ideal correlations $R(l)$ have the sum-of-sinusoids form (Eq. (2.21)), the last (i.e. r_2 -th) column of \mathbf{R} containing entries $R(L_1)$ to $R(L_1 + r_1 - 1)$ can be written as

$$\mathbf{R}(:, r_2) = \underbrace{\begin{bmatrix} 1 & \dots & 1 \\ e^{j\omega_1} & \dots & e^{j\omega_D} \\ \vdots & \ddots & \vdots \\ e^{j\omega_1(r_1-1)} & \dots & e^{j\omega_D(r_1-1)} \end{bmatrix}}_{\mathbf{A}_c} \begin{bmatrix} p_1 e^{j\omega_1 L_1} \\ p_2 e^{j\omega_2 L_1} \\ \vdots \\ p_D e^{j\omega_D L_1} \end{bmatrix}. \quad (4.3)$$

Similarly, the k -th column of \mathbf{R} can be expressed as

$$\mathbf{R}(:, k) = \mathbf{A}_c \mathbf{b}_k, \quad (4.4)$$

where \mathbf{b}_k is a D -dimensional vector with m -th entry $[\mathbf{b}_k]_m = p_m e^{j\omega_m(L_1+r_2-k)}$. Collecting such equations for all columns of \mathbf{R} (i.e. for $k = 1, \dots, r_2$), we get

$$\mathbf{R} = \mathbf{A}_c \mathbf{P} \mathbf{B}_c^H, \quad (4.5)$$

where \mathbf{A}_c is an $r_1 \times D$ matrix with entries $[\mathbf{A}_c]_{m,n} = e^{j\omega_n(m-1)}$, \mathbf{P} is a $D \times D$ diagonal matrix containing source powers p_1, \dots, p_D , and \mathbf{B}_c is an $r_2 \times D$ matrix with entries $[\mathbf{B}_c]_{m,n} = e^{-j\omega_n(L_1+r_2-m)}$. Since the DOAs $\omega_1, \dots, \omega_D$ are distinct, as long as $r_1 \geq D$ and $r_2 \geq D$ the Vandermonde matrices \mathbf{A}_c and \mathbf{B}_c have rank D . Thus, using Sylvester's rank inequality for matrices, it can be concluded that matrix \mathbf{R} also has rank D . Now, as \mathbf{A}_c and \mathbf{R} have the same rank D , and any vector \mathbf{x} satisfying $\mathbf{x}^H \mathbf{A}_c = 0$ also satisfies $\mathbf{x}^H \mathbf{R} = 0$ (from Eq. (4.5)), it can be concluded that matrices \mathbf{A}_c and \mathbf{R} have the same column spans.

In practice, we only have access to the matrix $\widehat{\mathbf{R}}$. We can estimate the column span of \mathbf{A}_c (and its orthogonal complement) through an SVD of $\widehat{\mathbf{R}}$:

$$\widehat{\mathbf{R}} = \widehat{\mathbf{U}} \widehat{\mathbf{\Sigma}} \widehat{\mathbf{V}}^H, \quad (4.6)$$

where $\widehat{\mathbf{U}}$ is an $r_1 \times r_1$ matrix with orthonormal columns, and $\widehat{\mathbf{\Sigma}}$ is an $r_1 \times r_2$ matrix containing singular values arranged in descending order along the principal diagonal. The first D columns of $\widehat{\mathbf{U}}$ provide an approximation to the column span of \mathbf{A}_c . The matrix $\widehat{\mathbf{U}}_n$ containing the last $r_1 - D$ columns of $\widehat{\mathbf{U}}$ provides an approximation to the ‘noise subspace’ (the orthogonal complement of \mathbf{A}_c). As long as $D < r_1$, we obtain a non-trivial noise subspace with a dimension of at least one. Thus, by using $\widehat{\mathbf{U}}_n$ in place of $\widehat{\mathbf{E}}_n$, and $\mathbf{a}_c(z)$ (where $[\mathbf{a}_c(z)]_i = z^{i-1}$) in place of $\mathbf{a}(z)$ in Eq. (2.15), we can get a coarray root-MUSIC polynomial, from which the DOAs can be estimated.

In summary, even when $\widehat{\mathbf{R}}$ is not a square matrix, we can apply SVD to estimate signal subspace and noise subspace from $\widehat{\mathbf{R}}$ and use coarray-based root-MUSIC to estimate DOAs using the one-sided ULA segment from the difference coarray. Note that $D < r_1$ and $D \leq r_2$ for the above-described method to work. Thus, to maximize the number of DOAs we can identify using this approach, $\min(r_1 - 1, r_2)$ should be as large as possible. As we only have the L contiguous correlation estimates $\widehat{R}(L_1), \dots, \widehat{R}(L_2)$ to form the Teopltitz matrix $\widehat{\mathbf{R}}$, r_1 and r_2 should also satisfy $r_1 + r_2 - 1 = L$. Under this constraint, it can be verified that the largest value for $\min(r_1 - 1, r_2)$ is obtained when $r_1 = \lfloor L/2 \rfloor + 1$ and $r_2 = \lceil L/2 \rceil$. Thus it is possible to identify a maximum of $r_1 - 1 = \lfloor L/2 \rfloor$ DOAs in this way by applying SVD to the matrix $\widehat{\mathbf{R}}$ of size $(\lfloor L/2 \rfloor + 1) \times \lceil L/2 \rceil$.

We would like to note that some papers, such as [83], [91], have reported DOA estimation performance using a subspace-based algorithm with CADiS arrays that have a one-sided ULA segment in the coarray. However, they do not describe or justify their method for identifying DOAs using such a one-sided ULA segment in coarray, or derive the maximum number of identifiable sources. In fact, the number of identifiable sources reported in [91] for CADiS array when using subspace-based DOA estimation is one less than $\lfloor L/2 \rfloor$ when L is even.

It is possible to use techniques other than coarray-based algorithms to identify more DOAs than the number of sensors. Some examples include the weighted least squares method [90], stochastic maximum likelihood estimation [121], and dictionary-based methods [156]. This chapter primarily focuses on introducing new sparse arrays that can handle strong mutual coupling and aperture constraints, rather than focusing on various algorithms that can be applied for DOA estimation. With suitable adaptations, it is possible to use other DOA estimation algorithms with the proposed arrays. This section demonstrated the adaptation of widely used coarray-MUSIC for one-sided ULA segments in the difference coarray. Other algorithms,

such as coarray interpolation and dictionary-based estimation, will be considered in the next chapter.

4.3 Proposed Weight-Constrained Sparse Arrays with $O(N)$ Aperture

In this section, we first systematically develop sparse arrays that have an aperture of $O(N)$ length and $w(1) = 0$. We will start by explaining a procedure to generate such arrays by properly dilating a ULA and adding a few extra sensors. We will then propose modifications to the construction to make other crucial weights ($w(2)$ and $w(3)$) small constants independent of N while keeping $w(1) = 0$. We will also develop sparse arrays that have $O(N)$ aperture and $w(1) = w(2) = 0$. Moreover, we will discuss further modifications to make $w(3)$ and $w(4)$ small constants independent of N . The coarray properties of the arrays proposed in this section are summarized in Table 4.2. These arrays will be called weight-constrained sparse arrays (WCSAs). We assume $N > 5$ to ensure that the general expressions for all arrays are valid.

4.3.1 Proposed Sparse Arrays with $O(N)$ Aperture and $w(1) = 0$

We want to create a sparse array with $O(N)$ aperture and $w(1) = 0$. We also want to ensure that the array has a large ULA segment in its coarray from lag two onwards for easy application of coarray root-MUSIC. To achieve this, we follow these steps:

1. Expand an $(N - 1)$ -sensor ULA by a factor of two to create a ‘2-sparse’ ULA $\{0, 2, \dots, 2(N - 2)\}$. This is called the array *dilation* step.
2. Place the N -th sensor at location $2N - 1$. This is called the array *augmentation* step.

Thus, we get the array

$$\mathbf{z}_1 = \left[\underbrace{0 \quad 2 \quad 4 \quad \dots \quad 2(N - 2)}_{\text{2-sparse ULA}} \quad 2N - 1 \right]. \quad (4.7)$$

Note that the coarray of the 2-sparse ULA consists only of the even lags from $-2(N - 2)$ to $2(N - 2)$. The augmented sensor is placed at a distance of three from the last sensor of the 2-sparse ULA and is at odd distances from the other $N - 1$ sensors. This creates lags $3, 5, \dots, 2N - 1$ in the coarray, which were missing from the coarray of the 2-sparse ULA. Thus, the coarray $\mathbb{D}_{\mathbf{z}_1}^+$ of \mathbf{z}_1 is

$$\mathbb{D}_{\mathbf{z}_1}^+ = \{0, 2, 3, 4, \dots, 2N - 4, 2N - 3, 2N - 1\}. \quad (4.8)$$

One-sided ULA segment in coarray

Table 4.2: Coarray properties of the weight-constrained sparse arrays with $O(N)$ aperture proposed in Sec. 4.3. Array aperture (A), coarray weights $w(l)$, ULA segment in the coarray, coarray holes, and maximum number of identifiable DOAs with coarray-MUSIC D_m are compared. ‘ A ’ in the second last column refers to the aperture listed in the second column. Array \mathbf{z}_6 has $w(1) = w(2) = 0$ and constant $w(3)$ and $w(4)$ independent of N .

Array	Aperture A	Coarray Weights					ULA Segment in Coarray			Holes in coarray	D_m
		$w(1)$	$w(2)$	$w(3)$	$w(4)$	$w(5)$	Start (L_1)	End (L_2)	Length (L)		
ULA	$N - 1$	$N - 1$	$N - 2$	$N - 3$	$N - 4$	$N - 5$	$-N + 1$	$N - 1$	$2N - 1$	None	$N - 1$
\mathbf{z}_1	$2N - 1$	0	$N - 2$	1	$N - 3$	1	2	$2N - 3$	$2N - 4$	1, $A - 1$	$N - 2$
$\mathbf{z}_2^{(1)}$	$3N - 5$	0	2	$N - 3$	1	1	2	$3N - 7$	$3N - 8$	1, $A - 1$	$\lfloor 1.5N \rfloor - 4$
$\mathbf{z}_2^{(2)}$	$3N - 3$	0	1	$N - 3$	1	1	2	$3N - 7$	$3N - 8$	1, $A - 3$, $A - 1$	$\lfloor 1.5N \rfloor - 4$
$\mathbf{z}_3^{(1)}$	$4N - 9$	0	2	1	$N - 4$	1	2	$4N - 13$	$4N - 14$	1, $A - 3$, $A - 1$	$2N - 7$
$\mathbf{z}_3^{(2)}$	$4N - 8$	0	1	2	$N - 4$	1	2	$4N - 13$	$4N - 14$	1, $A - 4$, $A - 2$, $A - 1$	$2N - 7$
\mathbf{z}_4	$3N$	0	0	$N - 3$	1	1	3	$3N - 7$	$3N - 9$	1, 2, $A - 6$, $A - 3$, $A - 2$, $A - 1$	$\lfloor (3N - 9)/2 \rfloor$
\mathbf{z}_5	$4N - 5$	0	0	2	$N - 4$	1	3	$4N - 13$	$4N - 15$	1, 2, $A - 7$, $A - 4$, $A - 2$, $A - 1$	$2N - 8$
\mathbf{z}_6	$5N - 12$	0	0	3	1	$N - 5$	3	$5N - 21$	$5N - 23$	1, 2, $A - 8$, $A - 5$, $A - 4$, $A - 2$, $A - 1$	$\lfloor (5N - 23)/2 \rfloor$

The array \mathbf{z}_1 has an aperture $A = 2N - 1$ and its coarray includes a ULA segment of length $L = 2N - 4$ starting from $L_1 = 2$ and ending at $L_2 = 2N - 3$. Since the coarray is symmetric, it also contains a ULA segment from $-L_2$ to $-L_1$. As explained in Sec. 4.2, we can identify up to $\lfloor L/2 \rfloor = N - 2$ DOAs by using coarray root-MUSIC with this array. Although an N -sensor ULA can identify one more DOA than \mathbf{z}_1 , the proposed array can tolerate mutual coupling better than the ULA, as we will see in Section 4.5, because it has $w(1) = 0$.

Note that the coarray $\mathbb{D}_{\mathbf{z}_1}^+$ has two holes located at positions 1 and $2N - 2$. It is not possible to eliminate the extra hole at position $2N - 2$. For any array that has $w(1) = 0$, $w(A - 1)$ must also be zero. This is because there are sensors at 0 and A for an array with aperture A . No sensor can be placed at either location 1 or $A - 1$, as that will create a sensor pair with separation 1. With no sensors at 1 and $A - 1$, $w(A - 1)$ must be zero. Thus, there are no holes in the coarray $\mathbb{D}_{\mathbf{z}_1}^+$ apart from those that arise as a necessary consequence of our design constraint $w(1) = 0$.

4.3.2 Further Reducing $w(2)$ and $w(3)$

Even though the array \mathbf{z}_1 can make $w(1)$ equal to zero, it has $w(2)$ equal to $N - 2$ because of the 2-sparse ULA segment in the array. To further reduce the mutual coupling, we need to construct an array where $w(2)$ is a small constant independent of the number of sensors, while also ensuring that $w(1)$ is zero. To achieve this, we can start with a ‘3-sparse’ ULA having $(N - 2)$ sensors and add two more sensors at a distance of either 2 or 4 units from the same end or opposite ends of the 3-sparse ULA.

Augmenting both sensors on the same side gives the array

$$\mathbf{z}_2^{(1)} = \left[0 \quad 3 \quad 6 \quad \dots \quad \underbrace{3(N - 3)}_{\text{3-sparse ULA}} \quad 3N - 7 \quad 3N - 5 \right], \quad (4.9)$$

and its coarray is

$$\mathbb{D}_{\mathbf{z}_2^{(1)}}^+ = \{0, \underline{2, 3, \dots, 3N - 8, 3N - 7, 3N - 5}\}. \quad (4.10)$$

The ULA segment in the coarray (underlined) is from $L_1 = 2$ to $L_2 = 3N - 7$. Similar to the previous design, the coarray has only two holes located at positions 1 and $3N - 6$ ($= A - 1$). An important advantage of this design over \mathbf{z}_1 is that $w(2) = 2$, independent of the number of sensors N used. This further reduces the impact of mutual coupling compared to the previous design \mathbf{z}_1 . However, $\mathbf{z}_2^{(1)}$ has larger aperture than \mathbf{z}_1 .

Instead of placing both additional sensors on the same side, they can be placed on opposite sides of the 3-sparse ULA to get the array

$$\mathbf{z}_2^{(2)} = \left[-2 \quad 0 \quad 3 \quad 6 \quad \dots \quad \underbrace{3(N-3)}_{\text{3-sparse ULA}} \quad 3N-5 \right]. \quad (4.11)$$

By doing so, $w(2)$ now becomes 1 instead of 2. The coarray still has ULA segment from 2 to $3N-7$, but the array aperture is now $3N-3$ instead of $3N-5$, and the coarray $\mathbb{D}_{\mathbf{z}_2^{(2)}}^+$ has three holes at 1, $3N-6$ ($= A-3$), and $3N-4$ ($= A-1$).

The arrays $\mathbf{z}_2^{(1)}$ and $\mathbf{z}_2^{(2)}$ make $w(2)$ a constant independent of the number of sensors N , but their $w(3)$ increases linearly with N because of the 3-sparse ULA segment in the array. To make $w(1) = 0$ and both $w(2)$ and $w(3)$ small constants, we propose the following construction. We use a 4-sparse ULA of $N-3$ sensors with appropriately placed three other sensors to get the array

$$\mathbf{z}_3^{(1)} = \left[-2 \quad 0 \quad 4 \quad \dots \quad \underbrace{4(N-4)}_{\text{4-sparse ULA}} \quad 4N-13 \quad 4N-11 \right]. \quad (4.12)$$

Its coarray is

$$\mathbb{D}_{\mathbf{z}_3^{(1)}}^+ = \{0, \underline{2, 3, 4, \dots, 4N-14, 4N-13, 4N-11, 4N-9}\}. \quad (4.13)$$

This array construction has constant $w(1) = 0$, $w(2) = 2$, $w(3) = 1$ independent of the number of sensors N , and its coarray has ULA segment from $L_1 = 2$ to $L_2 = 4N-13$. By rearranging the three additionally placed sensors, we get the following variation of this array configuration:

$$\mathbf{z}_3^{(2)} = \left[-5 \quad -2 \quad 0 \quad 4 \quad \dots \quad \underbrace{4(N-4)}_{\text{4-sparse ULA}} \quad 4N-13 \right]. \quad (4.14)$$

The advantage of this array over $\mathbf{z}_3^{(1)}$ is that $w(2)$ is reduced to 1. In the next subsection, to further reduce the impact of mutual coupling, we develop sparse arrays with $w(1) = w(2) = 0$.

4.3.3 Proposed Arrays with $O(N)$ Aperture and $w(1) = w(2) = 0$

We can construct $O(N)$ aperture arrays that have both $w(1) = w(2) = 0$ by using a similar construction mechanism of ULA dilation and augmentation. Consider an array with a 3-sparse ULA of $N-2$ sensors and two additional sensors placed at distances of 4 and 5 units from the opposite ends of the 3-sparse ULA:

$$\mathbf{z}_4 = \left[-5 \quad 0 \quad 3 \quad 6 \quad \dots \quad \underbrace{3(N-3)}_{\text{3-sparse ULA}} \quad 3N-5 \right]. \quad (4.15)$$

The sensor located at $3N - 5$ is positioned at a distance of 4 from one end of the 3-sparse ULA. This sensor produces differences $\{4, 7, 10, \dots, 3N - 5\}$ with the sensors of the 3-sparse ULA. On the other hand, the sensor at -5 is located at a distance of 5 from the opposite end of the 3-sparse ULA and creates differences $\{5, 8, 11, \dots, 3N - 4\}$ with the sensors of the 3-sparse ULA. When combined with the self-differences $\{0, 3, 6, \dots, 3(N - 3)\}$ created by the sensor pairs within the 3-sparse ULA, we get that the coarray of \mathbf{z}_4 is

$$\mathbb{D}_{\mathbf{z}_4}^+ = \{0, 3, 4, 5, \dots, 3(N - 3), 3N - 8, 3N - 7, 3N - 5, 3N - 4, 3N\}. \quad (4.16)$$

The largest difference of $3N$ is between the two augmented sensors.

This array with an aperture of length $3N$ has $w(1) = w(2) = 0$. Its coarray contains a ULA segment starting from $L_1 = 3$ and ending at $L_2 = 3N - 7$. It can identify up to $\lfloor (3N - 9)/2 \rfloor$ DOAs using coarray root-MUSIC. In addition to the holes at 1, and 2, the coarray has four additional holes located at $3N - 6$, $3N - 3$, $3N - 2$, and $3N - 1$.

4.3.4 Further Reducing $w(3)$ and $w(4)$

Although \mathbf{z}_4 has $w(1) = w(2) = 0$, its $w(3)$ increases as N increases. To make $w(3)$ constant we consider the following array:

$$\mathbf{z}_5 = \left[\begin{array}{ccccccccc} -5 & 0 & 4 & 8 & \dots & 4(N - 4) & 4N - 13 & 4N - 10 \end{array} \right]. \quad (4.17)$$

4-sparse ULA

Array \mathbf{z}_5 consists of a 4-sparse ULA of $N - 3$ sensors and three additional sensors placed at positions -5 , $4N - 13$, and $4N - 10$. The 4-sparse ULA creates self-differences $4 \llbracket 0, N - 4 \rrbracket$ in the coarray. The sensor at location -5 is at a distance 5 from the 4-sparse ULA segment, and creates differences $4 \llbracket 1, N - 3 \rrbracket + 1$. The sensor at location $4N - 13$ is at distance 3 from the 4-sparse ULA segment, and creates differences $4 \llbracket 1, N - 3 \rrbracket - 1$. The sensor at location $4N - 10$ is at a distance 6 from the 4-sparse ULA segment, and creates differences $4 \llbracket 1, N - 3 \rrbracket + 2$. Combined with the additional differences $4N - 8$ and $4N - 5$ between the augmented sensors, we get that the coarray of \mathbf{z}_5 is $\mathbb{D}_{\mathbf{z}_5}^+ = \{0, 3, 4, 5, \dots, 4N - 13, 4N - 11, 4N - 10, 4N - 8, 4N - 5\}$. The ULA segment in the coarray is from $L_1 = 3$ to $L_2 = 4N - 13$. The coarray has four holes in addition to the holes at positions 1 and 2.

The first three weights $w(l)$, $1 \leq l \leq 3$ of \mathbf{z}_5 do not increase with N . We can go a step further to also make $w(4)$ a constant by considering the following array:

$$\mathbf{z}_6 = \left[\begin{array}{ccccccccc} -7 & -4 & 0 & 5 & \dots & 5(N - 5) & 5N - 22 & 5N - 19 \end{array} \right]. \quad (4.18)$$

5-sparse ULA

Array \mathbf{z}_6 consists of a 5-sparse ULA of $N - 4$ sensors and four augmented sensors placed at positions -7 , -4 , $5N - 22$, and $5N - 19$. The 5-sparse ULA creates self-differences $5 \llbracket 0, N - 5 \rrbracket$. The sensor at location $5N - 19$ creates differences $5 \llbracket 1, N - 4 \rrbracket + 1$, the sensor at location -7 creates differences $5 \llbracket 1, N - 4 \rrbracket + 2$, the sensor at location $5N - 22$ creates differences $5 \llbracket 1, N - 4 \rrbracket - 2$, and the sensor at location -4 creates differences $5 \llbracket 1, N - 4 \rrbracket - 1$ with the sensors from the 5-sparse ULA segment. Combined with the additional differences $5N - 12$ and $5N - 15$ between the augmented sensors, we get that the coarray of \mathbf{z}_6 is

$$\mathbb{D}_{\mathbf{z}_6}^+ = \{0, \underline{3, 4, 5 \dots, 5N - 21}, 5N - 19, 5N - 18, 5N - 15, 5N - 12\}. \quad (4.19)$$

The ULA segment in the coarray (underlined) is from $L_1 = 3$ to $L_2 = 5N - 21$. The array has $w(1) = w(2) = 0$, $w(3) = 3$, and $w(4) = w(6) = w(7) = w(8) = w(9) = 1$, and $w(5) = N - 5$. Out of the first nine coarray weights $w(l)$, $1 \leq l \leq 9$, only $w(5)$ increases with the number of sensors N . There are five holes in addition to the holes at positions 1 and 2 in the coarray.

4.4 Proposed Weight-Constrained Nested Arrays with $O(N^2)$ Aperture

In this section, we develop sparse arrays with $O(N^2)$ aperture that have either $w(1) = 0$ or $w(1) = w(2) = 0$. For arrays in the previous section, we expanded a ULA and added a few extra sensors. For $O(N^2)$ aperture arrays, we will start by expanding a nested array. However, we will need to add several extra sensors to ensure that the coarray has a ULA segment from lag 2 or 3 onwards. These arrays will be called weight-constrained nested arrays (WCNAs). These arrays are well-suited for identifying a large number of DOAs under high mutual coupling, as we will see in Sec. 4.5. However, they have large apertures, and so they are not suitable when there is an aperture constraint. Coarray properties of the arrays described in this section are summarized in Table 4.3.

4.4.1 Weight-Constrained Nested Arrays with $w(1) = 0$

Consider a standard nested array [34] $\mathbf{z}_n = \mathbb{P}_1 \cup \mathbb{P}_2$ where, $\mathbb{P}_1 = \llbracket 1, N_1 \rrbracket$ and $\mathbb{P}_2 = (N_1 + 1) \llbracket 1, N_2 \rrbracket$. We assume that $N_1 > 3$, so that the general expressions are valid. The coarray of the nested array is $\mathbb{D}_{\mathbf{z}_n}^+ = \llbracket 0, (N_1 + 1)N_2 - 1 \rrbracket$. Now, consider the array

$$\mathbf{w}_1^{(1)} = (2\mathbb{P}_1) \cup (2\mathbb{P}_2) \cup \mathbb{P}_3, \quad (4.20)$$

Table 4.3: Coarray properties of the proposed weight-constrained nested arrays with $\mathcal{O}(N^2)$ aperture proposed in Sec. 4.4. For a given number of sensors N , the optimal values of N_1 and N_2 for nested array are $N_1 = \text{round}((N+1)/2)$, and $N_2 = N - \text{round}((N+1)/2)$ [34]. We assume $N_1 \geq 5$. The optimal values of N_1 and N_2 for $\mathbf{w}_1^{(1)}$ and $\mathbf{w}_1^{(2)}$ are given by Eq. (4.22), and that for \mathbf{w}_2 are given by Eq. (4.26). The maximum number of identifiable DOAs with coarray-MUSIC is $\lfloor L/2 \rfloor$.

Array	Aperture A	Coarray Weights					ULA Segment in Coarray		
		$w(1)$	$w(2)$	$w(3)$	$w(4)$	$w(5)$	Start (L_1)	End (L_2)	Length (L)
Nested	$(N_1 + 1)N_2 - 1$	N_1	$N_1 - 1$	$N_1 - 2$	$N_1 - 3$	$N_1 - 4$	$-A$	A	$2(N_1 + 1)N_2 - 1$
$\mathbf{w}_1^{(1)}$	$2(N_1 + 1)N_2 + 1$	0	N_1	N_2	$N_1 - 1$	1	2	$2(N_1 + 1)N_2 - 1$	$2(N_1 + 1)N_2 - 2$
$\mathbf{w}_1^{(2)}$	$2(N_1 + 1)(2N_2 - 1) + 1$	0	N_1	1	$N_1 - 1$	1	2	$2(N_1 + 1)N_2 - 1$	$2(N_1 + 1)N_2 - 2$
\mathbf{w}_2	$3(N_1 + 1)(2N_2 - 1) + 6$	0	0	N_1	1	N_2	3	$3(N_1 + 1)N_2 - 1$	$3(N_1 + 1)N_2 - 3$

where $\mathbb{P}_3 = 2\mathbb{P}_2 + 3$. This array $\mathbf{w}_1^{(1)}$ is formed by first expanding a nested array to create a ‘2-sparse’ nested array $(2\mathbb{P}_1) \cup (2\mathbb{P}_2)$ and appending it with another sparse ULA segment \mathbb{P}_3 containing N_2 sensors. \mathbb{P}_3 is just a shifted version of $2\mathbb{P}_2$ to the right by three units. Note that the coarray of the dilated nested array $(2\mathbb{P}_1) \cup (2\mathbb{P}_2)$ only contains even lags from $-2(N_1 + 1)N_2 + 2$ to $2(N_1 + 1)N_2 - 2$. Thus, we append \mathbb{P}_3 to fill up the missing odd lags (except lag 1, to ensure $w(1) = 0$). The array has a total of $N_1 + 2N_2$ sensors, and its aperture $A = 2(N_1 + 1)N_2 + 1$ is slightly larger than twice that of the original nested array \mathbf{z}_n . We will now prove that the coarray of $\mathbf{w}_1^{(1)}$ has a ULA segment from $L_1 = 2$ to $L_2 = 2(N_1 + 1)N_2 - 1$.

Lemma 4. The difference coarray $\mathbb{D}_{\mathbf{w}_1^{(1)}}^+$ of $\mathbf{w}_1^{(1)}$ contains a ULA segment from $L_1 = 2$ to $L_2 = 2(N_1 + 1)N_2 - 1$.

Proof. As the difference coarray of the nested array is $\mathbb{D}_{\mathbf{z}_n}^+ = [[0, (N_1 + 1)N_2 - 1]]$, it is easy to see that all even differences from 0 to $2(N_1 + 1)N_2 - 2$ are created by the sensors from $(2\mathbb{P}_1) \cup (2\mathbb{P}_2)$. Now consider the first sensor from \mathbb{P}_3 located at $2(N_1 + 1) + 3$. It creates odd differences $\{3, 5, \dots, 2(N_1 + 1) + 1\} = 2[[1, (N_1 + 1)]] + 1$ with the sensors from $2\mathbb{P}_1 \cup \{2(N_1 + 1)\}$. Similarly, the k -th sensor in \mathbb{P}_3 located at $2(N_1 + 1)k + 3$ creates differences of the form $2[[1, (N_1 + 1)]] + 2(N_1 + 1)(k - 1) + 1$ with the sensor from $2\mathbb{P}_1 \cup \{2(N_1 + 1)\}$. Putting together these differences for all sensors in \mathbb{P}_3 (i.e., for $k \in [[1, N_2]]$), we get that \mathbb{P}_3 generates all odd differences from $2[[1, (N_1 + 1)N_2]] + 1$. Furthermore, when $N_1 > 1$, there are no sensor pairs with unit separation, and thus $w(1) = w(A - 1) = 0$. Thus, we have proved that the difference coarray of $\mathbf{w}_1^{(1)}$ is

$$\mathbb{D}_{\mathbf{w}_1^{(1)}}^+ = \{0, \underline{2, 3, 4, \dots, 2(N_1 + 1)N_2 - 1}, 2(N_1 + 1)N_2 + 1\}, \quad (4.21)$$

which contains a ULA segment from $L_1 = 2$ to $L_2 = 2(N_1 + 1)N_2 - 1$. \square

The array has $w(1) = 0$, and the difference coarray $\mathbb{D}_{\mathbf{w}_1^{(1)}}$ has only two holes, one at location 1 and the other at location $A - 1$. The length of the ULA segment in the coarray is $L = L_2 - L_1 + 1 = 2(N_1 + 1)N_2 - 2$, and thus the number of identifiable DOAs is $\lfloor L/2 \rfloor = (N_1 + 1)N_2 - 1$, which is the same as the number of identifiable DOAs with the original nested array \mathbf{z}_n . However $\mathbf{w}_1^{(1)}$ uses N_2 additional sensors compared to \mathbf{z}_n . Furthermore, when $N_1 > 3$, it can be verified that $w(2) = N_1$, $w(3) = N_2$, $w(4) = N_1 - 1$, and $w(5) = 1$.

For a total number of sensors N , optimal N_1 and N_2 to maximize the ULA segment in the coarray of $\mathbf{w}_1^{(1)}$ are found by solving

$$\begin{aligned} \max_{N_1, N_2} \quad & 2(N_1 + 1)N_2 - 2 \\ \text{s.t.} \quad & N_1 + 2N_2 = N. \end{aligned} \quad (4.22)$$

The optimal solution to the above problem is given by $N_2 = \text{round}((N + 1)/4)$, and $N_1 = N - 2N_2$. With the optimal values of N_1 and N_2 , the maximum number of identifiable DOAs is $O(N^2)$ (approximately $(N + 1)^2/8 - 2$). Thus, the array $\mathbf{w}_1^{(1)}$ can identify $O(N^2)$ DOAs using coarray-MUSIC.

Note that \mathbb{P}_3 is a $2(N_1 + 1)$ -sparse ULA of N_2 sensors placed at a distance three to the right of the sensors at location $2(N_1 + 1)$. It was introduced to fill up the odd-valued differences missing from the coarray of the expanded nested array. These odd valued differences are created between the sensors from \mathbb{P}_3 and the sensors from the $(N_1 + 1)$ -sensor 2-sparse ULA segment given by $2\mathbb{P}_1 \cup \{2(N_1 + 1)\}$. Noting this, we can consider the following variation of the array

$$\mathbf{w}_1^{(2)} = (2\mathbb{P}_1) \cup (2\mathbb{P}_2) \cup \mathbb{P}'_3, \quad (4.23)$$

where $\mathbb{P}'_3 = 2\mathbb{P}_2 - 2N_2(N_1 + 1) - 1$. Here, the $2(N_1 + 1)$ -sparse ULA segment of N_2 sensors \mathbb{P}'_3 is placed three units to the left of the sensor at location 2, instead of placing it three units to the right of the sensor at $2(N_1 + 1)$. Thus, it creates the same set of odd-valued differences with the 2-sparse ULA segment $2\mathbb{P}_1 \cup \{2(N_1 + 1)\}$. Hence, it can be shown similar to Lemma 4 that array $\mathbf{w}_1^{(2)}$ also has the same ULA segment in its coarray from $L_1 = 2$ to $L_2 = 2(N_1 + 1)N_2 - 1$. However, the aperture of $\mathbf{w}_1^{(2)}$ is roughly twice that of $\mathbf{w}_1^{(1)}$, and its coarray has many holes, unlike that of $\mathbf{w}_1^{(1)}$ which had only two holes. By shifting the \mathbb{P}_3 to the other side of the nested array, $w(3)$ reduces from N_2 to 1. Thus, array $\mathbf{w}_1^{(2)}$ has reduced mutual coupling, at the cost of a larger aperture compared to $\mathbf{w}_1^{(1)}$.

4.4.2 Weight-Constrained Nested arrays with $w(1) = w(2) = 0$

To create a sparse array with $O(N^2)$ degrees of freedom and $w(1) = w(2) = 0$, we expand a nested array by a factor of 3 and augment it with two additional appropriately placed sparse ULA segments. Consider the array

$$\mathbf{w}_2 = (3\mathbb{P}_1) \cup (3\mathbb{P}_2) \cup \mathbb{P}_4 \cup \mathbb{P}_5, \quad (4.24)$$

where $\mathbb{P}_1 = [[1, N_1]]$, $\mathbb{P}_2 = (N_1 + 1) [[1, N_2]]$, and

$$\mathbb{P}_4 = 3\mathbb{P}_2 - 3N_2(N_1 + 1) - 1, \quad \mathbb{P}_5 = 3\mathbb{P}_2 + 5. \quad (4.25)$$

Here \mathbb{P}_4 and \mathbb{P}_5 are $3(N_1 + 1)$ -sparse ULA segments of N_2 sensors placed at distances of 4 and 5 units, respectively, from the opposite ends of the 3-sparse ULA segment $3\mathbb{P}_1 \cup \{3(N_1 + 1)N_2\}$. Array \mathbf{w}_2 has a total of $N_1 + 3N_2$ sensors, and its aperture $A = 3(N_1 + 1)(2N_2 - 1) + 6$, which is roughly six times that of the original nested array \mathbf{z}_n . Similar to Lemma 4, it can be shown that \mathbf{w}_2 has a ULA segment in its coarray from $L_1 = 3$ to $L_2 = 3(N_1 + 1)N_2 - 1$.

Lemma 5. The difference coarray $\mathbb{D}_{\mathbf{w}_2}^+$ of \mathbf{w}_2 contains a ULA segment from $L_1 = 3$ to $L_2 = 3(N_1 + 1)N_2 - 1$.

The proof is similar to that of Lemma 4, and details are omitted due to space constraints. The main idea is that the dilated nested array $(3\mathbb{P}_1) \cup (3\mathbb{P}_2)$ creates self-differences $3 \llbracket [0, (N_1 + 1)N_2 - 1] \rrbracket$. The sensors from \mathbb{P}_4 create differences $3 \llbracket [1, (N_1 + 1)N_2 - 1] \rrbracket + 1$ with the sensors from $3\mathbb{P}_1 \cup \{3(N_1 + 1)N_2\}$. The sensors from \mathbb{P}_5 create differences $3 \llbracket [1, (N_1 + 1)N_2 - 1] \rrbracket + 2$ with the sensors from $3\mathbb{P}_1 \cup \{3(N_1 + 1)N_2\}$. Combining the three sets of differences, a ULA segment from $L_1 = 3$ to $L_2 = 3(N_1 + 1)N_2 - 1$ is obtained.

Array \mathbf{w}_2 has $w(1) = w(2) = 0$. However, it has a large aperture and contains several holes in its difference coarray. The length of the one-sided ULA segment in its coarray is $L = 3(N_1 + 1)N_2 - 3$, and thus it can identify up to $\lfloor L/2 \rfloor \approx 1.5(N_1 + 1)N_2 - 2$ DOAs with coarray-MUSIC. However \mathbf{w}_2 uses $2N_2$ additional sensors compared to \mathbf{z}_n . It can be verified that when $N_1 > 3$, $w(3) = N_1$, $w(4) = 1$, and $w(5) = N_2$.

Given N sensors, the optimal N_1 and N_2 to maximize the ULA segment in the coarray of \mathbf{w}_2 are found by solving

$$\begin{aligned} \max_{N_1, N_2} \quad & 3(N_1 + 1)N_2 - 3 \\ \text{s.t.} \quad & N_1 + 3N_2 = N. \end{aligned} \tag{4.26}$$

The optimal solution to the above problem is given by $N_2 = \text{round}((N + 1)/6)$, and $N_1 = N - 3N_2$. With the optimal values of N_1 and N_2 , the maximum number of identifiable DOAs is $\mathcal{O}(N^2)$ (approximately $(N + 1)^2/8 - 2$). Thus, the array \mathbf{w}_2 has $w(1) = w(2) = 0$ and can identify $\mathcal{O}(N^2)$ DOAs using coarray-MUSIC.

4.5 Simulation Results

In this section, we compare our proposed weight-constrained arrays (with $\mathcal{O}(N)$ aperture array from Sec. 4.3 as well as $\mathcal{O}(N^2)$ aperture arrays from Sec. 4.4) with

other sparse arrays from the literature. Out of several generalizations of nested arrays [34], we choose the following four in our comparisons. Super nested arrays (SNA) [39], [101] have the same length of difference coarray as the parent nested arrays, but are more robust to mutual coupling because of reduced coarray weights at small lags. Therefore, we use SNA for comparison instead of nested arrays. We use a third-order super nested array (SNA3) if available, and if not, a second-order super nested array (SNA2) is used. Augmented nested arrays (ANA) [103] are formed by rearranging and distributing the sensors from the dense subarray of the nested array on the two sides of the sparse subarray of the nested array. Out of the four variations of ANAs, we use ANAII-2, which has the largest DOFs and small mutual coupling compared to the other variations. Recently proposed variations of nested arrays called dilated nested arrays (DNA) and displaced DNA (DDNA) [92] are also used in comparisons. All these modifications of the nested array require two parameters, N_1 and N_2 , for their construction. For a given total number of sensors, optimal values of N_1 and N_2 are used for each array. One key drawback of all these variations of nested arrays is that $w(2)$ of these arrays increases linearly with N_1 (and optimal N_1 is proportional to N).

We also include MRAs [33] in our comparison, but MRAs are available only for a limited number of sensors [33], [100]. In [84], the authors performed a restricted search for MRAs for $N \leq 20$ and tabulated the (approximate) MRAs they found using their restricted search. They also noted a general pattern of sensor locations in the arrays they obtained. The maximum inter-element spacing constraint (MISC) arrays [104] seem to be motivated by this general closed-form pattern noticed in [84]. In addition, the authors of [84] compiled another set of arrays, which we call constrained-MRAs (cMRA). These arrays maximize the ULA segment in the difference coarray under the constraint that $w(1) = 0$. The tabulated cMRAs in [84] are only an approximation to the ideal cMRAs because of the restricted search performed. As these arrays have $w(1) = 0$, they can effectively mitigate mutual coupling, as we will see in the simulations.

Out of several generalizations of coprime arrays (CPA) [35], we choose the following four in our comparisons. CADiS array [83] is generated by displacing and compressing one of the sparse ULAs of a coprime array. These arrays can have $w(l) = 0$ for several small values of l and hence can effectively mitigate mutual coupling, similar to the proposed arrays. However, as noted in Sec. 4.1.2, the ULA segment in the difference coarray of CADiS does not start immediately after the

few initial holes, unlike the proposed arrays, and hence the maximum number of identifiable DOAs can be smaller than our proposed arrays, as well will see. Furthermore, CADiS requires 3 design parameters (M , N , and p from [83] such that M and N are coprime and $p > 1$ divides M). For a given total number of sensors, or a fixed aperture, it is not easy to come up with optimal parameters that maximize the ULA segment in the coarray of CADiS, without enumerating all possibilities of design parameters (M , N , and p from [83]). Furthermore, for a certain number of sensors (such as 17, 23, and 29), there are no possible CADiS array constructions [91]. Another modification of coprime arrays we use for comparison is the thinned coprime array (TCA) [91]. It has the same aperture and DOFs as the original coprime array but with fewer sensors and smaller coarray weights than the original coprime array. Four variations of padded coprime arrays (PCA) are proposed in [105] by characterizing hole locations in a tailored CADiS array and filling up those holes by augmenting a small number of additional sensors. Out of these, extended PCA (ePCA) has the longest ULA segment in the coarray. Impressively, ePCA (with parameters M and N from [105]) can achieve $w(l) = 1, 1 \leq l \leq M - 1$ when M is odd. In some simulations, we also consider the enhanced and generalized coprime array (EGCA) [157], which can have a larger ULA segment in coarray and smaller mutual coupling than other variations of coprime arrays.

In Section 4.5.1, we compare the coarray properties for various arrays and illustrate coarray-MUSIC spectra obtained when estimating $D > N$ DOAs. We then conduct two types of Monte-Carlo simulations. The first set of simulations compares different arrays when the number of sensors N is fixed, and the array apertures can differ. The second set of simulations involves a fixed array aperture, while the number of sensors in the arrays can vary. Each data point in the graphs is averaged over 1000 Monte-Carlo trials in every Monte-Carlo simulation. In these trials, the source amplitudes $\mathbf{s}[k]$ and noise realizations $\mathbf{n}[k]$ are randomly chosen according to the signal model described in Chapter 2. All sources are assumed to have equal powers. To estimate DOAs with sparse arrays, we use the coarray root-MUSIC algorithm, as described in Sec. 4.2, using the largest ULA segment (possibly one-sided) in the difference coarray. As for the ULA, we apply direct root-MUSIC (i.e., using $\hat{\mathbf{R}}_{\mathbf{xx}}$), as it usually performs better than coarray root-MUSIC [87], [158]. The DOA estimation mean squared error (MSE) is calculated in the ω domain as

$$\frac{1}{1000D} \sum_{q=1}^{1000} \sum_{i=1}^D (\omega_i - \hat{\omega}_{iq})^2, \quad (4.27)$$

where $\{\hat{\omega}_{iq}\}_{i=1}^D$ are the estimated DOA in the q -th Monte-Carlo trial. The mutual coupling model described in Eq. (2.27) is used with $B = 10$, $c_1 = ce^{j\pi/3}$, and $c_l = (c_1/l)e^{-j(l-1)\pi/8}$, where $|c_1| = c \in [0, 1]$, represents the coupling strength. These values for coupling coefficients have been used in most of the papers on sparse arrays [39], [91], [92], [105], and we use the same values in all our simulations here for consistency. We provide some simulations with different coupling coefficients later in Sec. 4.5.5.

4.5.1 Array Properties and Coarray-MUSIC Spectra

We now compare the coarray properties of the proposed arrays with the well-known arrays discussed in the previous subsection for a fixed number of sensors, $N = 16$. For nested array variations, array designs with optimal N_1 and N_2 that maximize the number of identifiable DOAs are chosen. Similarly, an optimal pair of coprime integers that maximizes the number of identifiable DOAs is chosen for the coprime array. The same design parameters are chosen for ePCA. Out of the possible TCA configurations, we choose the array that can identify the most number of DOAs. Out of possible CADiS configurations, we choose the one with $w(l) = 0$ for $l = 1, 2$ for a fair comparison with the proposed arrays. Fig. 4.1 plots the coarray weights $w(l)$ for $0 \leq l \leq 40$. It is observed that although the proposed arrays have some large weights, the weights that contribute the most to the mutual coupling effect are either zero or relatively small. With fixed $N = 16$, each array has a different aperture A and the maximum number of identifiable DOAs using coarray root-MUSIC (D_m), as shown in Table 4.4. The proposed $O(N)$ aperture arrays have a relatively smaller aperture and can identify fewer DOAs than $O(N^2)$ aperture arrays. The coupling leakage values (\mathcal{L}) in Table 4.4 are calculated with $c = 0.3$ and $B = 10$ (Eq. (2.30)). The coupling leakage is found to be smallest for \mathbf{z}_6 , followed next by CADiS, \mathbf{w}_2 , and \mathbf{z}_5 . Coarray weights $w(l)$ for $1 \leq l \leq 5$ are also shown in the Table. Only the proposed arrays, CADiS, and cMRA have $w(1) = 0$. Although CADiS has a small mutual coupling, it can identify fewer DOAs compared to the proposed $O(N^2)$ aperture arrays $\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2 .

Next, we consider the task of estimating $D = 22$ ($> N = 16$) DOAs uniformly spaced in the ω domain from -0.9π to 0.9π with these arrays. SNR of 0 dB and 100 snapshots are considered. Note that ULA, \mathbf{z}_1 , and $\mathbf{z}_2^{(2)}$, and \mathbf{z}_4 cannot identify 22 DOAs with coarray-MUSIC as they have $D_m < 22$. These arrays will be included in the comparisons in the next subsections for simulations with smaller D . Fig. 4.2 shows the coarray-MUSIC spectra obtained for the rest of the arrays. We observe

Table 4.4: Properties of arrays with $N = 16$. The last 9 arrays (\mathbf{z}_1 to \mathbf{w}_2) are the new proposed arrays. Coupling leakage \mathcal{L} is calculated with $c = 0.3$ and $B = 10$. MSE for estimating 22 DOAs is the smallest for arrays with $w(1) = 0$ (see Sec. 4.5.1).

Array	A	\mathcal{L}	D_m	Coarray weights					MSE
				w(1)	w(2)	w(3)	w(4)	w(5)	
SNA2	71	0.2139	71	2	5	4	1	6	0.0174
ANAI-2	84	0.2071	84	2	5	2	3	4	0.0285
EGCA	88	0.1447	65	1	6	1	4	1	0.0772
DNA	77	0.1843	71	1	6	1	4	1	0.0716
DDNA	86	0.1738	72	1	5	1	3	1	0.0748
MISC	83	0.1865	83	1	6	1	4	2	0.093
MRA	90	0.2299	90	4	2	1	1	3	0.0315
cMRA	80	0.1489	38	0	5	2	4	3	0.0001
CPA	63	0.1997	39	2	2	2	9	2	0.0172
CADiS	88	0.1113	30	0	0	7	0	0	0.0001
TCA	75	0.1515	47	1	1	1	2	10	0.0658
ePCA	85	0.1623	65	1	2	1	8	1	0.0369
ULA	15	0.4443	15	15	14	13	12	11	-
\mathbf{z}_1	31	0.233	14	0	14	1	13	1	-
$\mathbf{z}_2^{(2)}$	45	0.1592	20	0	1	13	1	1	-
$\mathbf{z}_3^{(2)}$	56	0.1290	25	0	1	2	12	1	0.0002
\mathbf{z}_4	48	0.1506	19	0	0	13	1	1	-
\mathbf{z}_5	59	0.118	24	0	0	2	12	1	0.0005
\mathbf{z}_6	68	0.1061	28	0	0	3	1	11	0.0003
$\mathbf{w}_1^{(1)}$	73	0.1875	35	0	8	4	7	1	0.0003
$\mathbf{w}_1^{(2)}$	127	0.1779	35	0	8	1	7	1	0.0001
\mathbf{w}_2	126	0.117	34	0	0	7	1	3	0.0002

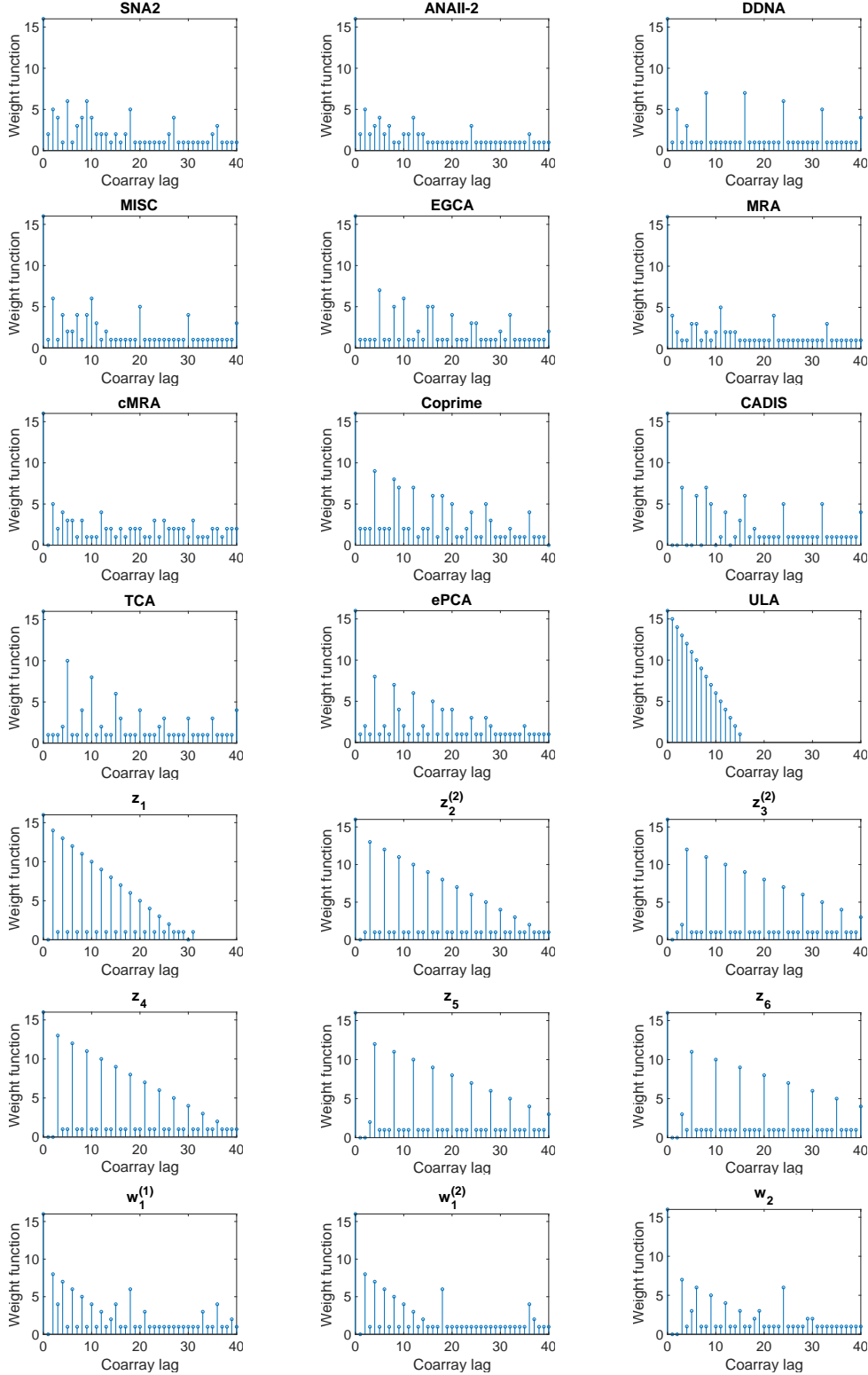


Figure 4.1: Coarray weights $w(l)$, $0 \leq l \leq 40$ for the arrays under consideration. All arrays have $N = 16$. Out of the proposed arrays shown here, \mathbf{z}_1 , $\mathbf{z}_2^{(2)}$, $\mathbf{z}_3^{(2)}$, $\mathbf{w}_1^{(1)}$, and $\mathbf{w}_1^{(2)}$ have $w(1) = 0$, and \mathbf{z}_4 , \mathbf{z}_5 , \mathbf{z}_6 , and \mathbf{w}_2 have $w(1) = w(2) = 0$. The only other arrays that have $w(1) = 0$ are CADiS [83] and (approximate) constrained MRAs (cMRA) [84]. See Sec. 4.5.1 for details.

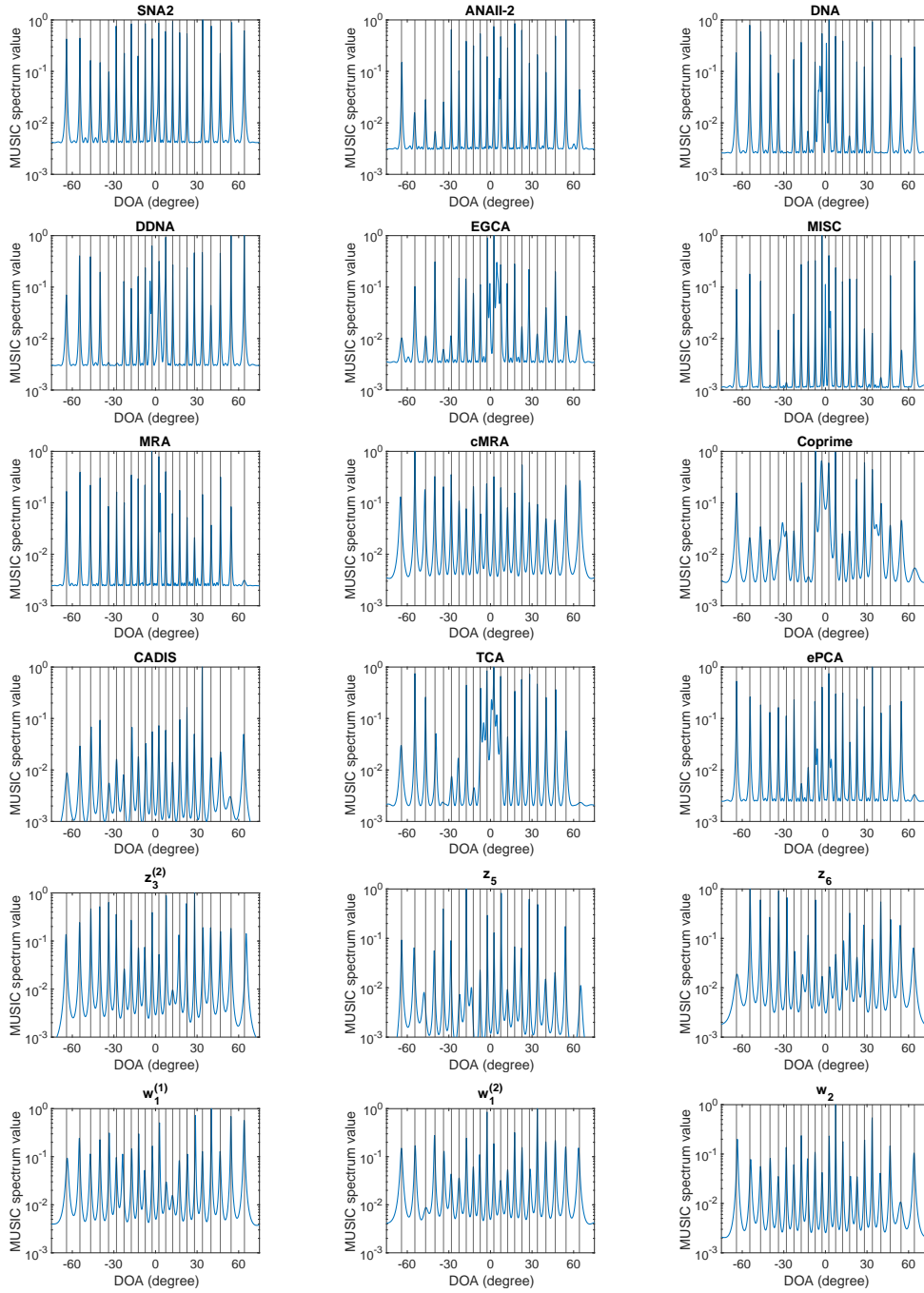


Figure 4.2: Coarray-MUSIC spectra obtained when identifying $D = 22$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with different 16-sensor arrays from Table 4.4. SNR of 0 dB, mutual coupling with $c = 0.3$ and $B = 10$, and 100 snapshots are considered. Black vertical lines correspond to the true DOA locations. Except for CADiS and cMRA, all other arrays from the literature do not correctly identify all 22 DOAs and miss out on at least one DOA. On the other hand, the proposed arrays are able to identify all DOAs correctly with a small MSE by producing coarray-MUSIC spectra having peak locations close to each true DOA.

that arrays that do not have $w(1) = 0$ do not produce correct peaks in the coarray-MUSIC spectra for some DOAs. On the other hand, the proposed arrays, CADiS, and cMRA can effectively mitigate the effect of mutual coupling and produce peaks in the coarray-MUSIC spectra corresponding to all 22 DOAs. The DOA estimation MSE is also small for these arrays with $w(1) = 0$, as shown in the last column of Table 4.4.

4.5.2 Monte-Carlo Simulations with a Fixed Number of Sensors

With the same set of arrays with $N = 16$ sensors, we perform Monte-Carlo simulations to compare DOA estimation performance. First, for the simulations in Fig. 4.3, we consider $D = 6$ DOAs spaced uniformly in the ω domain from -0.8π to 0.8π . We consider SNR of 5 dB, 500 snapshots, and coupling strength $c = 0.5$. Out of these three, we fix two quantities at a time and vary the third quantity to measure its impact on DOA estimation error. We see that in this situation, with a relatively small number of sensors and few DOAs, many of the proposed arrays perform worse than the other sparse arrays. However, notably, \mathbf{z}_6 performs the best in this situation for a large range of c , SNR, and snapshots. Array \mathbf{z}_6 performs well because it has $w(1) = w(2) = 0$, and small $w(3)$ and $w(4)$, which makes it robust to the presence of high mutual coupling. In particular, in Fig. 4.3(a), we see that the increase in MSE for the proposed arrays is gradual as the coupling strength c is increased. In contrast, for most of the other arrays, MSE increases drastically as c increases beyond a certain point. The sudden increase in MSE usually indicates some of the DOAs being misidentified (similar to the case in Fig. 4.2). Despite having a smaller aperture and shorter ULA segment in coarray compared to the other $O(N^2)$ aperture arrays, we see from Fig. 4.3(b) and Fig. 4.3(c) that \mathbf{z}_6 performs better when SNR is larger than 0 dB and more than 100 snapshots are used.

Next, we change the number of DOAs to $D = 20$ and repeat the above set of simulations. The DOAs are spaced uniformly from -0.9π to 0.9π in the ω domain. In Fig. 4.4 we see a clear separation between the arrays based on whether $w(1) = 0$ or not. Arrays with $w(1) = 0$ (irrespective of whether the aperture is $O(N)$ or $O(N^2)$) have smaller MSE than arrays for which $w(1) \neq 0$. The MSE for $\mathbf{z}_2^{(2)}$ is large because $D = 20$ DOAs is barely within the maximum number of identifiable DOAs (D_m) with this array (see Table 4.4). From Fig. 4.4(a) note that for $c \leq 0.2$, proposed arrays have larger MSE than most other arrays. For such small mutual coupling, the arrays with larger lengths of the ULA segment in the coarray have smaller MSE. This indicates that for small mutual coupling, making $w(1) = 0$ to reduce mutual

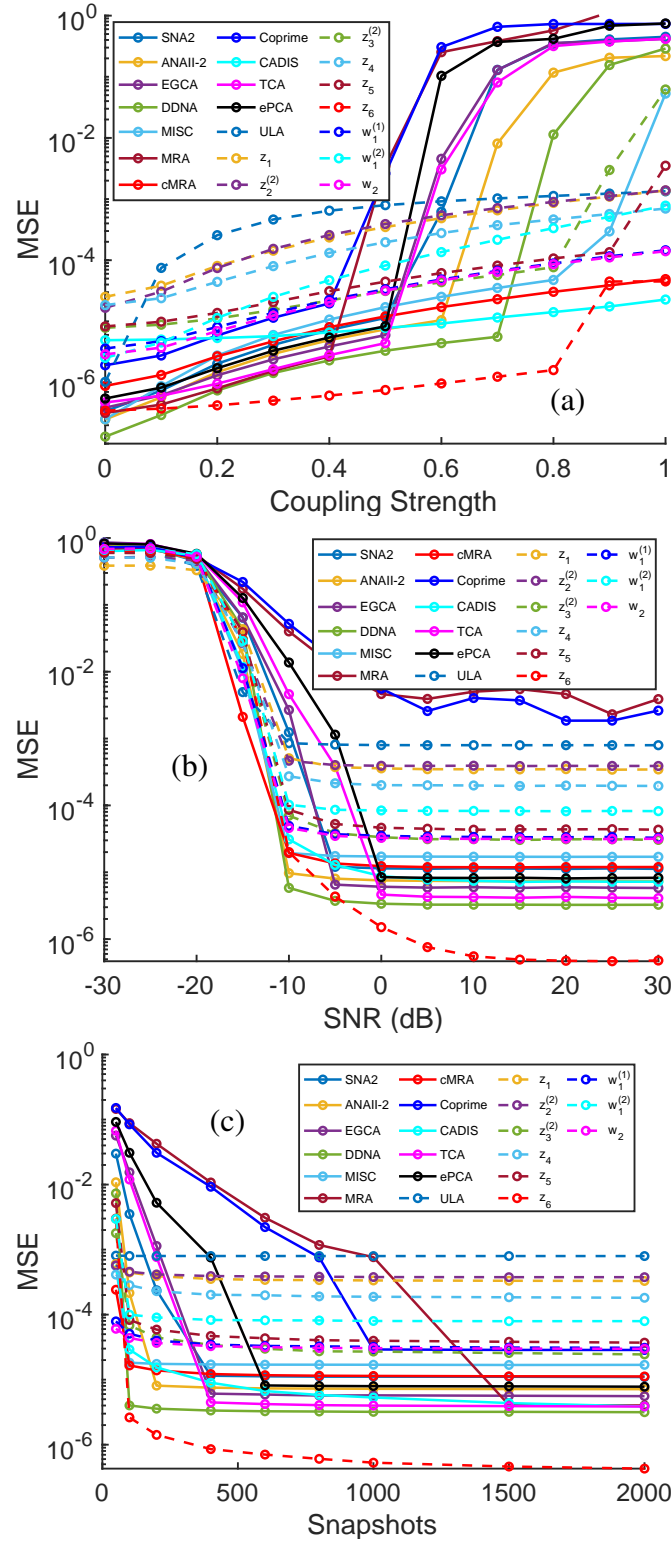


Figure 4.3: MSE when estimating $D = 6$ DOAs uniformly spaced in ω from -0.8π to 0.8π with 16-sensor arrays. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. Although many of the proposed arrays perform worse than other arrays from the literature, \mathbf{z}_6 is found to perform the best in most situations.

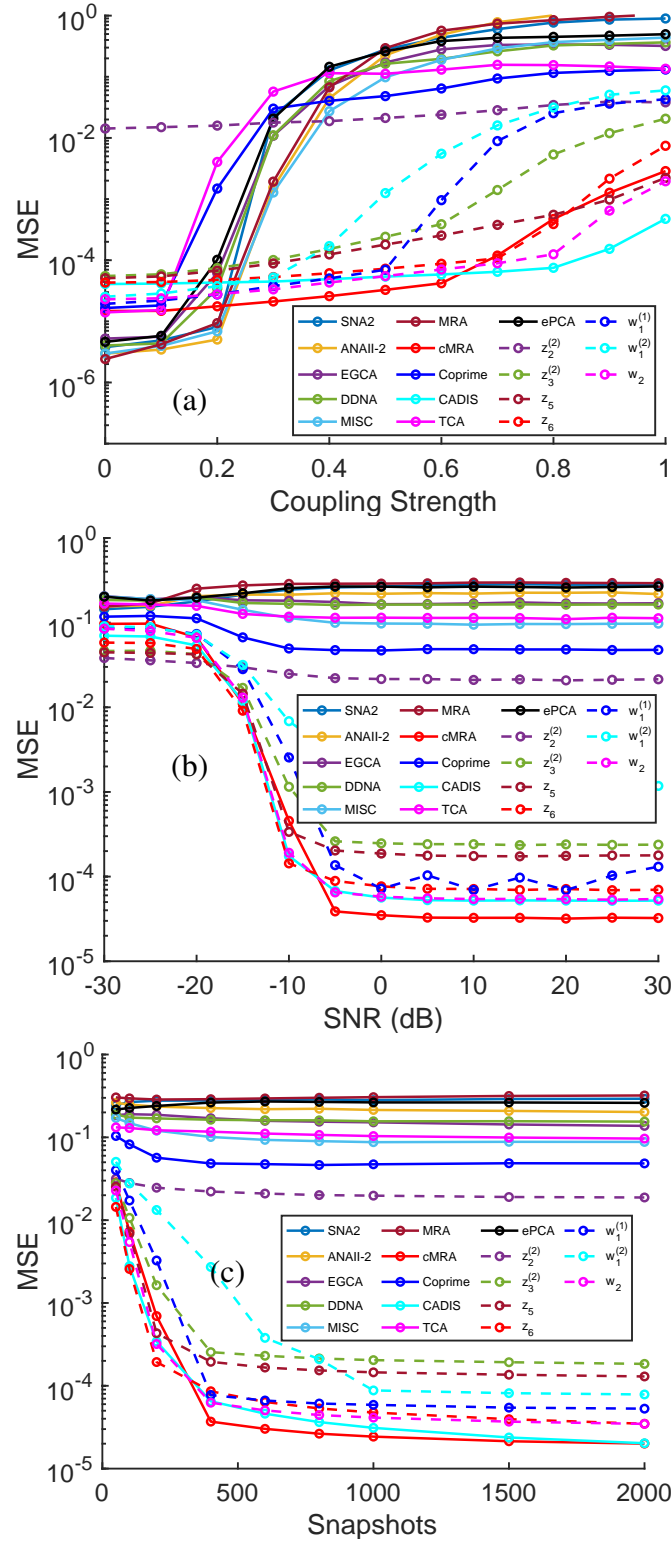


Figure 4.4: MSE when estimating $D = 20$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. Arrays with $w(1) = 0$ have smaller MSE, whereas arrays with $w(1) \neq 0$ cannot identify all the DOAs correctly, resulting in high MSE.

coupling is not as critical as it is when high mutual coupling is present ($c > 0.2$). We see a similar ordering of arrays in Figs. 4.4(b) and 4.4(c). The MSEs saturated to a high value are indicative of misidentified DOAs. This simulation makes it clear that under high mutual coupling and for larger D , $w(1) = 0$ is required to correctly identify the DOAs.

4.5.3 Monte-Carlo Simulations Under Aperture Constraint

As we explained in the introduction, array design may be constrained by the available aperture in practical applications. To compare array performance under such practical constraints, we consider a situation where a maximum aperture of length 100 and a maximum of 50 sensors are available. For each array type under consideration, we appropriately choose the number of sensors such that the array satisfies the aperture constraint of 100. For example, a super nested array has 19 sensors, ULA and \mathbf{z}_1 have 50 sensors each, and \mathbf{z}_3 has 27 sensors. Table 4.5 shows the number of sensors, the maximum number of identifiable DOAs with coarray-MUSIC (D_m), coupling leakage (\mathcal{L}) calculated with $c = 0.3$, $B = 10$, and the first 5 coarray weights. Note that the proposed $O(N)$ aperture arrays can fit more sensors in the available aperture than the $O(N^2)$ aperture arrays from the literature and at the same time, have a smaller coupling leakage than many of those arrays (except CADiS, which has $w(l) = 0$ for $1 \leq l \leq 4$).

In general, arrays with fewer sensors and sparser sensor arrangements can help minimize the impact of mutual coupling. However, a smaller number of sensors also means fewer observations per snapshot, which may result in less accurate DOA estimation. On the other hand, an array with a larger number of sensors provides more spatial samples of the impinging signals. But, under the aperture constraint, the sensors need to be closer to each other on average, which increases the effect of mutual coupling. Therefore, there is an interesting tradeoff in choosing the appropriate number of sensors and array geometry when the array aperture is constrained, and mutual coupling is present.

In Fig. 4.5, we examine the scenario with $D = 6$ DOAs located at -0.8π , -0.35π , -0.1π , 0.05π , 0.4π , and 0.8π in the ω domain. The DOAs are spaced non-uniformly, unlike the other simulation examples. We compare the MSE of DOA estimation for different arrays constructed under the $A \leq 100$ and $N \leq 50$ constraints. We consider an SNR of 5 dB, coupling strength of $c = 0.5$, and 500 snapshots, and vary one of these three quantities at a time to measure its impact on DOA estimation

Table 4.5: Properties of arrays when $A \leq 100$ and $N \leq 50$. Coupling leakage \mathcal{L} is calculated with $c = 0.3$ and $B = 10$. Many of the proposed $O(N)$ aperture arrays have more sensors but smaller \mathcal{L} compared to the $O(N^2)$ aperture arrays from the literature.

Array	N	\mathcal{L}	D_m	Coarray weights				
				w(1)	w(2)	w(3)	w(4)	w(5)
SNA3	19	0.1684	99	1	5	2	4	1
ANAI-2	17	0.2152	86	2	7	2	5	2
DNA	18	0.1909	89	1	8	1	6	1
DDNA	17	0.1692	80	1	5	1	3	1
MISC	17	0.1814	93	1	6	1	4	2
MRA	16	0.2299	90	4	2	1	1	3
cMRA	18	0.1722	48	0	8	2	7	2
Coprime	20	0.1757	59	2	2	2	2	11
CADiS	16	0.0712	25	0	0	0	0	6
TCA	17	0.1436	59	1	1	1	1	10
ePCA	18	0.1414	75	1	1	1	1	10
ULA	50	0.46	49	49	48	47	46	45
\mathbf{z}_1	50	0.2437	48	0	48	1	47	1
$\mathbf{z}_2^{(2)}$	34	0.1611	47	0	1	31	1	1
$\mathbf{z}_3^{(2)}$	27	0.1246	47	0	1	2	23	1
\mathbf{z}_4	33	0.157	45	0	0	30	1	1
\mathbf{z}_5	26	0.1179	44	0	0	2	22	1
\mathbf{z}_6	22	0.1031	43	0	0	3	1	17
$\mathbf{w}_1^{(1)}$	18	0.1801	44	0	8	5	7	1
$\mathbf{w}_1^{(2)}$	14	0.1649	27	0	6	1	5	1
\mathbf{w}_2	14	0.1088	25	0	0	5	1	3

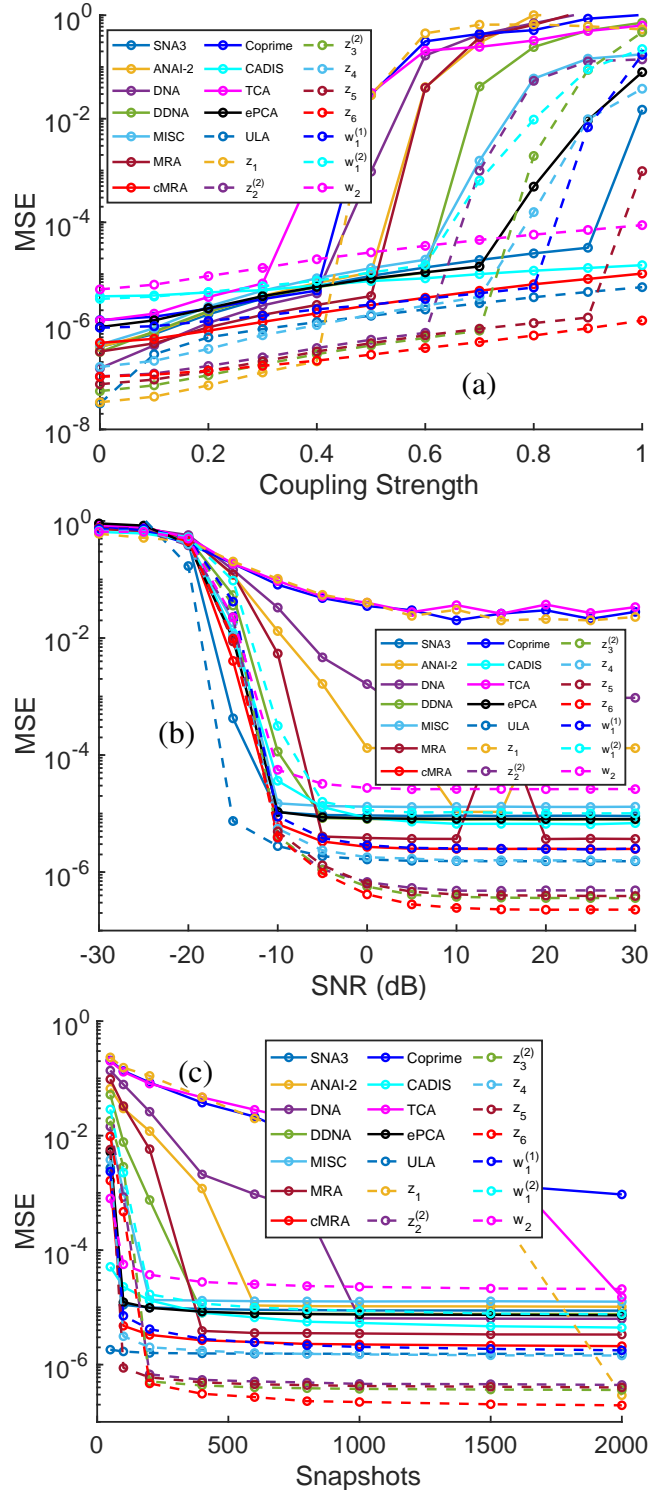


Figure 4.5: MSE when estimating $D = 6$ DOAs spaced non-uniformly in ω domain with different arrays under aperture constraint $A \leq 100$ and $N \leq 50$. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. The proposed $O(N)$ aperture arrays perform much better than other sparse arrays, because under the aperture constraint, they can fit in more sensors while maintaining small coupling leakage.

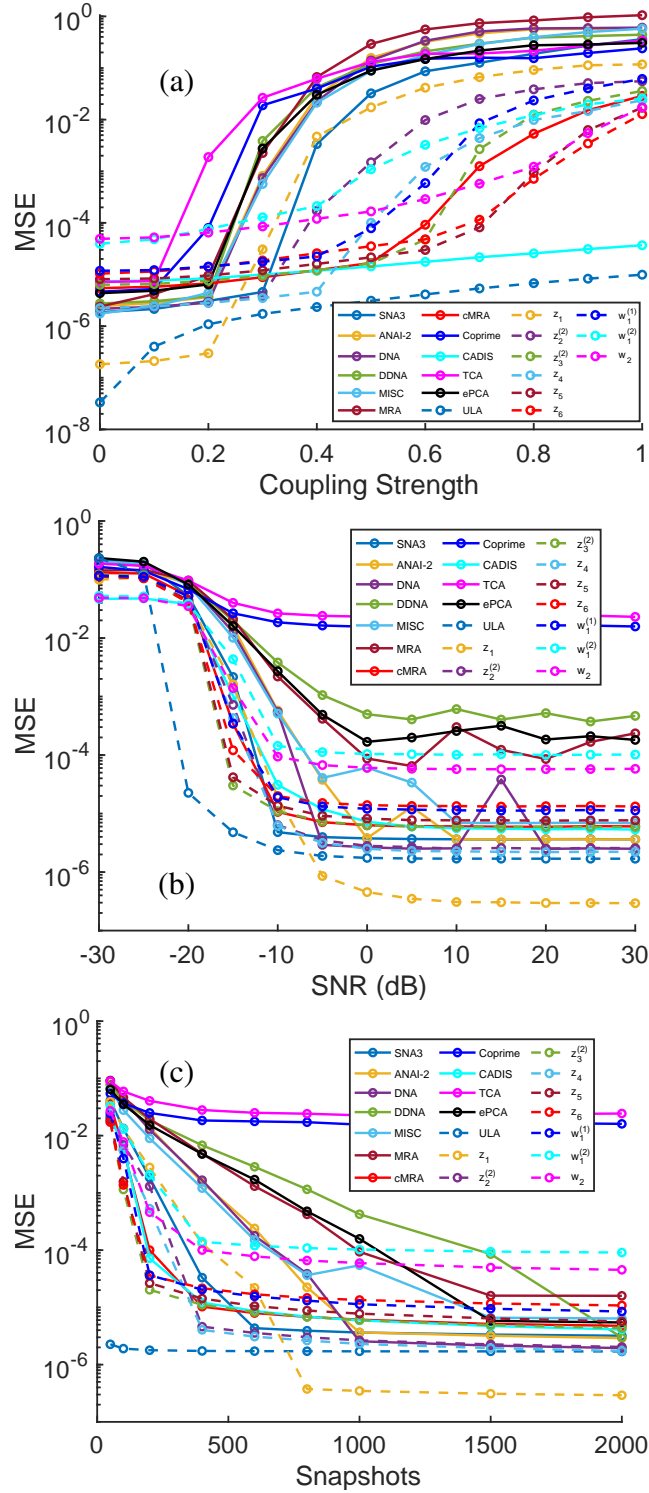


Figure 4.6: MSE when estimating $D = 20$ DOAs uniformly spaced in ω from -0.9π to 0.9π with different arrays under aperture constraint $A \leq 100$ and $N \leq 50$. (a) $\text{SNR} = 5$ dB and $K = 500$, (b) $c = 0.3$ and $K = 1000$, (c) $c = 0.3$ and $\text{SNR} = 5$ dB. Despite high coupling leakage \mathcal{L} , ULA performs the best under low SNR or a small number of snapshots. The proposed array z_1 performs well at higher SNRs or when a large number of snapshots are available. CADiS and ULA are found to perform well under very high mutual coupling.

performance. Based on the results, we observe that the proposed $O(N)$ arrays perform significantly better than all other $O(N^2)$ aperture arrays. In particular, from Fig. 4.5(a) it can be observed that $O(N)$ arrays perform better than other arrays even when $c = 0$ (i.e. when no mutual coupling is present). This is because $O(N)$ arrays have more sensors than $O(N^2)$ aperture arrays under aperture constraint. From Figs. 4.5(b) and 4.5(c) it can be observed that for SNR larger than -10 dB and more than 100 snapshots, many of the proposed $O(N)$ aperture sparse arrays perform better than other arrays.

Next, in Fig. 4.6, we consider $D = 20$ DOAs spaced uniformly in the ω domain from -0.9π to 0.9π . We consider the same arrays as in the previous simulation with $A \leq 100$ and $N \leq 50$. In Fig. 4.6(a), when $K = 500$ and SNR is 5 dB, ULA performs the best for most values of c despite its large coupling leakage and smaller aperture compared to the other arrays. A part of the reason for this may be that ULA has a large number of sensors, and it uses direct root-MUSIC (and not coarray root-MUSIC). CADiS is also found to perform well for higher values of c . However, in Fig. 4.6(b), where $c = 0.3$ and $K = 1000$, we see that \mathbf{z}_1 can perform the best when the SNR is -5dB or more. In Fig. 4.6(c), where $c = 0.3$ and SNR = 5 dB, we observe that for more than 800 snapshots \mathbf{z}_1 can perform better than ULA. In Figs. 4.6(b) and 4.6(c), when SNR is less than -10 dB, or when fewer than 800 snapshots are used, ULA is observed to be better than other arrays despite having large coupling leakage.

4.5.4 Additional Simulations with Weight-Constrained Nested Arrays

The proposed weight-constrained nested arrays with $O(N^2)$ aperture have an advantage when there is no aperture constraint, and a large number of DOAs need to be identified under high mutual coupling. This is demonstrated by the simulation in Fig. 4.7. Here, we consider arrays with $N = 25$ sensors. Note that cMRA [84] is not available for $N > 20$. Out of the possible CADiS array configurations for $N = 25$, we choose the array with $w(l) = 0$ for $1 \leq l \leq 4$ to reduce the mutual coupling effect. SNR of 5 dB, coupling strength $c = 0.3$, and 500 snapshots are considered. We vary the number of DOAs D and plot the MSE for different arrays. We can see that the proposed array \mathbf{w}_2 has the smallest MSE when $35 \leq D \leq 80$. Also, arrays $\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2 can identify more DOAs than CADiS.

It is clear that when we want to identify a large number of DOAs D without aperture constraint, WCNA's are better suited. In Fig. 4.8, we drop the WCSAs with $O(N)$

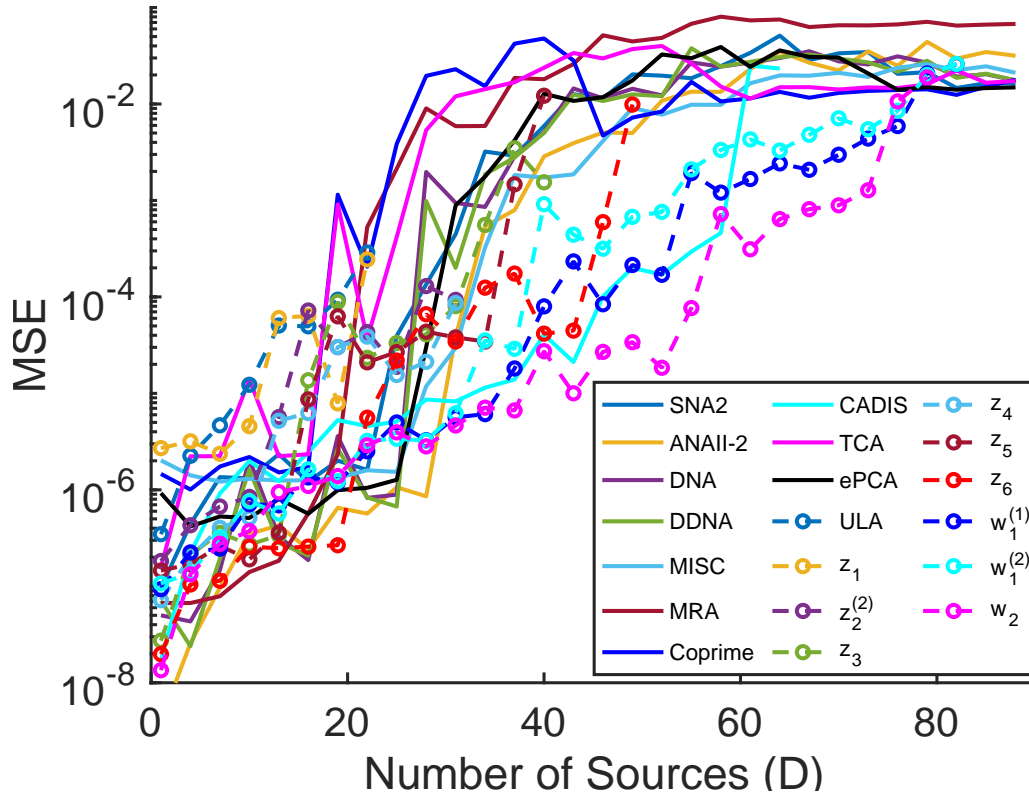


Figure 4.7: MSE when identifying D DOAs spaced uniformly in the ω domain from -0.9π to 0.9π with 25-sensor arrays. SNR = 5 dB, $K = 500$, and $c = 0.3$. Array \mathbf{w}_2 has the smallest MSE when $35 \leq D \leq 80$. Arrays $\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2 can identify more DOAs than CADiS.

aperture and only consider WCNA and other sparse arrays with $\mathcal{O}(N^2)$ aperture. $N = 30$ sensors, SNR of 5 dB, coupling strength $c = 0.4$, and 500 snapshots are considered. The number of DOAs D is varied on the X -axis. The CADiS array configuration that has the largest number of identifiable DOAs with coarray-MUSIC is used. We observe that the proposed weight-constrained nested array \mathbf{w}_2 with $w(1) = w(2) = 0$ has the smallest MSE and can effectively identify a large number of DOAs in the presence of mutual coupling with small MSE.

As mentioned in the introduction of this chapter, the CADiS array can have multiple possible configurations for a given number of sensors. Here we provide additional simulations to further compare the proposed WCNA with different CADiS configurations. In Fig. 4.9, for each value of N , we choose the CADiS array configuration with the longest ULA segment in the coarray. Nested CADiS [83] is not considered as it does not have $w(1) = 0$. We compare the maximum number of identifiable DOAs using coarray-MUSIC (D_m), coupling leakage \mathcal{L} when $c = 0.3$, and required

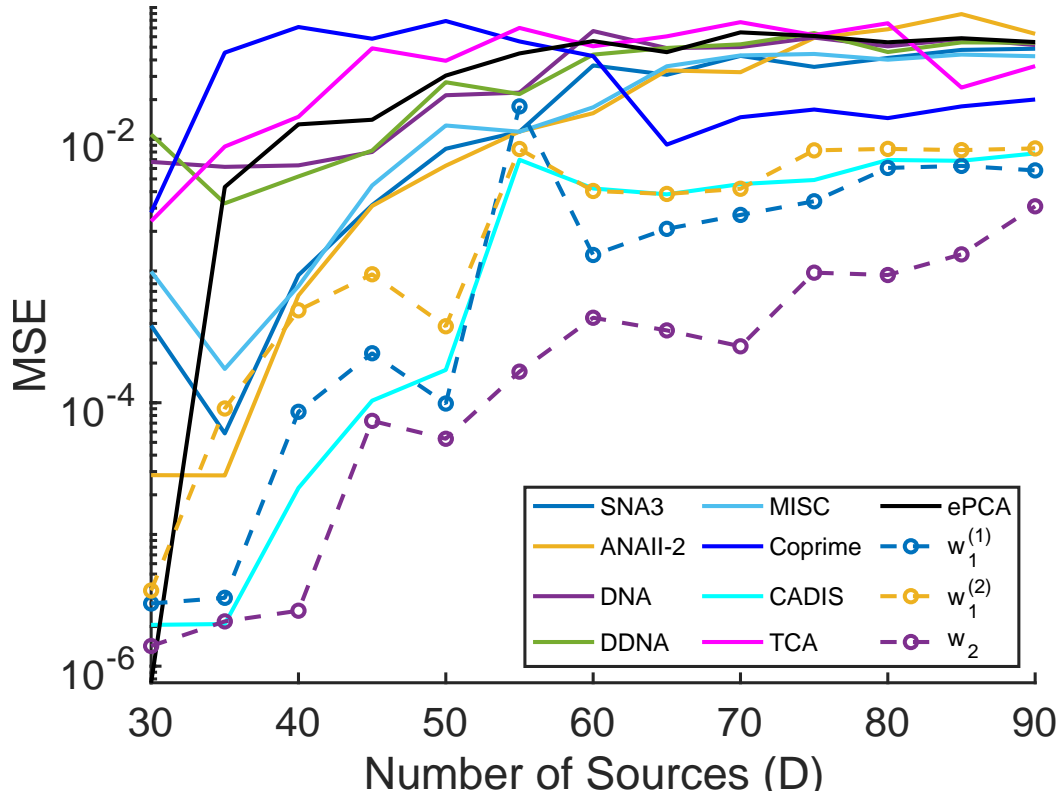


Figure 4.8: MSE when identifying D DOAs spaced uniformly in the ω domain from -0.9π to 0.9π with 30-sensor arrays. SNR = 5 dB, $K = 500$, and $c = 0.4$. Array \mathbf{w}_2 has the smallest MSE when $30 \leq D \leq 90$.

array aperture for the optimal CADiS, cMRA, and the proposed WCNAs. Notice from Fig. 4.9(a) that the proposed WCNAs can identify more DOAs than CADiS. Values of N for which CADiS has $D_m = 0$ are those for which CADiS is not available. From Fig. 4.9(b), note that array \mathbf{w}_2 has either comparable or smaller coupling leakage than CADiS for most values of N . However, arrays $\mathbf{w}_1^{(2)}$ and \mathbf{w}_2 require almost double the aperture than CADiS and $\mathbf{w}_1^{(1)}$ (Fig. 4.9(c)).

Next, in Fig. 4.10, for $N = 20$ sensors, we compare the proposed WCNAs with all possible (eight, in this case) configurations of CADiS arrays. Different CADiS array configurations provide different degrees of freedom and robustness to mutual coupling. However, we see here for the case of $N = 20$ that \mathbf{w}_2 still performs better than all CADiS configurations for $c \leq 0.5$. For larger c , some CADiS configurations having $w(l) = 0$ for several small values of l perform better than WCNAs. However, such CADiS configurations are observed to have significantly smaller D_m than WCNAs.

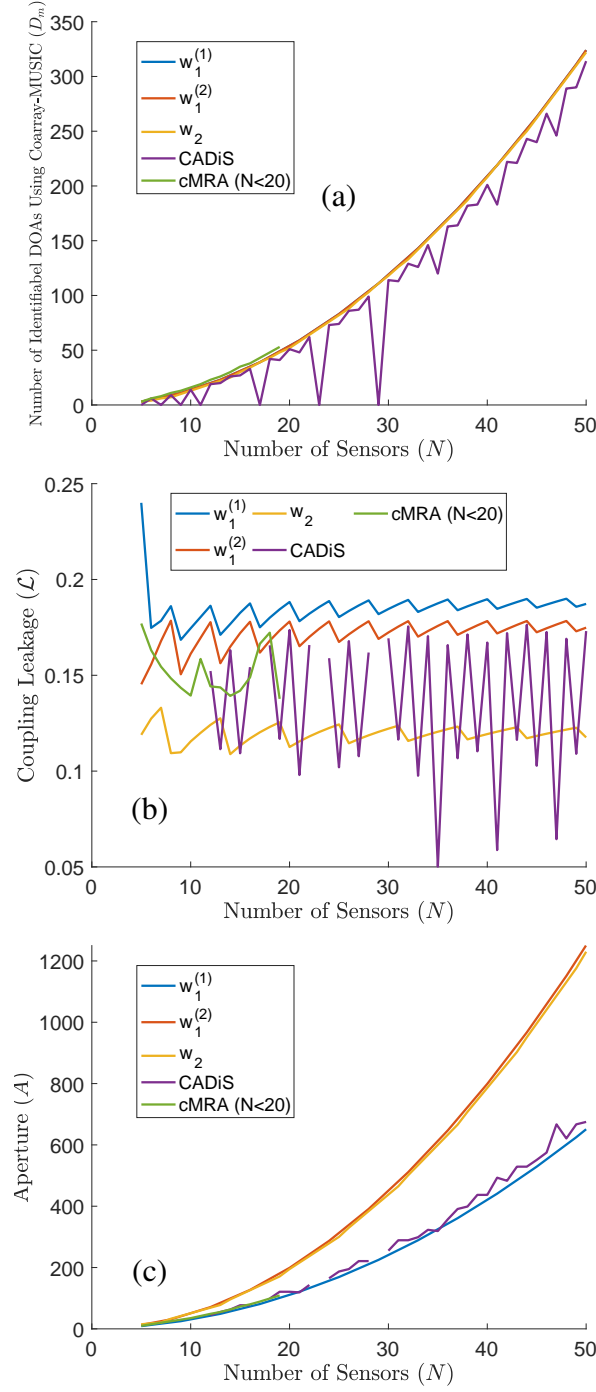


Figure 4.9: Comparing the proposed arrays $w_1^{(1)}$, $w_1^{(2)}$, and w_2 with CADiS and cMRA in terms of the number of identifiable DOAs using coarray-MUSIC (D_m), coupling leakage \mathcal{L} when $c = 0.3$, and required array aperture. For each value of N , the CADiS array configuration having the longest ULA segment in the coarray is considered. cMRA is available only for $N < 20$. The breaks in the plot for CADiS correspond to the values of N for which CADiS is not available. The proposed WCNA can identify more DOAs than CADiS with coarray-MUSIC, and w_2 has a comparable or smaller coupling leakage than CADiS for most values of N . However, apertures of $w_1^{(2)}$ and w_2 are roughly twice that of CADiS and $w_1^{(1)}$.

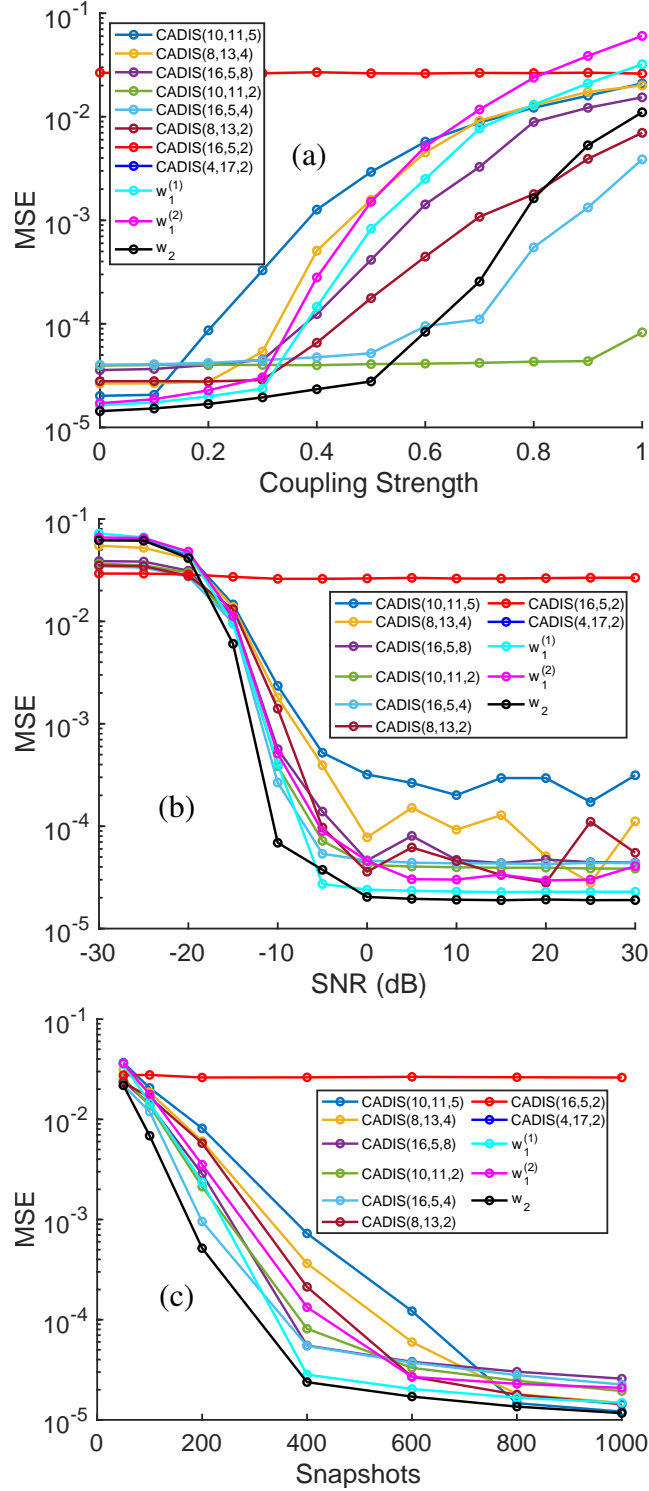


Figure 4.10: MSE when estimating $D = 30$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 20-sensor arrays. All possible configurations of CADiS are considered. The parameters of the CADiS arrays are mentioned in the brackets (M , N , and p from [83], respectively). (a) SNR = 5 dB and $K = 500$, (b) $c = 0.3$ and $K = 500$, (c) $c = 0.3$ and SNR = 5 dB. Array w_2 performs the best in this case when $c \leq 0.5$ for any number of snapshots and SNR.

4.5.5 Effect of Coupling Parameters and Source Correlations

In most of our simulations with mutual coupling, we have used coupling coefficients of the form $c_1 = ce^{j\pi/3}$, and $c_l = (c_1/l)e^{-j(l-1)\pi/8}$, and with $B = 10$. As mentioned before, these values for coupling coefficients have been used in most of the papers on sparse arrays [39], [91], [92], [105], and we use the same values in our simulations above for consistency. However, we have observed [154] that the DOA estimation performance may vary based on the phases of the coupling coefficients chosen. Here we present one example of this effect.

We compare some of the WCSAs with a few other sparse arrays from the literature under an aperture-constrained setting. We assume that a fixed aperture $A = 50$ and a maximum of 30 sensors are available. For each array type, we appropriately choose the number of sensors such that the array satisfies the aperture constraint. For example, the ULA has 30 sensors, \mathbf{z}_1 has 25 sensors, $\mathbf{z}_2^{(1)}$ has 18 sensors, and so on. Super nested array, coprime array, CADiS, and MISC array with the appropriate number of sensors that fit in the available aperture are chosen. Under this simulation setting, the array performances are found to vary a lot depending on the specific mutual coupling parameters chosen. Here we present three different examples. Mutual coupling with $B = 100$, $c_1 = \exp(j\pi/3)$ and $c_l = (c_1/l)\exp(-j(l-1)\pi/M)$ is assumed where M is 5, 7, and 8 for Figs. 4.11(a), 4.11(b), and 4.11(c) respectively. Four sources with SNR 0 dB and 1000 snapshots are considered. In Fig. 4.11(a), $\mathbf{z}_2^{(1)}$ performs better than other sparse arrays when $|c_1| \leq 0.6$, and better than ULA when $0.2 < |c_1| \leq 0.6$. In Figs. 4.11(b) and 4.11(c), it is difficult to pick a clear winner, but \mathbf{z}_1 performs better than $O(N^2)$ aperture sparse arrays in many situations. Further research is required to understand the sensitivity of different arrays and DOA estimation algorithms to the phases of mutual coupling coefficients.

Next, in Fig. 4.12, we also plot the MSE for the arrays from Table 4.5 (with $A \leq 100$ and $N \leq 50$) as the parameter B in the mutual coupling model is varied from 10 to 100. We observe that the general trend of the proposed arrays compared to existing arrays remains consistent as the parameter B is changed. This means that for different values of B , many of the proposed arrays can still perform better than the existing arrays.

As mentioned at the end of Sec. 2.5, although coarray-based DOA estimation assumes uncorrelated sources, correlated sources can be present in practical situations. To evaluate the DOA estimation performance of the weight-constrained arrays proposed in this chapter for correlated sources, we conduct additional simu-

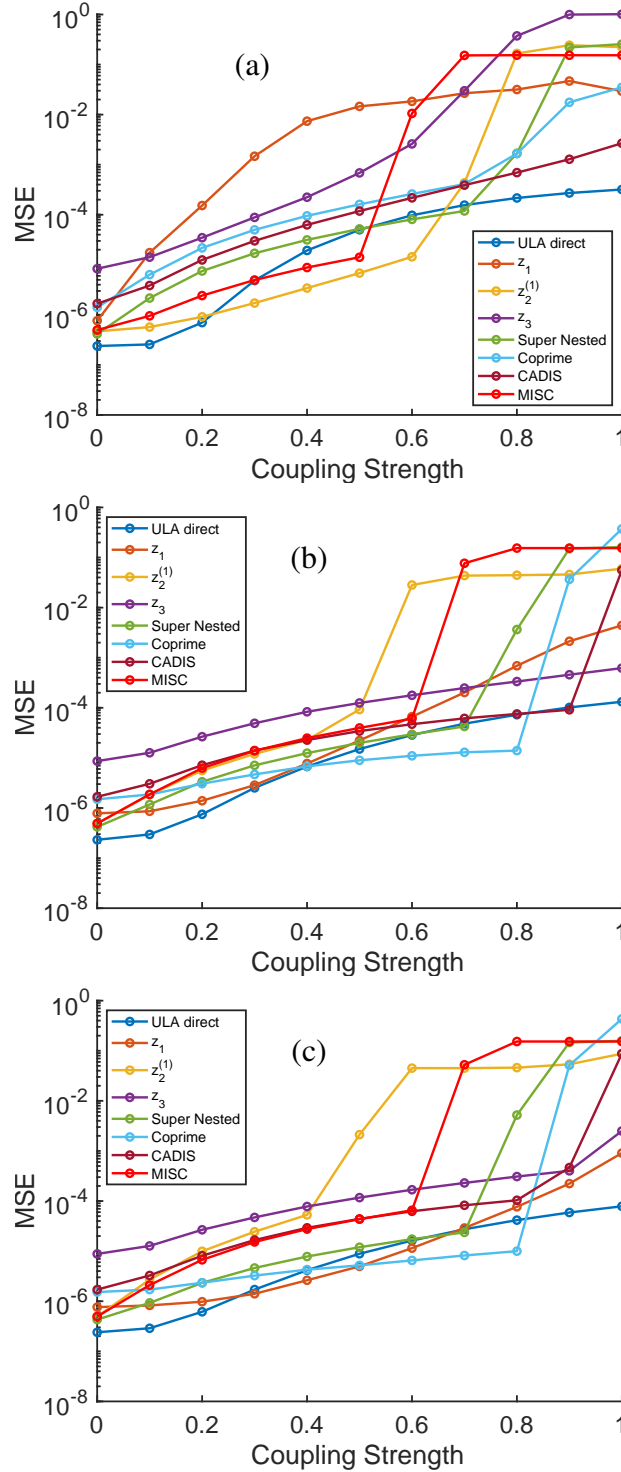


Figure 4.11: Comparing different arrays under the constraints $A \leq 50$ and $N \leq 30$. Mutual coupling coefficients are of the form $c_l = (c_1/l)\exp(-j(l-1)\pi/M)$. (a) $M = 5$, (b) $M = 7$, and (c) $M = 8$. $D = 4$, $K = 1000$, and $\text{SNR} = 0$ dB. Under different conditions, proposed sparse arrays \mathbf{z}_1 or $\mathbf{z}_2^{(1)}$ can perform better than other sparse arrays with $O(N^2)$ aperture. \mathbf{z}_3 in the figure legends refer to the array $\mathbf{z}_3^{(1)}$ from Table 4.2.

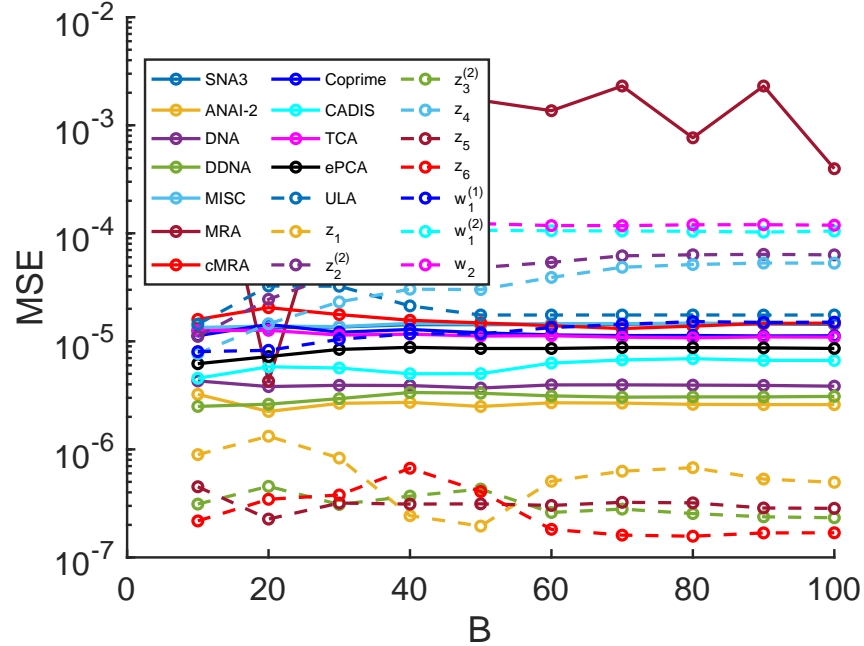


Figure 4.12: MSE when identifying $D = 6$ DOAs spaced uniformly in the ω domain from -0.8π to 0.8π for arrays with $A \leq 100$ and $N \leq 50$. SNR = 5 dB, $K = 500$, and $c = 0.5$. The general trend of the proposed arrays compared to existing arrays remains consistent as the parameter B is changed.

lation experiments. Using the 16-sensor arrays from Table 4.4, 10 DOAs uniformly spaced in the ω -domain from -0.9π to 0.9π are estimated. Although when the sources are correlated, the ideal correlations do not have the sum of sinusoids form as in Eq. (2.21), we still estimate $\hat{R}(l)$ as in Eq. (2.22) and construct the matrix by $\hat{\mathbf{R}}$ from Eq. (4.1) to estimate DOAs. In Fig. 4.13, the correlation coefficient between the second and fifth sources is set to $\rho e^{-j\pi/4}$, and the rest of the sources are uncorrelated. The magnitude of the correlation coefficient ρ is varied on the x -axis to measure its impact on DOA estimation MSE. We observe that in the absence of mutual coupling ($c = 0$), the rate of increase in MSE for most of the proposed arrays is not significantly different from the rate of increase in MSE for other sparse arrays. Moreover, in the presence of mutual coupling with $c = 0.4$, the proposed arrays still perform much better than other arrays from the literature, in the presence of a correlated source pair.

Next, in Fig. 4.14, we introduce a correlation coefficient of $\rho/2 \cdot e^{j\pi/3}$ between the seventh and ninth source, in addition to the correlation coefficient of $\rho e^{-j\pi/4}$ between the second and fifth source. Thus, there are now two correlated source pairs. These numerical results indicate that the proposed arrays can continue to

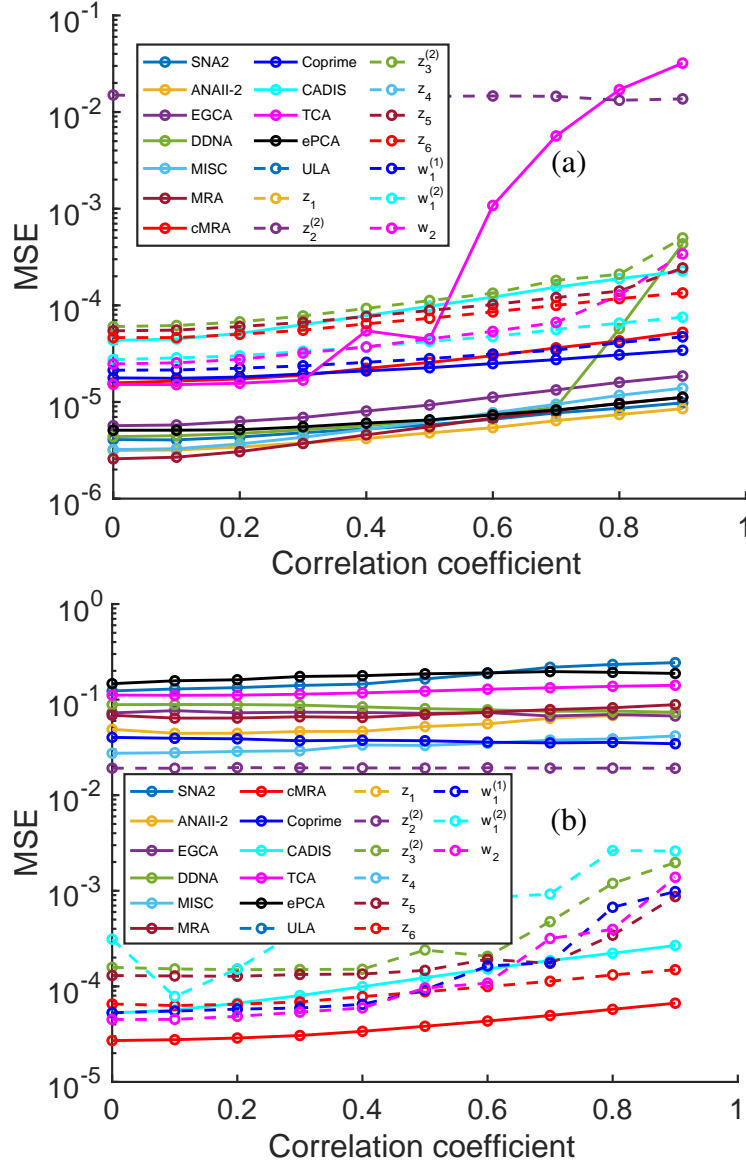


Figure 4.13: MSE when estimating $D = 10$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays when $\text{SNR} = 5$ dB, and $K = 500$. The correlation coefficient between the second and fifth source is taken to be $\rho e^{-j\pi/4}$. (a) $c = 0$, and (b) $c = 0.4$.

perform well in the presence of high mutual coupling and high correlations between a few pairs of sources.

4.5.6 Main Conclusions From Simulations

The observations from the above numerical simulations can be summarized as follows. These also serve as a guideline for the situations in which the proposed arrays can be used and can be expected to perform well.

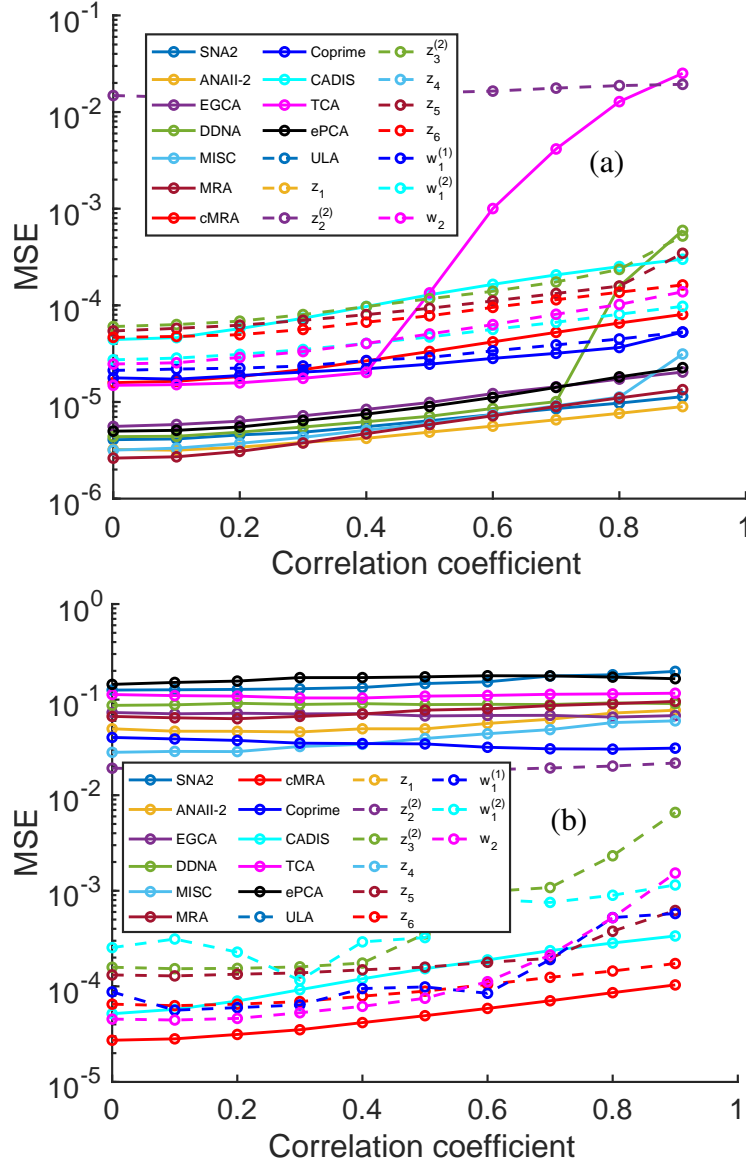


Figure 4.14: MSE when estimating $D = 10$ DOAs uniformly spaced in ω from -0.9π to 0.9π with 16-sensor arrays when SNR = 5 dB, and $K = 500$. The correlation coefficient between the second and fifth source is taken to be $\rho e^{-j\pi/4}$, and the correlation coefficient between the seventh and ninth source is $\rho/2 \cdot e^{j\pi/3}$. (a) $c = 0$, and (b) $c = 0.4$.

For a fixed number of sensors: When $c > 0.2$ and D is large, arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ are necessary to effectively mitigate the effect of mutual coupling, and correctly identify all DOAs (Fig. 4.2, Fig. 4.4). Arrays with $w(1) \neq 0$ perform poorly in this setting. However, there is no advantage to using arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ when $c < 0.2$. For a small number of DOAs, $\mathcal{O}(N)$ aperture array having small coarray weights (such as \mathbf{z}_6) can perform

well (Fig. 4.3). Proposed arrays with $O(N^2)$ are suitable for identifying a large number of DOAs with smaller errors in the presence of mutual coupling (Fig. 4.7). The proposed arrays also continue to perform well when a few source pairs are correlated (Figs. 4.13 and 4.14).

Under aperture constraint: Under fixed aperture, the proposed $O(N)$ aperture arrays can accommodate a larger number of sensors while still maintaining low coupling leakage (Table 4.5). This makes them effective regardless of the mutual coupling strength when there are only a few DOAs (Fig. 4.5). For a larger number of DOAs, some of the proposed arrays (such as \mathbf{z}_1) can perform well provided a high SNR and a larger number of snapshots are available (Fig. 4.6). At low SNRs and for a small number of snapshots, ULA is found to perform the best despite having high coupling leakage.

4.6 Concluding Remarks

In this chapter, we introduced new weight-constrained arrays that are robust to the presence of high mutual coupling. We argued that with a large number of sensors N , we may not always require $O(N^2)$ degrees of freedom, and $O(N^2)$ length aperture of such sparse arrays can be impractical for various applications. To better accommodate aperture constraints and mitigate the impact of high mutual coupling, we proposed two modifications to the array design criteria and presented several $O(N)$ aperture arrays with either $w(1) = 0$ or $w(1) = w(2) = 0$. Many of these arrays can still identify $D > N$ DOAs (such as $D = 2N$) using coarray-MUSIC on a one-sided ULA segment present in the coarray. These arrays are obtained by appropriately dilating a ULA and augmenting it with a few additional sensors. Through Monte-Carlo simulations, we demonstrated that these new arrays can outperform existing arrays from the literature under high mutual coupling or aperture constraint conditions. Additionally, we proposed another type of arrays with $w(1) = 0$ or $w(1) = w(2) = 0$ and $O(N^2)$ aperture, called weight-constrained nested arrays, which are suitable for identifying a large number of DOAs under high mutual coupling when aperture is not constrained.

The theme of ‘aperture-aware’ array design introduced in this work can be further explored to design sparse arrays systematically under aperture constraints. Several interesting array design questions arise under such a setting. For example, given a fixed aperture A and a fixed number of sensors $N < A$, it would be of interest to know the sensor arrangement that leads to the lowest mutual coupling and gives the best

DOA estimation performance. Furthermore, similar to the 1D weight-constrained arrays introduced in this chapter, 2D sparse arrays that have zero coarray weights at smaller lags are also of interest for future research.

Note that all the simulations in this section are based on coarray root-MUSIC. We chose coarray root-MUSIC because it has theoretical identifiability guarantees under ideal conditions, is conceptually simple to use, and does not require hyperparameter tuning like some other methods (such as dictionary-based methods) do. However, because of the DOA estimation algorithm based on a one-sided coarray segment, the proposed arrays are at a ‘disadvantage’ compared to the arrays having a *central* ULA segment in the coarray. For the arrays with *central* ULA segment in the coarray, both sides of the coarray are used to form the matrix $\hat{\mathbf{R}}$, whereas, for the arrays with central holes in their coarray (i.e., $w(1) = 0$), only one side of the coarray is used. Despite this limitation, the proposed arrays have a smaller MSE compared to other sparse arrays in the literature in the presence of mutual coupling under several simulation settings. This already demonstrates the value of our proposed arrays.

However, to further enhance the performance of the proposed arrays, other DOA estimation algorithms that can use both sides of the difference coarray should be considered. In the next chapter, we explore different algorithms that can do this. In particular, we propose ‘augmented root-MUSIC’ that can utilize the ULA segments on both sides of the coarrays with central holes. We have found that for some of the proposed arrays, the augmented root-MUSIC can have over an order of magnitude smaller MSE than coarray root-MUSIC that uses a one-sided ULA segment. Another option is to algorithmically fill up the central holes in the coarrays by using appropriate coarray interpolation techniques [159]–[161]. However, it can be challenging to incorporate mutual coupling into such algorithms. Once the central holes are appropriately filled through interpolation, we can use the central ULA segment in the coarray instead of the one-sided segment. These directions are explored in the next chapter.

IMPROVED DOA ESTIMATION ALGORITHMS AND GENERALIZED CONSTRUCTIONS FOR WEIGHT-CONSTRAINED ARRAYS

5.1 Introduction

In the last chapter, we highlighted some drawbacks of the traditional sparse array design principles. In particular, we argued that an $O(N^2)$ length aperture may not be suitable for practical applications with limited available aperture, especially when a large N is used. Furthermore, it was observed that although $w(l) > 0$ for $l = 1, 2, 3$, etc., ensure that the coarray contains a *central* ULA segment, such arrays may not work well in the presence of high mutual coupling. Thus, we proposed some modifications to the array design criteria and developed weight-constrained sparse arrays (WCSAs). These arrays are designed to have either $w(1) = 0$ or $w(1) = w(2) = 0$ and have $O(N)$ aperture, unlike the traditional sparse arrays. WCSAs are suitable in the presence of high mutual coupling and when a small aperture of $O(N)$ length is desired. Additionally, we also proposed weight-constrained nested arrays (WCNAs) that have either $w(1) = 0$ or $w(1) = w(2) = 0$ to reduce the impact of mutual coupling on DOA estimation, and have $O(N^2)$ aperture like traditional sparse arrays. WCNAs are suitable for identifying $O(N^2)$ DOAs in the presence of high mutual coupling. Because of zero coarray weights at lags 1 and 2, WCSAs and WCNAs are robust to high mutual coupling, as demonstrated through simulations in the previous chapter.

The coarrays of the weight-constrained arrays have central holes at lags 1 and 2 because of $w(1) = 0$ or $w(1) = w(2) = 0$, as shown in Fig. 5.1. There is a ULA segment of length L from lags L_1 to L_2 on the positive side of the difference coarrays of weight-constrained arrays. There is also a corresponding ULA segment on the negative side of the coarray from lags $-L_2$ to $-L_1$. In Chapter 4, to estimate the DOAs using weight-constrained arrays, we only used the one-sided ULA segment in the coarray starting from lag 2 or 3. Despite not using the ULA segment on the other side of the difference coarray, WCSAs and WCNAs are demonstrated to have smaller DOA estimation MSE under high mutual coupling in Sec. 4.5. This already shows the advantage of WCSAs and WCNAs under high mutual coupling. However,

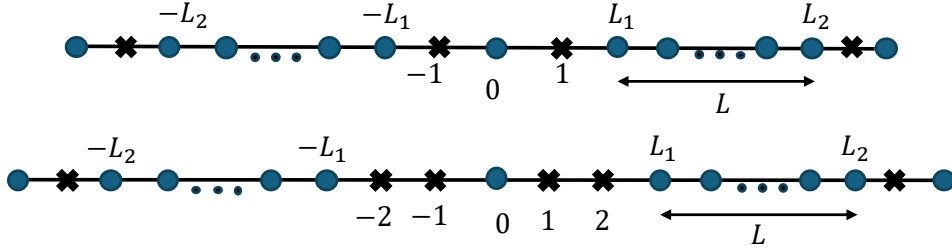


Figure 5.1: A representative demonstration of the coarrays of the weight-constrained arrays proposed in Chapter 4. Because of the central holes at lags 1 and 2, the coarrays do not contain a *central* ULA segment. There is a one-sided ULA segment on the positive side of the coarray from lags L_1 to L_2 , and a corresponding ULA segment on the negative side of the coarray from lags $-L_2$ to $-L_1$.

to further improve the DOA estimation performance of these arrays having central holes, it is important to develop algorithms that use both positive and negative sides of the difference coarrays. To address this, in this chapter, we consider a few alternate algorithms for DOA estimation using such arrays.

First, we propose coarray matrix augmentation, where the coarray Toeplitz matrices containing the correlations at lags from the ULA segment on the positive side and the negative side of the difference coarray are stacked to form a taller augmented matrix. We propose to use subspace-based algorithms such as root-MUSIC and ESPRIT with the augmented matrix and provide theoretical justification for doing so.

Next, we consider some other approaches from the literature that can also use both sides of the coarray having central holes, such as dictionary-based DOA estimation and covariance interpolation. We observe that the dictionary-based DOA estimation approach is not suitable when D is large, but the covariance interpolation can work well. Several interpolation algorithms are proposed in the literature, such as [160]–[165]. When there are holes in the difference coarray, these algorithms can interpolate the missing correlation values and form a matrix that is larger in size than what can be obtained by spatial smoothing or direct augmentation approaches (Sec. 2.5). Most of the previous papers have used interpolation for coprime arrays or other similar arrays that already contain a large *central* ULA segment. By filling the holes that appear *after* the ULA segment in the coarray, a few additional degrees of freedom from the non-ULA part of the coarray are utilized.

To the best of our knowledge, this is the first time covariance interpolation is being

proposed for the arrays with central holes in the coarray at lags such as 1 and 2. After interpolation, the estimated Toeplitz covariance matrix is used to perform root-MUSIC. We will demonstrate in our simulations that after interpolation, up to twice the number of DOAs can be identified using WCNAs, compared to what is possible with only using a one-sided ULA segment in the coarray. Thus, with interpolation, we mitigate the critical disadvantage of the non-central ULA segment in the difference coarray of WCNAs but still retain the advantage of robustness to mutual coupling because of $w(1) = 0$ or $w(1) = w(2) = 0$.

We want to note that although we focus on demonstrating the advantages of using interpolation for weight-constrained nested arrays in this chapter, similar advantages will also be observed for other arrays with central holes in the difference coarray. Some examples of such arrays from the literature include CADiS [83], generalized nested arrays (GNA) [152], and generalized non-redundant sparse arrays [166]. The algorithms discussed in this chapter also apply to such arrays.

In the last part of this chapter, we propose a generalized construction for weight-constrained sparse arrays. Out of WCSAs proposed in Chapter 4 (see Table 4.2), \mathbf{z}_1 has $w(1) = 0$, but $w(2)$ increases linearly with N . Arrays $\mathbf{z}_2^{(1)}$ and $\mathbf{z}_2^{(2)}$ have $w(1) = 0$ and $w(2)$ is a small constant, but $w(3)$ increases linearly with N . Arrays $\mathbf{z}_3^{(1)}$ and $\mathbf{z}_3^{(2)}$ have $w(1) = 0$, and $w(2)$ and $w(3)$ are small constants, but $w(4)$ increases linearly with N . By ensuring that the coarray weights at the initial few lags are small constants that do not depend on N , we reduce the impact of mutual coupling on DOA estimation.

To retain the advantages that arise from $O(N)$ aperture and $w(1) = 0$, and to reduce the impact of mutual coupling on DOA estimation even further, a natural question that arises is whether it is possible to construct an array such that:

1. The array aperture has $O(N)$ length and $w(1) = 0$.
2. For a given integer P , the coarray weights $w(l)$ for $2 \leq l \leq P$ do not increase with the number of sensors N .
3. The difference coarray contains a large ULA segment starting from lag $l = 2$.

We answer this question affirmatively in the last part of this chapter by proposing a generalized WCSA construction that provably satisfies the above requirements. The effects of changing the parameter P on different array properties, such as aperture,

number of identifiable DOAs using coarray domain root-MUSIC or ESPRIT, and coupling leakage, are analyzed. Improved constructions of sparse arrays under the constraints $w(1) = 0$ or $w(1) = w(2) = 0$ are of interest to further improve the DOA estimation accuracy under high mutual coupling.

Chapter outline. Sec. 5.2 proposes coarray matrix augmentation and provides theoretical justification for the proposed algorithms. Sec. 5.3 describes the interpolation algorithm that we propose to apply for weight-constrained arrays. In Sec. 5.4, we compare the DOA estimation performance and computation time of augmented root-MUSIC, augmented ESPRIT, covariance interpolation, and dictionary-based approach. In Sec. 5.5, we propose a generalized construction for WCSA. Then in Sec. 5.6, we compare the proposed generalized construction of WCSA with previously proposed constructions, as well as other well-known sparse arrays from the literature. Finally, Sec. 5.7 concludes the chapter. The content of this chapter is based on papers [167]–[169].

5.2 Augmented Coarray Root-MUSIC

To utilize both ULA segments present in the coarrays of WCSAs, in this chapter, we propose to use coarray-domain subspace algorithms such as coarray-MUSIC and coarray-ESPRIT using an augmented coarray matrix. This matrix is formed by stacking $\widehat{\mathbf{R}}^{(+)}$ and $\widehat{\mathbf{R}}^{(-)}$ as follows:

$$\widehat{\mathbf{R}}_{aug} = \begin{bmatrix} \widehat{\mathbf{R}}^{(-)} \\ \widehat{\mathbf{R}}^{(+)} \end{bmatrix}. \quad (5.1)$$

Here, $\widehat{\mathbf{R}}^{(+)}$ and $\widehat{\mathbf{R}}^{(-)}$ are the Toeplitz matrices of size $r_1 \times r_2$ where

$$r_1 = \lfloor L/2 \rfloor + 1, \text{ and } r_2 = \lceil L/2 \rceil. \quad (5.2)$$

The matrices are constructed based on the correlations from the ULA segment in the positive side of the coarray (i.e., $\widehat{R}(l)$ for $L_1 \leq l \leq L_2$) and the negative side of the coarray (i.e., $\widehat{R}(l)$ for $-L_2 \leq l \leq -L_1$), respectively. The (i, j) -th entry in $\widehat{\mathbf{R}}^{(+)}$ is given by

$$\widehat{\mathbf{R}}_{i,j}^{(+)} = \widehat{R}(L_1 + \lceil L/2 \rceil - 1 + i - j). \quad (5.3)$$

The (i, j) -th entry in $\widehat{\mathbf{R}}^{(-)}$ is given by

$$\widehat{\mathbf{R}}_{i,j}^{(-)} = \widehat{R}(-L_2 + \lceil L/2 \rceil - 1 + i - j). \quad (5.4)$$

Note that these matrices are not Hermitian and positive semidefinite, and are not even square matrices when L is even. The augmented matrix $\widehat{\mathbf{R}}_{aug}$ is a tall matrix of size $2r_1 \times r_2$.

It can be shown that under ideal situations (with no mutual coupling and a large number of snapshots), the augmented matrix \mathbf{R}_{aug} admits a rank- D matrix decomposition. This is based on Vandermonde-like decompositions of the individual Toeplitz matrices $\mathbf{R}^{(+)}$ and $\mathbf{R}^{(-)}$, as explained next. Here \mathbf{R}_{aug} , $\mathbf{R}^{(+)}$, and $\mathbf{R}^{(-)}$ are the matrices with ideal correlations $R(l)$ instead of the estimated correlations $\widehat{R}(l)$.

Similar to the decomposition of \mathbf{R} shown in Sec. 4.2, $\mathbf{R}^{(+)}$ can be decomposed as

$$\mathbf{R}^{(+)} = \mathbf{A}^{(+)} \mathbf{P} \mathbf{B}^H. \quad (5.5)$$

Here, $\mathbf{A}^{(+)}$ is an $r_1 \times D$ matrix with entries

$$\mathbf{A}_{m,n}^{(+)} = e^{j\omega_n(L_1+m-1)}. \quad (5.6)$$

\mathbf{P} is a $D \times D$ diagonal matrix containing source powers p_1, \dots, p_D . \mathbf{B} is an $r_2 \times D$ matrix with entries

$$\mathbf{B}_{m,n} = e^{-j\omega_n(r_2-m)}. \quad (5.7)$$

Similar to Eq. (5.5), $\mathbf{R}^{(-)}$ can be decomposed as

$$\mathbf{R}^{(-)} = \mathbf{A}^{(-)} \mathbf{P} \mathbf{B}^H, \quad (5.8)$$

where $\mathbf{A}^{(-)}$ is an $r_1 \times D$ matrix with entries $\mathbf{A}_{m,n}^{(-)} = e^{j\omega_n(-L_2+m-1)}$. Note that the matrix $\mathbf{A}^{(+)}$ is the manifold matrix corresponding to the first r_1 lags in the ULA segment on the positive side (i.e., $L_1 \leq l \leq L_1 + r_1 - 1$), and $\mathbf{A}^{(-)}$ is the manifold matrix corresponding to the first r_1 lags in the ULA segment on the negative side (i.e., $-L_2 \leq l \leq -L_2 + r_1 - 1$).

Combining the above decompositions of $\mathbf{R}^{(+)}$ and $\mathbf{R}^{(-)}$, we get the following decomposition for \mathbf{R}_{aug} :

$$\mathbf{R}_{aug} = \begin{bmatrix} \mathbf{R}^{(-)} \\ \mathbf{R}^{(+)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(-)} \\ \mathbf{A}^{(+)} \end{bmatrix} \mathbf{P} \mathbf{B}^H. \quad (5.9)$$

Next, we want to show that the column span of \mathbf{R}_{aug} is the same as the column span of the following virtual augmented array manifold matrix:

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{A}^{(-)} \\ \mathbf{A}^{(+)} \end{bmatrix}. \quad (5.10)$$

To see this, note that \mathbf{A}_{aug} and \mathbf{B} both have rank D when $D < r_1$ and all DOAs are distinct. Hence, using Sylvester's rank inequalities, it can be concluded that

\mathbf{R}_{aug} and \mathbf{A}_{aug} have the same rank. Combining this with the fact that any vector \mathbf{x} satisfying $\mathbf{x}^H \mathbf{A}_{aug} = 0$ also satisfies $\mathbf{x}^H \mathbf{R}_{aug} = 0$ (from Eq. (5.9)), it can be concluded that matrices \mathbf{A}_{aug} and \mathbf{R}_{aug} have the same column span.

In practice, we can only estimate this column span (or equivalently, its orthogonal complement) from the estimated matrix $\hat{\mathbf{R}}_{aug}$ with entries $\hat{R}(l)$. This can be done by performing SVD (singular value decomposition) of the matrix $\hat{\mathbf{R}}_{aug}$. The first D left singular vectors corresponding to the largest D singular values provide an estimate of the D -dimensional signal subspace (i.e., the column span of \mathbf{A}_{aug}), and the rest of the singular vectors span its orthogonal complement (i.e., the noise subspace). An important point to note here is that the signal and noise subspaces thus obtained are in \mathbb{C}^{2r_1} , whereas without the matrix augmentation proposed here (when only using one-sided ULA segment), the signal and noise subspaces are in \mathbb{C}^{r_1} . Furthermore, the number of ‘noise eigenvectors’ obtained after augmentation is $2r_1 - D$, which is more than the $r_1 - D$ noise eigenvectors that can be obtained without the matrix augmentation. Note that the maximum number of identifiable DOAs using the proposed augmentation is still $\lfloor L/2 \rfloor$. However, by estimating these signal and noise subspaces in a higher dimension, we hope to improve the DOA estimation accuracy of the arrays that have central holes in their difference coarrays.

Based on this estimated noise subspace, we can use root-MUSIC to construct an appropriate polynomial and estimate the DOAs based on the D roots of the polynomial that are closest to the unit circle. The root-MUSIC based on the noise subspace obtained from $\hat{\mathbf{R}}_{aug}$ is called augmented root-MUSIC. Instead of augmented root-MUSIC, it is also possible to use augmented-ESPRIT, with appropriately chosen subarrays and invariance required for ESPRIT.

5.3 Covariance Interpolation for WCNAs

Given a sparse array \mathbf{z} of aperture A , the interpolation task is to estimate an ideal Toeplitz covariance matrix of size $(A+1) \times (A+1)$ that a ULA of the same aperture would have produced. In the absence of mutual coupling and noise, this ideal covariance matrix corresponding to the fictitious ULA of aperture A is given by [1], [94]

$$\mathbf{R}_U = \mathbf{A}_U \mathbf{P} \mathbf{A}_U^H. \quad (5.11)$$

Here, \mathbf{A}_U is the array manifold corresponding to the ULA of aperture A , and \mathbf{P} is the source covariance matrix. As the manifold matrix of ULA \mathbf{A}_U is Vandermonde and \mathbf{P} is diagonal when sources are uncorrelated, \mathbf{R}_U is a rank- D , positive semi-definite,

Toeplitz matrix [1]. Note that in this situation, when the array output covariance is Toeplitz, the coarray covariance matrix $\widehat{\mathbf{R}}_{DA}$ described in Sec. 2.5 is the same as \mathbf{R}_U .

The output covariance $\widehat{\mathbf{R}}_{xx}$ (Eq. 2.6) of the sparse array provides estimates of some entries in \mathbf{R}_U , and it has been shown in [163] that asymptotically (with large K), the distribution of the error between the estimated values from $\widehat{\mathbf{R}}_{xx}$ and their ideal counterpart in \mathbf{R}_U can be approximated as follows:

$$\text{vec} \left(\widehat{\mathbf{R}}_{xx} - \mathbf{\Gamma} \mathbf{R}_U \mathbf{\Gamma}^T - \sigma_n^2 \mathbf{I} \right) \sim \text{AsN}(\mathbf{0}, \widehat{\mathbf{W}}). \quad (5.12)$$

Here $\mathbf{\Gamma}$ is a row selection matrix of size $N \times (A + 1)$ corresponding to the sensor positions of the sparse array \mathbf{z} . $\text{AsN}(\mathbf{0}, \widehat{\mathbf{W}})$ denotes an asymptotically normal distribution with mean $\mathbf{0}$ and covariance $\widehat{\mathbf{W}}$. The covariance matrix $\widehat{\mathbf{W}}$ is given by

$$\widehat{\mathbf{W}} = \widehat{\mathbf{R}}_{xx}^T \otimes \widehat{\mathbf{R}}_{xx} / K, \quad (5.13)$$

where \otimes denotes the Kronecker product between matrices. This fact is used to ‘whiten’ the error term and solve the following optimization problem for covariance interpolation [163]:

$$\min_{\mathbf{T}(\mathbf{u}) \succeq \mathbf{0}} \left\| \widehat{\mathbf{W}}^{-1/2} \text{vec}(\widehat{\mathbf{R}}_{xx} - \mathbf{\Gamma} \mathbf{T}(\mathbf{u}) \mathbf{\Gamma}^T) \right\|_2 + \lambda \text{tr}(\mathbf{T}(\mathbf{u})). \quad (5.14)$$

Here, the Hermitian Toeplitz matrix to be estimated is parameterized by its first column \mathbf{u} , and $\mathbf{T}(\mathbf{u})$ denotes the Hermitian Toeplitz matrix having \mathbf{u} as its first column. The first term in the optimization objective is the discrepancy between the correlation estimates that are available in $\widehat{\mathbf{R}}_{xx}$ and the corresponding entries from $\mathbf{T}(\mathbf{u})$. Because these error terms are correlated according to Eq. (5.12), a ‘whitening transform’ is applied by premultiplying $\text{vec}(\widehat{\mathbf{R}}_{xx} - \mathbf{\Gamma} \mathbf{T}(\mathbf{u}) \mathbf{\Gamma}^T)$ with $\widehat{\mathbf{W}}^{-1/2}$, and the l_2 norm of this whitened error vector is used as the first term in the objective. For simplicity, we drop the noise term $\sigma_n^2 \mathbf{I}$, but [163] proposes first to estimate noise variance $\hat{\sigma}_n^2$ and use it instead. To promote low-rank solutions, a convex relaxation of the rank penalty is added in the objective as a trace regularization term $\text{tr}(\mathbf{T}(\mathbf{u}))$. The positive semi-definiteness constraint on $\mathbf{T}(\mathbf{u})$ is enforced explicitly, and the hyperparameter λ selects the trade-off between the data fidelity term and trace regularization. In our simulations, λ is chosen to be 0.01. Similar to the simulations in [163], we have found that the performance of the proposed algorithm is not very sensitive to λ .

In our experimentation with different interpolation methods from the literature, we found that the above interpolation algorithm works better than others, especially

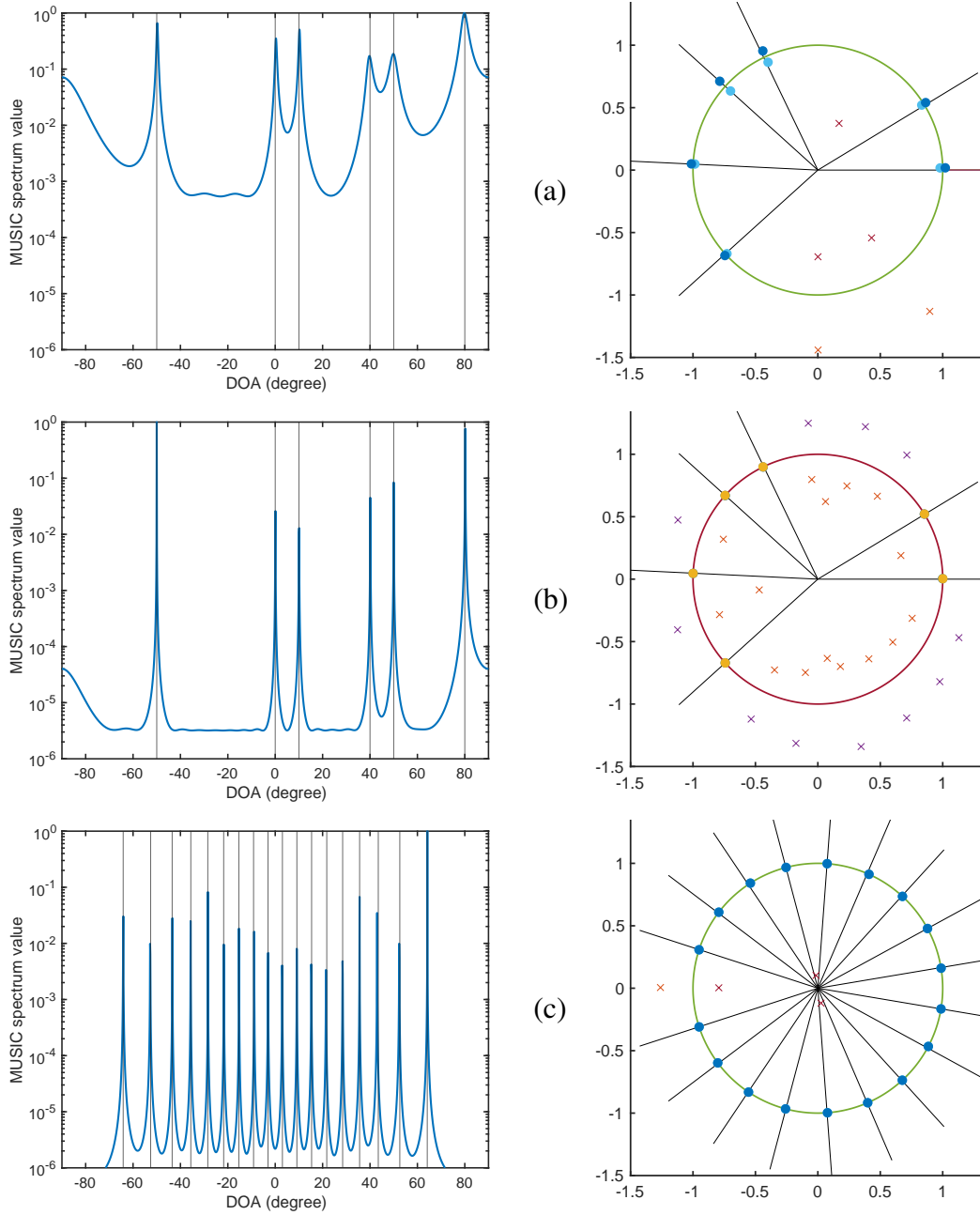


Figure 5.2: MUSIC spectra (left) and corresponding root-MUSIC plots (right) when estimating DOAs using WCNA $\mathbf{w}_1^{(1)}$ with $N = 8$. SNR is 0 dB, $K = 200$, and no mutual coupling is present. For root-MUSIC plots, the solid circles show D conjugate pairs of roots closest to the unit circle, and crosses denote other roots of the root-MUSIC polynomial. The black lines denote the angles corresponding to true DOAs $\{\omega_i\}_{i=1}^D$. (a) Estimating $D = 6$ non-uniformly spaced DOAs using one-sided ULA segment in coarray, as in Chapter 4, $\text{MSE} = 0.0990$. (b) Estimating the same six DOAs with interpolation, $\text{MSE} = 1.3785 \times 10^{-4}$. The MUSIC spectrum has much sharper peaks in this case. (c) Estimating $D = 18$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with interpolation, $\text{MSE} = 5.9150 \times 10^{-5}$.

when no mutual coupling is present. So, we use this to interpolate the coarrays of WCNA. This interpolation method does not average the correlation entries from $\widehat{\mathbf{R}}_{\mathbf{xx}}$ that correspond to the same lag. Recall that averaging of the entries from $\widehat{\mathbf{R}}_{\mathbf{xx}}$ that correspond to the same lag is done in Eq. (2.22) for computing the coarray covariance matrix. As discussed in Sec. 6.2, this averaging has been observed to be the reason for the loss of efficiency for coarray-based DOA estimation algorithms [90], [125], [158]. Furthermore, instead of simply using the l_2 norm of the error term $\text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T)$ as it is, the above method whitens this error term $\text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T)$ by premultiplying it with $\widehat{\mathbf{W}}^{-1/2}$ based on its asymptotic distribution. This is a unique feature of the above interpolation algorithm that is not used in other interpolation methods. Later in Chapter 6, we will experimentally evaluate the efficiency of DOA estimates obtained by this method.

For a demonstrative example of covariance interpolation, we consider the WCNA $\mathbf{w}_1^{(1)}$ with $N = 8$ sensors and demonstrate how the array can effectively be utilized by interpolation (Eq. (5.14)). Without interpolation, using a one-sided ULA segment in its coarray, $\mathbf{w}_1^{(1)}$ can identify up to $D_m = 9$ DOAs. First, consider six DOAs $-50^\circ, 10^\circ, 0^\circ, 40^\circ, 50^\circ$, and 80° with SNR 0 dB and 200 snapshots. Although no mutual coupling is considered here for this simulation, it will be considered in the other simulations later. Fig. 5.2(a) shows the MUSIC spectrum and root-MUSIC plot obtained when using a one-sided ULA segment in the coarray. The mean squared error in the ω -domain is 0.0990, and the MUSIC spectrum is not very sharp. In contrast, the MUSIC spectrum after interpolation in Fig. 5.2(b) is much sharper, and in the root-MUSIC plot, phases of the roots closest to the unit circle match well with the true DOAs (corresponding to the black lines), unlike the previous case. The MSE after interpolation is 1.3785×10^{-4} , which is about three orders of magnitude smaller than that without interpolation. To solve the optimization problem (5.14), we used CVX, a package for specifying and solving convex programs [170], [171]. Next, we consider $18 (= 2D_m)$ DOAs uniformly spaced in the ω domain. These DOAs cannot be estimated using only the one-sided ULA segment from the coarray of $\mathbf{w}_1^{(1)}$. The MUSIC and root-MUSIC plots from Fig. 5.2(c) obtained after interpolation show that all the DOAs are correctly identified. The MSE, in this case, is 5.9150×10^{-5} .

5.4 Simulations Comparing Proposed Algorithms

In this section, we compare the performance of the augmented root-MUSIC algorithm and other DOA estimation methods that use both sides of the difference

coarray through Monte-Carlo simulations. We consider different WCSAs (\mathbf{z}_1 to \mathbf{z}_6), WCNAAs ($\mathbf{w}_1^{(1)}$, $\mathbf{w}_1^{(2)}$, and \mathbf{w}_2). Please refer to Sec. 4.3 and Sec. 4.4 for the details of these arrays. We also consider CADiS array [83] and cMRA [84] from the literature that have central holes in their difference coarrays. All arrays have 16 sensors, and the mutual coupling model described in Eq. (2.27) is used with $B = 10$, $c_1 = ce^{j\pi/3}$, and $c_l = (c_1/l)e^{-j(l-1)\pi/8}$, where $|c_1| = c \in [0, 1]$, represents the coupling strength. The source amplitudes $\{\mathbf{s}[k]\}_{k=1}^K$ and noise $\{\mathbf{n}[k]\}_{k=1}^K$ are randomly sampled in each Monte-Carlo trial.

5.4.1 Augmented Root-MUSIC

Figs. 5.3 and 5.4 show the Monte-Carlo MSEs averaged over 1000 Monte-Carlo trials when $D = 6$ and $D = 20$, respectively. Out of the three quantities, SNR = 5dB, the number of snapshots $K = 500$, and coupling strength $c = 0.4$, two are kept fixed, and the other is varied on the X -axis. The DOA estimation MSEs with coarray root-MUSIC using only the one-sided segments in the coarray are denoted in solid lines, whereas the MSEs when using augmented root-MUSIC are denoted in dashed lines. It can be observed that in both cases ($D < N$ and $D > N$), when SNR is larger than -10 dB, more than 400 snapshots are used, and coupling strength $c < 0.6$, most arrays have significantly smaller MSE with augmented root-MUSIC than one-sided root-MUSIC. There is about an order of magnitude improvement in the MSE when augmented root-MUSIC is used. This demonstrates the advantage of the proposed matrix augmentation to use the ULA segments on both sides of the difference coarrays when the coarray has central holes. However, the polynomial formed in augmented root-MUSIC has a much higher degree than the root-MUSIC polynomial formed using only a one-sided ULA segment in the coarray. This significantly increases the computation time of the augmented root-MUSIC. In Table 5.1, we compare the computation time of different methods discussed in this section. Compared to the computation time of coarray root-MUSIC using a one-sided ULA segment in the coarray, the augmented root-MUSIC required about 12 times more computation time for the array \mathbf{z}_5 .

Finally, to observe the improvement offered over the root-MUSIC performed one-sided ULA segment, we rerun the simulations from Fig. 4.3 of Chapter 4 using the augmented root-MUSIC algorithm. The resulting plots are shown in Fig. 5.5. Note that in Fig. 4.3, only array \mathbf{z}_6 performed better than other arrays from the literature. Whereas in Fig. 5.5 above using augmented root-MUSIC, many of the proposed arrays now have over an order of magnitude smaller MSE than earlier and perform at

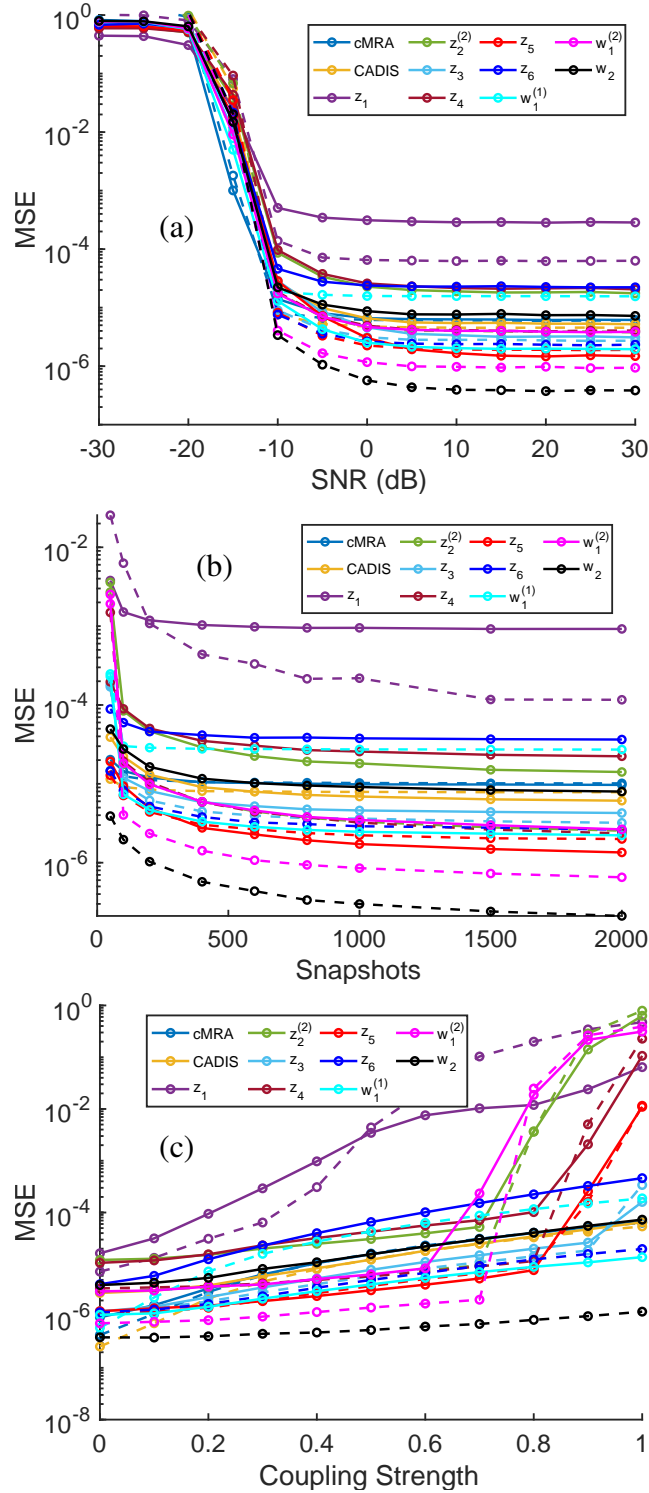


Figure 5.3: Coarray root-MUSIC MSE for estimating $D = 6$ DOAs uniformly separated in ω from -0.9π to 0.9π with 16-sensor arrays. Solid lines show MSE obtained when using a one-sided ULA segment in the coarray (i.e., matrix $\hat{\mathbf{R}}^{(+)}$), and dashed lines show MSE obtained when using both sides of the ULA segments in the coarray (i.e., matrix $\hat{\mathbf{R}}_{aug}$). (a) $c = 0.4$ and $K = 500$, (b) $c = 0.4$ and SNR = 5 dB, (c) SNR = 5 dB and $K = 500$. Augmented root-MUSIC with $\hat{\mathbf{R}}_{aug}$ improves the DOA estimation MSE compared to $\hat{\mathbf{R}}^{(+)}$ for most of the arrays.

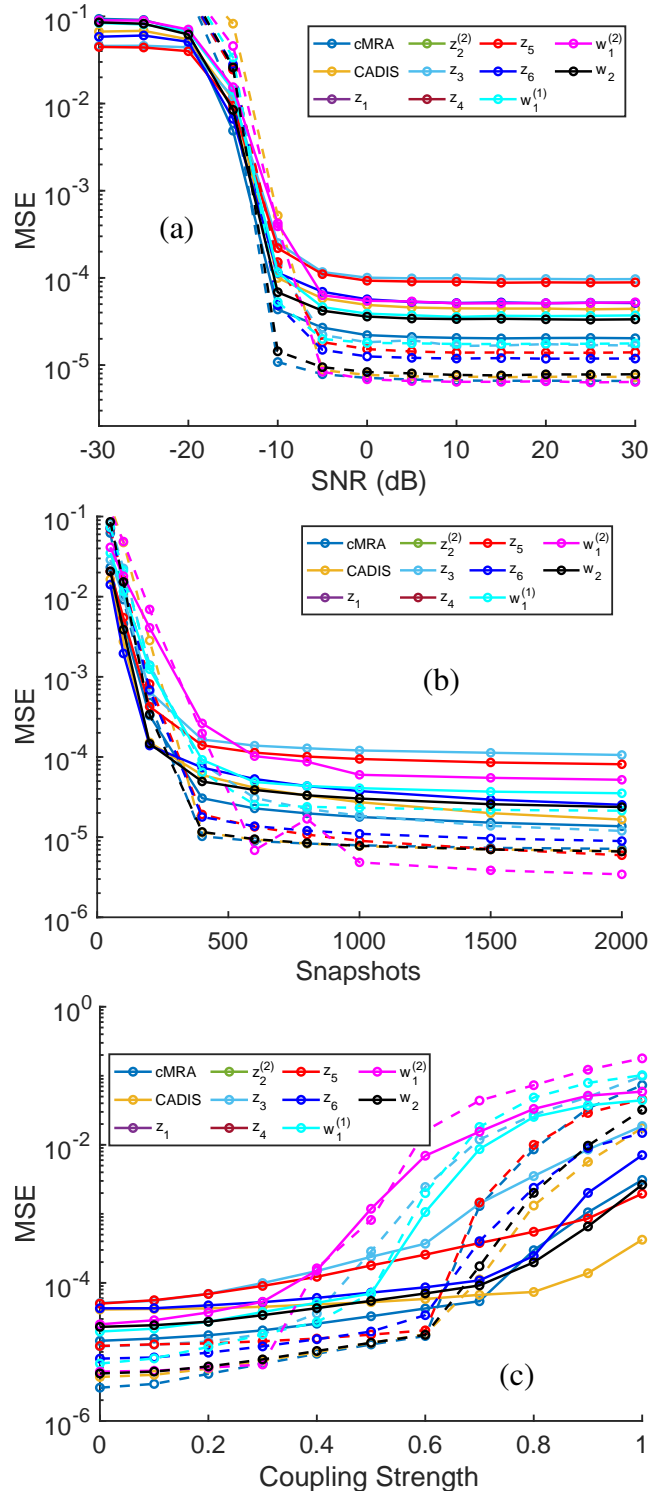


Figure 5.4: Simulation settings are the same as in Fig. 5.3, but with $D = 20$ instead of $D = 6$. Augmented root-MUSIC with $\hat{\mathbf{R}}_{aug}$ improves the DOA estimation MSE compared to $\hat{\mathbf{R}}^{(+)}$ for WCSAs and WCNAs, as well as CADiS and cMRA over a wide range of SNR, snapshots, and coupling strength.

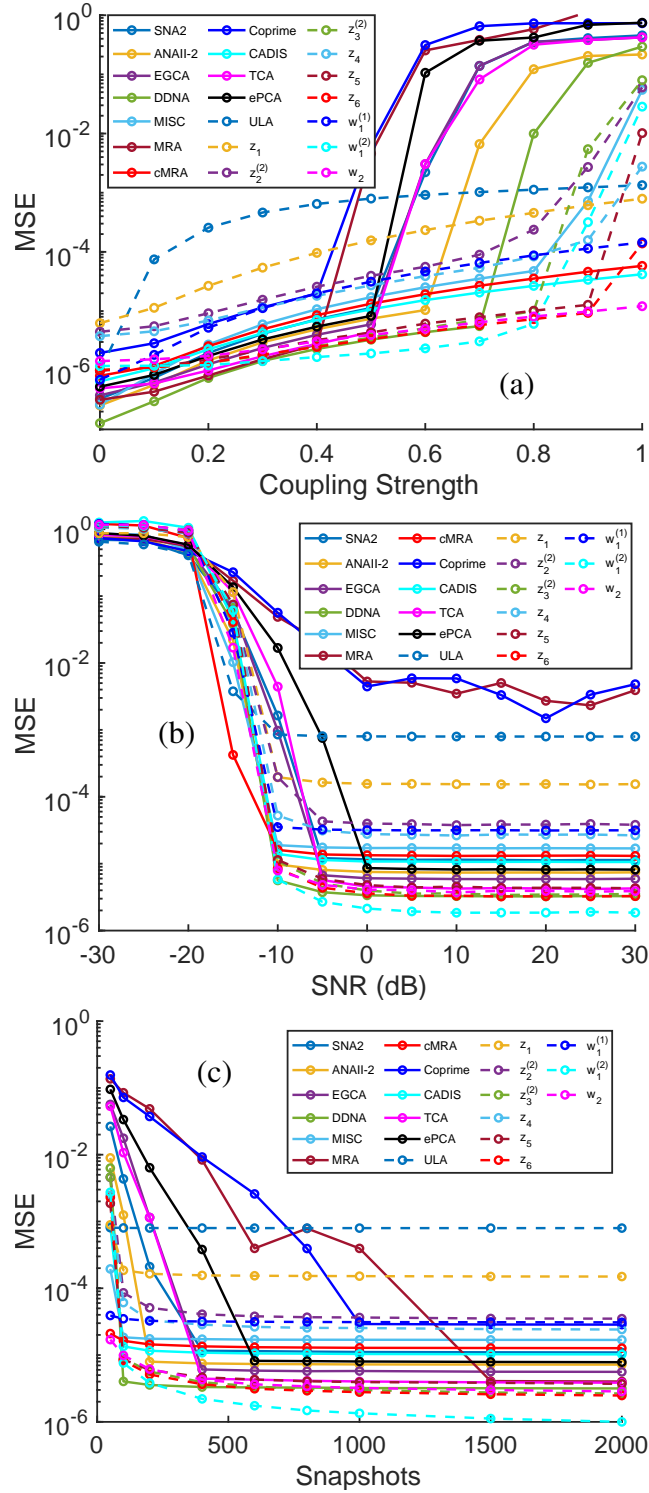


Figure 5.5: MSE when estimating $D = 6$ DOAs uniformly spaced in ω from -0.8π to 0.8π with 16-sensor arrays using augmented root-MUSIC. (a) SNR = 5 dB and $K = 500$, (b) $c = 0.5$ and $K = 500$, (c) $c = 0.5$ and SNR = 5 dB. These plots are analogous to the plots from Fig. 4.3. We observe here that because of the use of both ULA segments present in the coarrays of weight-constrained sparse arrays, now many of the proposed arrays can perform at par or better than other arrays from the literature.

Table 5.1: Comparison of computation times of different DOA estimation methods applied to array \mathbf{z}_5 with $N = 16$, averaged over 400 Monte-Carlo runs. The runtime of coarray root-MUSIC using a one-sided ULA segment in the coarray is taken as the reference value (x).

DOA estimation method	Runtime
Coarray root-MUSIC (One-sided ULA segment)	x
Augmented root-MUSIC	11.8x
Coarray ESPRIT	0.4x
Augmented ESPRIT	0.6x
Dictionary-based estimation	1057x
Interpolation followed by root-MUSIC	240x

par or better than arrays from the literature. This further highlights the importance of developing improved DOA estimation algorithms to leverage the advantage of weight-constrained arrays under mutual coupling.

5.4.2 Augmented ESPRIT

Next, as explained in the previous section, it is also possible to estimate the DOAs using ESPRIT applied on the augmented coarray matrix $\hat{\mathbf{R}}_{aug}$. Fig. 5.6 shows the coarray ESPRIT and augmented ESPRIT MSEs averaged over 1000 Monte-Carlo trials when $D = 20$. These plots show that the improvement in MSE obtained with augmented ESPRIT is much smaller than the improvement observed with augmented root-MUSIC in Figs. 5.3 and 5.4. Further investigations into why we do not observe significant improvement with augmented ESPRIT will be of interest in the future. However, augmented ESPRIT requires a much smaller computation time than that of augmented root-MUSIC, as seen in Table 5.1. Notably, although the size of the augmented matrix $\hat{\mathbf{R}}_{aug}$ is doubled in one of the dimensions compared to the one-sided coarray matrix $\hat{\mathbf{R}}^{(+)}$ previously used, the increase in the computation time for augmented ESPRIT is much smaller compared to the increase in runtime observed for augmented root-MUSIC. The reason for this is that the root-finding step dominates the computation time of augmented root-MUSIC, whereas augmented ESPRIT does not require finding roots. The computation time of SVD of an $m \times n$ matrix is $O(\min(mn^2, nm^2))$ and thus increases only linearly from $\hat{\mathbf{R}}^{(+)}$ to $\hat{\mathbf{R}}_{aug}$ when one of the dimensions of the matrix is doubled. However, the computation time of root-finding algorithms typically scales cubically with the degree of the polynomial, and its effect is observed in the increase in the computation time of augmented root-MUSIC.

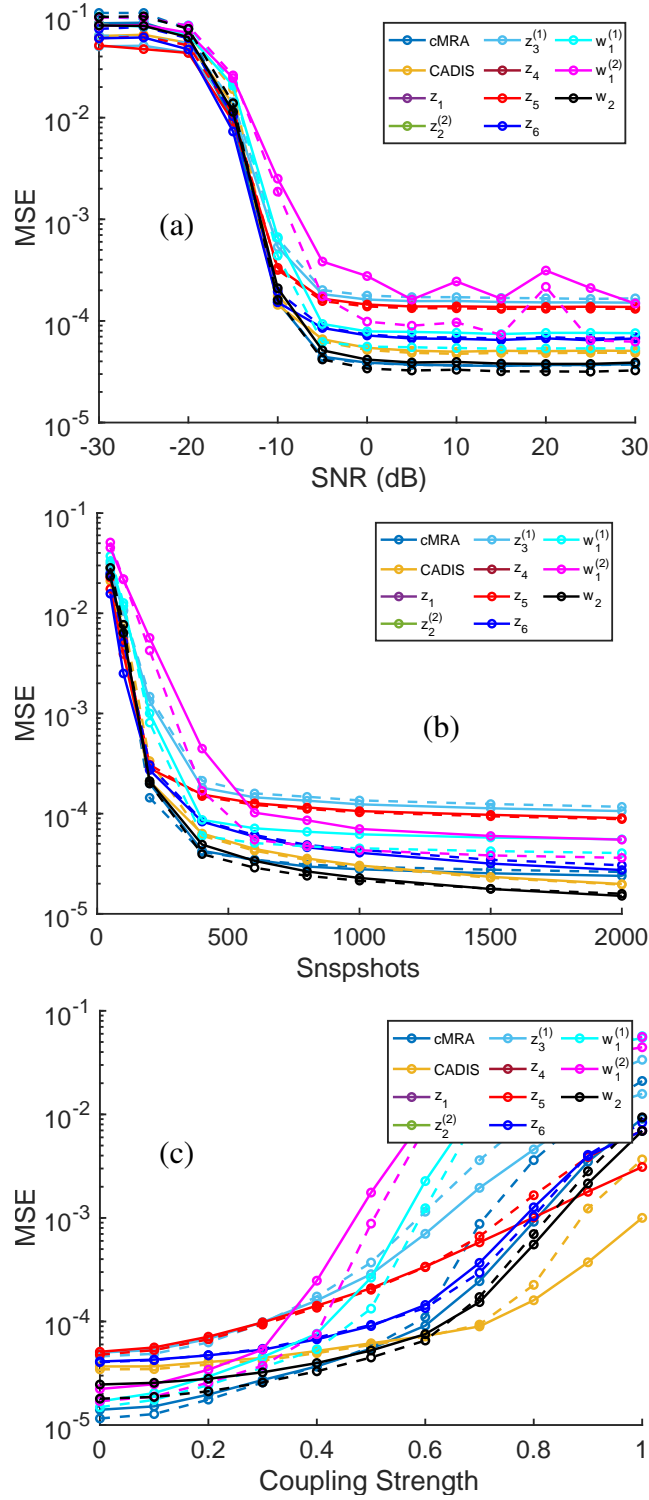


Figure 5.6: Simulation settings are the same as in Fig. 5.4, but using coarray ESPRIT and augmented ESPRIT instead of coarray root-MUSIC and augmented root-MUSIC. Improvement in MSE obtained with augmented ESPRIT is much smaller than the improvement observed with augmented root-MUSIC.

Instead of using subspace-based methods for DOA estimation considered thus far, it is possible to use other DOA estimation methods that use both sides of the difference coarray. We consider two such methods here, namely, the dictionary-based DOA estimation and covariance interpolation (Sec. 5.3). These approaches are based on solving an optimization problem and require heavy computation, as demonstrated by the much higher computation time in Table 5.1.

5.4.3 Coarray-Domain Dictionary-Based DOA estimation

The coarray-domain dictionary-based approach to DOA estimation is based on solving the following optimization problem:

$$\min_{\bar{\mathbf{p}}} \|\widehat{\mathbf{r}}_{\mathbf{x}} - \mathbf{D}\bar{\mathbf{p}}\|_2^2 + \lambda \|\bar{\mathbf{p}}\|_1. \quad (5.15)$$

where $\widehat{\mathbf{r}}_{\mathbf{x}} = \text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}})$, and the dictionary \mathbf{D} is constructed based on M sampled directions $\omega_1, \dots, \omega_M$ as

$$\mathbf{D} = \begin{bmatrix} \mathbf{b}(\omega_1) & \mathbf{b}(\omega_2) & \dots & \mathbf{b}(\omega_M) & \text{vec}(\mathbf{I}) \end{bmatrix}, \quad (5.16)$$

where $\mathbf{b}(\omega) = \mathbf{a}^*(\omega) \otimes \mathbf{a}(\omega)$ is the Kronecker product of the steering vector corresponding to direction ω with its conjugate. We use $M = 2000$ and a regularization parameter $\lambda = 0.01$ in the simulations. CVX [170], [171] is used to solve the optimization problem. We observe that under high mutual coupling, the dictionary-based DOA estimation works well when D is small, but it does not work well when D is large. In Figs. 5.7(a) and 5.7(b), the dictionary-based spectra obtained for array \mathbf{z}_5 with $N = 16$ sensors are shown, when estimating $D = 6$ and $D = 20$ DOAs, respectively. SNR 5 is dB, $K = 500$ snapshots are used, and the coupling strength is $c = 0.4$. We observe that there are several spurious peaks in the dictionary-based spectrum when $D = 20$ and the DOAs are not correctly identified. In Figs. 5.7(c) and 5.7(d), we plot the MSEs averaged over 200 Monte-Carlo trials as coupling strength c is varied, when $D = 6$ and $D = 20$, respectively. When $D = 20$, most arrays have high MSE when $c > 0.2$. CADiS and \mathbf{w}_2 are observed to have the lowest MSE. These arrays are more robust to mutual coupling because of $w(1) = w(2) = 0$. Unlike the subspace-based DOA estimation algorithms, the accuracy of the dictionary-based approach depends on hyperparameters M and λ that may require appropriate hyperparameter tuning. Furthermore, the dictionary-based approach requires several orders of magnitude higher computation time (Table 5.1).

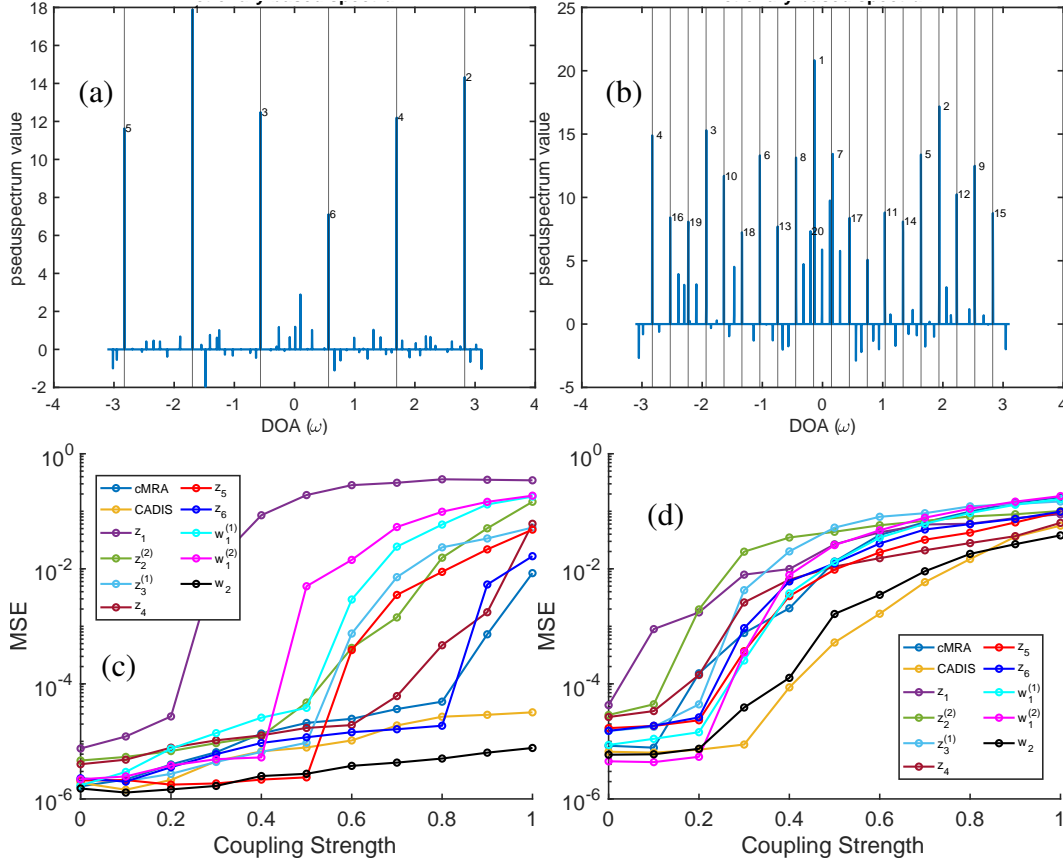


Figure 5.7: Dictionary-based DOA estimation. (a) and (b) show the dictionary-based spectra obtained for array \mathbf{z}_5 with $N = 16$ sensors, $K = 500$, SNR 5 dB, and $c = 0.4$ when $D = 6$ and $D = 20$, respectively. There are several spurious peaks when $D = 20$ and the DOAs are not correctly identified. (c) and (d) show the variation in Monte-Carlo MSE averaged over 200 trials, as the coupling strength c is varied on the X-axis, when $D = 6$ and $D = 20$, respectively. When $D = 20$, most arrays have high MSE when $c > 0.2$.

5.4.4 Covariance Interpolation

As explained in Sec. 5.3, the covariance interpolation algorithmically interpolated the holes at lags such as 1 and 2 so that we still get a large two-sided ULA segment in the coarray. Here, we provide an example to demonstrate its effectiveness. Fig. 5.8 shows the MUSIC spectra obtained using WCNA $\mathbf{w}_1^{(1)}$ with $N = 8$ sensors when SNR is 0 dB, $K = 200$, and $c = 0.2$. When estimating $D = 6$ non-uniformly spaced DOAs using one-sided ULA segment in coarray without interpolation (Fig. 5.8(a)), the MUSIC spectrum is not sharp, and MSE is high. Whereas after interpolation (Fig. 5.8(b)), the MSE is reduced by about two orders of magnitude. Furthermore, it is possible to identify more than $\lfloor L/2 \rfloor$ DOAs after interpolation, as seen in Fig. 5.8(c).

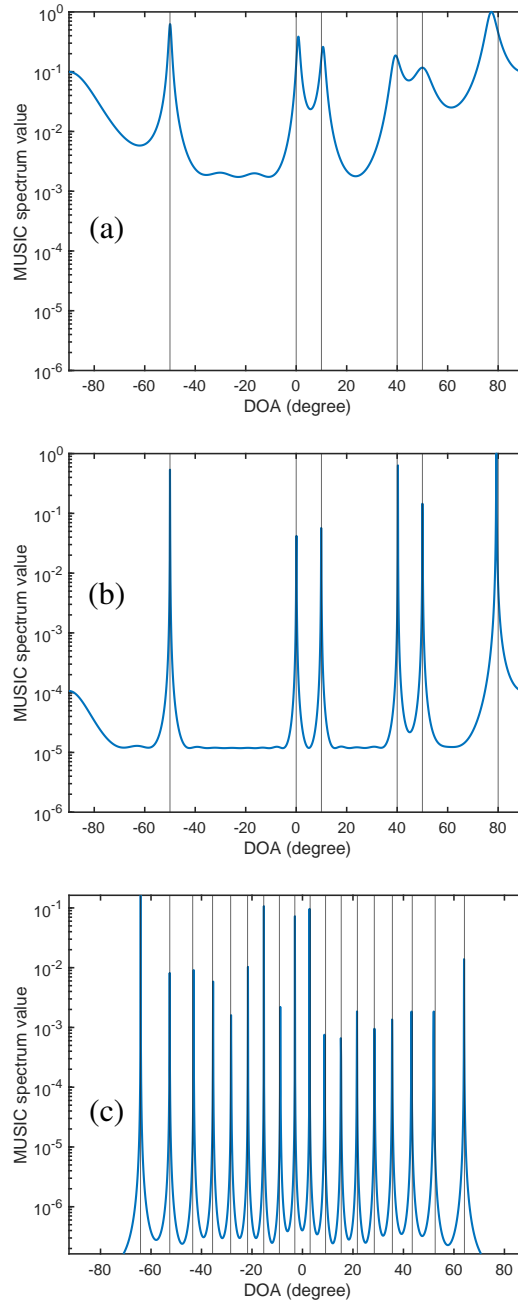


Figure 5.8: MUSIC spectra when estimating DOAs using WCNA $\mathbf{w}_1^{(1)}$ with $N = 8$, SNR 0 dB, $K = 200$, and $c = 0.2$. (a) Estimating $D = 6$ non-uniformly spaced DOAs located at -50° , 10° , 0° , 40° , 50° , and 80° using one-sided ULA segment in coarray, as in the previous chapter, $\text{MSE} = 0.0977$. (b) Estimating the same six DOAs with interpolation, $\text{MSE} = 1.00 \times 10^{-3}$. The MUSIC spectrum has much sharper peaks in this case. (c) Estimating $D = 18$ DOAs uniformly spaced in ω domain from -0.9π to 0.9π with interpolation, $\text{MSE} = 1.23 \times 10^{-4}$.

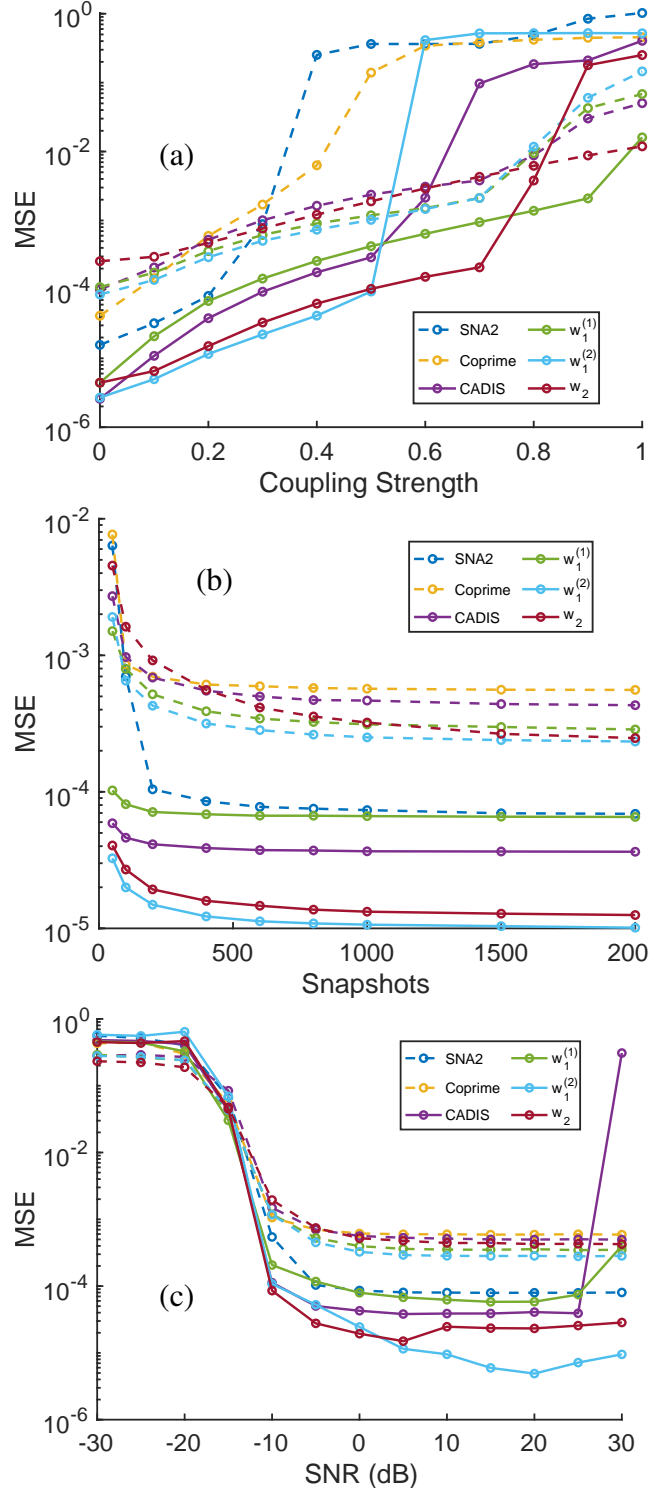


Figure 5.9: MSE when estimating $D = 6$ DOAs spaced uniformly in ω domain from -0.8π to 0.8π . The solid lines denote the MSE obtained after interpolation, and the dashed lines denote the MSE obtained (without interpolation) using the longest ULA segment in the coarray. (a) SNR = 5 dB and $K = 500$, (b) SNR = 5 dB and $c = 0.2$, (c) $K = 500$ and $c = 0.2$.

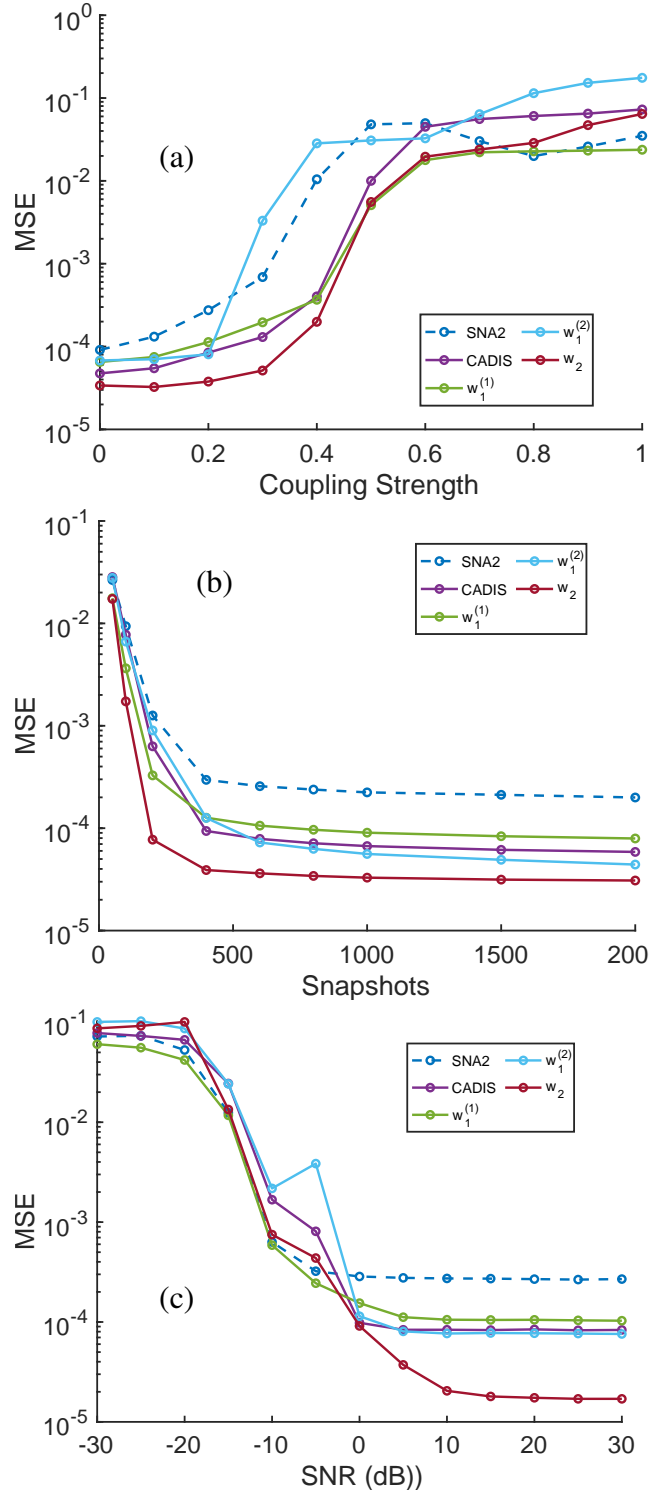


Figure 5.10: MSE when estimating $D = 16$ DOAs spaced uniformly in ω domain from -0.9π to 0.9π . Coarray root-MUSIC is used for SNA, whereas interpolation (Eq. (5.14)) followed by root-MUSIC is used for WCNAs and CADiS. The coprime array with 8 sensors cannot identify 16 DOAs (even after interpolation) and is excluded. (a) SNR = 5 dB and $K = 500$, (b) SNR = 5 dB and $c = 0.2$, (c) $K = 500$ and $c = 0.2$.

Next, we perform Monte-Carlo simulations for both cases when $D < D_m$ (Fig. 5.9) and $D > D_m$ (Fig. 5.10). We consider super nested array (SNA) [39], (extended) coprime array [35], [94], CADiS array [83], and the three WCNA explained in the previous section, all with $N = 8$ sensors. Each simulation point in the plots is averaged over 1000 Monte-Carlo runs with randomly chosen $\{\mathbf{s}[k], \mathbf{n}[k]\}_{k=1}^K$. We also consider the coupling coefficient with values $c_1 = c \cdot e^{j\pi/3}$ and $c_l = (c_1/l)e^{-j(l-1)\pi/8}$ for $2 \leq l \leq 10$. Even in the presence of mutual coupling, we solve the same optimization in Eq. (5.14). CADiS array also has central holes in its coarray [83], so we apply the above interpolation to CADiS as well.

In Fig. 5.9, six DOAs are considered. The solid lines denote the MSE obtained after interpolation (Eq.(5.14)), and the dashed lines denote the MSE obtained when using the longest ULA segment in the difference coarray (possibly one-sided). Out of the three quantities, SNR = 5dB, snapshots $K = 500$, and coupling strength $c = 0.2$, two are kept fixed and the third is varied on the X -axis. In Fig. 5.9(a) for $c \leq 0.5$, the WCNA $\mathbf{w}_1^{(2)}$ and \mathbf{w}_2 after interpolation have the lowest MSEs. Similarly, in Fig. 5.9(b) we observe that for all different numbers of snapshots, $\mathbf{w}_1^{(2)}$ and \mathbf{w}_2 perform the best. Array $\mathbf{w}_1^{(1)}$ performs worse than CADiS here in Fig. 5.9(b) when $c = 0.2$, but in Fig. 5.9(a) we see that $\mathbf{w}_1^{(1)}$ can perform better than CADiS when $c > 0.5$. Lastly, Fig. 5.9(c) shows that as SNR increases $\mathbf{w}_1^{(2)}$ can perform significantly better than \mathbf{w}_2 .

Here, we would like to note that a surprising behavior of an increase in MSE at high SNR is observed when interpolation is used in the presence of mutual coupling. This effect is observed rather dramatically in Fig. 5.9(c) for the CADiS array, but the increase in MSE can still be seen to some extent for WCNA at higher SNRs. This unusual effect seems to be because of the particular choice of the optimization problem (Eq. (5.14)). In our simulations, we have noticed that this unusual behavior can go away by appropriately modifying the optimization problem by dropping/compensating for $\widehat{\mathbf{W}}$ in Eq. (5.14). However, with the modified optimization, overall DOA estimation accuracy is lower using the optimization from Eq. (5.14). Thus, in this chapter, we still use the interpolation technique from [163] even in the presence of mutual coupling, and a careful study of this increase in MSE at high SNR is deferred to future work. Robust interpolation methods that explicitly account for the presence of mutual coupling in their optimization objective should be developed in the future.

Next, in Fig. 5.10, we consider 16 DOAs and perform similar Monte-Carlo simula-

tions as in Fig. 5.9. Note that none of the arrays, except SNA, could have identified 16 DOAs using coarray root-MUSIC without interpolation. Monte-Carlo simulations from Fig. 5.10 demonstrate that it is indeed possible to correctly identify more than $D_m (= \lfloor L/2 \rfloor)$ DOAs with a small MSE, after interpolation. The array \mathbf{w}_2 is observed to perform the best. This demonstrates the ability of WCNA to effectively handle mutual coupling by enforcing $w(1) = w(2) = 0$ and at the same time estimate more DOAs with interpolation.

5.5 Generalized Construction of Weight-Constrained Sparse Arrays

In this section, we propose a generalized construction of WCSAs where, in addition to $O(N)$ aperture and $w(1) = 0$, the coarray weights $w(l)$ for $2 \leq l \leq P$ are small constants that do not increase with N . Here P can be any integer satisfying $2 < P < N$. Similar to the construction procedure of the previously proposed WCSAs, we start with a $(P + 1)$ -sparse ULA of $N - P$ sensors:

$$\mathbb{P}_1 = (P + 1)[[0, N - P - 1]]. \quad (5.17)$$

The rest of the P sensors are appended appropriately to ensure that the resulting array contains a large ULA segment in the coarray. Note that the differences of the form $(P + 1)k$ for each $k \in [[0, N - P - 1]]$ are created by self-differences of sensors within \mathbb{P}_1 . The missing differences of the form $(P + 1)k + q$ where $1 \leq q \leq P$ should be generated when the rest of the P sensors are augmented, while ensuring that no two sensors are at a distance of 1 unit from each other. The following array construction can achieve this:

$$\mathbf{z}_* = \mathbb{P}_1 \cup \mathbb{P}_2 \cup \mathbb{P}_3 \cup \mathbb{P}_4, \quad (5.18)$$

where,

$$\mathbb{P}_2 = -2[[1, \lfloor P/2 \rfloor]], \quad (5.19)$$

$$\mathbb{P}_3 = (P + 1)(N - P - 1) + 1 + 2[[1, \lfloor (P - 1)/2 \rfloor]], \quad (5.20)$$

$$\mathbb{P}_4 = \begin{cases} -P - 2 & \text{if } P \text{ is odd,} \\ (P + 1)(N - P - 1) + P + 2 & \text{if } P \text{ is even.} \end{cases} \quad (5.21)$$

Here, \mathbb{P}_2 is a 2-sparse ULA of $\lfloor P/2 \rfloor$ sensors placed at a distance of 2 units to the left of \mathbb{P}_1 . \mathbb{P}_3 is a 2-sparse ULA of $\lfloor (P - 1)/2 \rfloor$ sensors placed at a distance of 3 units to the right of the \mathbb{P}_1 . Finally, \mathbb{P}_4 is a single sensor placed at a distance of $P + 2$ units on the left of \mathbb{P}_1 if P is odd, and on the right if P is even. Fig. 5.11 shows the

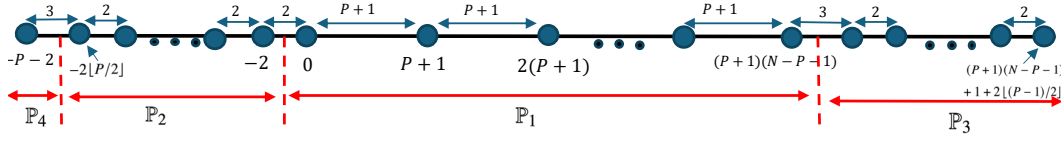


Figure 5.11: The generalized construction of weight-constrained sparse array \mathbf{z}_* from Eq. (5.18) when P is odd.

proposed generalized construction of the weight-constrained sparse arrays when P is odd.

Note that all interelement spacings are ensured to be at least two units by construction, thus ensuring $w(1) = 0$. Also, it can be verified that the aperture of the array \mathbf{z}_* is

$$A = (P + 1)(N - P + 1), \quad (5.22)$$

and thus, for a fixed P , it has $O(N)$ length, as desired. Now we prove that the difference coarray of \mathbf{z}_* contains a large ULA segment starting from lag 2:

Lemma 6. The difference coarray $\mathbb{D}_{\mathbf{z}_*}^+$ of \mathbf{z}_* contains a ULA segment from $L_1 = 2$ to $L_2 = (P + 1)(N - P) - 1$.

Proof. Note that for each $k \in [[0, N - P - 1]]$:

- \mathbb{P}_1 creates self-lags of the form $(P + 1)k$ in the coarray.
- The m -th sensor in \mathbb{P}_2 located at $-2m$ is at a distance of $2m$ from the left end of the $(P + 1)$ -sparse ULA \mathbb{P}_1 . Thus, it creates differences of the form $\{(P + 1)k + 2m\}$ with the sensors from \mathbb{P}_1 .
- Similarly, n -th sensor in \mathbb{P}_3 located at $(P + 1)(N - P - 1) + 2n + 1$ is at a distance of $2n + 1$ from the right end of the $(P + 1)$ -sparse ULA \mathbb{P}_1 . Thus, it creates differences of the form $\{(P + 1)k + 2n + 1\}$ with the sensors from \mathbb{P}_1 .
- Considering all values of m in $[[1, \lfloor P/2 \rfloor]]$ for \mathbb{P}_2 and all values of n in $[[1, \lfloor (P - 1)/2 \rfloor]]$, it can be seen that all differences of the form $\{(P + 1)k + q\}$ where $q \in [[2, P]]$ are generated between the sensors from $\mathbb{P}_2 \cup \mathbb{P}_3$ and \mathbb{P}_1 .
- Finally, the sensor from \mathbb{P}_4 is at a distance of $P + 2$ units from \mathbb{P}_1 (either from the left or right end, depending on whether P is odd or even). Thus it

creates differences of the form $\{(P+1)k + P + 2\} = \{(P+1)(k+1) + 1\}$ for $k \in [[0, N - P - 1]]$ with the sensors from \mathbb{P}_1 .

Combining the above sets of differences for all values of $k \in [[0, N - P - 1]]$, we get that all lags in the range $[[2, (P+1)(N-P) - 1]$ are created in the difference coarray. Thus, the difference coarray $\mathbb{D}_{\mathbf{z}_*}^+$ of \mathbf{z}_* contains a ULA segment from $L_1 = 2$ to $L_2 = (P+1)(N-P) - 1$. \square

Remark. Note that the ULA segment in the coarray of \mathbf{z}_* may extend beyond L_2 for some combinations of N and P . In other words, the coarray $\mathbb{D}_{\mathbf{z}_*}^+$ does not necessarily have a hole at location $L_2 + 1 = (P+1)(N-P)$. We have observed in our examples that for odd values of P , there is indeed a hole in the coarray at lag $(P+1)(N-P)$, whereas, for even values of P , there is no hole at lag $(P+1)(N-P)$. For even values of P , only two holes (at locations 1 and $A-1$) are observed in the coarray $\mathbb{D}_{\mathbf{z}_*}^+$.

Now, to see that the coarray weights $w(l)$ for $2 \leq l \leq P$ do not increase with the number of sensors N , consider the following two observations. First, as N increases, only the size of sparse ULA \mathbb{P}_1 increases, and the relative positions of \mathbb{P}_2 , \mathbb{P}_3 , and \mathbb{P}_4 from the endpoints of sparse ULA \mathbb{P}_1 remain the same. Secondly, all sensors in \mathbb{P}_1 except the first sensor at 0 and the last sensor at location $(P+1)(N-P-1)$ are at least $P+1$ distance apart from any other sensors and hence do not contribute to $w(l)$ for $2 \leq l \leq P$. Combining these two observations, it can be concluded that the coarray weights $w(l)$ for $2 \leq l \leq P$ do not increase with the number of sensors N . Furthermore, the coarray weights $w(l)$ for $2 \leq l \leq P$ remain small constants that do not depend on N . For example, when $4 < P < N-1$, it can be verified that $w(2) = P-2$, $w(3) = 2$, $w(4) = P-4$, and $w(5) = 2$.

For a given number of sensors N , there is an interesting tradeoff in choosing an optimal value of P . Note that both the aperture and L_2 are quadratic functions of P , and are maximized when P is approximately $N/2$. Thus, increasing P increases the aperture and the length of the ULA segment in the coarray up to a certain point. When N is large, increasing P increases the number of coarray weights $w(l)$ for $2 \leq l \leq P$ having small values, thus contributing to a reduction in mutual coupling. However, on the other hand, the values of coarray weights corresponding to even lags increase with increasing P , contributing to an increase in mutual coupling.

Fig. 5.12 illustrates these behaviors. For different numbers of sensors, we plot the array aperture, the number of identifiable DOAs using coarray MUSIC ($D_m =$

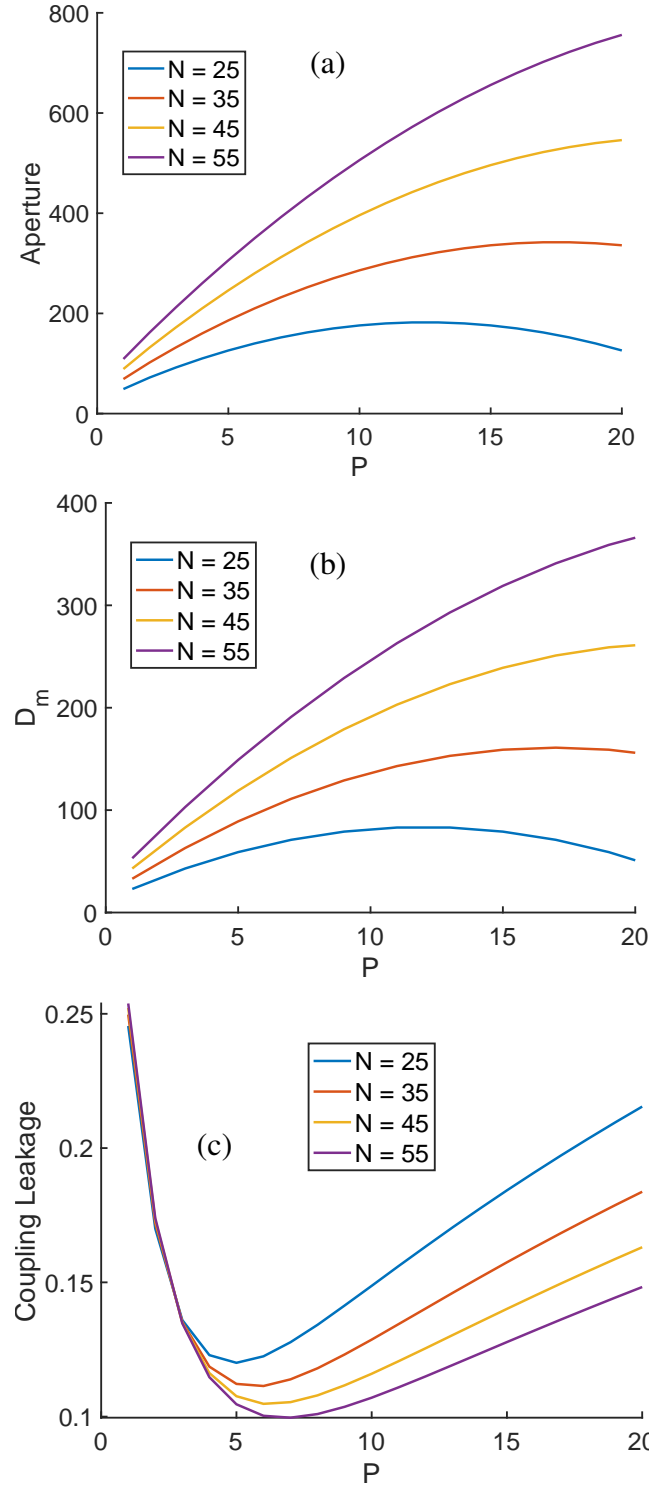


Figure 5.12: The effect of varying the parameter P on the aperture A , maximum number of identifiable DOAs using coarray MUSIC $D_m = \lfloor L/2 \rfloor$, and coupling leakage \mathcal{L} of the proposed generalized construction of WCSA \mathbf{z}_* . The aperture and D_m peak around $P \approx N/2$, and the coupling leakage is lowest for $5 \leq P \leq 7$ when $25 \leq N \leq 55$.

$\lfloor L/2 \rfloor$), and the coupling leakage as P is varied on the X -axis. Coupling coefficients with $B = 1000$ and $|c_1| = 0.3$ are considered to calculate coupling leakage. As expected, the aperture and D_m increase with increasing P up to a point and then decrease. The coupling leakage \mathcal{L} decreases rapidly at smaller values of P and then increases as P is increased further. The optimum P is observed to be between 5 and 8 when $25 \leq N \leq 55$. As we will see in the simulation examples in the next section, generalized WCSAs \mathbf{z}_* with values of P between 5 and 8 indeed have small DOA estimation errors.

5.6 Simulations with Generalized Construction of WCSAs

Now, we compare the performance of the proposed generalized construction of WCSA with previously proposed WCSAs and other sparse arrays through Monte-Carlo simulations. Each simulation point is averaged over 1000 Monte-Carlo runs with randomly chosen $\{\mathbf{s}[k], \mathbf{n}[k]\}_{k=1}^K$. For ULA, direct MUSIC is used (when $D < N$), whereas for all sparse arrays, coarray-ESPRIT is used to estimate DOAs. For the proposed WCSA \mathbf{z}_* , the ULA segment in the coarray from $L_1 = 2$ to $L_2 = (P + 1)(N - P) - 1$ is used. The mutual coupling parameters with $B = 10$, $c_1 = ce^{j\pi/3}$, and $c_l = (c_1/l)e^{-j(l-1)\pi/8}$ are used.

First, we compare the performance of the proposed WCSA \mathbf{z}_* with the WCSAs \mathbf{z}_1 to \mathbf{z}_6 proposed in Sec. 4.3 for estimating $D = 25$ DOAs using $N = 30$ sensors. Note that some of the previously proposed WCSA (\mathbf{z}_4 , \mathbf{z}_5 , and \mathbf{z}_6) have $w(1) = w(2) = 0$, whereas rest of the arrays have only $w(1) = 0$. From Fig. 5.13, we observe that the proposed WCSA \mathbf{z}_* has a smaller MSE than previously proposed WCSA for several different values of P for a wide range of SNR and snapshots and when the coupling strength is lower than 0.5. In particular, the reduction in MSE for \mathbf{z}_* with $P = 8$ is observed to be the most significant.

Next, for a fixed number of sensors $N = 16$, we compare the DOA estimation error of the proposed array \mathbf{z}_* for $4 \leq P \leq 8$ with different well-known sparse arrays from the literature, such as super nested array (SNA) [39], [101], augmented nested array (ANAI-2) [103], extended generalized coprime array (EGCA) [157], displaced dilated nested array (DDNA) [92], MISC arrays [104], minimum redundancy array (MRA) [33], [84], (approximate) constrained MRAs (cMRAs) [84], extended coprime array [35], CADiS array [83], thinned coprime array (TCA) [91], and extended padded coprime array (ePCA) [105]. Out of these arrays, cMRA has $w(1) = 0$, and CADiS can have $w(l) = 0$ for several small values of l . Fig. 5.14 clearly shows

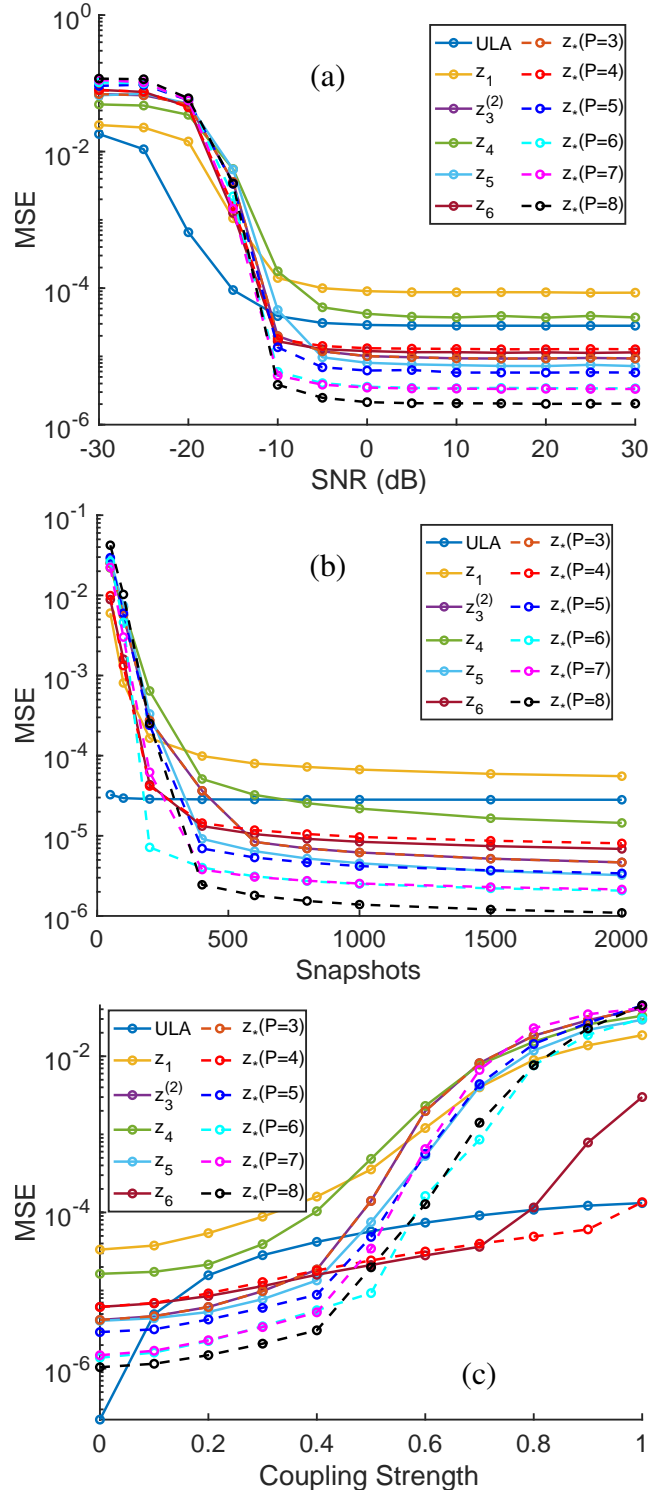


Figure 5.13: Comparing coarray ESPRIT MSE for estimating $D = 25$ DOAs uniformly separated in ω from -0.95π to 0.95π with 30-sensor arrays. (a) $c = 0.3$, $K = 500$, (b) SNR = 5 dB, $c = 0.3$, and (c) SNR = 5 dB, $K = 500$. The proposed WCSA \mathbf{z}_* with several different values of P has lower MSE than previously proposed WCSAs (\mathbf{z}_1 to \mathbf{z}_6) for a wide range of SNR and snapshots when the mutual coupling strength is lower than 0.5.

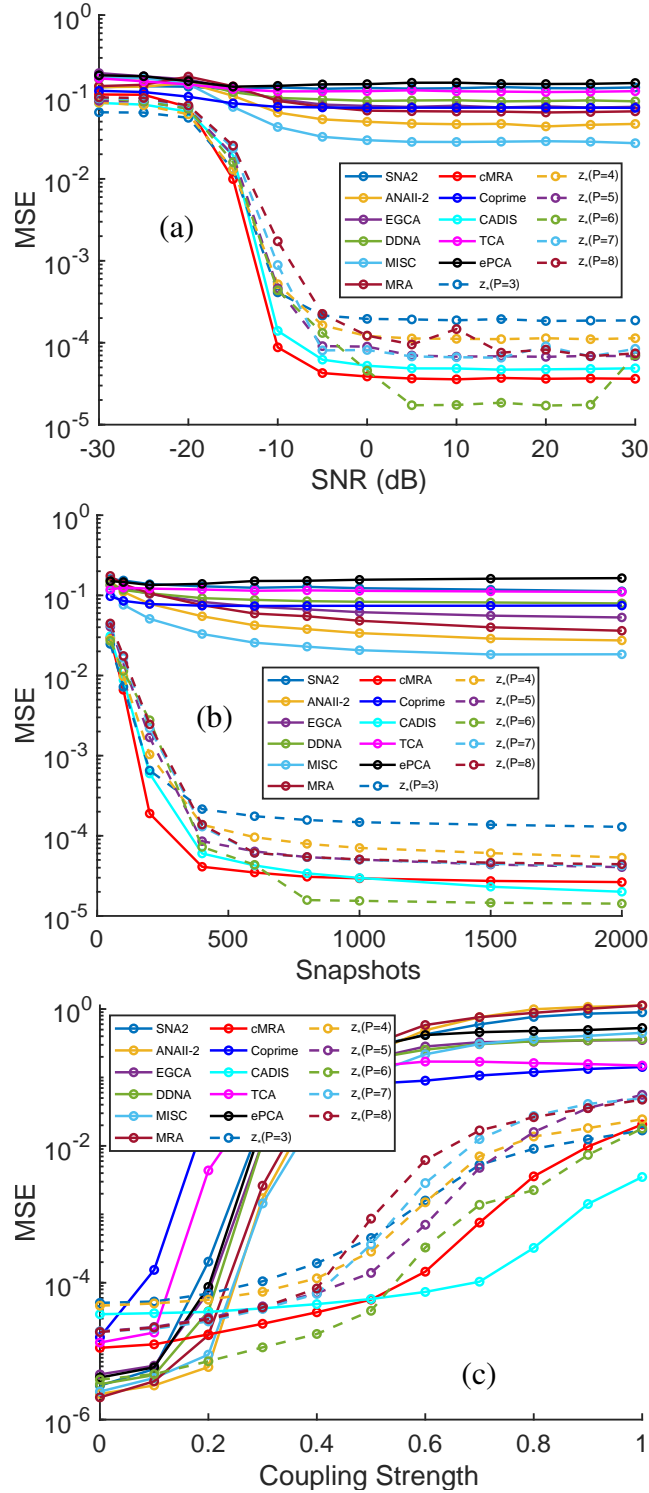


Figure 5.14: Comparing coarray ESPRIT MSE for estimating $D = 20$ DOAs uniformly separated in ω from -0.9π to 0.9π with 16-sensor arrays. (a) $c = 0.4$, $K = 500$, (b) SNR = 5 dB, $c = 0.4$, and (c) SNR = 5 dB, $K = 500$. The proposed WCSA z_* , CADiS, and cMRA are the arrays that have $w(1) = 0$ and they can correctly identify the DOAs with small MSE when the coupling strength is larger than 0.2.

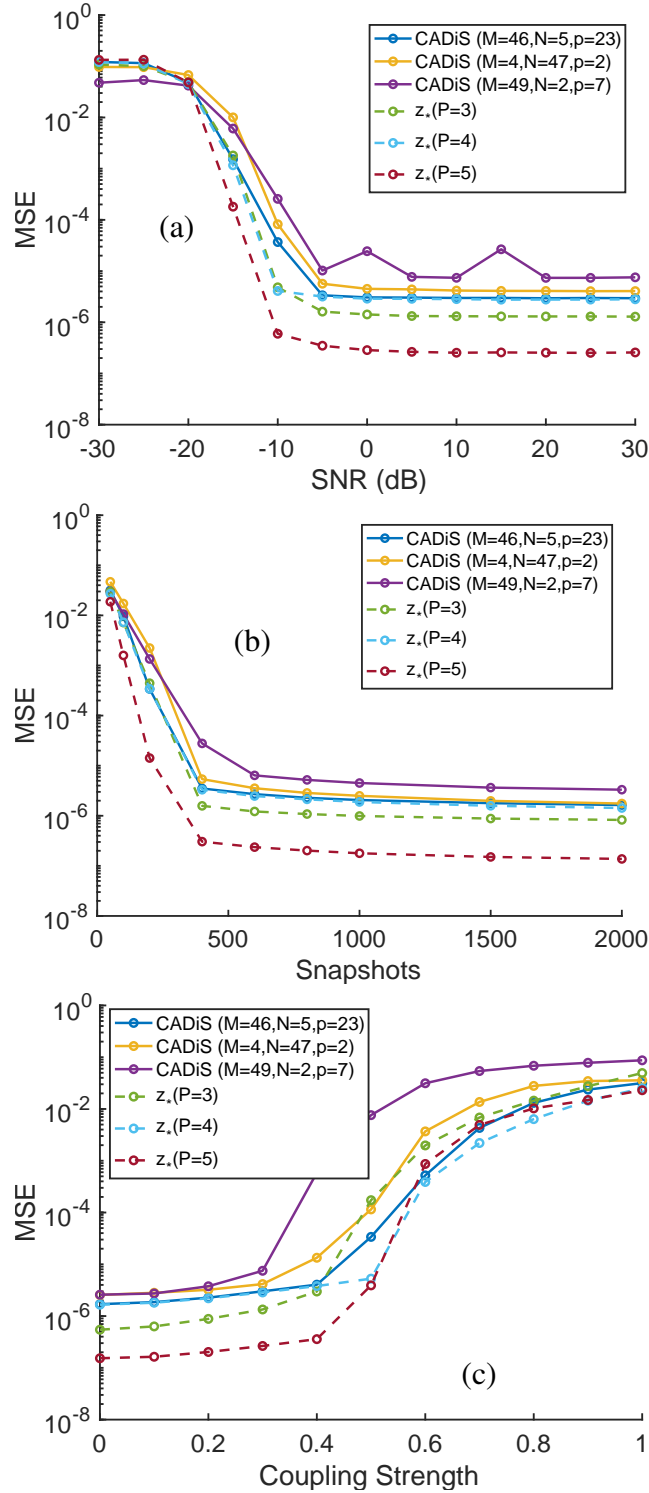


Figure 5.15: Comparing coarray ESPRIT MSE of the arrays with $N = 50$ under the aperture constraint $A \leq 300$ when estimating $D = 25$ DOAs separated uniformly in ω from -0.9π to 0.9π . All possible configurations of CADiS satisfying the constraints are compared with three configurations of the proposed array satisfying the constraints. (a) $c = 0.3$, $K = 500$, (b) SNR = 5dB, $c = 0.3$, (c) SNR = 5dB, $K = 500$. The proposed array z_* with $P = 5$ is observed to have the lowest MSE.

that arrays that do not have $w(1) = 0$ perform poorly when $c > 0.2$. The proposed array can perform even better than cMRA and CADiS in some situations (here, for example, \mathbf{z}_* with $P = 6$).

Similar to MRAs, no closed-form expression is available for cMRAs. Sensor positions are only available for $N < 20$ [84] and are found by extensive numerical search. CADiS can have several possible configurations for a given number of sensors, however, the optimal configuration parameters (M , N , and p from [83]) may be difficult to choose without enumerating all possibilities. CADiS is also observed to have more holes in the coarray, whereas the proposed arrays are observed to have either 2 or 4 holes in the coarray. Furthermore, the CADiS array with N sensors has an aperture of $O(N^2)$ length, whereas the proposed WCSA has an aperture of $O(N)$ length. Thus, the proposed array may have an advantage in the presence of an aperture constraint, as demonstrated by the next simulation.

In Fig. 5.15, we consider the situation where $N = 50$ and a maximum aperture of length 300 is available. There are three possible configurations of CADiS satisfying this constraint, and they are compared with three configurations of the proposed WCSA satisfying the constraints. It is observed that the proposed WCSAs have lower MSE than possible CADiS configurations and, in particular, the proposed WCSA \mathbf{z}_* with $P = 5$ performs the best. Furthermore, the lengths of the ULA segments in the coarray of the proposed array with $P = 4$ and $P = 5$ (228 and 268, respectively) are equal to or longer than those of all three CADiS configurations (228, 144, and 99). This demonstrates that for a given number of sensors N and available aperture A , the proposed array \mathbf{z}_* can place the sensors in the available aperture better than CADiS and produce a longer ULA segment in the difference coarray.

5.7 Concluding Remarks

In the first part of this chapter, we proposed coarray matrix augmentation and covariance interpolation for effectively using the non-central ULA segments in the coarrays of weight-constrained arrays. We proved that the ideal augmented coarray matrix \mathbf{R}_{aug} has the same column span as the augmented manifold matrix \mathbf{A}_{aug} , and proposed applying root-MUSIC and ESPRIT using the augmented coarray matrix. Simulation results demonstrated that augmented root-MUSIC significantly improves the accuracy of DOA estimation compared to using only the one-sided ULA segment of arrays that have central holes in their difference coarrays. Although augmented

ESPRIT showed only marginal improvement in DOA estimation MSE, it has a much lower computation time, as it does not involve a computationally dominant root-finding step.

With covariance interpolation, we experimentally demonstrated that the arrays could identify up to twice the number of DOAs compared to what is possible using only the one-sided ULA segment in the difference coarray. Even when the number of DOAs is small, the MSE obtained after interpolation is over an order of magnitude smaller than that without interpolation. This mitigates the disadvantage of the non-central ULA segments in the coarrays of WCNA, but retains the advantage of robustness to mutual coupling because of $w(1) = 0$ or $w(1) = w(2) = 0$.

In the simulations, we observed an undesirable increase in MSE at high SNRs when the interpolation algorithm in Eq. (5.14) is used. In the future, further investigations are required to understand the reasons behind this behavior. Furthermore, a ‘coupling-aware’ interpolation algorithm that can incorporate the coupling matrix \mathbf{C} in the optimization objective would be of interest. One possible way of doing this is considered in Sec. 6.6. Solving the optimization problem for interpolation is computationally heavy and not suitable for real-time applications, especially when the number of sensors N is large. Thus, developing computationally efficient algorithms to effectively utilize the arrays with non-central ULA segments in coarrays would be beneficial.

In the last part of this chapter, we proposed a generalized construction for weight-constrained sparse arrays that have $\mathcal{O}(N)$ aperture and $w(1) = 0$. Furthermore, for a given integer $2 < P < N$, the proposed array configuration satisfies the property that $w(l) \leq P-2$ for $2 \leq l \leq P$. As these coarray weights do not grow with an increasing number of sensors N , the impact of mutual coupling on DOA estimation is reduced. We proved that the difference coarray of the proposed construction \mathbf{z}_* contains a large ULA segment from lag $L_1 = 2$ to $L_2 = (P+1)(N-P) - 1$. Monte-Carlo simulations demonstrated that the proposed arrays can perform better than WCSAs proposed in Chapter 4 and several well-known sparse arrays from the literature. Furthermore, for a fixed P , the aperture of the proposed weight-constrained sparse array \mathbf{z}_* increases linearly with N , which is helpful under the aperture constraint.

In the future, it would be of interest to develop generalized constructions of WCSAs such that $w(1) = w(2) = 0$ and $w(l)$ for $3 \leq l \leq P$ are small constants independent of N . However, we have observed that with the currently adopted construction mechanism of dilating a ULA and appending it with a few additional sensors, this

is possible only when $P < 5$. A similar limitation, because of the currently adopted construction mechanism, is also encountered for constructing weight-constrained arrays with $w(1) = w(2) = w(3) = 0$. A different array construction would be required to achieve the above.

ON THE EFFICIENCY OF COARRAY-DOMAIN DOA ESTIMATION ALGORITHMS

6.1 Introduction

In Chapter 2, we briefly reviewed the coarray-domain DOA estimation algorithms. The key step for estimating the DOAs via coarray-domain processing is the construction of a ‘coarray covariance matrix’ based on the known entries in the sample covariance matrix of the array output ($\widehat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}$). First, array output correlations are estimated at each lag l in the difference coarray according to Eq. (2.22). Then, the coarray covariance matrix is usually formed either by employing spatial smoothing to obtain the matrix $\widehat{\mathbf{R}}_{SS}$ (Eq. (2.31)) or through a direct augmentation approach to obtain the matrix $\widehat{\mathbf{R}}_{DA}$ (Eq. (2.33)). When the difference coarray $\mathbb{D}_{\mathbf{z}}$ of the array \mathbf{z} contains central ULA segment of length $2L - 1$ from lags $l = -(L - 1)$ to $l = L - 1$, the size of these matrices is $L \times L$. Subsequently, popular subspace-based DOA estimation algorithms such as MUSIC, root-MUSIC, and ESPRIT are applied using $\widehat{\mathbf{R}}_{SS}$ or $\widehat{\mathbf{R}}_{DA}$. Because the two matrices are related to each other [93] by the relation in Eq. (2.34), the same DOA estimates are obtained by using either of the two matrices.

In Chapter 2, we also discussed some properties and trends of CRB and analytical MSE expressions for direct-MUSIC and coarray-MUSIC derived in [86], [87], [122]. It was noted [87] that the MSE of DOA estimation using coarray-MUSIC with sparse arrays does not converge to CRB, even asymptotically, when more than one source is present. This means that coarray-MUSIC is **inefficient**, even asymptotically with a large number of snapshots and SNR. This contrasts with the direct MUSIC, which does achieve the CRB asymptotically. However, these observations in [87] regarding coarray-MUSIC inefficiency were made for sparse arrays (nested array, coprime array, and MRA), but not for a ULA. Although there is no advantage in using coarray-MUSIC over direct-MUSIC for ULA (in terms of degrees of freedom), such analysis is useful to gain insights into the efficiency of coarray-MUSIC. Furthermore, [87] studied the effect of SNR on the efficiency of sparse arrays, however, the effect of the number of snapshots on efficiency was not considered. It is worth mentioning that some recent papers have pointed out

the asymptotic inefficiency of coarray-MUSIC for DOA estimation and suggested alternative coarray-based algorithms. Among these alternatives are the weighted least squares (WLS) [90] method, which provably achieves CRB asymptotically, an augmented Toeplitz matrix reconstruction approach [89], and a weighted proxy covariance matrix construction approach [124].

Thus, in the first part of this chapter, we conduct a series of experiments to investigate when and how the coarray-MUSIC loses its efficiency for a simple scenario involving a ULA and fewer sources than the number of sensors. In this case, direct-MUSIC already achieves CRB asymptotically, as the number of snapshots increases. However, we observe that coarray-MUSIC suffers from significant performance loss even in this case, similar to what has been observed for sparse arrays. It is worth noting that the only difference between coarray-MUSIC and direct-MUSIC for ULA is the diagonal averaging of the array output covariance matrix, and hence, this observation is surprising. This observation for ULA suggests that the loss of efficiency is an artifact of the coarray-MUSIC algorithm, rather than the array sparsity. We also consider two different variations of the coarray covariance matrix, namely, the ‘tall’ and ‘fat’ coarray covariance matrices that can be used for DOA estimation instead of the standard square coarray covariance matrix.

Through further experiments, we encounter some surprising observations. We observe that the coarray-MUSIC MSE for two-source situations does not decrease monotonically with the separation between the two DOAs. In another experiment, we analyze the relative importance of each array output correlation. We start with the ideal array output correlations and perturb one of the entries at a time to see its effect on DOA estimation MSE. Our findings show that perturbing array output correlations at a lag around $N/2$ results in the largest MSE. These surprising observations provide new insights into the complex relationship of coarray-MUSIC MSE on several parameters, such as array geometry, DOA separation, and accuracy of the estimated array output correlations.

In the second part of this chapter, we further examine the crucial step of coarray-domain matrix formation to mitigate the source of inefficiency in coarray-domain DOA estimation. Note that the $L \times L$ matrices $\hat{\mathbf{R}}_{SS}$ and $\hat{\mathbf{R}}_{DA}$ act as substitutes for the output covariance matrix that a fictitious ULA with L physical sensors would have produced. So instead of constructing the matrix using spatial smoothing or direct augmentation, we use the Toeplitz covariance matrix construction approach proposed in [163] to estimate the larger covariance matrix of the fictitious ULA.

This approach formulates an optimization problem to construct a large Toeplitz covariance matrix, upon which subspace-based DOA estimation algorithms can be applied. We demonstrate that by using this approach for the covariance matrix construction, it is possible to obtain large improvements in the DOA estimation MSE. Our numerical experiments demonstrate that when $D < N$, the DOA estimation MSE is close to CRB, and does not saturate at large SNR, unlike that of the coarray-MUSIC. When $D > N$, MSE can be much smaller than that of coarray-MUSIC, and, in some situations, can even be better than the CRB, because of the potential bias in the DOA estimates produced by this method. For the situations where mutual coupling is present, and the coupling matrix \mathbf{C} is known (or can be reliably estimated), we propose a modification to the Toeplitz covariance matrix construction approach. The numerical simulations show that with this modification to incorporate the knowledge of \mathbf{C} , the DOA estimation MSE can be very close to the CRB, and in some situations can even be smaller than the CRB, similar to the case without mutual coupling.

We would like to note that several optimization problems with a similar flavor to that in [163] have been proposed in the literature for coarray interpolation [160]–[162], [164], [165]. However, to the best of our knowledge, these methods have not been investigated in terms of the efficiency of DOA estimates obtained using them. Our experiments demonstrate that even for arrays such as ULA, nested arrays, and MRAs that do not have holes in the difference coarray and hence there is no ‘need’ for ‘interpolating’ the coarray, constructing the covariance matrix according to the optimization in [163] can greatly improve the DOA estimation MSE. Importantly, unlike these other optimization problems for coarray interpolation, the method from [163] involves a key step of ‘error whitening’. As we will see later in the simulations, the whitening step is observed to be a major contributor to the reduction of MSE. In this chapter, we want to establish a simpler alternative to some of the recently proposed methods from the literature, such as [90], [172], and highlight the importance of developing other ways of constructing a coarray covariance matrix for efficient DOA estimation using sparse arrays.

Chapter Outline: Sec. 6.2 investigates coarray-MUSIC efficiency for ULA. Following that, in Section 6.3, we explain the tall and fat variations of the coarray covariance matrix, along with examples of DOA estimation using these matrices. Next, in Sec. 6.4, we describe the Toeplitz covariance matrix construction approach and explain the key features of this approach that contribute to the improvement in

DOA estimation MSE. Sec. 6.5 provides simulations with different sparse arrays that demonstrate the effectiveness of this approach in reducing the DOA estimation MSE. Sec. 6.6 proposes a modification to the Toeplitz covariance matrix construction approach to account for the presence of mutual coupling, and provides simulations with different sparse arrays. Finally, Sec. 6.7 concludes this chapter. The content from Sec. 6.2 to Sec. 6.5 of this chapter is based on papers [158], [173].

6.2 Efficiency of Coarray-MUSIC for ULA

In previous investigations on the efficiency of coarray-MUSIC, [87], [124], one question that remained unexplored was whether the coarray-MUSIC approach achieves CRB at least for the ULA. Since the N -sensor ULA can only identify up to $N - 1$ sources unambiguously using second-order statistics [1], we will focus on the case where $D < N$. The difference coarray \mathbb{D} of the ULA is comprised of all lags from $-(N - 1)$ to $N - 1$. Therefore, in the case of ULA, transitioning to the coarray domain does not increase degrees of freedom. As direct (element space) MUSIC already asymptotically achieves CRB when $D < N$, the goal here is not to find a better algorithm but to gain further insight into when and how the coarray-MUSIC loses its efficiency. Thus, the first part of this chapter experimentally investigates the asymptotic efficiency of coarray-MUSIC for ULA when $D < N$.

For the N -sensor ULA, the correlations $\hat{R}(l)$ from Eq. (2.22) can be expressed as

$$\hat{R}(l) = \frac{1}{N - |l|} \sum_{i-j=l} [\hat{\mathbf{R}}_{\mathbf{xx}}]_{i,j}. \quad (6.1)$$

Thus, for the case of ULA, the coarray covariance matrix $\hat{\mathbf{R}}_{DA}$ (Eq. (2.33)) formed for coarray-MUSIC is simply the ‘Toeplitzized’ (diagonally averaged) version of $\hat{\mathbf{R}}_{\mathbf{xx}}$.

For the ULA, the ideal covariance matrix $\mathbf{R}_{\mathbf{xx}}$ (Eq. (2.6)) is Toeplitz when the sources are uncorrelated. Thus, intuitively it may seem that the Toeplitzization of $\hat{\mathbf{R}}_{\mathbf{xx}}$ to form $\hat{\mathbf{R}}_{DA}$ would not worsen the results, but the opposite is true. We performed a Monte-Carlo experiment for an 11-sensor ULA ($N = 11$) to estimate six DOAs at $-60^\circ, -40^\circ, 0^\circ, 15^\circ, 40^\circ$, and 60° . The experimental MSE (averaged over 1000 Monte-Carlo runs) and analytical MSE are compared with CRB for both direct-MUSIC (using $\hat{\mathbf{R}}_{\mathbf{xx}}$) and coarray-MUSIC (using $\hat{\mathbf{R}}_{DA}$). From Fig. 6.1 we observe that experimental (Monte-Carlo) MSE matches analytical MSE for both direct-MUSIC and coarray-MUSIC. However, the direct-MUSIC MSE practically equals

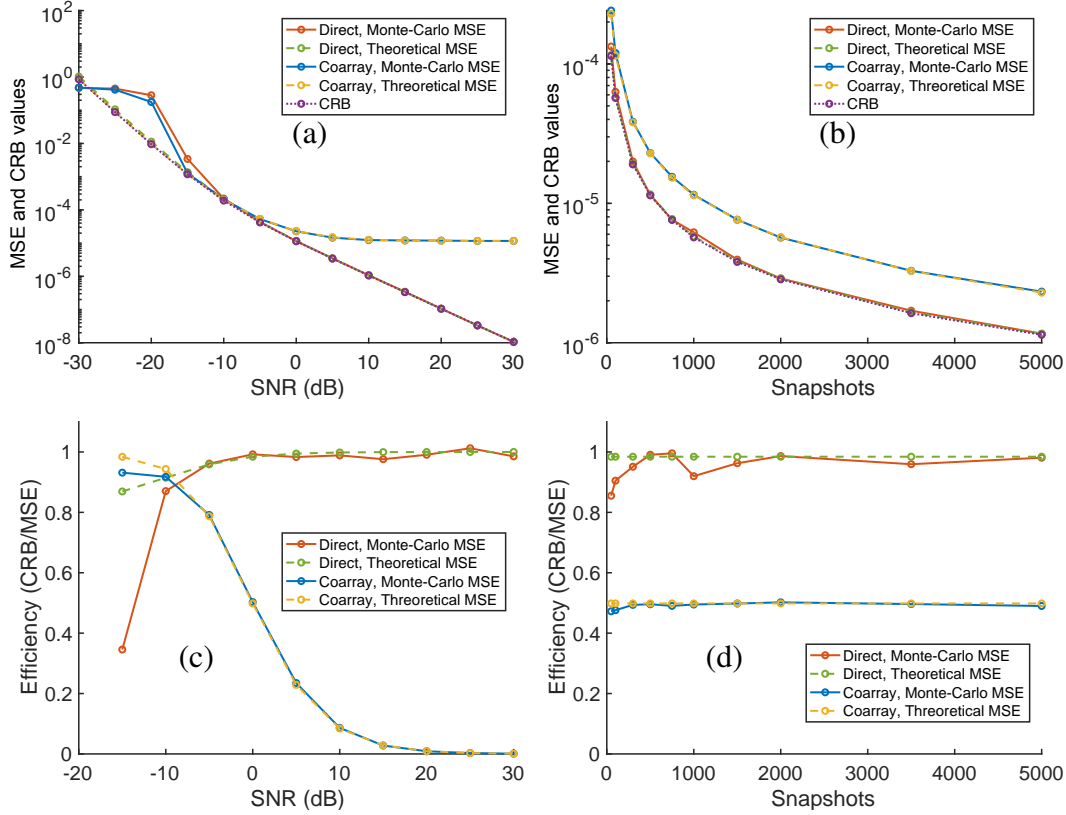


Figure 6.1: Experimental (Monte-Carlo) MSE, theoretical MSE [87], [122], and CRB [86] for direct-MUSIC (using $\hat{\mathbf{R}}_{xx}$) and coarray-MUSIC (using $\hat{\mathbf{R}}_{DA}$) when (a) SNR is varied when the number of snapshots $K = 500$, (b) number of snapshots is varied when SNR = 0 dB. Corresponding efficiency (Eq. (2.35)) plots are shown in (c) and (d) respectively.

CRB, whereas the coarray-MUSIC MSE does not approach CRB asymptotically with a large SNR or a large number of snapshots.

In Fig. 6.1(a), we see that the MSE saturates when the number of snapshots is fixed and SNR is varied, but the CRB doesn't. This is similar to what has been observed in the case of sparse arrays when $D < N$ in [87]. Additionally, Fig. 6.1(c) shows that the efficiency of coarray-MUSIC even for ULA goes to zero at higher SNR. However, we would also like to note an important detail that the efficiency of coarray-MUSIC is observed to be better than direct-MUSIC when SNR is smaller than -10 dB. Furthermore, in Fig. 6.1(b), we observe that for the other case when the number of snapshots is varied keeping SNR fixed, the coarray-MUSIC MSE does not approach CRB even for a large number of snapshots, although neither MSE nor CRB saturates in this case.

The interesting thing to note here is that for the ULA, the only difference between

direct-MUSIC and coarray-MUSIC is the diagonal averaging of $\widehat{\mathbf{R}}_{\mathbf{xx}}$. However, the DOA estimation performance changes drastically. Even though the ideal array output covariance $\mathbf{R}_{\mathbf{xx}}$ is Toeplitz, the Toeplitzization of the observed $\widehat{\mathbf{R}}_{\mathbf{xx}}$ adversely affects DOA estimation performance.

6.3 Two Variations of Coarray Covariance Matrix

From the previous section, we can conclude that the inefficiency in DOA estimation with sparse arrays using coarray-MUSIC is not due to the sparse arrays themselves but rather an issue with the coarray-MUSIC algorithm. Although there are some other recently proposed coarray-based algorithms that perform better than coarray-MUSIC, they are more complex and require solving an optimization problem, which increases computational complexity. Examples of such algorithms are the weighted least squares method [90], which can achieve CRB asymptotically, the Toeplitz matrix reconstruction approach [89], and the weighted proxy covariance matrix construction approach [124]. As coarray-MUSIC is a simple modification of the direct-MUSIC algorithm, it remains a popular choice for coarray-based DOA estimation due to its simplicity.

In this section, we explore alternatives to the standard coarray covariance matrix $\widehat{\mathbf{R}}_{DA}$. Recall that, when the difference coarray \mathbb{D} contains a uniform segment from $-(L-1)$ to $L-1$, the matrix $\widehat{\mathbf{R}}_{DA}$ is an $L \times L$ square Toeplitz matrix. Performing MUSIC with $\widehat{\mathbf{R}}_{DA}$ produces a noise subspace of dimension $(L-D)$ in \mathbb{C}^L . The accuracy of DOA estimation depends on the accuracy of the noise subspace estimate. The two alternatives for $\widehat{\mathbf{R}}_{DA}$ that we examine are of different dimensions and provide a noise subspace in a different space than \mathbb{C}^L . Since we assume the number of sources D to be known, we can construct either a tall matrix ($\widehat{\mathbf{R}}_{DA}^{tall}$) or a fat matrix ($\widehat{\mathbf{R}}_{DA}^{fat}$), as explained next.

$\widehat{\mathbf{R}}_{DA}^{tall}$ is a $(2L-D) \times D$ tall Toeplitz matrix with entries

$$[\widehat{\mathbf{R}}_{DA}^{tall}]_{i,j} = \widehat{R}(i-j+D-L). \quad (6.2)$$

A full singular value decomposition (SVD) of $\widehat{\mathbf{R}}_{DA}^{tall}$ gives

$$\widehat{\mathbf{R}}_{DA}^{tall} = \widehat{\mathbf{U}}^{tall} \widehat{\mathbf{\Sigma}}^{tall} \left(\widehat{\mathbf{V}}^{tall} \right)^H, \quad (6.3)$$

where $\widehat{\mathbf{U}}^{tall}$ is a $(2L-D) \times (2L-D)$ matrix with orthonormal columns and $\widehat{\mathbf{\Sigma}}^{tall}$ is a $(2L-D) \times D$ matrix containing singular values arranged in descending order along the principal diagonal. It can be shown that under ideal conditions (large

snapshots and $\sigma_n = 0$), the first D columns of $\widehat{\mathbf{U}}^{tall}$ will have the same column span as the $(2L - D) \times D$ matrix \mathbf{A}_v^{tall} with entries

$$[\mathbf{A}_v^{tall}]_{k,l} = e^{j\omega_l k}. \quad (6.4)$$

The remaining $(2L - 2D)$ columns of $\widehat{\mathbf{U}}^{tall}$ approximate the ‘noise subspace’, which is the orthogonal complement of \mathbf{A}_v^{tall} . The noise subspace can be used to obtain an estimate of the DOAs $\{\omega_i\}_{i=1}^D$ through the use of the MUSIC spectrum or by rooting the root-MUSIC polynomial. Note that by obtaining the SVD of $\widehat{\mathbf{R}}_{DA}^{tall}$, we get a $(2L - 2D)$ -dimensional noise subspace in \mathbb{C}^{2L-D} . This is in contrast with the standard coarray-MUSIC, which provides $(L - D)$ -dimensional noise subspace in \mathbb{C}^L . DOA estimation using $\widehat{\mathbf{R}}_{DA}^{tall}$ will be called ‘tall-MUSIC’ or ‘tall root-MUSIC’.

Similarly, we can construct a $(D + 1) \times (2L - D - 1)$ dimensional fat Toeplitz matrix $\widehat{\mathbf{R}}_{DA}^{fat}$ with entries

$$[\widehat{\mathbf{R}}_{DA}^{fat}]_{i,j} = \widehat{R}(i - j + L - D - 1). \quad (6.5)$$

Applying SVD to the matrix $\widehat{\mathbf{R}}_{DA}^{fat}$ gives

$$\widehat{\mathbf{R}}_{DA}^{fat} = \widehat{\mathbf{U}}^{fat} \widehat{\mathbf{\Sigma}}^{fat} \left(\widehat{\mathbf{V}}^{fat} \right)^H, \quad (6.6)$$

where $\widehat{\mathbf{U}}^{fat}$ is a $(D + 1) \times (D + 1)$ matrix, and entries of $\widehat{\mathbf{\Sigma}}^{fat}$ are arranged in descending order. The first D columns of $\widehat{\mathbf{U}}^{fat}$ provide an estimate of the column span of $(D + 1) \times D$ matrix with entries $[\mathbf{A}_v^{fat}]_{k,l} = e^{j\omega_l k}$. The last column of $\widehat{\mathbf{U}}^{fat}$, which is orthogonal to the rest of the columns of $\widehat{\mathbf{U}}^{fat}$, is a vector approximating the ‘noise subspace’ in \mathbb{C}^{D+1} . Thus, through SVD of $\widehat{\mathbf{R}}_{DA}^{fat}$, we obtain a single ‘noise vector’ in a much lower dimension $\mathbb{C}^{(D+1)}$. DOA estimation using $\widehat{\mathbf{R}}_{DA}^{fat}$ will be called ‘fat-MUSIC’ or ‘fat root-MUSIC’.

6.3.1 Computational Complexity

The two variations of coarray-MUSIC presented above have an advantage in terms of computational complexity. For standard coarray-MUSIC, the computational complexity for the eigendecomposition of $\widehat{\mathbf{R}}_{DA}$ is $O(L^3)$. However, the computational complexity of finding SVD of an $m \times n$ matrix is $O(\min(mn^2, nm^2))$. Therefore, the complexity of computing SVD of $\widehat{\mathbf{R}}_{DA}^{tall}$ is $O((2L - D)D^2) = O(LD^2)$, and the complexity of computing SVD of $\widehat{\mathbf{R}}_{DA}^{fat}$ is also $O(LD^2)$. Obtaining DOA estimates by rooting the root-MUSIC polynomial adds an additional component to the overall computational complexity. The polynomial for tall root-MUSIC is of degree $2(2L - D) - 2$, and for fat root-MUSIC, it is of degree $2D$. The computational

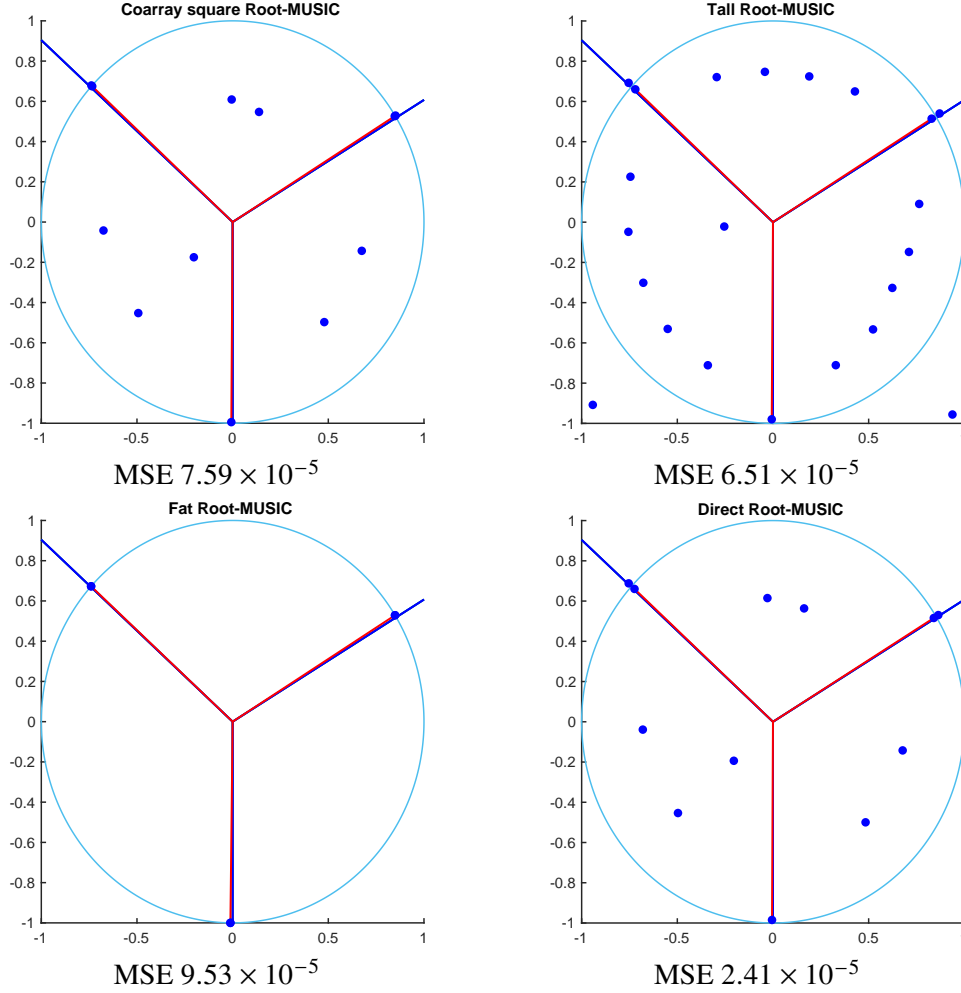


Figure 6.2: Estimating three DOAs -30° , 10° , and 50° with an 11-sensor ULA, 200 snapshots, and SNR 0 dB. (a) Standard coarray-MUSIC (using $\hat{\mathbf{R}}_{DA}$), (b) tall-MUSIC (using $\hat{\mathbf{R}}_{DA}^{tall}$), (c) fat-MUSIC (using $\hat{\mathbf{R}}_{DA}^{fat}$), and (d) direct-MUSIC (using $\hat{\mathbf{R}}_{xx}$). The phases (angle with X -axis) of the blue and the red lines indicate the true DOAs ω_i and estimated DOAs $\hat{\omega}_i$ respectively.

complexity of rooting a degree- n polynomial is usually $O(n^3)$, but some efficient algorithms with $O(n^2 \log n)$ computational complexity are also available [174]. Thus, accounting for the matrix decomposing and polynomial rooting, fat root-MUSIC has the best overall computational complexity. In our MATLAB simulations, we observed that fat-MUSIC has a runtime reduction of over 8-fold compared to standard coarray-MUSIC when $N = 30$ and $D = 2$. Moreover, it has an over 35-fold reduction in runtime when $N = 60$ and $D = 6$.

6.3.2 Simulation Examples with Tall and Fat Coarray-MUSIC

First, we consider three DOAs and an 11-sensor ULA. The ground-truth values of the DOAs to be estimated are -30° , 10° , and 50° . To estimate the DOAs, we use

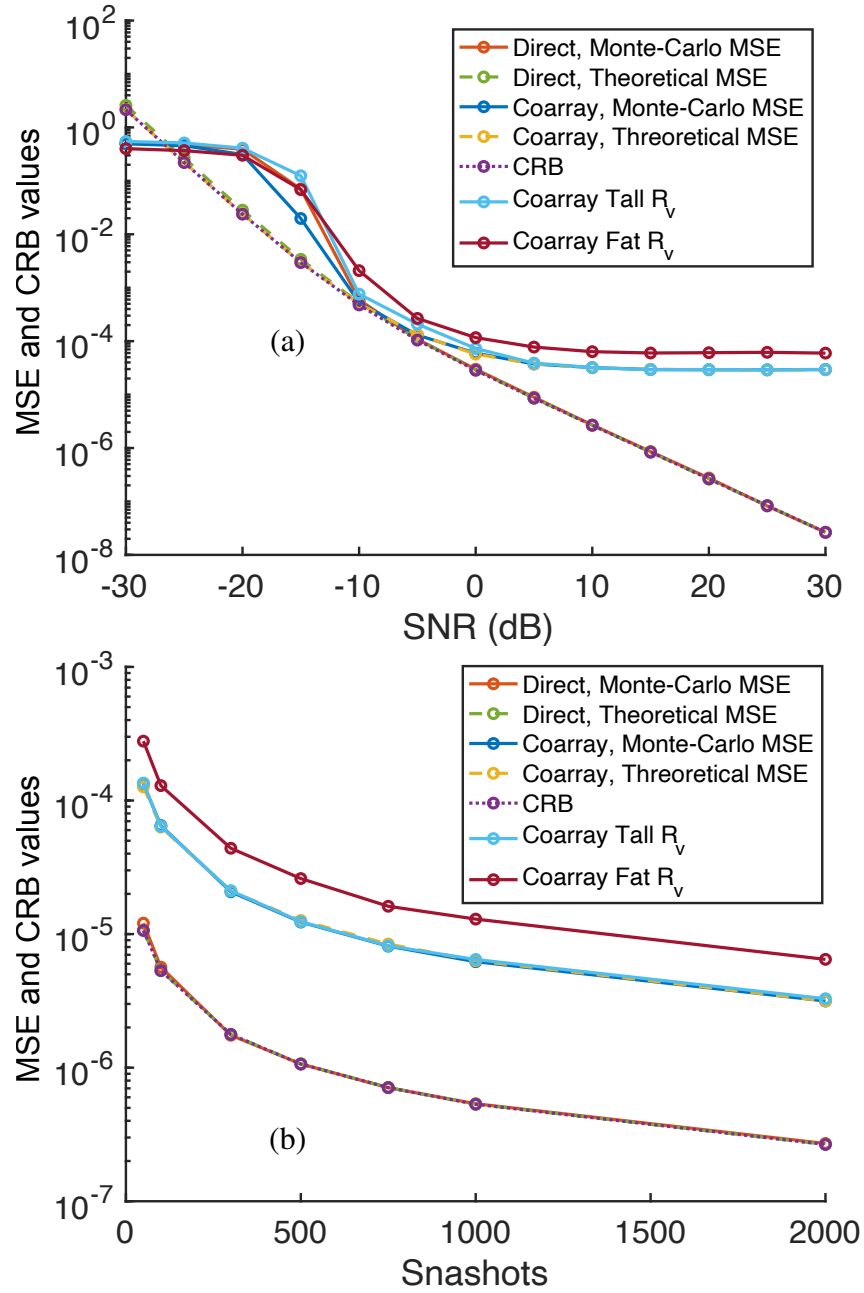


Figure 6.3: Estimating six DOAs $-60^\circ, -40^\circ, 0^\circ, 15^\circ, 40^\circ$, and 60° with an 11-sensor ULA. Direct-MUSIC, coarray-MUSIC, tall-MUSIC, and fat-MUSIC are considered. (a) SNR is varied when the number of snapshots $K = 200$, (b) the number of snapshots is varied when SNR = 10 dB. Tall-MUSIC performs at par with coarray-MUSIC, and fat-MUSIC performs slightly worse. Fat-MUSIC has the best computational complexity.

four methods: standard coarray-MUSIC ($\hat{\mathbf{R}}_{DA}$), tall-MUSIC ($\hat{\mathbf{R}}_{DA}^{tall}$), fat-MUSIC ($\hat{\mathbf{R}}_{DA}^{fat}$), and direct-MUSIC ($\hat{\mathbf{R}}_{xx}$). The simulation involves 200 snapshots and an SNR of 0 dB. Fig. 6.2 shows the root-MUSIC plots obtained with each method. We observe that ‘tall root-MUSIC’ has more zeros than the standard coarray root-MUSIC, whereas the ‘fat root-MUSIC’ does not have any extra zeros apart from those corresponding to the DOAs. Although direct-MUSIC performs the best, ‘tall root-MUSIC’ performs slightly better than standard coarray root-MUSIC in this example.

Next, we perform a Monte-Carlo simulation to compare the DOA estimation performance of tall-MUSIC and fat-MUSIC with that of coarray-MUSIC. We consider an 11-sensor ULA and six DOAs -60° , -40° , 0° , 15° , 40° , and 60° . The simulation is conducted with 200 snapshots in Fig. 6.3(a), where SNR is varied, and with SNR of 10 dB in Fig. 6.3(b), where the number of snapshots is varied. We observe that the tall-MUSIC performs on par with the standard coarray-MUSIC, but no better on average, while the fat-MUSIC performs slightly worse. Hence, using the tall and fat variation of the coarray covariance matrix does not seem to enhance the DOA estimation performance fundamentally. However, tall-MUSIC $\hat{\mathbf{R}}_{DA}^{tall}$ can be utilized as a viable alternative to the standard coarray-MUSIC, and fat-MUSIC can be used when reduced computational complexity is desired.

Next, in Fig. 6.4, we explore scenarios involving two sources. Specifically, we consider a setup where one DOA is fixed at 0° , while we vary the location of the other DOA. We use an 11-sensor ULA, 400 snapshots, and a 10 dB SNR. We compare the DOA estimation MSE of direct-MUSIC, coarray-MUSIC, tall-MUSIC, and fat-MUSIC. The results from Fig. 6.4(a) indicate that the MSEs of the three coarray-based methods are highly variable compared to that of direct-MUSIC. For some DOA separations, the MSE of coarray-MUSIC and tall-MUSIC is close to the CRB, but for other separations, the MSE is over an order of magnitude greater than the CRB.

In Fig. 6.4(b), we plot the efficiency of direct-MUSIC and coarray-MUSIC against DOA separation. We find that the coarray-MUSIC efficiency has peaks for certain DOA separations and is not monotonically increasing as the DOA separation is increased. It should be noted that the efficiency of direct-MUSIC also has peaks at similar locations as that of coarray-MUSIC; however, the difference in the efficiency values at peaks and valleys is negligible for direct-MUSIC, but that for coarray-MUSIC is very drastic. Lastly, in Fig. 6.4(c), we use a 21-sensor ULA instead of

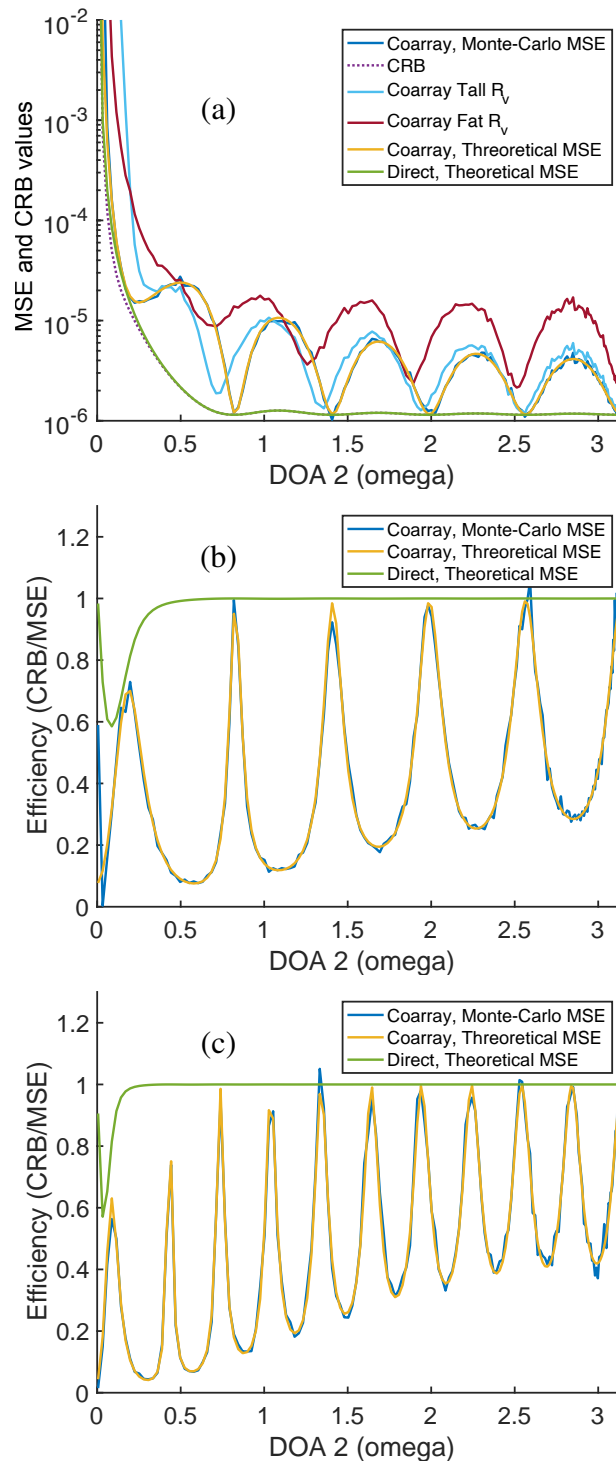


Figure 6.4: Estimating 2 DOAs with an 11-sensor ULA, with 400 snapshots, and SNR 10 dB. One DOA is fixed at 0° , and the other DOA is varied. (a) MSE with direct-MUSIC, coarray-MUSIC, tall-MUSIC, and fat-MUSIC. (b) Efficiency for direct-MUSIC and coarray-MUSIC. (c) Using a 21-sensor ULA with everything else the same as in (b). Coarray-MUSIC efficiency is observed to be much more sensitive to DOA separation than direct-MUSIC.

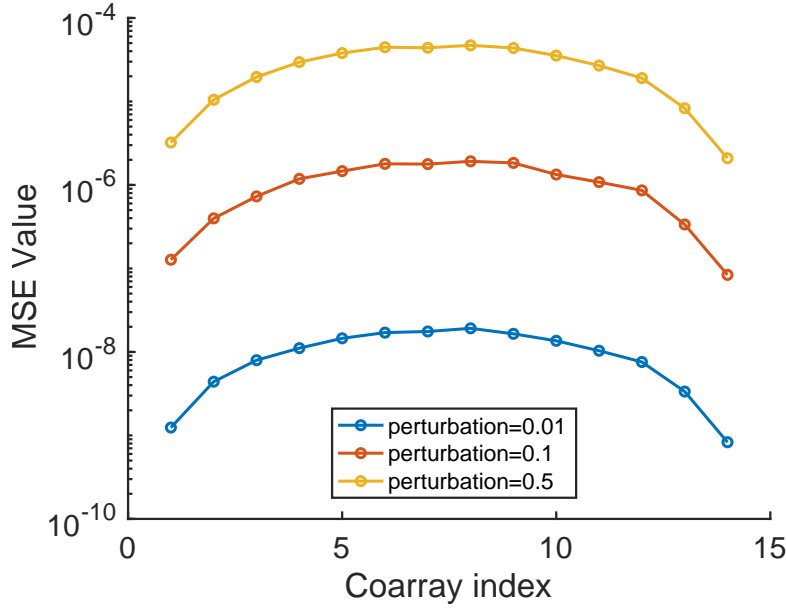


Figure 6.5: DOA estimation MSE for 15-sensor ULA when array output correlations are perturbed from their ideal value, one at a time. Three different perturbation variances, 0.01, 0.1, and 0.5, are considered. MSE is largest when correlations at lags around $N/2$ are perturbed, and smallest for $l_0 = 1$ and $l_0 = N - 1$.

an 11-sensor ULA and observe that the number of peaks has roughly doubled. The separation between the locations of consecutive peaks is observed to be roughly $2\pi/N$ in both cases.

Lastly, we conduct an additional experiment. We obtain the ideal array output correlations $R(l)$ from the entries of \mathbf{R}_{xx} (Eq. (2.6)). By applying MUSIC with \mathbf{R}_{xx} , DOAs can be recovered precisely. To understand how the perturbation in each of the array output correlations affects the DOA estimation performance, we take $\hat{R}(l_0) = R(l_0) + \epsilon$ and $\hat{R}(-l_0) = R(-l_0) + \epsilon^*$ for a specific lag index $l_0 > 0$, and the rest of the array output correlations are kept unperturbed. Here, ϵ is random complex Gaussian perturbation with variance σ_p^2 . We then form the Toeplitz coarray covariance matrix $\hat{\mathbf{R}}_{DA}$ as in Eq. (2.33) and perform DOA estimation using coarray-MUSIC. In Fig. 6.5, we plot the MSE averaged over 500 Monte-Carlo runs for each perturbation index, considering a 15-sensor ULA and 3 DOAs. We find that perturbing the ideal correlation at different lags affects the DOA estimation MSE differently. Rather surprisingly, MSE is largest when correlations at lags around $N/2$ are perturbed, and smallest for $l_0 = 1$ and $l_0 = N - 1$.

6.4 Optimization-Based Construction of Coarray Covariance Matrix

In Sec. 6.2, we experimentally demonstrated that coarray-MUSIC is inefficient even for ULA, indicating that the particular way of constructing the coarray covariance matrix by populating the estimated correlation entries $\hat{R}(l)$ is the cause of inefficiency. In this section, we consider an optimization-based approach for constructing a Toeplitz covariance matrix from [163]. In Chapter 5, we had used this method for estimating the DOAs with the weight-constrained arrays (see Eq. (5.14)). For such arrays with holes in the difference coarray, the Toeplitz covariance matrix construction approach was interpreted as a ‘covariance interpolation algorithm’, similar to [160]–[162], [164], [165]. While the purpose of using the Toeplitz covariance matrix construction approach in Chapter 5 was to interpolate the holes in the coarrays of weight-constrained arrays, the investigations in the rest of this chapter are around the DOA estimation efficiency of the Toeplitz covariance matrix construction approach. We will see in the next section that this way of constructing the coarray covariance matrix is not only beneficial for the arrays having holes in the difference coarrays (such as coprime arrays and weight-constrained arrays), but can also greatly improve the DOA estimation MSE for sparse arrays such as nested arrays and MRAs that do not have holes in their difference coarrays and hence do not ‘need’ their coarrays to be ‘interpolated’ in the traditional sense.

Given the sample covariance matrix $\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}$ of a sparse array \mathbf{z} with aperture A , the Toeplitz covariance matrix construction approach from [163] aims at recovering the ideal Hermitian Toeplitz covariance matrix \mathbf{R}_U that a fictitious ULA of aperture A would have produced. That is, even though the physical array is a sparse array over an aperture of length A , it aims to recover the covariance matrix that a ULA over that aperture would have produced. An optimization problem is proposed in [163] for this. As in the last chapter (Eq. 5.14), we use the following slightly modified version of the optimization problem for simplicity:

$$\min_{\mathbf{T}(\mathbf{u}) \succeq 0} \|\widehat{\mathbf{W}}^{-1/2} \text{vec}(\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} - \mathbf{\Gamma} \mathbf{T}(\mathbf{u}) \mathbf{\Gamma}^T)\|_2 + \lambda \text{tr}(\mathbf{T}(\mathbf{u})). \quad (6.7)$$

As the ideal Hermitian Toeplitz matrix to be estimated (\mathbf{R}_U) can be parameterized by its first column, the optimization problem is formulated in terms of a parameter vector \mathbf{u} , and $\mathbf{T}(\mathbf{u})$ denotes the Hermitian Toeplitz matrix having \mathbf{u} as its first column. $\mathbf{\Gamma}$ is a row selection matrix of size $N \times (A + 1)$ corresponding to the sensor positions of the sparse array \mathbf{z} , i.e. $\mathbf{\Gamma}_{i,j} = 1$ if $n_i = j$, and zero otherwise. Matrix $\widehat{\mathbf{W}} = \hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}^T \otimes \hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} / K$, where \otimes is the Kronecker product. Note that $\text{vec}(\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} - \mathbf{\Gamma} \mathbf{T}(\mathbf{u}) \mathbf{\Gamma}^T)$ is the discrepancy between the correlation estimates that are available in $\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}$ and

the corresponding entries from $\mathbf{T}(\mathbf{u})$. The premultiplication by $\widehat{\mathbf{W}}^{-1/2}$ is motivated by the asymptotic distribution of the terms in this error vector [163], [175]:

$$\text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{R}_U\mathbf{\Gamma}^T - \sigma_n^2\mathbf{I}) \sim \text{AsN}(\mathbf{0}, \widehat{\mathbf{W}}). \quad (6.8)$$

Thus, the first term in the optimization objective $\|\widehat{\mathbf{W}}^{-1/2} \text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T)\|_2$ is the l_2 norm of the ‘whitened’ error vector. We drop the noise term $\sigma_n^2\mathbf{I}$ for simplicity, but [163] proposes first to estimate noise variance $\hat{\sigma}_n^2$ and use it instead. The second term in the objective is trace regularization $\text{tr}(\mathbf{T}(\mathbf{u}))$, and is added to the objective as a convex relaxation of the rank penalty. The positive semi-definiteness constraint on $\mathbf{T}(\mathbf{u})$ is enforced explicitly, and the hyperparameter λ selects the trade-off between the data fidelity term and trace regularization. In our simulations, λ is chosen to be 0.01. Similar to the simulations in [163], we have found that the performance of the proposed algorithm is not very sensitive to λ , as long as λ is not too large. The two key features of this method, which we think are important for the improvement in DOA estimation observed in the simulations, are explained next.

First, this approach does not average the correlation entries from $\widehat{\mathbf{R}}_{\mathbf{xx}}$ corresponding to the same lag to form the correlation sequence $\widehat{R}(l)$. This averaging to form the correlation sequence $\widehat{R}(l)$ is the key step before the formation of coarray covariance matrices $\widehat{\mathbf{R}}_{SS}$ and $\widehat{\mathbf{R}}_{DA}$, and it is what fundamentally motivates the coarray-domain DOA estimation. In Sec. 6.2 and in papers such as [90], [125], this averaging has been recently observed to be the reason behind the inefficiency of coarray-based DOA estimation algorithms. By not averaging the entries from $\widehat{\mathbf{R}}_{\mathbf{xx}}$ that correspond to the same lag, this method retains the information present in $\widehat{\mathbf{R}}_{\mathbf{xx}}$ as it is.

Second, instead of simply using the l_2 norm of the error term $\text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T)$ as it is, the above method whitens this error term by premultiplying it with $\widehat{\mathbf{W}}^{-1/2}$ based on its asymptotic distribution. To demonstrate the importance of this whitening step, in our simulations, we also consider the following optimization instead of Eq. (6.7):

$$\min_{\mathbf{T}(\mathbf{u}) \succeq 0} \|\text{vec}(\widehat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T)\|_2 + \lambda \text{tr}(\mathbf{T}(\mathbf{u})). \quad (6.9)$$

Note that Eq. (6.9) still does not average the correlation entries from $\widehat{\mathbf{R}}_{\mathbf{xx}}$ corresponding to the same lag and is a reasonable way of formulating an optimization problem to construct a coarray covariance matrix. However, it does not take into account the correlations between the entries in the error vector. To demonstrate the importance of this, we will compare the DOA estimation MSE obtained by using the covariance matrices estimated according to Eq. (6.7) and Eq. (6.9). To the best of

our knowledge, most of the interpolation algorithms from the literature [160]–[162], [164], [165] do not take into account the correlations between the entries in the error vector, and do not perform error whitening as done in Eq. (6.7).

6.5 Efficiency of the Optimization-Based Approach for Different Arrays

Now we compare the DOA estimation MSE obtained from the three methods of constructing the coarray covariance matrix, namely, direct augmentation ($\hat{\mathbf{R}}_{DA}$), interpolation with whitening (Eq. (6.7)), and interpolation without whitening (Eq. (6.9)). To solve the optimization problems, we use CVX, a package for specifying and solving convex programs [170], [171]. Then, root-MUSIC is applied with these three matrices to estimate the DOAs. First, consider the simplest case of a ULA with $N = 12$ sensors. For a ULA, $\hat{\mathbf{R}}_{DA}$ is simply a ‘toeplitzized’ (i.e., diagonally averaged) version of $\hat{\mathbf{R}}_{xx}$. Although direct root-MUSIC (using $\hat{\mathbf{R}}_{xx}$) is efficient, after toeplitzization (i.e., when using $\hat{\mathbf{R}}_{DA}$), DOA estimation is no longer efficient. Note that Eq. (6.7) and Eq. (6.9) also create Toeplitz matrices similar to $\hat{\mathbf{R}}_{DA}$, but they are formed in a different way by solving optimization problems rather than arranging the estimated correlation in a Toeplitz manner. Fig. 6.6 plots the DOA estimation MSEs obtained with the three methods along with CRB when there are $D = 10$ sources separated uniformly in the ω -domain and 500 snapshots are used. Each simulation point in the plots is averaged over 500 Monte-Carlo runs with randomly chosen $\{\mathbf{s}[k], \mathbf{n}[k]\}_{k=1}^K$. For the ULA (top left in Fig. 6.6), as SNR is varied on the X-axis, we observe that the DOA estimation MSE when using the matrix obtained with Eq. (6.7) does not saturate and practically equals CRB when SNR is larger than -10 dB. This is a huge improvement compared to the coarray root-MUSIC, which has saturating MSE for large SNR. Thus, at large SNR, by using the matrix obtained from Eq. (6.7) instead of $\hat{\mathbf{R}}_{DA}$, the efficiency κ improves from nearly 0 to nearly 1. Also note that the DOA estimation MSE when using the matrix obtained with Eq. (6.9) shows no improvement compared to the coarray root-MUSIC. This clearly highlights the key importance of the whitening used in Eq. (6.7).

In the rest of the plots in Fig. 6.6, the same quantities are compared for different well-known sparse arrays from the literature (see Sec. 4.5 for the acronyms of the different arrays) as well as the weight-constrained arrays proposed in Chapter 4. All arrays have $N = 12$ sensors. We observe a similar large improvement in the MSE for all these sparse arrays when the matrix obtained from Eq. (6.7) is used instead of direct augmentation. Thus, the Toeplitz covariance matrix construction approach (with whitening) almost completely mitigates the inefficiency of coarray

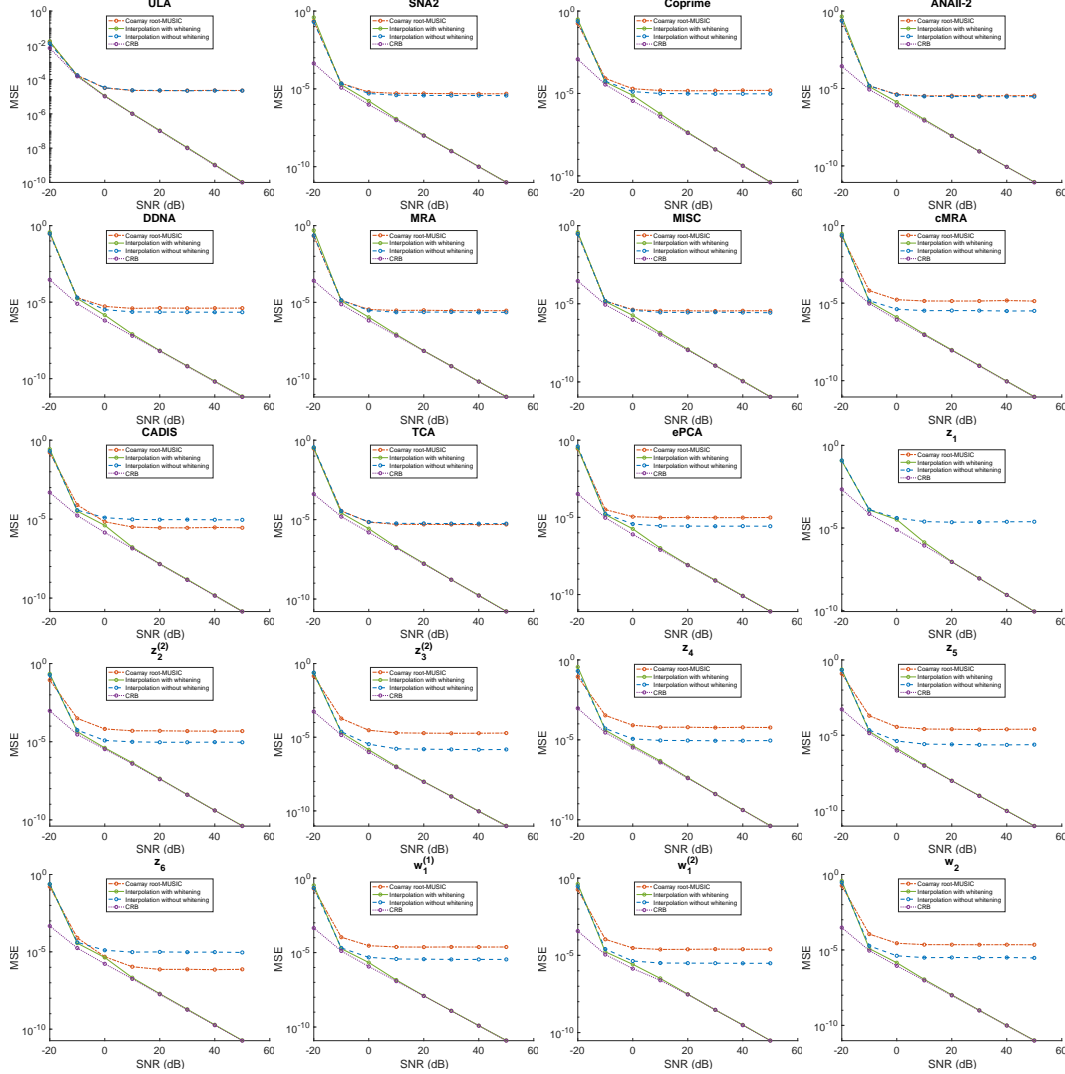


Figure 6.6: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 10$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots.

root-MUSIC for sparse arrays when $D < N$, changing the efficiency κ from nearly 0 to nearly 1 when SNR is large.

Next, in Fig. 6.7 we consider the task of estimating $D = 20$ DOAs with 12-sensor sparse arrays. We observe that for this case where $D > N$, the covariance matrix construction approach with whitening provides a significant improvement in MSE over coarray-MUSIC. Although CRB saturates at high SNR in this case when $D > N$, the MSE obtained with the matrix from Eq. (6.7) is close to CRB.

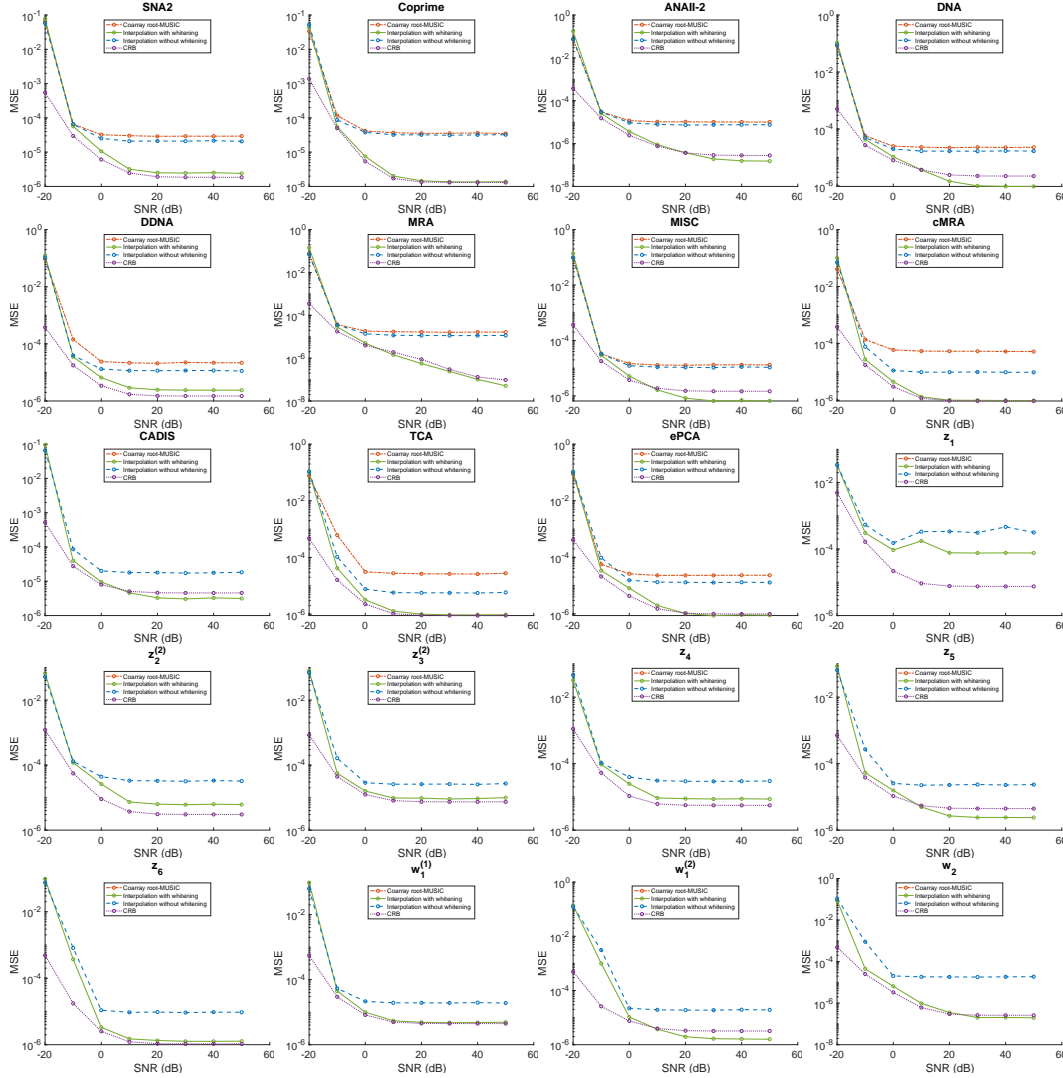


Figure 6.7: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 20$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots.

When whitening is not used, the improvement over coarray root-MUSIC is small. Interestingly, also note that for many arrays in Fig. 6.7, the MSE when using the interpolation with whitening is smaller than CRB. This behavior can be attributed to the fact that the DOA estimates obtained with the matrix from Eq. (6.7) can potentially be biased. One of the sources of bias is the regularization term used in the optimization problem. Some other methods from the literature, such as [172], have also reported attaining MSE lower than the CRB because of biased DOA estimates.

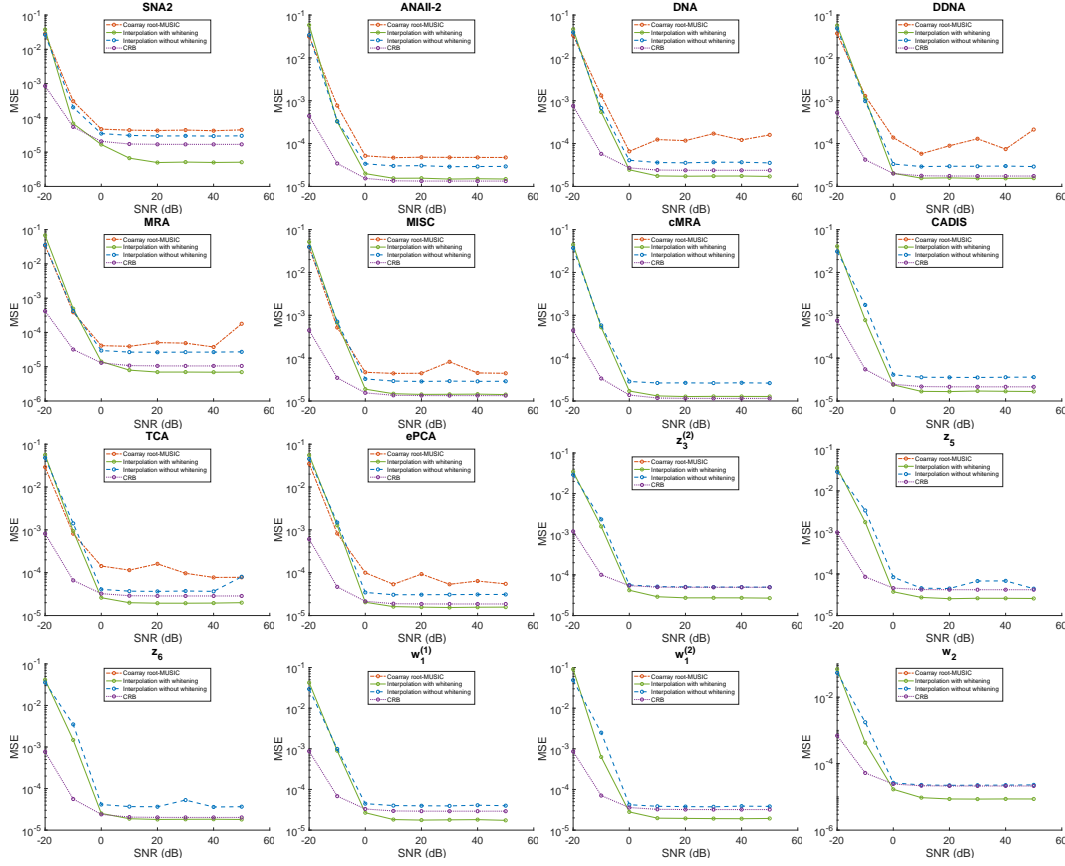


Figure 6.8: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), and the CRB when estimating $D = 30$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots.

This behavior can be seen even more pronounced in Fig. 6.8 where $D = 30$ DOAs are estimated with 12-sensor arrays. The arrays that have an aperture less than 30 cannot identify 30 DOAs even with the larger Toeplitz covariance matrix obtained after solving the optimization problems, and are omitted from Fig. 6.8. For the rest of the arrays, we see that while the interpolation without whitening does offer some improvement over coarray root-MUSIC, interpolation with whitening offers much greater improvement and has MSE smaller than CRB. Further investigations are required to understand this behavior, and will be of interest for the future.

6.6 Modification of the Optimization-Based Approach in the Presence of Mutual Coupling

In the previous section (Sec. 6.5), we considered the situations without any mutual coupling. In this section, we explore how the situation changes when mutual coupling is present. One option is to still solve the optimization problems in Eq. (6.7) and Eq. (6.9) even if the coupling is present. In Chapter 5, we used this approach even in the presence of mutual coupling. However, if the coupling matrix \mathbf{C} is known (or can be reliably estimated), the optimization problem can be modified to improve the DOA estimation accuracy. In the presence of mutual coupling, the array output is given by Eq. (2.27) and thus, the ideal array output covariance matrix is

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} = \mathbf{C}\mathbf{A}\mathbf{R}_{\mathbf{s}\mathbf{s}}\mathbf{A}^H\mathbf{C}^H + \sigma_n^2\mathbf{I}. \quad (6.10)$$

Hence, we propose to solve the following modified optimization problem from Eq. (6.7) when the coupling matrix \mathbf{C} is known:

$$\min_{\mathbf{T}(\mathbf{u}) \succeq 0} \quad ||\widehat{\mathbf{W}}^{-1/2} \text{vec}(\widehat{\mathbf{R}}_{\mathbf{x}\mathbf{x}} - \mathbf{C}\mathbf{T}\mathbf{T}(\mathbf{u})\mathbf{I}^T\mathbf{C}^H)||_2 + \lambda \text{tr}(\mathbf{T}(\mathbf{u})). \quad (6.11)$$

Note that this way of formulating the optimization problem compensates for the effect of mutual coupling. It also retains the two important advantages mentioned in the previous section, namely, (a) not averaging the correlation entries from $\widehat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}$ corresponding to the same lag and (b) using the asymptotic distribution to whiten the correlated error terms, using the $\widehat{\mathbf{W}}$ matrix.

Now we compare the DOA estimation MSE obtained with the above ‘coupling compensated interpolation’ formulation of the optimization problem with the other ways of constructing coarray covariance matrix, namely, direct augmentation (i.e., just using $\widehat{\mathbf{R}}_{DA}$), interpolation with whitening (Eq. (6.7)), and interpolation without whitening (Eq. (6.9)). Out of these methods, only the coupling compensated interpolation approach assumes the knowledge of \mathbf{C} . To the best of our knowledge, the CRB for DOA estimation in the presence of mutual coupling has not been derived in the literature. In the simulations below, for a reference, we plot the CRB derived in [86], [87] that assumes no mutual coupling is present. The CRB derived under the assumption of no mutual coupling is a lower bound for the CRB when mutual coupling is present.

The coupling coefficients of the form

$$c_1 = 0.2e^{j\pi/3}, \quad c_l = (c_1/l)e^{-j(l-1)\pi/8} \quad \text{for } 2 \leq l \leq 10 \quad (6.12)$$

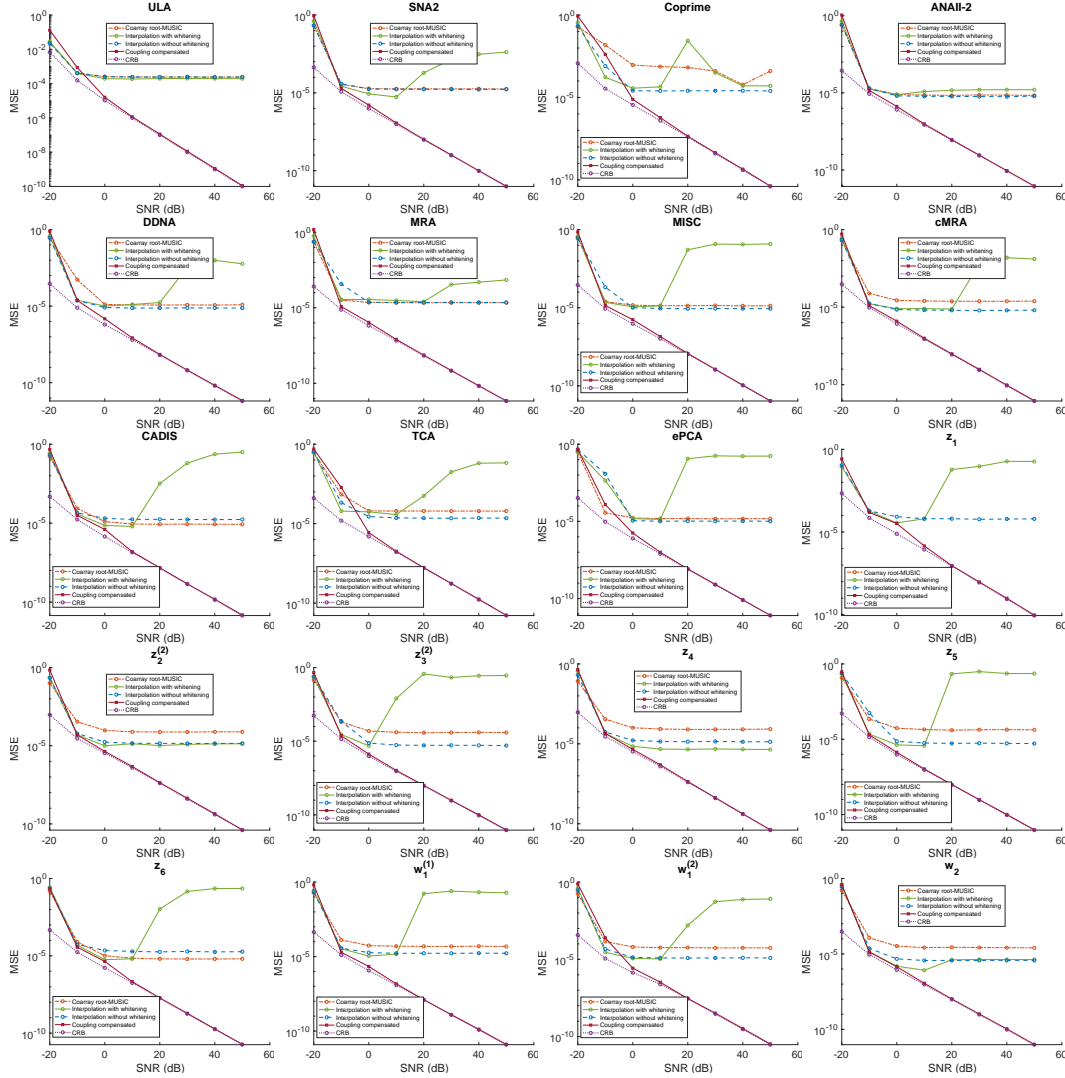


Figure 6.9: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11)), and the CRB when estimating $D = 10$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.

are assumed in the simulations. Arrays have $N = 12$ sensors, and $K = 500$ snapshots are used. Each simulation point in the plots is averaged over 500 Monte-Carlo runs with randomly chosen $\{\mathbf{s}[k], \mathbf{n}[k]\}_{k=1}^K$. Fig. 6.9 plots the DOA estimation MSEs obtained with the four methods along with CRB when there are $D = 10$ sources separated uniformly in the ω -domain.

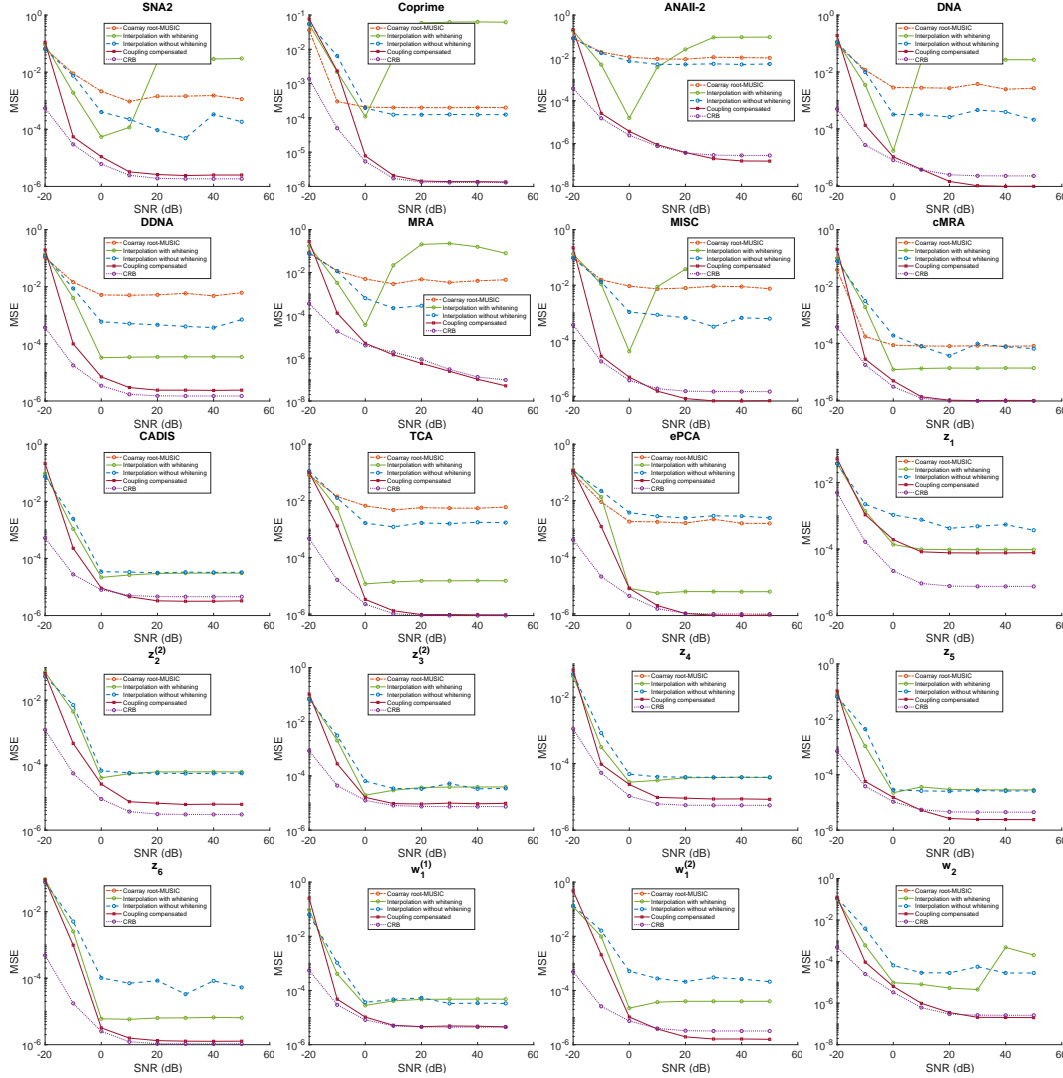


Figure 6.10: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11)), and the CRB when estimating $D = 20$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.

Simulations in the previous section (without mutual coupling) demonstrated that the interpolation with whitening (Eq. (6.7)) achieved CRB when $D < N$. However, in Fig. 6.9, we observe that when coupling is present, interpolation with whitening does not achieve CRB. On the contrary, for several arrays, the MSE obtained with the matrix from Eq. (6.7) increases at higher SNR. Similar behavior was also observed in the last chapter in Fig. 5.9 for $D < N$. In the optimization problem from Eq.

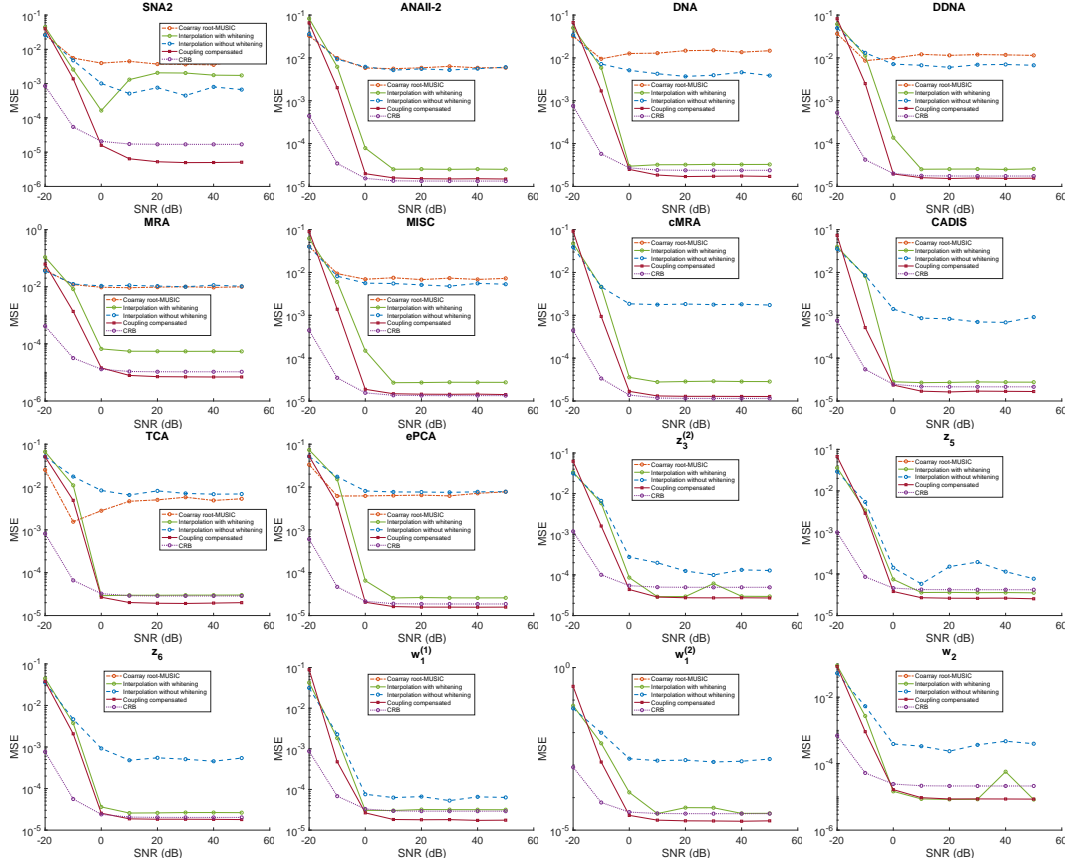


Figure 6.11: Comparing DOA estimation MSE with coarray root-MUSIC (using matrix $\hat{\mathbf{R}}_{DA}$), interpolation with whitening (using matrix obtained by solving Eq. (6.7)), interpolation without whitening (using matrix obtained by solving Eq. (6.9)), coupling compensated interpolation (using matrix obtained by solving Eq. (6.11)), and the CRB when estimating $D = 30$ DOAs separated uniformly in the ω -domain form -0.9π to 0.9π using $N = 12$ sensor arrays and $K = 500$ snapshots. Mutual coupling with parameters as described in Eq. (6.12) is considered in these simulations.

(6.7), the effect of the coupling matrix \mathbf{C} is present in the term $\hat{\mathbf{W}}^{-1/2}$, but it is not accounted for in the term $\mathbf{\Gamma}\mathbf{T}(\mathbf{u})\mathbf{\Gamma}^T$. This mismatch manifests in a surprising manner, and the MSE is observed to increase at higher SNR. It would be of interest for the future to understand why this increase in MSE at high SNR is observed for only some arrays.

On the other hand, the optimization problem from Eq. (6.9) does not involve whitening (premultiplication with $\hat{\mathbf{W}}^{-1/2}$). The DOA estimation MSE obtained with this matrix saturates similarly to the coarray root-MUSIC for all arrays, and it also does not increase at higher SNR. Thus, when the mutual coupling matrix \mathbf{C} is not known, interpolation without whitening Eq. (6.9) can be a better alternative

than interpolation with whitening Eq. (6.7), especially for higher SNRs.

When the coupling matrix \mathbf{C} is known, we use the matrix obtained from coupling compensated interpolation (Eq. (6.11)). It is observed in Fig. 6.9 that this provides a large improvement in DOA estimation MSE. The MSE does not saturate and is very close to CRB, indicating that the efficiency is almost unity. To the best of our knowledge, such a result on efficient DOA estimation in the presence of mutual coupling has not been reported in the literature before.

Similar to Figs. 6.7 and 6.8, we consider the task of estimating $D = 20$ and $D = 30$ DOAs in Figs. 6.10 and 6.11 respectively, but when the mutual coupling described in Eq. (6.12) is considered. The DOA estimation error when using interpolation without whitening saturates at high SNR. The interpolation with whitening shows a similar trend to that in Fig. 6.9, where the MSE can worsen at higher SNRs for some arrays. The coupling compensated interpolation method provides the smallest MSE, and it is observed to be smaller than CRB for many arrays. Achieving MSE smaller than CRB is possible when there is bias in the estimates. These results demonstrate that the DOA estimation algorithms that can compensate for the effect of mutual coupling can lead to large improvements in the DOA estimation.

6.7 Concluding Remarks

In this chapter, we conducted several experiments to investigate the efficiency of coarray-MUSIC and a Toeplitz covariance matrix construction approach for DOA estimation. In the first part, we demonstrated that, similar to sparse arrays, coarray-MUSIC is asymptotically inefficient for ULAs as well. This behavior is surprising because the only difference between coarray-MUSIC and direct-MUSIC for ULAs is the diagonal averaging of the array output covariance matrix. This observation for ULA indicates that the loss of efficiency is an artifact of the coarray-MUSIC algorithm rather than a result of array sparsity. In scenarios with two sources, coarray-MUSIC demonstrated increased sensitivity to DOA separation compared to direct-MUSIC. We examined two variations of coarray-MUSIC: tall-MUSIC and fat-MUSIC. These variations allow for the extraction of the noise subspace in a different dimensional space. While tall-MUSIC exhibits a comparable MSE to coarray-MUSIC, and fat-MUSIC significantly reduces computational complexity, neither modification improves efficiency. Lastly, we observed that perturbing the ideal array output correlations at a lag around $N/2$ leads to the largest MSE for coarray-MUSIC. These observations provide new insights into the complex interplay

between various factors, such as array geometry, DOA separation, and the accuracy of the estimated array output correlations, all of which affect the MSE of coarray-MUSIC. Further research is necessary to better understand the behavior of coarray-MUSIC MSE.

In the later part of the chapter, we explored an alternative way of constructing a Toeplitz covariance matrix proposed in [163]. This approach from [163] is observed to greatly improve the DOA estimation performance with sparse arrays when no mutual coupling is present. When the number of DOAs D is smaller than the number of sensors N , this approach is found to produce efficient DOA estimates at large SNRs for many different sparse arrays. When $D > N$, we observed that this approach still provides a large improvement in MSE, and for some arrays, the MSE can be even smaller than the CRB because of the biased DOA estimates. Whitening of the correlated error terms is found to be an important factor in the MSE improvement obtained with this optimization-based approach. However, when mutual coupling is present, solving the optimization problem from [163] is observed to work poorly when SNR is large. For this situation, we proposed a way of compensating for the effect of mutual coupling if the coupling matrix \mathbf{C} is known. The coupling compensated interpolation is observed to be efficient when $D < N$ and can have an MSE smaller than the CRB for $D > N$.

Recently, a few other methods, such as WLS [90] and a stochastic maximum likelihood based method [172], have also been proposed in the literature to mitigate the inefficiency of coarray-domain DOA estimation. However, these methods are quite elaborate and can be difficult to implement. On the other hand, the approaches based on direct augmentation and spatial smoothing are conceptually straightforward and easy to implement, given the popularity and simplicity of the subspace-based DOA estimation algorithms like root-MUSIC. But they produce inefficient DOA estimates. Thus, it is of great interest to investigate other ways of constructing coarray covariance matrices that retain the conceptual simplicity of estimating the Toeplitz covariance matrix, followed by applying root-MUSIC, and still achieve efficient DOA estimation. While the optimization-based approach we considered in this chapter is one such method that shows promising results, it is computationally expensive compared to coarray-MUSIC because of the optimization-based solution. In the future, it would be of interest to explore other possibilities of constructing the coarray covariance matrix that are computationally efficient and also produce efficient DOA estimates. A deep learning based methodology for subspace rep-

resentation learning has been developed very recently [176], [177]. It would also be of interest to understand the bias introduced by the optimization problems and theoretically analyze the analytical MSE expression for the Toeplitz covariance matrix construction approach. Furthermore, a careful study of the optimization-based approach and the increase in MSE at high SNR in the presence of mutual coupling will be of great interest in the future. Robust interpolation methods that explicitly account for the presence of mutual coupling in their optimization objective should also be developed in the future.

PERIODICITY-AWARE SIGNAL DENOISING USING CAPON-OPTIMIZED RAMANUJAN FILTER BANKS AND PRUNED RAMANUJAN DICTIONARIES

7.1 Introduction

Signal denoising is a widely studied problem in signal processing [75], [76], [178]–[182]. Due to the random nature of the noise, the noise-free signal cannot be recovered just from its noisy measurements in the absence of any other information. A key to successful denoising is to appropriately incorporate some a-priori information about the nature of the signal in the denoising process. For example, it is well-known that most of the energy of natural images is concentrated in only a few dominant wavelet coefficients. This makes wavelet denoising [179] a suitable denoising method for natural images. Noisy periodic signals have a line spectrum in the frequency domain, added with noise, as shown in Fig. 1.1(b). Discrete Fourier Transform (DFT) based denoising works well for harmonic signals if the component frequencies lie on the DFT frequency grid. Invertible transforms like DFT and wavelet transform can be readily used to denoise appropriate classes of signals. Such denoising methods usually threshold the transformed signal according to some scheme, and then perform the inverse transformation to get a denoised signal [179], [183], [184].

Instead of an invertible transformation, an overcomplete set can also be used for representing a class of signals. For an appropriately chosen overcomplete set, the signals can be represented as sparse linear combinations of the vectors in the representation set, and denoising can then be posed as a sparse vector recovery problem. Orthogonal matching pursuit (OMP) [180], [185], and LASSO [186] are examples of popular algorithms to solve such sparse vector recovery problems. Instead of explicitly choosing an appropriate overcomplete set for sparse signal representation, it is also possible to learn a representation set beforehand from clean data, or directly from the patches of the noisy signal to be denoised. One such well-known method for dictionary learning is K-SVD [74]. It has been adapted for denoising natural images [75], [76].

In this chapter, we consider the class of discrete-time periodic signals. Periodic

signals form an important class of signals as many naturally occurring signals can be approximated as noisy periodic signals. The structural properties of the underlying periodic signal often carry important information about the phenomenon being observed. Some examples of periodic signals are segments of tonal music, voiced speech, and ECG. Discrete-time periodic sequences also occur in nature. These include tandem repeats in the DNA [42] that are indicative of genetic disorders and protein repeats [187] in amino acids that are indicative of their structural and binding properties.

When such signals are contaminated with noise, denoising them is an important processing step in many applications. Although there are numerous methods in the literature for signal and image denoising, most of them do not work well for periodic signals, as we will demonstrate through our simulations. This is because they are not designed with the a-priori information that the underlying signal is periodic. These methods do not respect the periodic structure of the signal and produce a non-periodic signal as the denoised output. It was demonstrated in [53] that the DFT representation of periodic signals is not useful to identify the periods if the component frequencies do not lie on the DFT frequency grid, especially when only a short signal length is available. Non-harmonic analysis (NHA) was developed in [181], particularly for denoising the sum of sinusoid signals. It poses denoising as a mean squared error minimization problem. However, this class of signals may not be discrete-time periodic with an integer period, which is the focus of this chapter. Furthermore, NHA requires a good initialization, which the DFT may not be able to provide. In order to overcome these limitations, we propose a new ‘periodicity-aware’ denoising method for discrete-time periodic signals.

7.1.1 Scope and Outline

To develop a ‘periodicity-aware’ denoising method for discrete-time periodic signals, we use representation tools developed for periodic signals based on Ramanujan sums and subspaces. These include Ramanujan filter banks (RFB) [63], [64] and Ramanujan dictionaries [53]. This choice is motivated by the fact that Ramanujan-sum representations have already been shown to be well-suited for identifying multiple periodic components in discrete time signals [53], [63], [64]. These are reviewed in Sec. 7.2. In Sec. 7.3, we describe our proposed hybrid signal denoising framework in detail. The need for a hybrid framework arises because a perfect reconstruction synthesis counterpart for the Ramanujan analysis filter bank is not known. Thus in our framework, analysis is done by RFB, and synthesis is done by a pruned

Ramanujan dictionary. This guarantees that the denoised signal is composed of integer-periodic components with periods smaller than a pre-selected value. To our knowledge, no other denoising method is known to have this property. We demonstrate that the proposed method produces the best SNR gain for periodic signals and outperforms many traditional denoising techniques. It produces a stable SNR gain over a large range of input SNR values, even when the signal length is not very large.

In Sec. 7.4, we further optimize each filter in the analysis bank based on incoming data using the Capon formulation used in array signal processing for finding an optimal beamformer. Since the filters are multi-banded, we need to use multi-constrained Capon type optimization to get the optimal filter coefficients based on the data. We show that this significantly improves the denoising performance when the signal length is sufficiently large. In Sec. 7.5, we discuss a recursive way to compute inverses of several autocorrelation matrices required for the entire bank of Capon filters, to reduce the computations. In Sec. 7.6 we prove several multirate properties of Ramanujan subspace signals and Ramanujan sums. In particular, we show that after decimation, a q -th Ramanujan subspace signal remains in the same subspace S_q if and only if the decimation ratio M is coprime to q . This property motivates the decimation of the filter bank outputs in the denoising framework, as it still approximately preserves the average per-sample signal energy of the outputs (as we shall show). Such decimation of filter outputs reduces the required computations when the filters are implemented using their polyphase representation [188]. Lastly, in 7.7, we provide extensive Monte-Carlo simulations to compare various denoising methods. We discuss the effect of hyperparameter values on denoising performance and the applicability of the proposed method in denoising ECG signals. We also discuss the applicability of the proposed framework for the period estimation and component separation tasks and provide simulation examples.

The main contributions of this chapter are as follows:

1. We propose the idea of using the Ramanujan analysis filter bank followed by a synthesis dictionary for denoising periodic signals, and demonstrate that it can achieve a good denoising performance.
2. We propose the use of optimized analysis bank filters with the same passband locations as Ramanujan filter banks. These filters are optimized based on data, similar to Capon beamformers. Each filter in the RFB has multiple look directions (at ‘coprime frequencies’) and is optimized to maximize its output

signal to interference plus noise ratio (SINR). We use a multi-passband style Capon beamformer algorithm to design the analysis bank.

3. We discuss the computational cost incurred to implement the Capon-optimized filters and provide a way to reduce computations by recursively calculating inverses of several autocorrelation matrices using Levinson's recursion from linear prediction theory.
4. We derive new theoretical properties of Ramanujan subspace signals, in the context of decimation. We show that the downsampled versions of Ramanujan subspace signals belong in the same space under some coprimality conditions. We show that such decimation creates no aliasing and preserves the average per-sample signal energy.
5. We introduce a subspace of signals that can be considered to be a nonzero bandwidth approximation of the infinitesimal-bandwidth ideal Ramanujan subspace signals. We discuss the applicability of the above decimation result to such signals. We show that this helps to further reduce computations by downsampling filter bank outputs for both RFB and Capon-optimized filter banks.
6. We provide extensive Monte-Carlo simulation results to validate the major new developments described above. We compare different variants of the proposed methods with each other and with existing methods. We will see that the proposed method, which uses Capon-optimized filters, works significantly better than existing methods. We also investigate the effect of different hyperparameters in the proposed method on denoising performance in detail.

The content of this chapter is based on papers [189], [190].

7.2 Ramanujan Filter Banks and Dictionaries

It has been well-established in the past decade that Ramanujan sums are particularly useful to identify component periods of a discrete-time periodic signal [52], [62], [65]. The q -th Ramanujan sum $c_q(n)$ was defined for every positive integer q in Eq. (1.15). It can be equivalently expressed as

$$c_q(n) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} e^{j2\pi kn/q} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} W_q^{-kn} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} W_q^{kn}. \quad (7.1)$$

Note that in this chapter, we prefer to have the summation index in the definition of Ramanujan sums from $k = 0$ to $k = q - 1$ instead of $k = 1$ to $k = q$. Ramanujan sums exhibit several useful properties. Despite having complex exponentials in its definition, the q -th Ramanujan sum takes only real integer values for all n and is periodic with period q [52], [62], [65]. The q -th Ramanujan subspace S_q [52] is the class of signals of the form

$$x(n) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_q^{kn}, \quad (7.2)$$

where α_k can be any complex numbers. It has been shown that S_q is a $\phi(q)$ -dimensional subspace. It contains period q signals that have non-zero DFT value only at the ‘coprime frequencies’ $2\pi k/q$ where $0 \leq k \leq q - 1$ and k is coprime to q . It can be shown that the $\phi(q)$ shifted versions of $c_q(n)$ also span S_q , i.e. any $x(n) \in S_q$ can be expressed as

$$x(n) = \sum_{l=0}^{\phi(q)-1} \beta_l c_q(n-l) \quad (7.3)$$

for appropriate coefficients β_l . Note that the signals in S_q have infinite length in the time domain and infinitesimal bandwidth in the frequency domain. A review of Ramanujan sums, Ramanujan subspaces, and their properties can be found in [52], and an overview was given in [65].

7.2.1 Nested Periodic Dictionaries

Consider a signal $\mathbf{x} \in \mathbb{R}^N$ that can have periodic components with periods from 1 to P_{max} . We assume in this chapter that P_{max} , the bound on the maximum component period of the signal, is known a-priori. Then \mathbf{x} can be represented as [53], [191]

$$\mathbf{x} = \mathbf{D}\mathbf{b}, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_{P_{max}} \end{bmatrix}. \quad (7.4)$$

Here, \mathbf{H}_q are the matrices of size $N \times \phi(q)$ whose columns are the q -th Ramanujan sum and its $\phi(q)$ shifted versions truncated to length N . Thus, \mathbf{H}_q spans the signals from the q -th Ramanujan subspace S_q that are truncated to length N . This dictionary is called the Ramanujan dictionary [53]. The coefficient vector \mathbf{b} has non-zero entries at locations corresponding to the subspaces in which the periodic components of \mathbf{x} lie. The size of \mathbf{D} is $N \times \Phi(P_{max})$, where

$$\Phi(q) = \sum_{k=1}^q \phi(k). \quad (7.5)$$

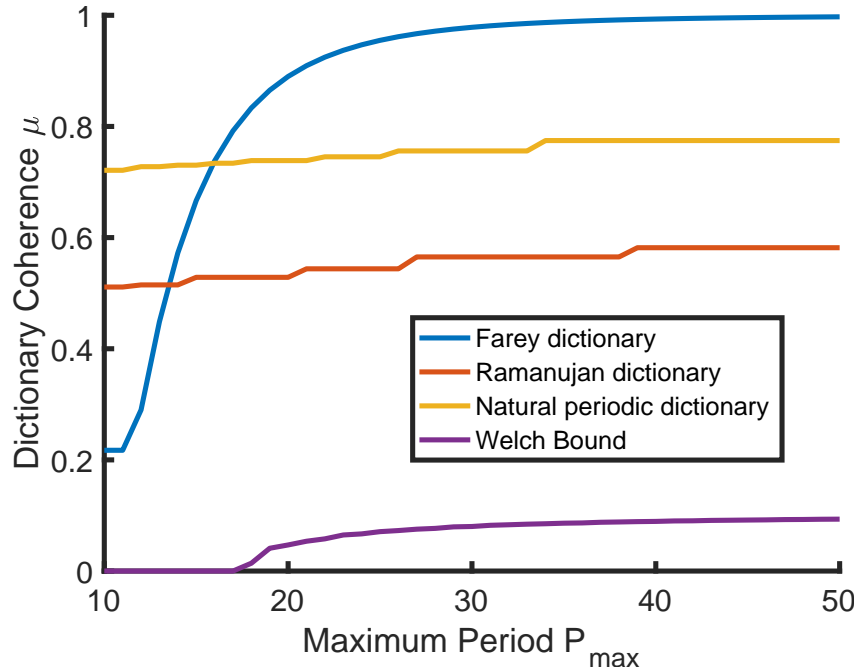


Figure 7.1: Dictionary coherence values of Ramanujan, Farey, and natural periodic dictionaries compared against the Welch bound as P_{max} is varied, keeping the number of rows constant at 100.

As $\Phi(P_{max})$ grows $O(P_{max}^2)$ [191], \mathbf{D} is a fat matrix if the number of samples N is not too large. Since \mathbf{D} is a fixed known matrix, obtaining \mathbf{b} from \mathbf{x} can be considered a sparse vector recovery problem. Two other dictionary choices apart from the Ramanujan dictionary are the Farey dictionary [191] and the natural periodic dictionary [53]. The Farey dictionary has complex-valued entries, whereas the natural periodic dictionary has only zeros and ones. For details, see [53]. For successful sparse signal recovery, it is desirable to have the representation matrix with low mutual coherence [192]. When the columns of a dictionary \mathbf{D} have a unit l_2 norm, the mutual coherence is defined as the maximum absolute value of the inner product between any two distinct columns

$$\mu(\mathbf{D}) = \max_{i \neq j} |\mathbf{d}_i^H \mathbf{d}_j|, \quad (7.6)$$

where \mathbf{d}_i is the i -th column of \mathbf{D} . For a dictionary of size $N \times M$, $\mu(\mathbf{D})$ is lower-bounded as follows [193]:

$$\mu(D) \geq \sqrt{(M - N)/N(M - 1)}. \quad (7.7)$$

This is known as the Welch bound. Fig. 7.1 shows the coherence values for Ramanujan, Farey, and natural periodic dictionaries when the number of rows is

100 and the maximum period P_{max} is varied from 10 to 50. Out of the three, the Ramanujan dictionary has the lowest mutual coherence and hence may be considered to be a better choice to analyze periodic signals. Thus, in this work, we use the Ramanujan dictionary for signal reconstruction. That is, we take \mathbf{H}_q as

$$\mathbf{H}_q = \begin{bmatrix} | & & | & & | \\ c_q(n) & c_q(n-1) & \dots & c_q(n-(\phi(q)-1)) \\ | & & | & & | \end{bmatrix}. \quad (7.8)$$

The columns of \mathbf{H}_q are the first $\phi(q)$ shifted versions of the q -th Ramanujan sum $c_q(n)$, truncated to length N . We will also demonstrate in the next Chapter 9 that under favorable circumstances, K-SVD [74] can learn Ramanujan and Farey dictionaries from the noisy, sparse coefficient data generated from them even if no periodicity structure is imposed in the learning stage.

7.2.2 Ramanujan Filter Banks

The dictionary method is not suitable when the periodicity structure of the signal evolves over time. Ramanujan filter banks (RFB) [63], [64], [194] are proposed for such a scenario. RFB consists of FIR filters that correspond to each period from 1 to P_{max} . The q -th Ramanujan filter is given by

$$C_q^{(l)}(z) = \sum_{n=0}^{lq-1} c_q(n)z^{-n}, \quad 1 \leq q \leq P_{max}. \quad (7.9)$$

Its filter coefficients are the first l periods of the q -th Ramanujan sum. Thus, the filter length of the q -th filter is $N_q = lq$. A qualitative plot of the frequency response of the 9th Ramanujan filter is shown in Fig. 7.2. $C_q^{(l)}(z)$ has $\phi(q)$ passbands around the coprime frequencies.

Increasing l reduces the width of the passbands and the frequency response approaches the ideal Ramanujan filter $C_q(e^{j\omega})$, which consists of a set of Dirac-delta functions with infinitesimal bandwidth. When a signal $x(n)$ is passed through the q -th Ramanujan filter, the filter outputs are convolutions of $x(n)$ with filter coefficients $c_q(n)$, $0 \leq n \leq lq - 1$. When l is large, the outputs can be considered to approximately belong to S_q . A period vs. time plane plot [63], [64] is generated by plotting the windowed filter output energy as a function of time. These plots are similar to time-frequency plots. Indices of the filters whose outputs have high energy indicate the component periods of the signal. Periodicities in proteins [187] and DNAs [42] have been analyzed successfully using RFB. Multiplier-less imple-

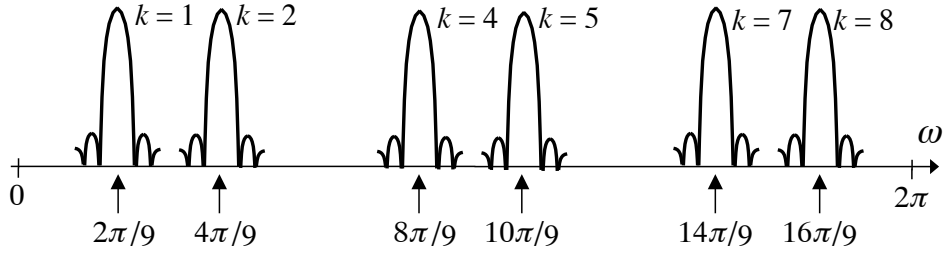


Figure 7.2: A qualitative plot of the frequency response magnitude of the FIR Ramanujan filter $C_9^{(l)}(e^{j\omega})$, when l is finite [194].

mentation [194] and efficient implementation by factoring common zeros [195] have also been proposed.

7.3 Hybrid Analysis-Synthesis Framework

The RFB has one FIR filter $C_q^{(l)}(z)$ corresponding to each q in $1 \leq q \leq P_{max}$. The RFB is not like traditional analysis filter banks for which we can typically find a stable synthesis filter bank yielding perfect reconstruction [188], primarily because of the multiband structure of the Ramanujan filters. Such a perfect reconstruction filter bank scheme would have been useful in denoising, but since it does not exist, we use a different type of synthesis mechanism to obtain the denoised signal. We propose to use RFB for analysis and Ramanujan dictionaries for synthesis in our denoising framework. Thus, we have a **hybrid** of an analysis filter bank and a synthesis dictionary. Later in Sec. 7.6, we also derive some decimation properties of Ramanujan subspace signals and discuss their application in the context of denoising.

7.3.1 Ramanujan Dictionaries for Synthesis

The synthesis dictionary in our denoising framework is generated based on the outputs of the RFB. To explain the basic idea, consider a noiseless signal $\mathbf{x} \in \mathbb{R}^N$ whose periodic components do not change over time. Consider an ideal Ramanujan filter bank with the ideal Ramanujan filters $C_q(e^{j\omega})$ for $1 \leq q \leq P_{max}$, instead of $C_q^{(l)}(z)$. Suppose the filters with indices $q = r_1, r_2, \dots, r_m$ produce non-zero outputs when \mathbf{x} is passed through the ideal RFB. Due to the ideal nature of the filters assumed here, we can conclude that the signal has non-zero frequency components possibly only at $2\pi k_{ij}/r_i$, where $(k_{ij}, r_i) = 1$, $i = 1, \dots, m$. In other words, the signal \mathbf{x} is a linear combination of the signals from the Ramanujan subspaces $S_{r_1}, S_{r_2}, \dots, S_{r_m}$. Since the columns of \mathbf{H}_q (Eq. (7.4)) span the signals from S_q that are truncated to

length N , \mathbf{x} can be expressed as

$$\mathbf{x} = \widehat{\mathbf{D}}\mathbf{b}, \text{ where } \widehat{\mathbf{D}} = \begin{bmatrix} \mathbf{H}_{r_1} & \mathbf{H}_{r_2} & \cdots & \mathbf{H}_{r_m} \end{bmatrix}. \quad (7.10)$$

Here $\widehat{\mathbf{D}}$ is a matrix of size $N \times \sum_{i=1}^m \phi(r_i)$ and \mathbf{b} is a $\sum_{i=1}^m \phi(r_i)$ dimensional coefficient vector. We shall call this operation of retaining only a few subspaces from the complete Ramanujan dictionary \mathbf{D} as **dictionary pruning**. That is, $\widehat{\mathbf{D}}$ is a pruned version of \mathbf{D} in (7.4), and the pruning is based on the output of the RFB. If \mathbf{x} consists of only a few component periods in the range $1 \leq q \leq P_{max}$, the matrix $\widehat{\mathbf{D}}$ will have a much lesser number of columns than $\Phi(P_{max})$. In summary, a signal \mathbf{x} that has no components with a period larger than P_{max} , lies in the column space of an appropriate pruned Ramanujan dictionary $\widehat{\mathbf{D}}$ as shown in Eq. (7.10).

In this work, we want to get an estimate of the noise-free version \mathbf{x} of a signal from its given noisy version $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where \mathbf{e} is additive noise to the noiseless periodic signal \mathbf{x} . The task now is to appropriately combine the information at RFB output with the noisy input signal \mathbf{y} , to extract a cleaner version $\widehat{\mathbf{x}}$ of the input. We will do this by using the appropriately pruned dictionary for synthesis. This is a unique aspect of our method.

7.3.2 Proposed Denoising Framework

Let the noisy version $\mathbf{y} = \mathbf{x} + \mathbf{e}$ be input to a practical RFB (i.e., with finite l) having filters up to P_{max} . We assume that P_{max} , the largest possible period that signal components can have, is known. As described earlier, filters with large l have narrower passband widths. In this case, the sidelobes of the filters can be assumed to be sufficiently suppressed in the passbands of other Ramanujan filters. So, under moderate noise conditions, it is reasonable to expect the filters with indices r_1, r_2, \dots, r_m to produce the outputs with much higher energies than the other filters in RFB. The small non-zero outputs of the other filters are largely due to the noise present. So now we threshold the output energies to attenuate the noise components. An appropriately chosen threshold should ideally retain only the subspace indices r_1, r_2, \dots, r_m . These are the filter index values that would have produced non-zero outputs if the noiseless signal \mathbf{x} was passed through the RFB instead of \mathbf{y} . Once the indices r_i are known, a pruned synthesis dictionary $\widehat{\mathbf{D}}$ like in Eq. (7.10) can be formed.

Now the only remaining thing is to estimate the coefficient vector \mathbf{b} . Once we have an estimate $\widehat{\mathbf{b}}$ for the unknown coefficient vector \mathbf{b} , the denoised signal is taken to

be

$$\hat{\mathbf{x}} = \widehat{\mathbf{D}}\hat{\mathbf{b}}. \quad (7.11)$$

Since the input noisy signal \mathbf{y} is known to us, we can obtain $\hat{\mathbf{b}}$ by solving the following l_1 regularized least squares optimization problem:

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b}} \|\mathbf{y} - \widehat{\mathbf{D}}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1. \quad (7.12)$$

The first term in the objective function ensures that the denoised signal has some resemblance to the input noisy signal \mathbf{y} , as expected in a denoising task. The second regularization term prefers solutions with smaller magnitudes of entries in \mathbf{b} . Such l_1 -regularized optimization problems are known to promote sparse solutions. λ is a regularization hyperparameter that can either be chosen heuristically or based on the expected noise variance. This optimization problem can be solved efficiently by convex optimization algorithms such as [196].

Since the denoised output $\hat{\mathbf{x}}$ lies in the column space of $\widehat{\mathbf{D}}$, it is composed of integer-periodic components with periods corresponding to r_1, \dots, r_m . Thus, $\hat{\mathbf{x}}$ is also guaranteed to be integer-periodic when enough samples are considered. We can ‘see’ this periodic behavior if we plot $\hat{\mathbf{x}}$ for a large enough number of samples. More samples of $\hat{\mathbf{x}}$ can be obtained by extending the pruned synthesis dictionary $\widehat{\mathbf{D}}$ to include more rows, and using the same coefficient vector $\hat{\mathbf{b}}$. The output being composed of integer-periodic components with periods smaller than P_{max} is a unique advantage of our method.

It is possible to estimate $\hat{\mathbf{b}}$ by solving the following optimization problem:

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b}} \|\mathbf{b}\|_0 \text{ s.t. } \|\mathbf{y} - \widehat{\mathbf{D}}\mathbf{b}\|_2^2 \leq \epsilon. \quad (7.13)$$

Here, ϵ is again a noise-dependent hyperparameter. This l_0 norm minimization problem is an NP-hard problem. However, it can be solved heuristically by sparse coding algorithms such as OMP [180]. In our numerical simulations, we noticed that both versions of estimating $\hat{\mathbf{b}}$ (i.e., Eq. (7.12) and Eq. (7.13)) led to very similar SNR gains, so we consistently use the computationally efficient method (Eq. (7.12)) in our simulations.

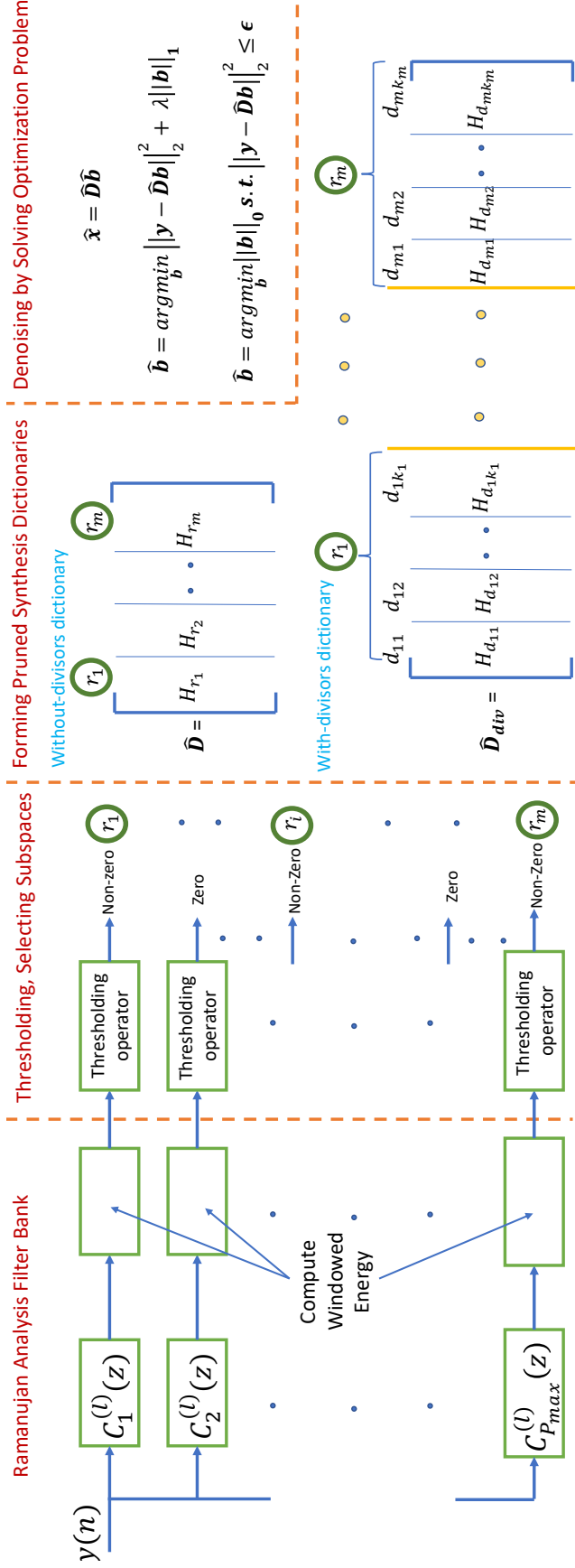


Figure 7.3: Four stages of the proposed periodicity-aware denoising framework that uses Ramanujan analysis filter bank and adaptive synthesis dictionary. See Sec. 7.3 for details.

7.3.3 Details

Consider the two operations that directly contribute to the denoising effect. First is the thresholding of the outputs of RFB, followed by dictionary pruning to eliminate certain subspaces from signal representation. And second, the optimization problems (Eq. (7.12) or Eq. (7.13)) solved to get $\hat{\mathbf{b}}$. These sparsity-promoting optimization problems with l_1 and l_0 norms can further drop some of the less significant subspaces from the representation of $\hat{\mathbf{x}}$ by setting some entries in $\hat{\mathbf{b}}$ to be zero.

A good value for the RFB threshold depends on the noise variance. In practice, it is usually found out heuristically. It is, however, possible that some of the low-energy harmonics of a *signal* produce lower energy outputs at the corresponding filter as compared to the *noise* energy at other filters. For example, for a period-10 signal having both period-2 and period-5 components, the subspaces retained after the pruning step may only be S_5 and S_{10} if the period-2 component is of low energy and gets thresholded. Since we have two denoising contributors, we can carefully balance the two to reduce such unwanted dropping of low-energy harmonics.

One option is to set the RFB output threshold conservatively. This reduces the possibility of low-energy harmonics being dropped. Here, the hope is that several other noise subspaces retained at this stage will be dropped by the optimization algorithm in the next stage. However, we did not find this option as effective in our simulations.

Another option is to set RFB thresholds slightly higher. This would almost completely eliminate all the noise subspaces. However, many of the low-energy harmonics might also get thresholded in the process. To rectify this to some extent, we manually reintroduce the ‘divisor subspaces’ that correspond to the harmonics. More precisely, in addition to the subspaces corresponding to the indices r_1, r_2, \dots, r_m , we also add subspaces corresponding to the **divisors** of r_1, r_2, \dots, r_m in the synthesis dictionary. If r_i has k_i divisors $d_{i1}, d_{i2}, \dots, d_{ik_i}$, the synthesis dictionary is given by

$$\hat{\mathbf{D}}_{div} = [\underbrace{\mathbf{H}_{d_{11}}, \dots, \mathbf{H}_{d_{1k_1}}}_{\text{divisors of } r_1}, \dots, \underbrace{\mathbf{H}_{d_{m1}}, \dots, \mathbf{H}_{d_{mk_m}}}_{\text{divisors of } r_m}]. \quad (7.14)$$

This way, the lost subspace S_2 in the above example will be reintroduced, since 2 is a divisor of 10.

For the given set of indices r_1, r_2, \dots, r_m , the dictionary generated according to Eq. (7.10) will be referred to as a ‘**without-divisors**’ dictionary, whereas the dictionary

generated according to Eq. (7.14) will be referred to as a ‘**with-divisors**’ dictionary. To summarize the above example of the period-10 signal where $r_1 = 5$ and $r_2 = 10$, the ‘without-divisors’ and ‘with-divisors’ synthesis dictionaries are

$$\widehat{\mathbf{D}} = \begin{bmatrix} \mathbf{H}_5 & \mathbf{H}_{10} \end{bmatrix}, \quad (7.15)$$

$$\widehat{\mathbf{D}}_{div} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_5 & \mathbf{H}_{10} \end{bmatrix}. \quad (7.16)$$

Consider another example, where a signal has non-zero energy output for $r_1 = 1$, $r_2 = 7$, and $r_3 = 12$ after thresholding. Then the two types of synthesis dictionaries will be

$$\widehat{\mathbf{D}} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_7 & \mathbf{H}_{12} \end{bmatrix}, \quad (7.17)$$

$$\widehat{\mathbf{D}}_{div} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \mathbf{H}_6 & \mathbf{H}_7 & \mathbf{H}_{12} \end{bmatrix}. \quad (7.18)$$

because 2, 3, 4, and 6 are divisors of 12. Fig. 7.3 shows the complete ‘periodicity-aware’ denoising framework using Ramanujan analysis filter bank and pruned synthesis dictionary. The proposed framework uses the a-priori knowledge that the noiseless signal is composed of integer-periodic components with periods up to P_{max} , and thus produces an appropriate signal as the output composed of integer-periodic components.

7.3.4 Experimental Results

In this section, we compare the denoising performance of our proposed framework with some of the well-known denoising methods from the literature. We use the SNR gain metric for comparisons. It is defined as the increase in the signal-to-noise ratio (SNR) of the denoised signal compared to that of the noisy signal. Before presenting Monte-Carlo simulations, we will present a specific instance of denoising. Fig. 7.4(a) shows a unit-norm period-12 signal. Fig. 7.4(b) shows the noisy signal obtained by adding white Gaussian noise. The SNR of the noisy signal is 0 dB. We use 100 samples of the signal for denoising. The hyperparameters of the proposed framework are chosen as follows: $P_{max} = 40$, $l = 10$, and $\lambda = 0.01$. The RFB threshold is set to 0.4 times the maximum energy output of the filter bank. So, the retained subspace indices $\mathbf{H}_{r_1}, \dots, \mathbf{H}_{r_m}$ are such that the energy of the r_i -th filter is at least 0.4 times the largest filter output energy. Later in this subsection, we also present Monte-Carlo simulation results (Fig. 7.5), where we perform a hyperparameter search for different input SNRs to obtain a good set of hyperparameters.

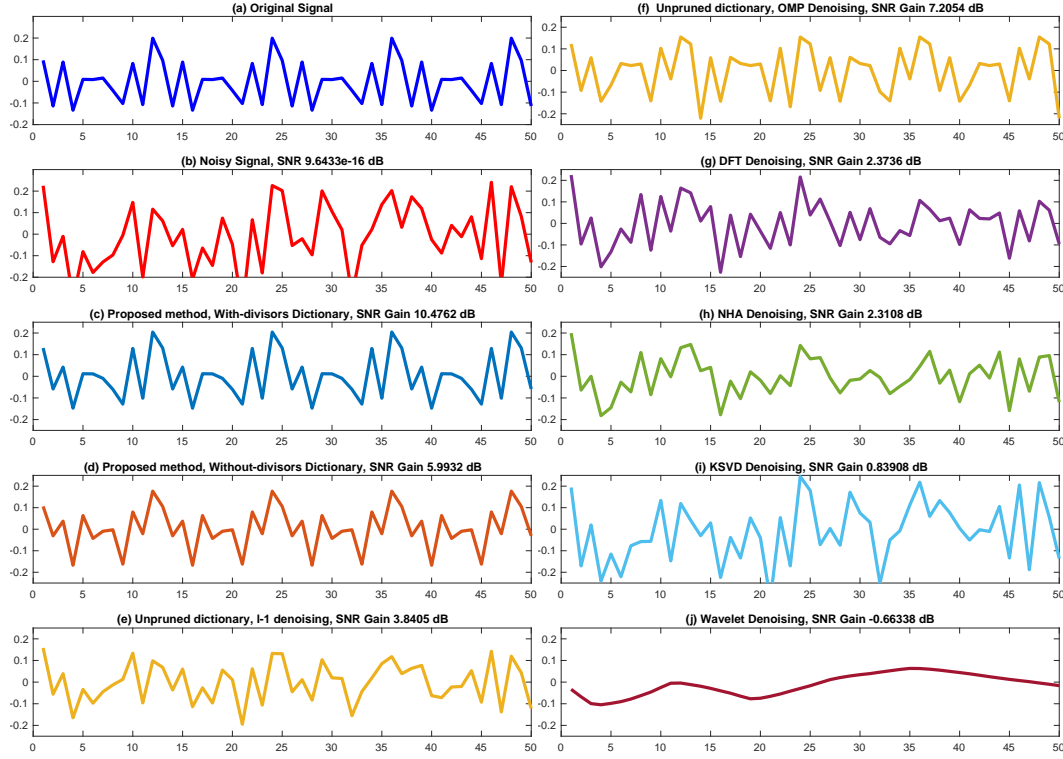


Figure 7.4: Comparison of different denoising methods for a period-12 signal. See Sec. 7.3.4 for details and discussion. (a) Original signal $x(n)$, (b) noisy version $y(n)$, and (c)-(j) denoised versions $\hat{x}(n)$ using various methods.

The denoised signal is obtained by solving Eq. (7.12). When the ‘with-divisors’ synthesis dictionary (Eq. (7.14)) is used, we get the best denoising signal as shown in Fig. 7.4(c). It has the highest SNR gain of over 10 dB. Note that the denoised signal is exactly periodic and visually quite similar to the original signal. The ‘without-divisors’ synthesis dictionary (Eq. (7.10)) loses out on a lower energy harmonic and produces a much lower SNR gain (Fig. 7.4(d)). This justifies the inclusion of divisor subspaces in our synthesis dictionary. Denoising with the l_1 reconstruction (Eq. (7.12)) and the l_0 reconstruction (Eq. (7.13)) were found to produce similar SNR gains with pruned dictionaries, so we consistently use l_1 reconstruction in our proposed method. Next, Fig. 7.4(e) shows the denoised signal obtained by l_1 reconstruction using the full (i.e., unpruned) Ramanujan dictionary as in Eq. (7.4). The low SNR gain with the unpruned dictionary justifies the need for the crucial dictionary pruning step. The OMP denoising with the unpruned dictionary (Fig. 7.4(f)) provided a good SNR gain in this case, but on average it is found to perform poorer than the ‘without-divisors’ dictionary method as we will see later in Fig. 7.5.

Comparing the denoised output of the proposed method with the results from Fig. 7.4(g-j), we see that our method clearly outperforms DFT, NHA [181], wavelet [179], and K-SVD [76] based denoising methods. None of these methods yields a truly periodic signal. The DFT denoising (Fig. 7.4(g)) does not perform that well here because the signal period, 12, is not a divisor of the signal length 100, and thus the signal frequency components do not lie on the DFT frequency grid.

The NHA-based denoising method [181] requires a good initialization from DFT to perform well. Also, it suffers from error propagation due to the iterative addition of frequency components in its denoising steps. These reasons limit the denoising ability of NHA (Fig. 7.4(h)). The data-adaptive K-SVD denoising technique also performs poorly (Fig. 7.4(i)). One possible reason is that with the small signal length, K-SVD is not able to learn a good dictionary from the patches of the noisy signal. The wavelet denoising (Fig. 7.4(j)) produces a completely unacceptable denoising result. Wavelets provide a good representation for signals with time-localized supports, such as, for example, bump signals [184], whereas the periodic signals are very different from this.

Note that implementing many of the denoising methods requires choosing some hyperparameters. For example, our proposed method has three hyperparameters, namely, the regularization level λ , the RFB threshold, and the filter lengths $N_q = lq$ that are controlled by the hyperparameter l . The P_{max} value is assumed to be known a-priori. Similarly, OMP denoising requires a threshold ϵ , NHA requires similar iteration terminating hyperparameters, and KSVD-based denoising requires choosing the number of dictionary columns and signal patch size.

For the Monte-Carlo simulations that we perform next, we have access to the noiseless ground truth signal of the synthetically generated dataset. Thus, we perform a hyperparameter search for each of the denoising methods at every input SNR by joint search over appropriate sets of values. For the proposed method, we jointly search energy threshold level and regularization level λ by joint search from the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ respectively, while keeping $l = 10$ fixed. This hyperparameter search process based on the synthetic data can be considered analogous to the ‘validation phase’ from machine learning based methods performed to tune the hyperparameters. In practice, we only receive a noisy signal. Even though we do not have access to the noiseless ground truth signal in this case, we assume that an estimate of the noise level (i.e., SNR of the noisy signal) is available. Then, to perform denoising, we can use the values of

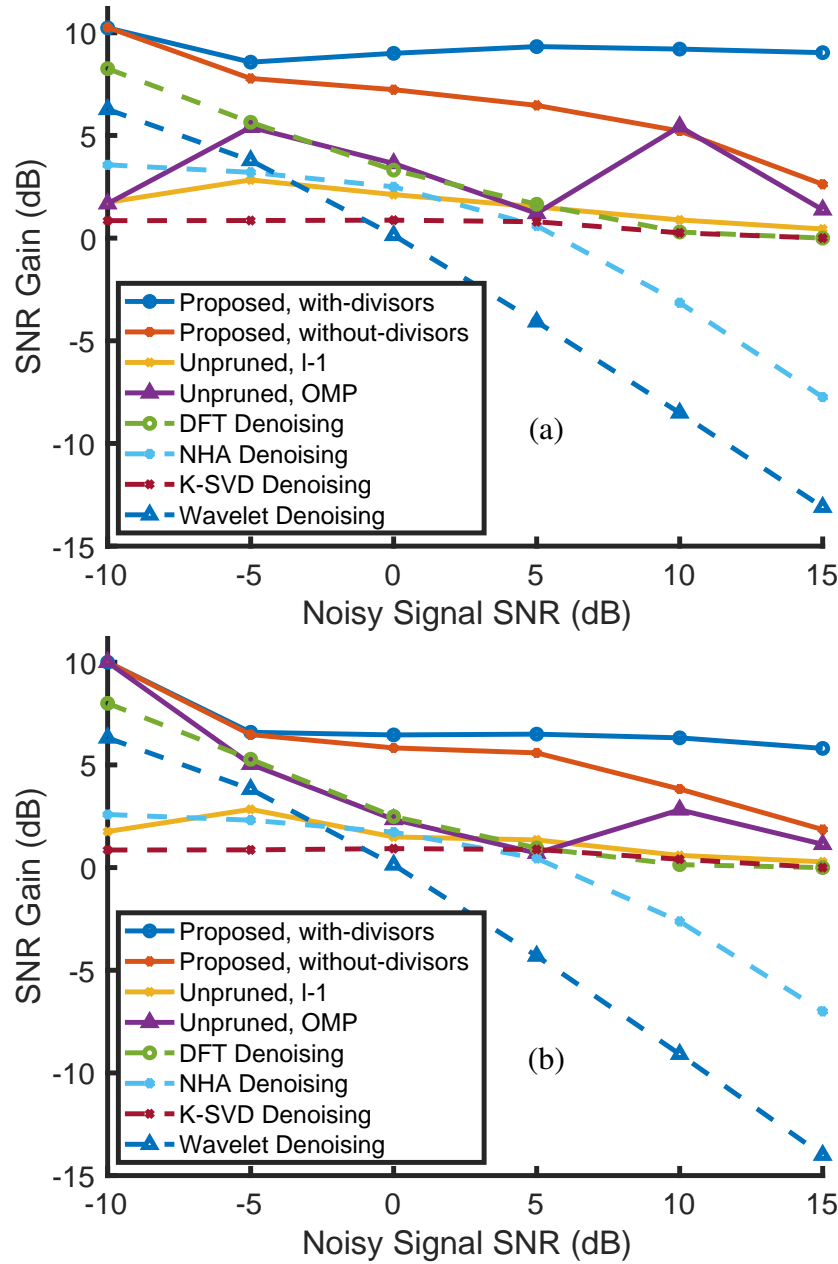


Figure 7.5: Comparison of SNR gains offered by different denoising methods averaged over 1000 Monte-Carlo runs consisting of (a) period-12 signals and (b) sum of period-7 and period-12 (effective period 84) signals.

hyperparameters that provided the best SNR gain at the particular input SNR in the hyperparameter search phase.

Now we perform Monte-Carlo simulations by generating 1000 period-12 signals randomly and adding Gaussian noise to them. Fig. 7.5(a) shows the SNR gains offered at various noise levels in the input (when hyperparameters found by the

hyperparameter search, as explained above, are used). The proposed method that uses the ‘with-divisors’ dictionary offers the best SNR gain that is stable over a large range of input SNR values, followed next by the ‘without-divisors’ dictionary method. To compare the denoising performance for signals with multiple periodic components, we also considered signals that are combinations of period-7 and period-12 signals. Since the effective period of such signals is $7 \times 12 = 84$, data with only 100 samples hardly even ‘looks’ periodic. Even in this case, the proposed denoising method demonstrated results superior to the other methods as seen in Fig. 7.5(b).

7.4 Optimizing Ramanujan Filters Using Capon Beamforming Principles

In our denoising framework, the energy of the q -th RFB filter output indicates the presence or absence of the period q component. Ideally, we would want the outputs of the q -th Ramanujan filter to be nonzero only when the signal contains the coprime frequency components corresponding to q (i.e., $2\pi k/q$ where k and q are coprime). However, the passbands of practical Ramanujan filters $C_q^{(l)}(z)$ have non-zero bandwidth (Fig. 7.2). So it is possible that $C_q^{(l)}(z)$ does not completely attenuate frequency components that correspond to other periods and lie close to some $k \cdot 2\pi/q$, $(k, q) = 1$. As an example, consider a period-18 signal with SNR 2 dB. Fig. 7.6(a) shows the period vs. time plane plot produced by the RFB. Many filters around index $q = 18$ and $q = 9$ (a harmonic of period-18 signal) produce high energy outputs. In addition to this, several other filters corresponding to larger periods also produce high-energy outputs. In such cases, it may not be possible to find a good threshold that retains only the appropriate subspaces in the synthesis dictionary.

One way to prevent this is to make the filter passbands narrower by increasing l for $C_q^{(l)}(z)$. However, this increases the time domain filter length ql , resulting in poorer time domain localization and increased transient output samples. To avoid this, we propose to adapt each filter in the RFB to the incoming data. We want the q -th filter to pass the set of $\phi(q)$ coprime frequencies unattenuated and almost completely attenuate all other frequency components present in the signal. To achieve this, we use the principle of minimum variance distortionless response (MVDR) beamformer or Capon’s method [1], [197] from array signal processing. We show that optimizing each filter this way greatly improves the period vs. time plane plots. As a preview, Fig. 7.6(b) shows the period vs. time plane plot obtained at the output of the Capon-optimized filter bank. It is much cleaner than the period

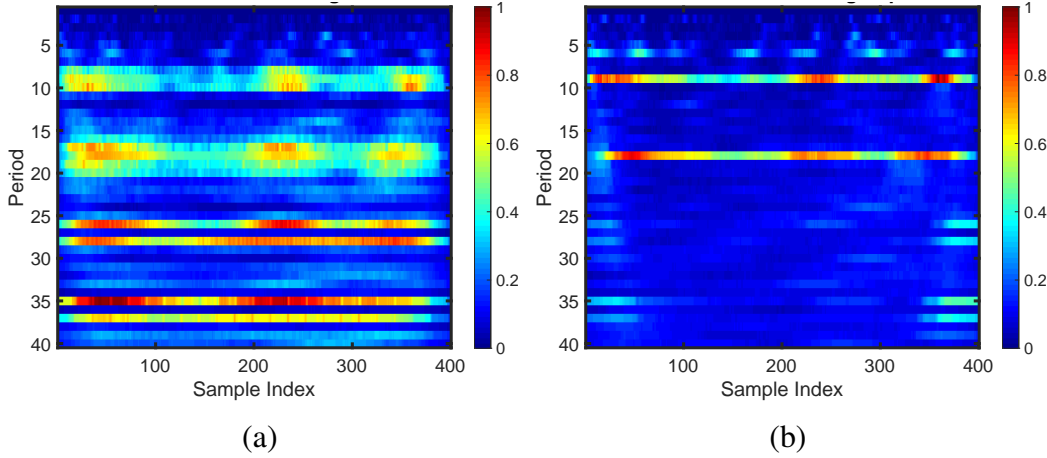


Figure 7.6: Period vs. time plane plot [63], [64] of a period-18 signal, produced by (a) RFB and (b) Capon-optimized filter bank.

vs. time plane plot produced by RFB and has significant energy outputs only for filters with $q = 18$, $q = 9$, and $q = 6$, which are the correct harmonics of the period-18 signal under consideration. This data-adaptive method of optimizing filters in the analysis bank reduces the possibility of selecting incorrect Ramanujan subspaces and thus improves the average denoising SNR gain, as we will see.

7.4.1 Review of Capon Beamforming

In a typical setting in the field of array signal processing, an N -sensor uniform linear array (ULA) is used for receiving signals from different spatial directions. The inter-sensor spacing is assumed to be $\mu/2$, where μ is the wavelength of the incoming electromagnetic waves. Suppose we want to receive the signal from a particular direction θ_0 in space, but there are interfering signals coming from other directions $\theta_1, \theta_2, \dots, \theta_D$ as well. In this setting, the array output $\mathbf{x} \in \mathbb{C}^N$ is modeled as [1]

$$\mathbf{x} = c_0 \mathbf{a}(\omega_0) + \mathbf{A} \mathbf{c} + \mathbf{e}. \quad (7.19)$$

Here, the ‘frequency variables’ $\omega_0, \omega_1, \dots, \omega_D \in [-\pi, \pi)$ are related to spatial angles of arrivals $\theta_0, \theta_1, \dots, \theta_D$ as $\omega_i = \pi \sin \theta_i$. The steering vector corresponding to direction ω is defined as $\mathbf{a}(\omega) = [1, e^{j\omega}, \dots, e^{j(N-1)\omega}]^T$, and the $N \times D$ array manifold matrix \mathbf{A} is defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(\omega_1) & \mathbf{a}(\omega_2) & \dots & \mathbf{a}(\omega_D) \end{bmatrix}. \quad (7.20)$$

Here \mathbf{e} represents noise, and $\mathbf{c} = [c_1, \dots, c_D]^T$ where the source amplitudes c_i are assumed to be random variables.

A beamformer $\mathbf{h} \in \mathbb{C}^N$ computes a linear combination $y = \mathbf{h}^H \mathbf{x}$ of the array output \mathbf{x} . Note that $\mathbf{h}^H \mathbf{a}(\omega) = H^*(e^{j\omega})$, where $H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}$ is the discrete-time Fourier transform of the beamformer \mathbf{h} . With this,

$$y = \mathbf{h}^H \mathbf{x} = c_0 H^*(e^{j\omega_0}) + \sum_{i=1}^D c_i H^*(e^{j\omega_i}) + \mathbf{h}^H \mathbf{e}. \quad (7.21)$$

To receive the signal from a particular direction θ_0 , the beamformer should have a fixed gain $H(e^{j\omega_0}) = 1$ for the ‘look direction’ ω_0 and $H(e^{j\omega_i}) = 0$ for $i = 1, \dots, D$. Since the interfering directions are not known, the Capon beamformer aims to reduce the combined effect of interference and noise in a statistical sense based on the covariance of the received data. Assuming that source amplitudes c_i are zero-mean random variables independent of each other and also independent of the zero-mean noise \mathbf{e} , the mean square value of the beamformer output y is given by

$$\begin{aligned} \mathbb{E}[|y|^2] &= \mathbf{h}^H \mathbf{R}_{\mathbf{xx}} \mathbf{h} \\ &= p_0 |H(e^{j\omega_0})|^2 + \sum_{i=1}^D p_i |H(e^{j\omega_i})|^2 + \mathbf{h}^H \mathbf{R}_{\mathbf{ee}} \mathbf{h}, \end{aligned} \quad (7.22)$$

where $\mathbf{R}_{\mathbf{xx}}$ is the covariance of array output, $p_i = \mathbb{E}[c_i c_i^H]$ is the power of the i -th source, and $\mathbf{R}_{\mathbf{ee}}$ is noise covariance. Under the look direction constraint $H(e^{j\omega_0}) = 1$, minimizing the effect of interferers and noise is equivalent to minimizing $\mathbb{E}[|y|^2]$. Thus, finding the optimal beamformer is posed as the following optimization problem:

$$\min_{\mathbf{h} \in \mathbb{C}^N} \mathbf{h}^H \mathbf{R}_{\mathbf{xx}} \mathbf{h} \quad \text{s.t.} \quad \mathbf{h}^H \mathbf{a}(\omega_0) = 1. \quad (7.23)$$

It can be shown [1], [197] that when $\mathbf{R}_{\mathbf{xx}}$ is invertible, the solution to this problem is given by

$$\mathbf{h} = \frac{\mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{a}(\omega_0)}{\mathbf{a}^H(\omega_0) \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{a}(\omega_0)}. \quad (7.24)$$

In practice, we only have an estimate $\hat{\mathbf{R}}_{\mathbf{xx}}$ for the covariance matrix, obtained from a finite number of snapshots. So, $\hat{\mathbf{R}}_{\mathbf{xx}}$ which is used in place of $\mathbf{R}_{\mathbf{xx}}$ in (7.24) to get the beamformer coefficients.

7.4.2 Multi-Band Capon Formulation to Optimize Analysis Filters

Note that the optimal beamformer \mathbf{h} can be equivalently interpreted as a standard N -tap FIR filter with impulse response $h(n)$ having a passband around the look direction ω_0 . When N is large enough, some of the zeros of the optimal Capon

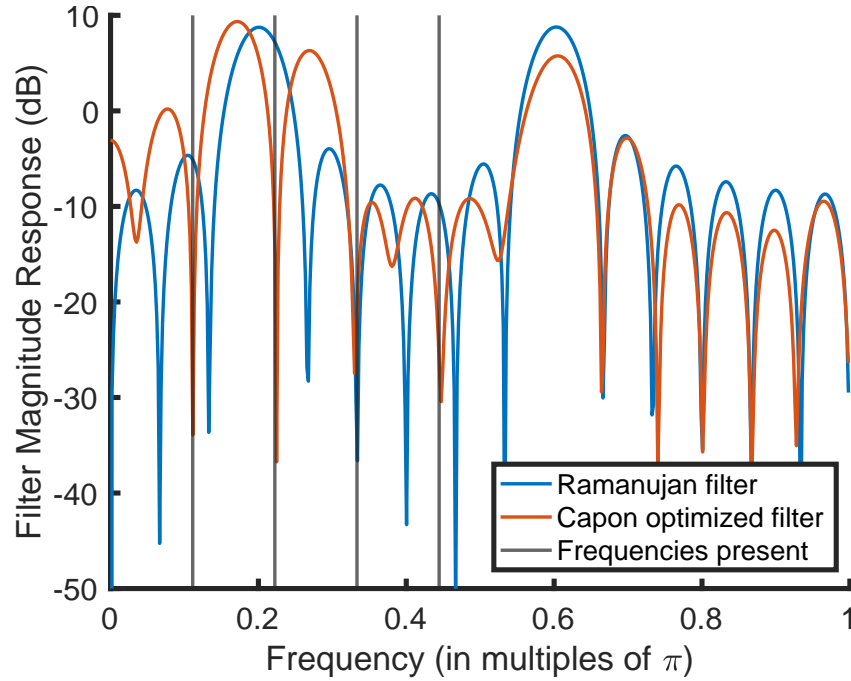


Figure 7.7: Magnitude response of the Capon-optimized filter \mathbf{h}_{10} closely approximates the Ramanujan filter $C_{10}^{(l)}(z)$ in the passbands and puts zeros very close to the other frequency components present in the signal (denoted with black vertical lines).

beamformer typically lie very close to the unit circle in the z -plane with phase values that are very close to the interfering signal frequencies ω_i . Thus, the interferences are almost completely attenuated by the Capon beamformer.

Now consider a q -th Ramanujan filter in our denoising framework. We want its output to contain only the coprime frequency components corresponding to the period q , and attenuate the frequency components corresponding to other periods. This is very similar to the Capon beamformer setting. The frequency components corresponding to other periods are like unknown interferences. The difference, however, is that we want the filter frequency magnitude to be unity at multiple frequencies $k \cdot 2\pi/q$ with $(k, q) = 1$ and not just one ‘look direction’ ω_0 . We can get such a multi-banded filter by solving the following multi-constrained Capon optimization problem:

$$\begin{aligned} \mathbf{h}_q &= \arg \min_{\mathbf{h} \in \mathbb{C}^{N_q}} \mathbf{h}^H \mathbf{R}_q \mathbf{h} \\ \text{s.t. } \quad &\mathbf{h}^H \mathbf{a}(\omega_k) = 1, \omega_k = 2\pi k/q, \\ &1 \leq k \leq q, \quad (k, q) = 1. \end{aligned} \tag{7.25}$$

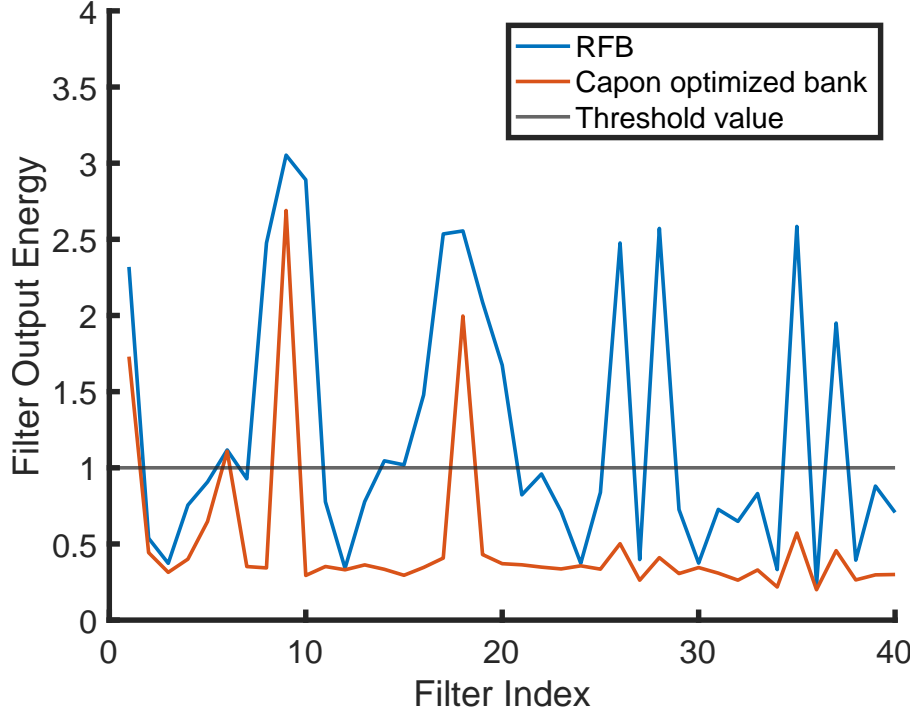


Figure 7.8: Total output energies for each filter in RFB and Capon-optimized banks. The Capon-optimized filter bank does not produce spurious peaks for higher period filters, thus making a good choice of threshold easy.

Here the q -th filter \mathbf{h}_q has length N_q . \mathbf{R}_q is the estimated $N_q \times N_q$ correlation matrix of the incoming data:

$$\mathbf{R}_q = \frac{1}{N - N_q + 1} \sum_{i=1}^{N-N_q+1} \mathbf{y}(i : i + N_q - 1) \mathbf{y}^H(i : i + N_q - 1), \quad (7.26)$$

where \mathbf{y} is the length- N noisy periodic signal to be denoised. Such a multi-constrained Capon formulation has been studied in the past [198]–[200]. The optimal solution to (7.25) is given by

$$\mathbf{h}_q = \mathbf{R}_q^{-1} \mathbf{A}_q (\mathbf{A}_q^H \mathbf{R}_q^{-1} \mathbf{A}_q)^{-1} \mathbf{1}_{\phi(q)}. \quad (7.27)$$

In our application, $\mathbf{A}_q = \begin{bmatrix} \mathbf{a}(\omega_1) & \dots & \mathbf{a}(\omega_{\phi(q)}) \end{bmatrix}$ where, $\omega_1, \dots, \omega_{\phi(q)}$ are the $\phi(q)$ coprime passband frequencies and $\mathbf{1}_{\phi(q)}$ denotes the $\phi(q)$ -dimensional vector containing all ones.

7.4.3 Denoising using Capon-Optimized Filter Bank

Before comparing the denoising performance with RFB and Capon-optimized filter bank, let's compare their filter magnitude responses. In our examples, we set the

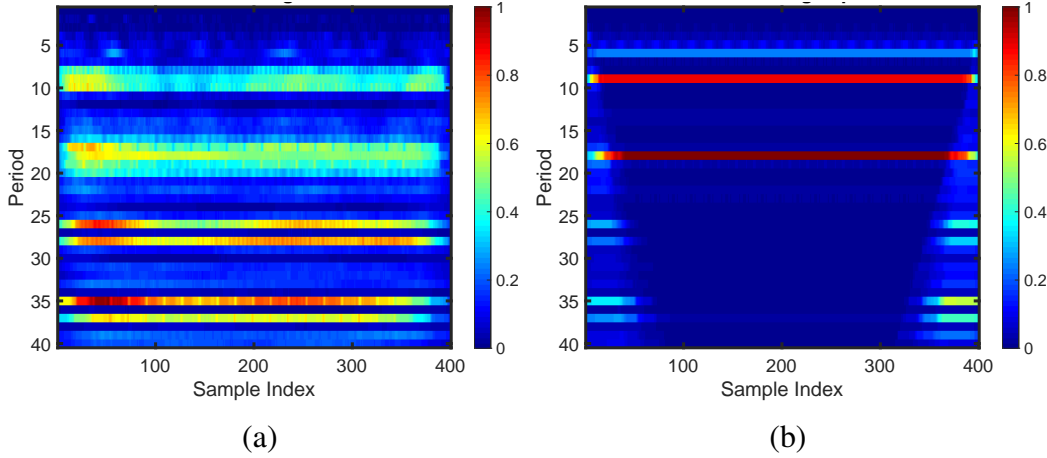


Figure 7.9: Period vs. time plane plot after denoising the period-18 signal considered in Fig. 7.6. Denoising and plotting Period vs. time plane plots using (a) RFB and (b) Capon-optimized filter bank.

length of q -th filters in both RFB and Capon-optimized filter banks to be $N_q = 3q$. The RFB filters are independent of the signal under consideration, whereas Capon-optimized filters depend on the data correlation matrix. Consider again the period 18 signal with 3 harmonic components and SNR of 2 dB that was used in Fig. 7.6. Fig. 7.7 shows the magnitude responses of the Ramanujan filter and the Capon-optimized filter (Eq. (7.27)) corresponding to $q = 10$ for this signal. The Capon-optimized filters achieve high attenuation at the frequencies corresponding to signal components that correspond to periods other than q . For this reason, the Capon-optimized filters provided much cleaner period vs. time plane plots, as was seen in Fig. 7.6. Note that in the passbands, the magnitude responses of the q -th Ramanujan filter and q -th Capon-optimized filter were found to be very similar in most cases.

Fig. 7.8 shows the total output energy produced by each filter in the RFB and Capon-optimized filter banks. Unlike the RFB, the Capon-optimized filter bank does not produce spurious high-energy peaks, especially for the filters corresponding to higher periods. This makes the selection of a good threshold to retain appropriate subspaces in the synthesis dictionary easier, as shown in the figure. As one would expect, this further improves the denoising performance. Fig. 7.9 shows the period vs. time plane plots of the signal considered in Fig. 7.6 after denoising. Denoising and plotting the period vs. time plane plots is done with RFB in Fig. 7.9(a) and Capon-optimized bank in Fig. 7.9(b). We can see that the period vs. time plane plot for the signal denoised with Capon-optimized bank has practically zero energy

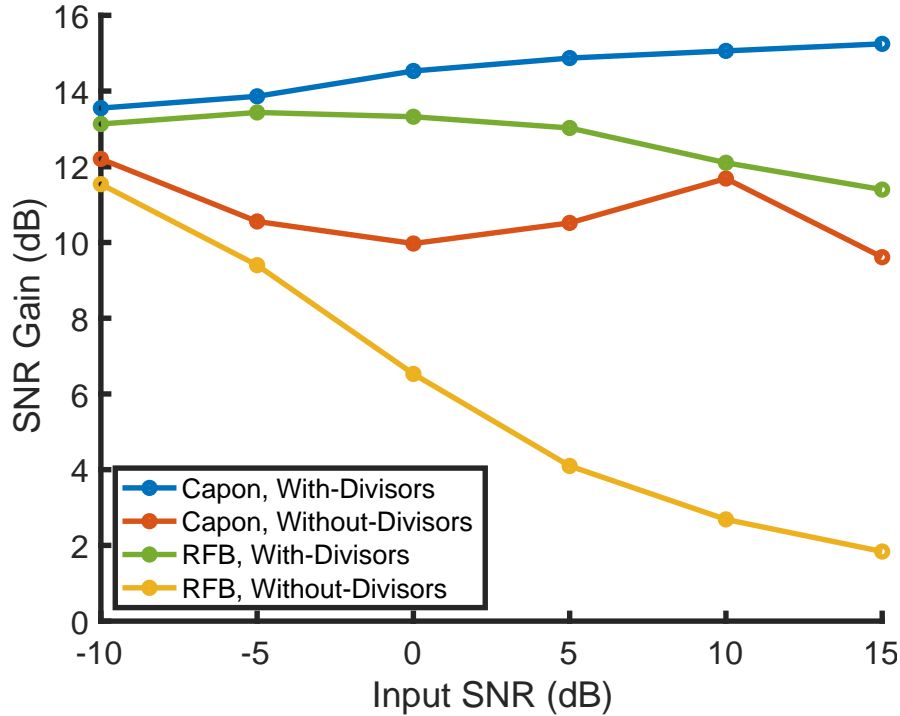


Figure 7.10: Comparison of SNR gains obtained by the hybrid denoising system when the analysis bank is (i) RFB and (ii) Capon-optimized. The plots are obtained by averaging the performance over 1000 Monte Carlo runs for period-12 signals.

outputs for filters other than those corresponding to 18, 9, and 6, which are the harmonics of the period-18 signal considered. The small non-zero energies at the start and the end for the other filters are due to transient portions of the convolution outputs, where not all filter coefficients are involved in the convolution. The filter length $N_q = 3q$ increases with the filter index q , and so does the length of transients.

Fig. 7.10 shows the average SNR gain obtained with the Capon-optimized filter bank and RFB filter bank for 1000 randomly generated period-12 signals of length 400. We consider denoising using both the ‘with-divisors’ dictionary and the ‘without-divisors’ dictionary for each filter bank. The filter length is chosen to be $N_q = 3q$ for both filter banks. Note that the filter length used here is much smaller than that used in the previous simulations from Sec. 7.3.4. We see that the combination of the Capon-optimized filter bank and the ‘with-divisors’ dictionary consistently provides the best average SNR gain between 12-15 dB. Several other comparative simulations are presented in Sec. 7.7.

7.5 Discussions on Computational Complexity

The difference in computational complexities of the RFB-based denoising framework and the Capon-based denoising framework arises because of the way filters are designed. Implementing Ramanujan filters (Eq. (7.9)) is not computationally expensive, as they have data-independent fixed filter coefficients defined based on Ramanujan sums. On the other hand, implementing Capon-optimized filters first requires computing filter coefficients based on data. Computing the q -th Capon-optimized filter coefficients requires computing the $N_q \times N_q$ data correlation matrix \mathbf{R}_q (Eq. (7.26)) and then evaluating Eq. (7.27). Inverting \mathbf{R}_q requires $\mathcal{O}(q^3)$ computations. Furthermore, inverting the expression $\mathbf{A}_q^H \mathbf{R}_q^{-1} \mathbf{A}_q$ from (7.27) also requires $\mathcal{O}(\phi(q)^3) = \mathcal{O}(q^3)$ computations. This is the cost we have to pay for the improved denoising performance.

For streaming (or non-stationary) data, it is possible to update the \mathbf{R}_q^{-1} term iteratively as we receive new samples instead of explicitly evaluating \mathbf{R}_q^{-1} , to save some computational cost. We do not consider streaming (or non-stationary) data in this chapter, as it requires several careful modifications to the current method that vary depending on the nature of how periodicity structure changes over time. For the stationary signals (whose periodicity structure does not change over time) considered in this chapter, the ideal correlation matrix of the signal is Toeplitz. So instead of generating separate $N_q \times N_q$ autocorrelation matrices for each q as in Eq. (7.26), we can generate a single $N_{P_{max}} \times N_{P_{max}}$ autocorrelation matrix $\mathbf{R}_{P_{max}}$ by setting $q = P_{max}$ in Eq. (7.26). We then ‘Toeplitzize’ it by averaging the entries along all the diagonals of the matrix to get a Toeplitz estimate $\hat{\mathbf{R}}$ for the autocorrelation matrix as follows:

$$\hat{\mathbf{R}}(i, j) = \sum_{\substack{1 \leq i', j' \leq N_{P_{max}} \\ i' - j' = i - j}} \mathbf{R}_{P_{max}}(i', j'). \quad (7.28)$$

Let $\hat{\mathbf{R}}_q$ denote the $N_q \times N_q$ leading principal submatrix of $\hat{\mathbf{R}}$. This $\hat{\mathbf{R}}_q$ can be taken as an estimate of the ideal $N_q \times N_q$ Toeplitz autocorrelation matrix and can be used in place of \mathbf{R}_q in (7.27). When the signal length is large enough, the submatrices of $\hat{\mathbf{R}}$ will closely approximate the \mathbf{R}_q computed according to Eq. (7.26). There is an important computational advantage to using this replacement. It is possible to find inverses of all $\hat{\mathbf{R}}_q$ from $q = 1$ to $q = P_{max}$ in $\mathcal{O}(P_{max}^3)$ computation by leveraging Levinson’s recursion from the linear prediction theory [201] as we explain next.

In linear prediction theory, the task is to predict the next signal sample for $x(n)$ from

its past N samples:

$$\hat{x}(n) = \sum_{i=1}^N -a_{N,i}^* x(n-i), \quad e(n) = x(n) - \hat{x}(n). \quad (7.29)$$

Here, $\hat{x}(n)$ is the predicted value, and $e(n)$ is the prediction error. The N -th order optimal linear prediction coefficients satisfy

$$\hat{\mathbf{R}}_{N+1} \mathbf{a}_N = \begin{bmatrix} \epsilon_N & 0 & 0 & \dots & 0 \end{bmatrix}^T, \quad (7.30)$$

where $\hat{\mathbf{R}}_{N+1}$ is the $(N+1) \times (N+1)$ Toeplitz autocorrelation matrix of $x(n)$, $\mathbf{a}_N = \begin{bmatrix} 1 & a_{N,1} & a_{N,2} & \dots & a_{N,N} \end{bmatrix}^T$ contains the N prediction coefficients $\{a_{N,i}\}_{i=1}^N$, and $\epsilon_N = \mathbb{E}[|e(n)|^2]$ is the mean squared prediction error. Eq. (7.30) is known as the augmented normal equation. It is well-known [201] that using Levinson's recursion it is possible to find the optimal predictor coefficient vectors \mathbf{a}_1 to \mathbf{a}_N and corresponding prediction errors ϵ_1 to ϵ_N in $\mathcal{O}(N^2)$ computations. Putting together the augmented normal equations for orders 1 to N , it can be shown [201] that

$$\Delta_{N+1}^H \hat{\mathbf{R}}_{N+1} \Delta_{N+1} = \Lambda_{N+1}, \quad (7.31)$$

where

$$\Delta_{N+1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_{N,1} & 1 & \dots & 0 \\ a_{N,2} & a_{N-1,1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,N} & a_{N-1,N-1} & \dots & 1 \end{bmatrix} \quad (7.32)$$

is a lower triangular matrix containing predictor coefficients of all orders, and $\Lambda_{N+1} = \text{diag}(\epsilon_N, \epsilon_{N-1}, \dots, \epsilon_0)$ is a diagonal matrix containing prediction errors. From Eq. (7.31) we get

$$\hat{\mathbf{R}}_{N+1}^{-1} = \Delta_{N+1} \Lambda_{N+1}^{-1} \Delta_{N+1}^H = \sum_{n=0}^{n=N} \frac{\bar{\mathbf{a}}_n \bar{\mathbf{a}}_n^H}{\epsilon_n}, \quad (7.33)$$

where $\bar{\mathbf{a}}_n$ is the $(n+1)$ -th column of Δ_{N+1} . Using this, we can compute the inverses recursively as follows:

$$\hat{\mathbf{R}}_{N+1}^{-1} = \frac{\mathbf{a}_N \mathbf{a}_N^H}{\epsilon_N} + \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_N^{-1} \end{bmatrix}. \quad (7.34)$$

Here, the N -th recursive step requires only $\mathcal{O}(N^2)$ computations as all the predictor coefficients and prediction errors can be computed by Levinson's recursion in $\mathcal{O}(N^2)$. Thus, going back to our discussion on implementing the Capon-optimized

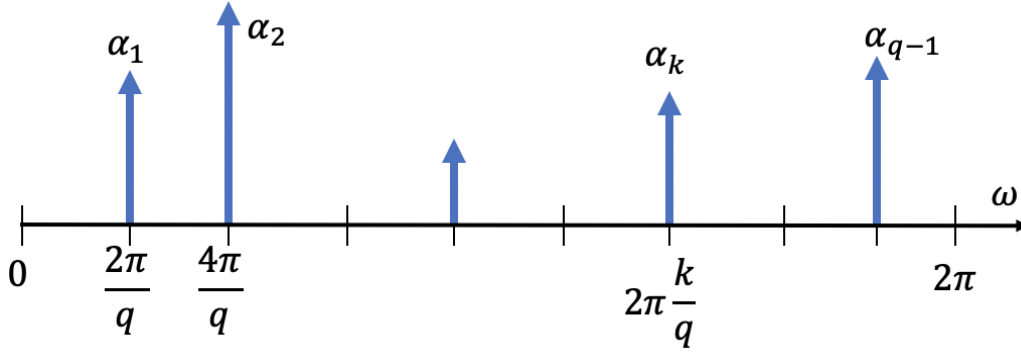


Figure 7.11: Frequency domain representation of a signal $x(n) \in S_q$. Decimated signal $x(Mn)$ still belongs to S_q with permuted α_k s if and only if $(M, q) = 1$. See Lemma 8.

analysis filters from Eq. (7.27), using the Toeplitz submatrices $\widehat{\mathbf{R}}_q$ in place of \mathbf{R}_q allows us to compute the $\widehat{\mathbf{R}}_q^{-1}$ for $q = 1$ to P_{max} with $O(P_{max}^3)$ computations. This saves a considerable number of computations as we need to evaluate (7.26) for a single $q = P_{max}$, ‘Toeplitzize’ it, and then efficiently find inverses for all leading principal submatrices recursively.

However, computing the inverse of $(\mathbf{A}_q^H \mathbf{R}_q^{-1} \mathbf{A}_q)$ required in the expression (7.27) still requires $O(\phi(q)^3)$ for every q . The presence of matrices \mathbf{A}_q prevents the recursive efficient computation of these inverses.

7.6 Multirate Properties of Ramanujan Subspace Signals

In this section, we derive some multirate properties of Ramanujan subspace signals. These properties allow us to decimate the outputs of the filters in the Ramanujan filter bank, without losing information. Thus, we can use some of the multirate properties to further reduce the required computations by decimating the filter bank outputs in the proposed denoising framework. These ideas are also applicable to the Capon-optimized version of the RFB. Strictly speaking, the theoretical results are applicable only to signals in S_q which have infinite time duration and infinitesimal bandwidth (Fig. 7.11). But, we also discuss the applicability of the results for the outputs of the realizable analysis filters, which have nonzero bandwidth around the coprime frequencies.

For this section, we will require a basic result from number theory, which we restate from [61] using our notations and conventions. As [61] does not provide explicit proof for the result, we include it here for clarity and completeness.

Lemma 7 (Thm. 58 from [61]). Let $M, q \in \mathbb{N}$ be such that $(M, q) = 1$. Then the two sets $\mathcal{A} = \{k \mid 0 \leq k \leq q-1, (k, q) = 1\}$ and $\mathcal{B} = \{kM \bmod q \mid 0 \leq k \leq q-1, (k, q) = 1\}$ are identical.

Proof. If $(k, q) = 1$ for some $1 \leq k \leq q-1$, we have $(kM, q) = 1$ as $(M, q) = 1$. Thus $(kM - ql, q) = 1$ for any $l \in \mathbb{Z}$ and in particular $((kM \bmod q), q) = 1$. Since $0 \leq kM \bmod q \leq q-1$, we can conclude that $\mathcal{B} \subset \mathcal{A}$. Since \mathcal{A} has $\phi(q)$ distinct entries, it suffices to show that all entries in \mathcal{B} are distinct. If they are not, let $0 \leq k_1 < k_2 \leq q-1$ be such that

$$k_1 M \bmod q = k_2 M \bmod q. \quad (7.35)$$

This implies $(k_2 - k_1)M = ql$ for some $l \in \mathbb{N}$, i.e.,

$$\frac{(k_2 - k_1)}{l} = \frac{q}{M} \text{ for some } l \in \mathbb{N}. \quad (7.36)$$

Since $0 < k_2 - k_1 < q-1$, Eq. (7.36) contradicts $(M, q) = 1$. Thus, \mathcal{B} has $\phi(q)$ distinct entries and $\mathcal{A} = \mathcal{B}$. \square

7.6.1 Decimating Ramanujan Subspace Signals

In this subsection, we explore what happens when an ideal infinite length $x(n) \in S_q$ is decimated. We have the following result.

Lemma 8 (Coprime decimation property of Ramanujan subspace signals). Let $x(n) \in S_q$, the q -th Ramanujan subspace, and M be the decimation ratio. Then the decimated signal $y(n) = x(Mn)$ still belongs to S_q if and only if $(M, q) = 1$.

Proof. If part. Since $x(n) \in S_q$, it can be represented as in Eq. (7.2). Since $W_q^m = W_q^{m-lq} = W_q^{(m \bmod q)}$ for any $l \in \mathbb{Z}$, we get

$$x(Mn) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_q^{kMn} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_q^{(kM \bmod q)n}$$

Now let $k' = kM \bmod q$. Using Lemma 7, we can conclude that when $(M, q) = 1$,

$$x(Mn) = \sum_{\substack{k'=0 \\ (k',q)=1}}^{q-1} \bar{\alpha}_{k'} W_q^{k'n} \in S_q. \quad (7.37)$$

where $\bar{\alpha}_{k'} = \alpha_k$. This concludes *if part* of the proof. The *only if part* is a consequence of Lemma 10 proved later. \square

The following are a few important observations based on the above lemma and its proof:

- From Eq. (7.37) we can see that the coprime decimation of a Ramanujan subspace signals $x(n)$ only permutes the coefficients α_k among the coprime frequency locations.
- Since the permutation of coefficients α_k does not produce any loss of information, it follows that there is no loss of information (or ‘aliasing’) when $x(n) \in S_q$ is decimated by M such that $(M, q) = 1$.
- One way to recover $x(n)$ back from $y(n) = x(Mn)$ is to decimate $y(n)$ again by L such that $ML \bmod q = 1$. When M and q are coprime, such an L always exists because of Euclid’s lemma [61]. Indeed, $\hat{x}(n) = y(Ln) = x(MLn) = x((ML \bmod q)n) = x(n)$. The second last equality follows because $x(n) \in S_q$ has period q .

Ramanujan sum $c_q(n)$ belongs in S_q and has $\alpha_k = 1 \forall k$ when expressed in the form of Eq. (7.2). Thus, as a corollary of Lemma 8, we have the following interesting coprime decimation identity for Ramanujan sums.

Corollary 1 (Coprime decimation identity for Ramanujan sums). $c_q(Mn) = c_q(n)$ for any M such that $(M, q) = 1$.

We know that the Ramanujan subspace S_q does not include all period q signals [52]. The set of all period- q signals does not form a subspace, as the addition of period- q signals can result in a signal with a period that is a divisor of q . However, the set of all signals with repetition index q does form a subspace. We denote it as R_q . Recall that for a periodic signal $x(n)$, the period P is the smallest integer such that $x(n + P) = x(n) \forall n$, whereas any multiple of P is a repetition index of $x(n)$. If $x(n) \in R_q$, its period (smallest repetition interval) can be any divisor of q . Thus,

$$R_q = \bigoplus_{q_i | q} S_{q_i}, \quad (7.38)$$

where \oplus denotes the outer sum. Since Ramanujan subspaces are orthogonal to each other [52], every $x(n) \in R_q$ can be uniquely expressed as

$$x(n) = \sum_{q_i | q} x_{q_i}(n), \quad (7.39)$$

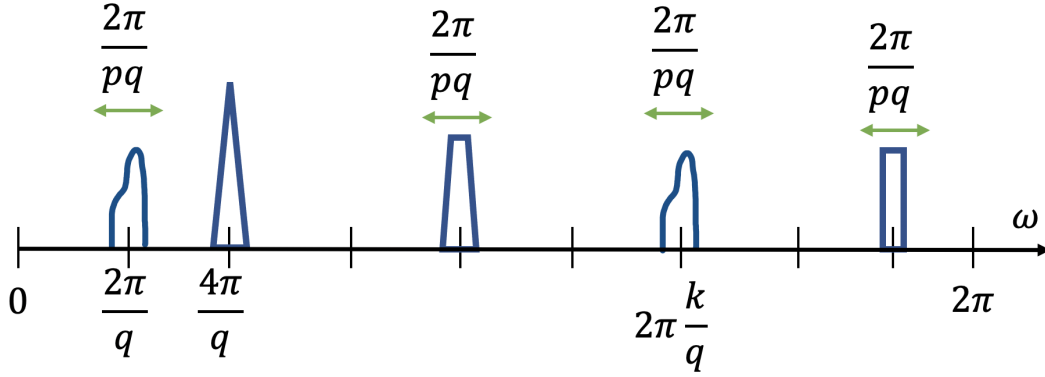


Figure 7.12: A signal $x(n) \in S_q^{(p)}$ has its Fourier transform concentrated around the coprime frequencies $2\pi k/q$ with $(k, q) = 1$.

where $x_{q_i}(n) \in S_{q_i}$ are (orthogonal) projections of $x(n)$ onto S_{q_i} . Now note that if $(M, q) = 1$ then $(M, q_i) = 1$ for all divisors q_i of q . So using Lemma 8 we get

$$x(Mn) = \sum_{q_i|q} x_{q_i}(Mn) = \sum_{q_i|q} x_{q_i}(n) = x(n). \quad (7.40)$$

So we have proved the following result.

Corollary 2 (Coprime decimation property for periodic signals). For any $x(n) \in R_q$, if $(M, q) = 1$ then $x(Mn) \in R_q$.

Now let $S_q^{(p)}$ be the class of multi-banded signals whose discrete-time Fourier transform is concentrated around the coprime frequency locations (Fig. 7.12). More precisely, the discrete-time Fourier transform of a signal $x(n) \in S_q^{(p)}$ with $p \geq 1$ satisfies

$$X(e^{j\omega}) = 0 \text{ if } \left| \omega - k \frac{2\pi}{q} \right| > \frac{\pi}{pq} \quad \forall k \text{ s.t. } (k, q) = 1. \quad (7.41)$$

Notice that each of the $\phi(q)$ bands has a bandwidth no greater than $2\pi/(pq)$. This subspace can be considered a finite-bandwidth approximation of S_q . Consider a signal $x(n) \in S_q^{(\alpha M)}$ where $(M, q) = 1$ and $\alpha \geq 1$. As an extension of Lemma 8, it then follows that $x(Mn)$ will belong to $S_q^{(\alpha)}$. Decimation by M expands the bandwidth of each of the $\phi(q)$ bands by a factor of M (and scales the magnitudes by a factor of $1/M$) so that each band now has a bandwidth of $2\pi/\alpha q$, instead of $2\pi/\alpha q M$. The stretched bands also have the center frequencies $2\pi k/q$, with $(k, q) = 1$, but in some permuted order. Since each of the $\phi(q)$ bands was narrow enough ($\alpha \geq 1$), there is no aliasing when the signal is decimated. In fact, if q is

even, the minimum separation between center frequencies $2\pi k/q$, with $(k, q) = 1$ is $4\pi k/q$. So signals in $S_q^{(\alpha M)}$ with $\alpha \geq 0.5$ can also be decimated by a factor of M without causing aliasing. This can be interpreted as a discrete-time sampling theorem for the class of signals $S_q^{(\alpha M)}$. It is possible to recover the original signal from the decimated version, albeit through a somewhat complex reconstruction procedure involving multiple filters. Summarizing, we have proved the following.

Lemma 9 (Coprime decimation of finite-bandwidth approximation of Ramanujan subspace signals). If $x(n) \in S_q^{(\alpha M)}$ with $(M, q) = 1$ and $\alpha \geq 1$, then $x(Mn) \in S_q^{(\alpha)}$ and no aliasing occurs with such decimation.

An analogous result can also be stated for $R_q^{(\alpha M)}$, which is the finite-bandwidth approximation of the subspace of signals with repetition index q . A signal $x(n) \in R_q^{(p)}$ with $p \geq 1$ if

$$X(e^{j\omega}) = 0 \text{ if } \left| \omega - k \frac{2\pi}{q} \right| > \frac{\pi}{pq} \text{ for } k = 0, 1, \dots, q-1. \quad (7.42)$$

The discrete-time Fourier transform of signals in $R_q^{(\alpha M)}$ has bandwidths of $2\pi k/(\alpha M q)$ around frequency locations $2\pi k/q$. For such signals, we have the following result.

Corollary 3 (Coprime decimation of finite-bandwidth approximation of periodic signals). If $x(n) \in R_q^{(\alpha M)}$ with $(M, q) = 1$ and $\alpha \geq 1$, then $x(Mn) \in R_q^{(\alpha)}$ and no aliasing occurs with such decimation.

Next, we examine what happens with non-coprime decimation.

Lemma 10 (Non-coprime decimation property of Ramanujan subspace signals). Let $x(n) \in S_q$, the q -th Ramanujan subspace, and M be the decimation ratio such that $(M, q) = g$. Then the decimated signal $y(n) = x(Mn) \in S_{q'}$, where $q' = q/g$.

Proof. With $x(n) \in S_q$ represented as in Eq. (7.2), the decimated signal is

$$\begin{aligned} y(n) = x(Mn) &= \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_q^{kMn} \\ &= \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_{q'}^{kM'n} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_{q'}^{(kM' \bmod q')n}, \end{aligned} \quad (7.43)$$

where $M' = M/g$. Note that $(M', q') = 1$. Since $(k, q) = 1$ and q' is a divisor of q , we have $(k, q') = 1$. This gives us $(kM', q') = 1$ and so $((kM' \bmod q'), q') = 1$.

Thus the terms $W_{q'}^{kM'n}$ are effectively of the form $W_{q'}^{k'n}$ where $0 \leq k' \leq q' - 1$ and $(k', q') = 1$. This gives us

$$y(n) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_{q'}^{(kM' \bmod q')n} = \sum_{\substack{k'=0 \\ (k',q')=1}}^{q'-1} \bar{\alpha}_{k'} W_{q'}^{k'n}. \quad (7.44)$$

Here, $\bar{\alpha}_{k'} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \mathbb{1}_{k \sim k'} \alpha_k$ where $\mathbb{1}_{k \sim k'} = 1$ if $(kM' \bmod q') = k'$, and zero otherwise. From the RHS of Eq. (7.44), it is clear that $y(n) \in S_{q'}$. \square

When $g = 1$, q and M are coprime, and we get the *if* statement of Lemma 8. Furthermore, the *only if* part of Lemma 8 is now obvious in the light of Lemma 10. For the special case of $c_q(n) \in S_q$, we have the following conjecture:

Conjecture 1. [Non-coprime decimation of Ramanujan sums] Let $(M, q) = g$, and $q' = q/g$. Then

$$c_q(Mn) = \frac{\phi(q)}{\phi(q')} c_{q'}(n). \quad (7.45)$$

The conjecture was numerically checked to be true for many different values of q and M , however, a proof is not yet known. When $g = 1$, $q' = q$ and we get back the coprime decimation identity for Ramanujan sums: $c_q(Mn) = c_q(n)$ (Corollary 1).

7.6.2 Decimating Filter Bank Outputs to Reduce Computations

Lemma 8 and its extension to signals from space $S_q^{(\alpha M)}$ where $(M, q) = 1$ and $\alpha \geq 1$ open up avenues to decimate the RFB filter outputs in our denoising framework. For this, first consider the following observations:

- Coprime decimation of signals from $S_q^{(\alpha M)}$ permutes the stretched (and scaled by a factor of $1/M$) frequency bands of the original signal. As there is no aliasing, the total energy of the decimated signal is $1/M$ times the total energy of the undecimated signal. Since the number of samples in the decimated signal is also reduced by a factor of M , the average per-sample energy of the decimated signal is the same as that of the undecimated signal.
- The outputs $y_q(n)$ of $C_q^{(l)}(z)$ can be approximately considered to be signals from spaces $S_q^{(\alpha M)}$, for appropriate M and α . This approximation is reasonable when l is not too small, so most of the energy of the filter output is indeed concentrated around the coprime frequencies. A similar approximation can also be considered for the outputs of Capon-optimized filters.

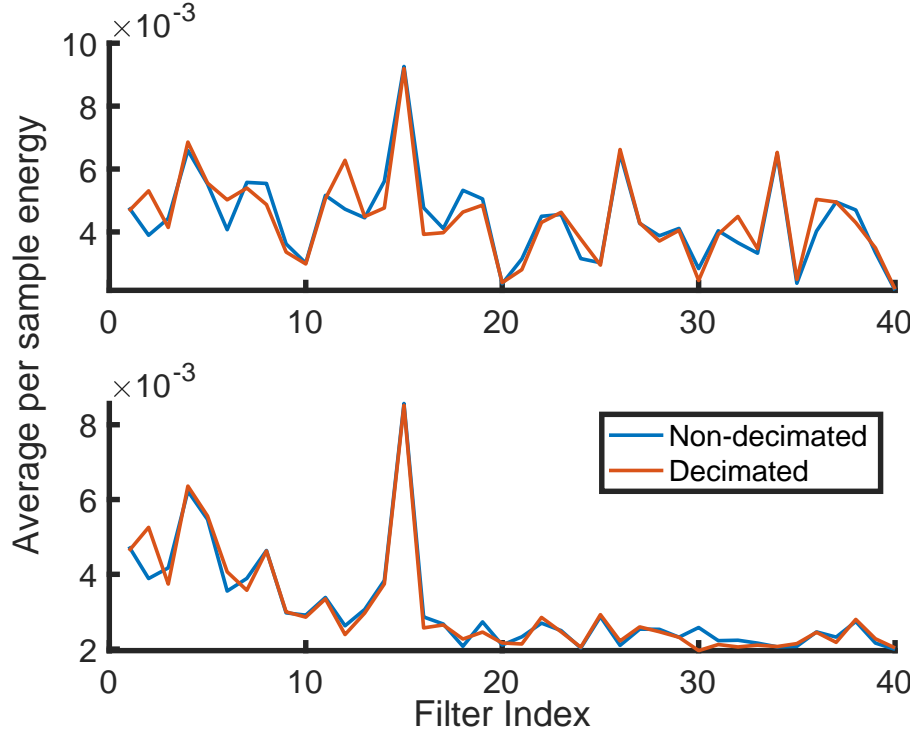


Figure 7.13: Coprime decimation approximately preserves the average per-sample energy of the filter outputs for both RFB (top) and Capon-optimized filter bank (bottom).

- The selection of subspaces in the dictionary pruning stage of our denoising method is only based on the energies of the filter bank outputs.

These observations suggest that it is reasonable to decimate the output of the q -th filter by a decimation ratio M_q such that $(M_q, q) = 1$. We experimentally found that the average per-sample energies of decimated outputs of RFB and Capon-optimized banks indeed reasonably approximate the corresponding average per-sample energies of undecimated outputs. Fig. 7.13 shows an example of a period-15 signal having SNR 0 dB. The average per-sample energies of decimated and undecimated outputs match reasonably well, especially so for the Capon-optimized filter bank. M_q , in this case, was chosen to be the smallest integer larger than unity that is coprime to q (see Eq. (7.46)).

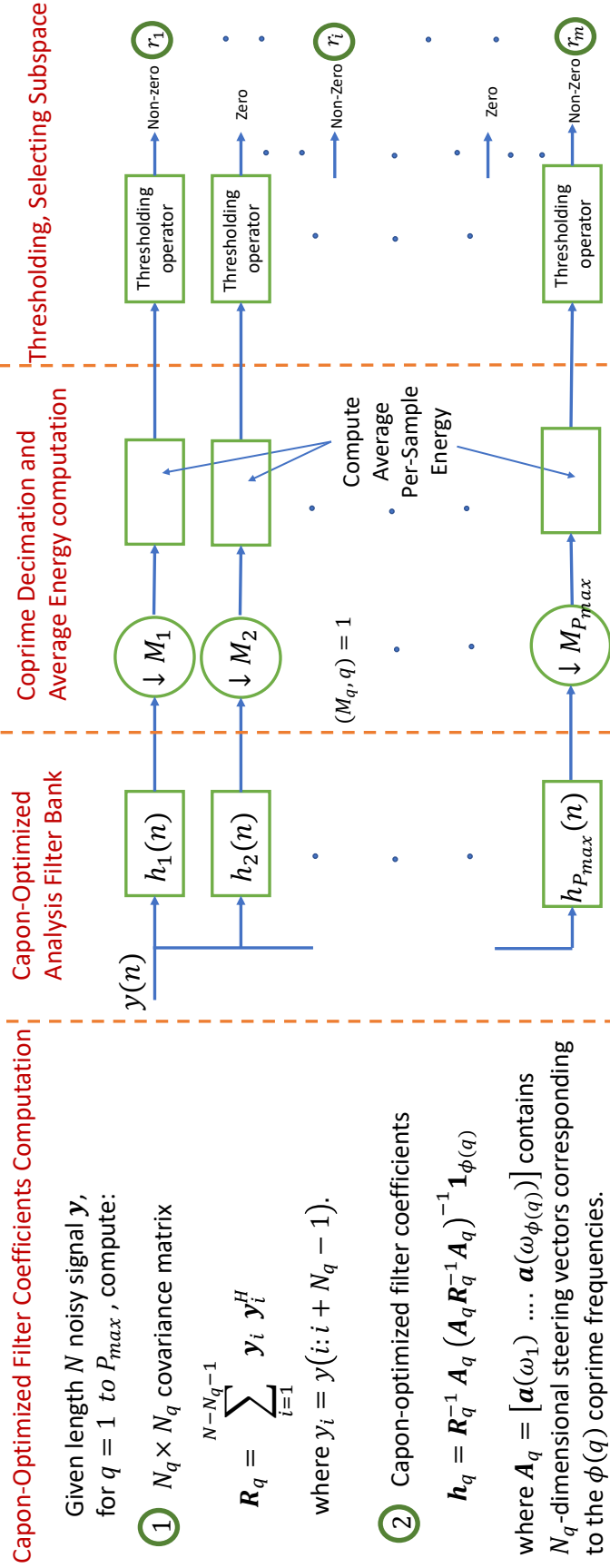


Figure 7.14: Initial stages of proposed periodicity-aware denoising framework that uses Capon-optimized filters and coprime decimation. The later stages of forming a pruned synthesis dictionary and denoising by solving the optimization problem are the same as in Fig. 7.3.

As each filter is now followed by a decimator, we can implement the filters in polyphase form, and using the multirate identities move decimators before the polyphase filters [188]. Each polyphase filter then operates at a lower rate, and the number of computations performed by the q -th filter is reduced by a factor of M_q . Fig. 7.14 shows the initial stages of the proposed periodicity-aware denoising framework that uses Capon-optimized filters and coprime decimation. The later stages of forming a pruned synthesis dictionary and denoising by solving the optimization problem are the same as in Fig. 7.3.

7.7 Comparative Simulations with Decimation, and Other Investigations

In this section, we provide several Monte-Carlo simulations comparing the different variants of our proposed denoising method and other well-known denoising algorithms. Please see Sec. 7.3.4 for details of the denoising methods with which we compare our proposed denoising method. Here, the signal length is taken to be 400, and the filter length of the q -th filter is $N_q = 3q$. For each simulation setting, we generate 1000 random periodic signals of length 400 and add white Gaussian noise to them. Then we perform a joint hyperparameter search as explained in Sec. 7.3.4, to choose a good set of hyperparameter values for each denoising method. The effect of varying different parameters of the proposed framework on the denoising performance is investigated in Sec. 7.7.1. We also briefly discuss the applicability of the proposed framework for other tasks, such as period estimation and component separation, later in this section.

As discussed before, there are 4 variants of our proposed method depending on the choice of filter bank (Ramanujan filters and Capon-optimized filters) and choice of synthesis dictionary ('with-divisors' and 'without-divisors'). When sufficient signal length is available, the Capon-optimized banks are always found to perform better than RFB for denoising. Also, the 'with-divisors' synthesis dictionary is found to be a better choice than the 'without-divisors' synthesis dictionary. Furthermore, it is possible to choose the coprime decimation ratios M_q -s (see Sec. 7.6.2) for efficient implementation in many different ways.

Fig. 7.15(a) shows SNR gains for different denoising methods averaged over 1000 different period-15 signals. Here we use the 'smallest coprime decimation', i.e., the output of the q -th filter is decimated by M_q^s :

$$M_q^s = \min_{M>1} M \quad \text{s.t.} \quad (M, q) = 1. \quad (7.46)$$

Similar to previous simulations, the combination of the Capon-optimized filter bank

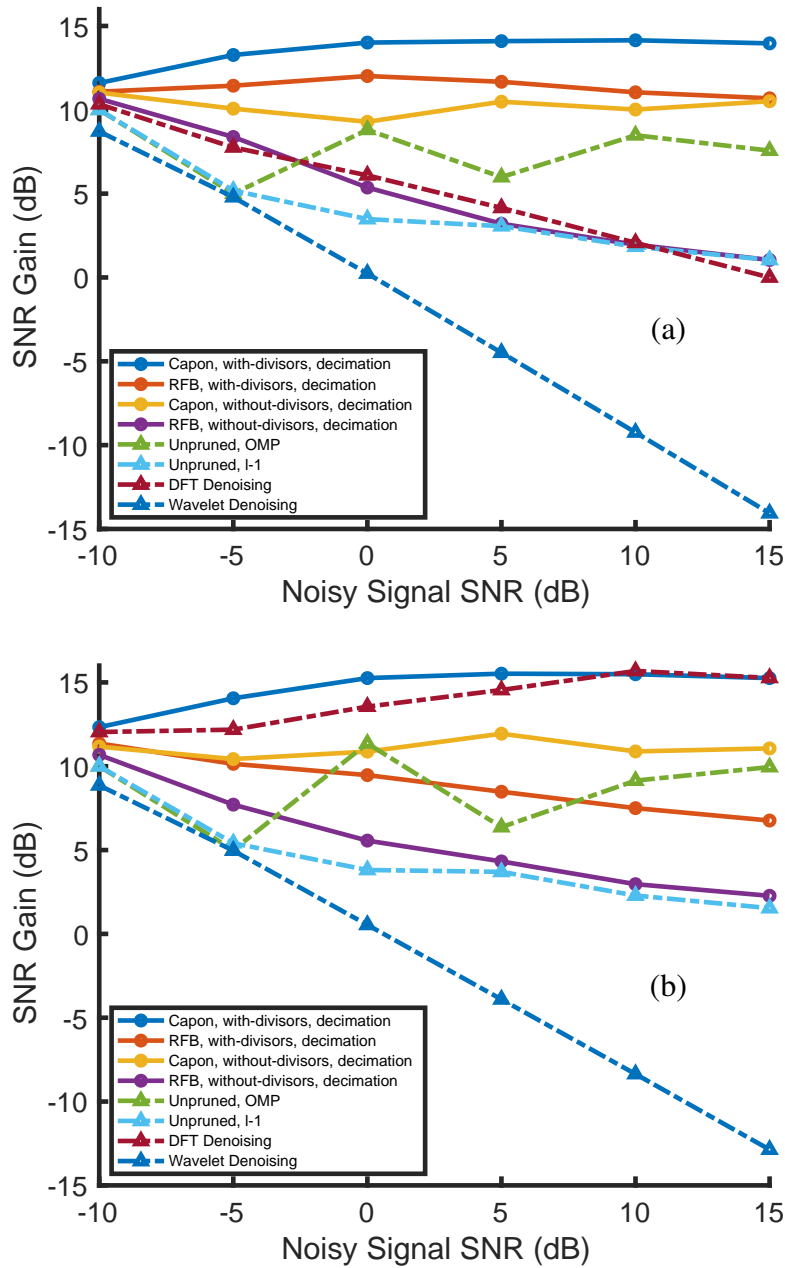


Figure 7.15: Comparing the proposed denoising method that uses coprime decimation with other denoising methods. SNR gains averaged over 1000 Monte-Carlo runs. (a) Period-15 signals, decimation ratio M_q^5 (Eq. 7.46) (b) period-10 signals, decimation ratio M_q^{10} (Eq. 7.47).

and the ‘with-divisors’ synthesis dictionary gives the best SNR gain, followed next by the combination of RFB and the ‘with-divisors’ dictionary. Fig. 7.15(b) considers 1000 different period-10 signals. Here, the decimation ratio is taken as M_q^{10} , where

$$M_q^L = \max_{M \leq L} M \quad \text{s.t.} \quad (M, q) = 1. \quad (7.47)$$

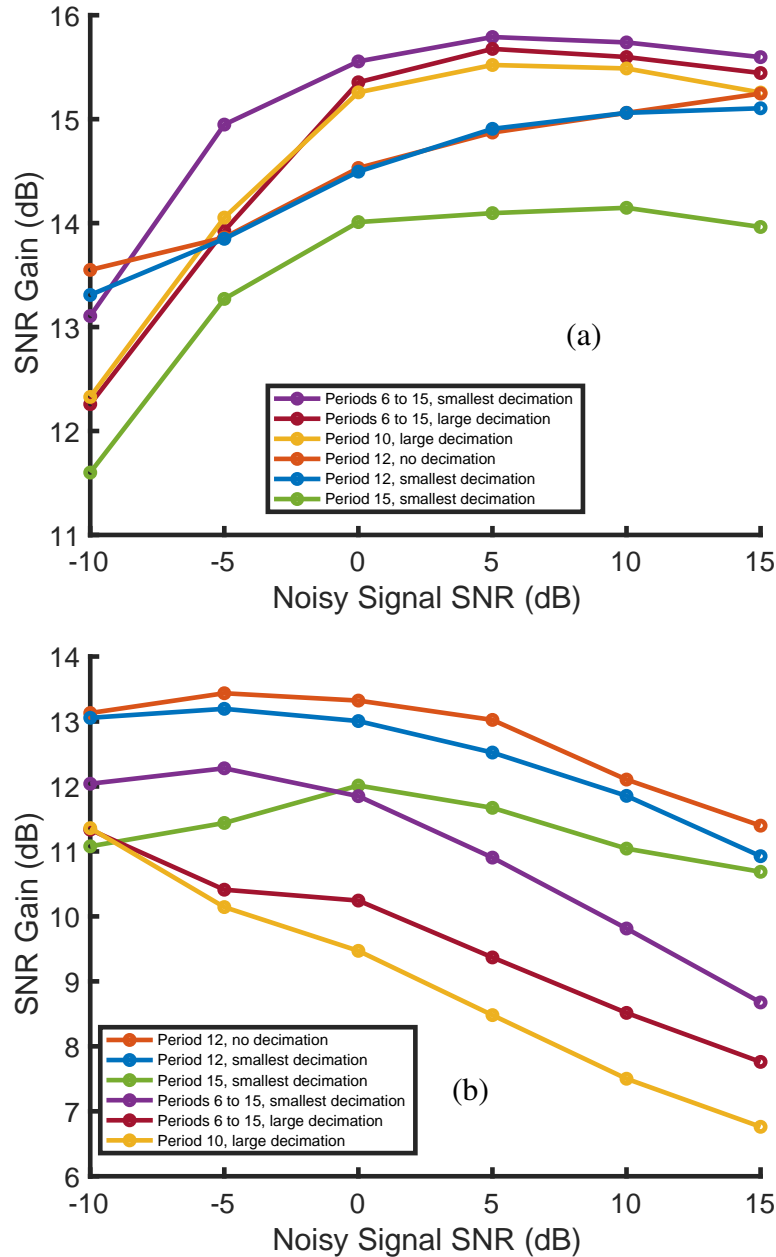


Figure 7.16: Comparing the effect of decimation values and signal period on denoising performance when denoising with (a) Capon-optimized filter bank and 'with-divisors' dictionary (b) RFB and 'with-divisors' dictionary. SNR gains averaged over 1000 Monte-Carlo runs.

chooses the largest integer coprime to q that is not larger than L as the decimation rate for the q -th filter. In this case, the DFT performs very well as the signal period, 10, is a divisor of the signal length 400, so the signal frequencies lie on the DFT grid. However, the proposed method that uses Capon-optimized filters and the 'with-divisors' dictionary still performs marginally better. In this case, the

combination of the Capon-optimized bank and the ‘without-divisors’ dictionary has better SNR gain than the combination of RFB and the ‘with-divisors’ dictionary, unlike the previous case of period-15 signals.

We now compare the effect of different values of decimation and the period of the signals under consideration. In Fig. 7.16 we consider three decimation schemes, namely, no decimation, smallest coprime decimation (M_q^s), and larger coprime decimation (M_q^{10}), and four period schemes, namely, period-15, period-12, period-10, and periods 6 to 15. For the ‘periods 6 to 15’ case, 100 random signals of each period are included, totaling 1000 signals. Fig. 7.16(a) uses Capon-optimized filters, whereas Fig. 7.16(b) uses RFB. The ‘with-divisors’ dictionary is used for signal reconstruction in both cases.

The plots reveal many interesting trends. Firstly, note that SNR gains obtained with Capon-optimized filters increase with noisy signal SNR at first and then saturate, whereas SNR gains obtained with RFB decrease with noisy signal SNR. From the relative positioning of trends within Fig. 7.16(a), we see that the primary factor that affects the SNR gains with Capon-optimized filters is the signal period. Signals with relatively smaller periods (6 to 15, and 10) have larger SNR gains, and the signals with larger periods (12 and 15) have smaller SNR gains. SNR gains for the period-12 signal with no decimation and with the smallest coprime decimation (M_q^s) are practically the same, indicating that small coprime decimation does not degrade the denoising performance. SNR gains for period 6 to 15 signals with large coprime decimation (M_q^{10}) are only marginally less than those with the smallest coprime decimation (M_q^s) at higher SNRs. The high average SNR gain for period 6 to 15 signals indicates that with Capon-optimized filters, it is possible to get a good set of hyperparameters that work well with signals having a range of different periods.

On the other hand, we see some different trends from Fig. 7.16(b) when RFB is used instead of Capon-optimized filters. Here, the primary factor that affects the SNR gains is the decimation. No decimation plots are on the top, followed by the smallest coprime decimation plots. Large coprime decimation plots are at the bottom. Furthermore, SNR gains for period 6 to 15 signals are comparatively smaller, even with small decimation. This indicates that with a common set of hyperparameters, it is difficult to get good denoising for a range of periods when RFB is used.

Summarizing, using Capon-optimized filters in our proposed denoising framework is found to have some important advantages. They provide consistently better

performance than using Ramanujan filters. They are also more robust to coprime decimation than Ramanujan filters. Furthermore, they do not have diminishing SNR gain as the noisy signal SNR increases. Finally, compared to Ramanujan filters, they can better denoise signals with a range of different periods with a common set of hyperparameters.

7.7.1 Effect of Hyperparameters

In this subsection, we discuss how the different hyperparameters in our proposed method affect the denoising performance. In the above simulations, we used $N_q = 3q$ and performed a hyperparameter search for the energy threshold value and regularization level λ . For the hyperparameter search, we compute the SNR gain obtained by each combination of values from the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ for threshold and λ respectively, averaged over 1000 signals having periods from 6 to 15. We report the best combination of hyperparameters along with the corresponding average SNR gain in Tables 7.1 and 7.2. All four variants, based on the choice of filter bank and synthesis dictionary, are included. All methods use the smallest coprime decimation M_q^s (Eq. (7.46)). As expected, it is observed that when using the ‘with-divisors’ dictionary, a higher threshold is preferred compared to the case of the ‘without-divisors’ dictionary. This is because some of the subspaces lost due to a higher threshold can be reintroduced by the ‘with-divisors’ dictionary. Also, note that the values of λ when using a Capon-optimized filter bank are the same for ‘with-divisors’ and ‘without-divisors’ dictionaries.

Next, in Fig. 7.17, we demonstrate how the SNR gain changes as we vary the different hyperparameters of the four variants of the proposed denoising framework. The average SNR gain for 1000 noisy signals with periods from 6 to 15, with SNR 0 dB, is considered. The default values of the parameters are as follows: threshold = 0.5, $\lambda = 0.01$, $l = N_q/q = 3$, $P_{max} = 40$, signal length = 400, and decimation M_q^s (Eq. (7.46)). We vary one of these parameters at a time, keeping the others fixed in each sub-figure.

In Fig. 7.17(a), we see that the best SNR gains for Capon-optimized banks are obtained at lower threshold values than those for RFB. This is so because Capon filters better filter out the noise and other harmonic components than Ramanujan filters, as discussed previously (Fig. 7.8). From Fig. 7.17(b), we see that SNR gain is not very sensitive to λ when $\lambda \leq 0.1$, and decreases rapidly when λ is increased

Table 7.1: Values of the hyperparameters (filter bank threshold and regularization level λ) selected by joint search over the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ respectively for 1000 random signals with periods from 6 to 15 when using the RFB in the first stage of the proposed denoising framework.

SNR (dB)	RFB, without-divisors			RFB, with-divisors		
	Threshold	λ	SNR gain	Threshold	λ	SNR gain
-10	1.0	0.1	11.35	1.0	0.33	12.04
-5	0.9	0.033	10.18	0.9	0.033	12.28
0	0.8	0.01	7.82	0.9	0.01	11.85
5	0.5	0.01	5.21	0.8	0.001	10.90
10	0.3	0.033	2.86	0.8	0.001	9.81
15	0.1	0.033	1.66	0.7	0.0001	8.68

Table 7.2: Values of the hyperparameters (filter bank threshold and regularization level λ) selected by joint search over the sets $\{0, 0.1, 0.2, \dots, 1\}$ and $\{0.0001, 0.001, 0.01, 0.033, 0.1, 0.33, 1\}$ respectively for 1000 random signals with periods from 6 to 15 when using the Capon-optimized filter bank in the first stage of the proposed denoising framework.

SNR (dB)	RFB, without-divisors			RFB, with-divisors		
	Threshold	λ	SNR gain	Threshold	λ	SNR gain
-10	1.0	0.1	11.35	1.0	0.33	12.04
-5	0.9	0.033	10.18	0.9	0.033	12.28
0	0.8	0.01	7.82	0.9	0.01	11.85
5	0.5	0.01	5.21	0.8	0.001	10.90
10	0.3	0.033	2.86	0.8	0.001	9.81
15	0.1	0.033	1.66	0.7	0.0001	8.68

from 0.1. Next, from Fig. 7.17(c), we see that Ramanujan filters need a much larger filter length to match the performance of Capon-optimized filters. However, a larger filter length is practical only if a large data length is available. For Capon-optimized filters, $l = 3$ already gives a high SNR gain which does not increase much when l is further increased. This was the reason for fixing $l = 3$ in our previous simulations.

In Fig. 7.17(d), we vary the P_{max} value, which was assumed to be 40 throughout

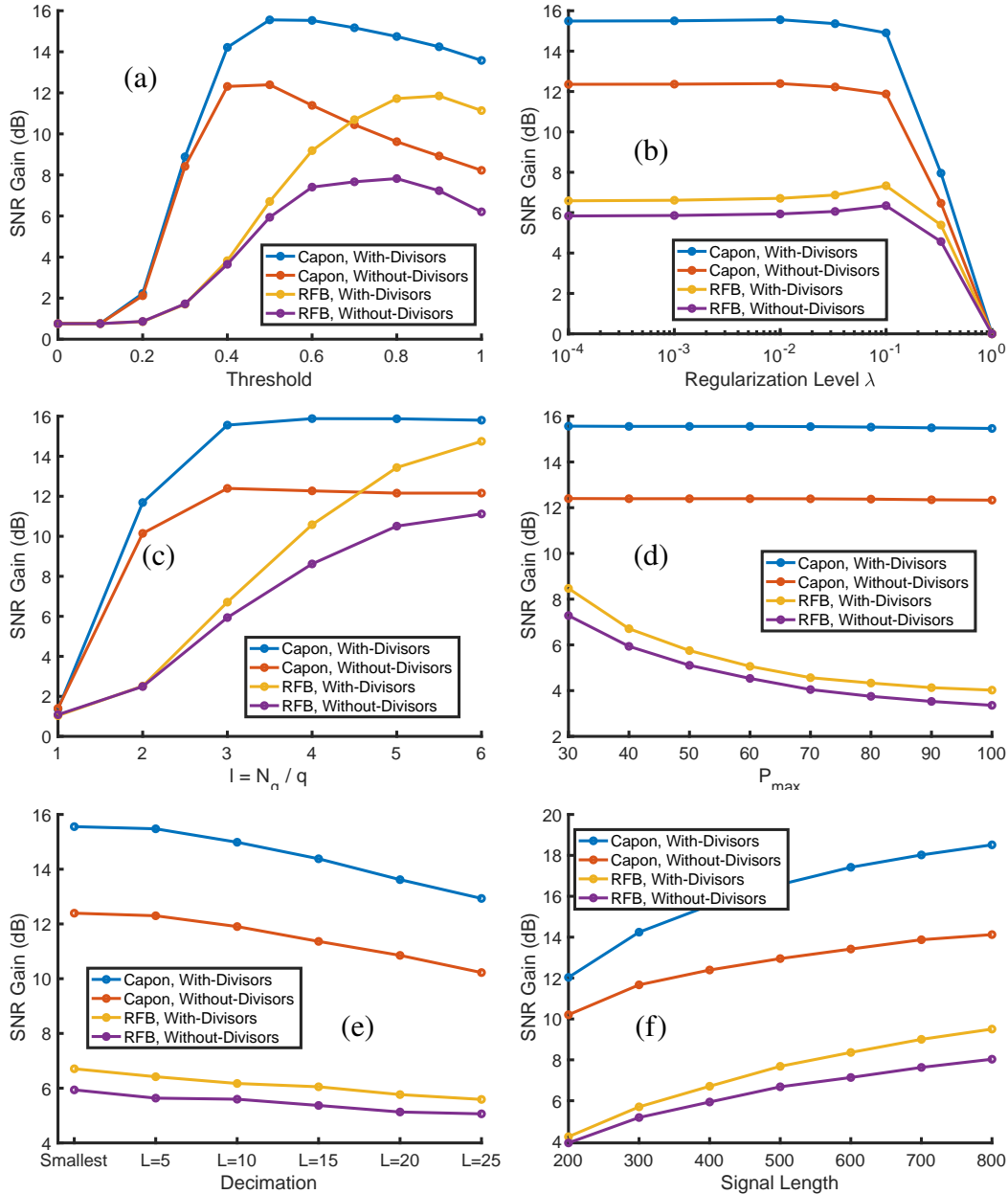


Figure 7.17: (a)-(f) Effect of hyperparameters on average SNR gain of 1000 signals having periods from 6 to 15 and SNR 0 dB. The default values of the parameters are as follows: threshold = 0.5, $\lambda = 0.01$, $l = N_q/q = 3$, $P_{max} = 40$, and smallest coprime decimation M_q^s (Eq. (7.46)). One of the parameters is varied in each sub-figure, keeping the others fixed.

the chapter. We see that changing the P_{max} value does not affect the performance with the Capon-optimized bank, whereas performance with RFB decreases. This is an important advantage of Capon-optimized banks, especially when a good a-priori estimate of P_{max} may not be available. Fig. 7.17(e) shows the effect of change in decimation on SNR gain. The smallest coprime decimation M_q^s (Eq. (7.46))

and larger decimations M_q^L (Eq. (7.47)) with $L = 5, 10, 15, 20, 25$ are considered. We see that increasing the decimation values decreases the SNR gains, but not too severely. Lastly, in Fig. 7.17(f), we see that as the signal length is increased, the denoising performance improves, as expected.

7.7.2 Period Estimation and Component Separation

In this subsection, we discuss the applicability of the proposed framework for period estimation and component separation. However, note that this is not the primary objective of the proposed framework, and a good set of hyperparameters obtained previously for denoising may not be optimal for the period estimation or component separation tasks.

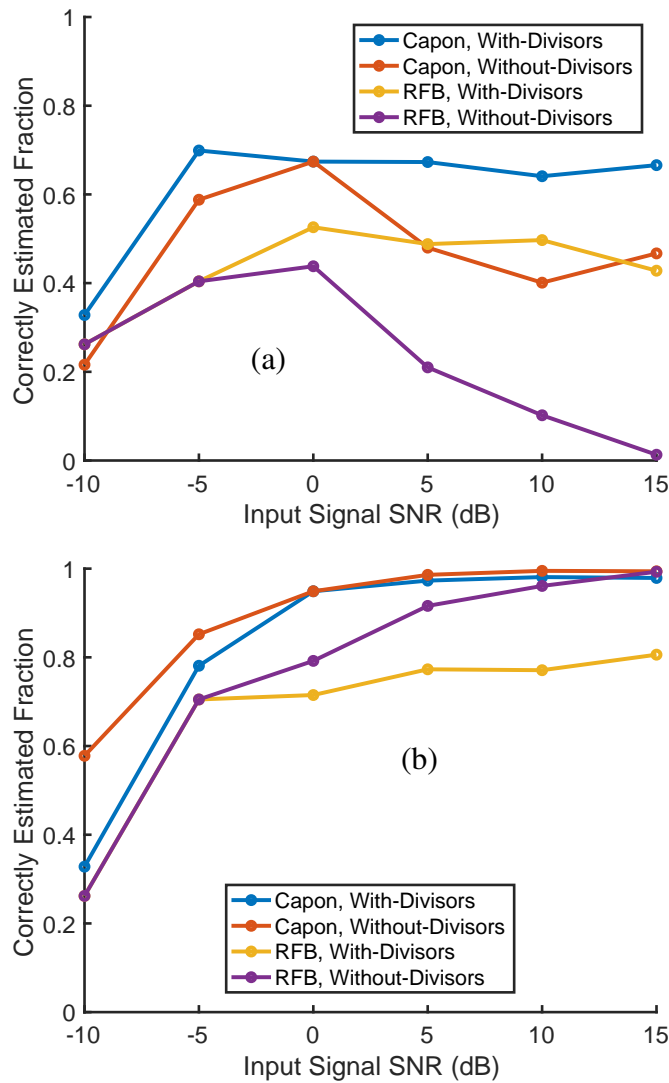


Figure 7.18: Probability of correct (a) period estimation (b) repetition index estimation for 1000 signals with periods from 6 to 15.

As discussed before, the indices r_1, \dots, r_m correspond to the periods of the components contributing to the reconstructed signal $\hat{\mathbf{x}}$ (see Eq. (7.11)). In practice, we noticed that all the retained subspaces in the synthesis dictionary have non-zero contributions to the reconstructed signal. This is because, with a good set of hyperparameters and a large signal length, the appropriately pruned synthesis dictionary is usually tall when the signal has a small number of periodic components. In this case, the period of the denoised signal $\hat{\mathbf{x}}$ is given by $\text{lcm}(r_1, r_2, \dots, r_m)$, where lcm denotes the least common multiple [63], [64]. Also note that the operation of introducing divisor subspaces in Eq. (7.14) does not change the overall period of the reconstructed signal $\hat{\mathbf{x}}$, as the lcm of the divisors d_{11}, \dots, d_{mk_m} is the same as the lcm of r_1, \dots, r_m .

In Fig. 7.18(a), we plot the fraction of the 1000 signals with periods from 6 to 15 for which the period of the denoised signal estimated by $\text{lcm}(r_1, r_2, \dots, r_m)$ is the same as that of the original noiseless signal. We see that the Capon-optimized bank that uses the ‘with-divisors’ dictionary has higher period estimation accuracy. Here, for period estimation, we used the parameter values obtained through the hyperparameter search (Tables 7.1 and 7.2) in the context of denoising. If the parameters were optimized for period estimation instead, period estimation performance would be even better. Fig. 7.18(b) shows the fraction of the signals for which the repetition index is correctly estimated. We say that the repetition index is correctly estimated when the indices r_1, \dots, r_m contain the correct indices corresponding to the components of the noiseless signal, possibly along with some additional indices. This fraction is observed to approach 1 as SNR is increased.

Next, we consider the task of periodic component separation. However, note that there is a fundamental limitation in separating periodic components that share a common harmonic, regardless of the method used. For example, if a signal $x(n)$ has period-12 and period-9 components $x_{12}(n)$ and $x_9(n)$, they cannot be uniquely separated. This is because we can add any period-3 signal to $x_{12}(n)$ and subtract the same signal from $x_9(n)$ without changing $x(n)$, and the periods of the component signals. Similarly, the period-1 (or ‘DC component’) cannot be uniquely split between component periods. We can separate the periodic components only when the signal components are zero-mean and the component periods are mutually coprime. See [52] for more details. In what follows, the period- P component is estimated as

$$\hat{\mathbf{x}}_P = \begin{bmatrix} \mathbf{H}_{P_1} & \dots & \mathbf{H}_{P_L} \end{bmatrix} \hat{\mathbf{b}}_P, \quad (7.48)$$

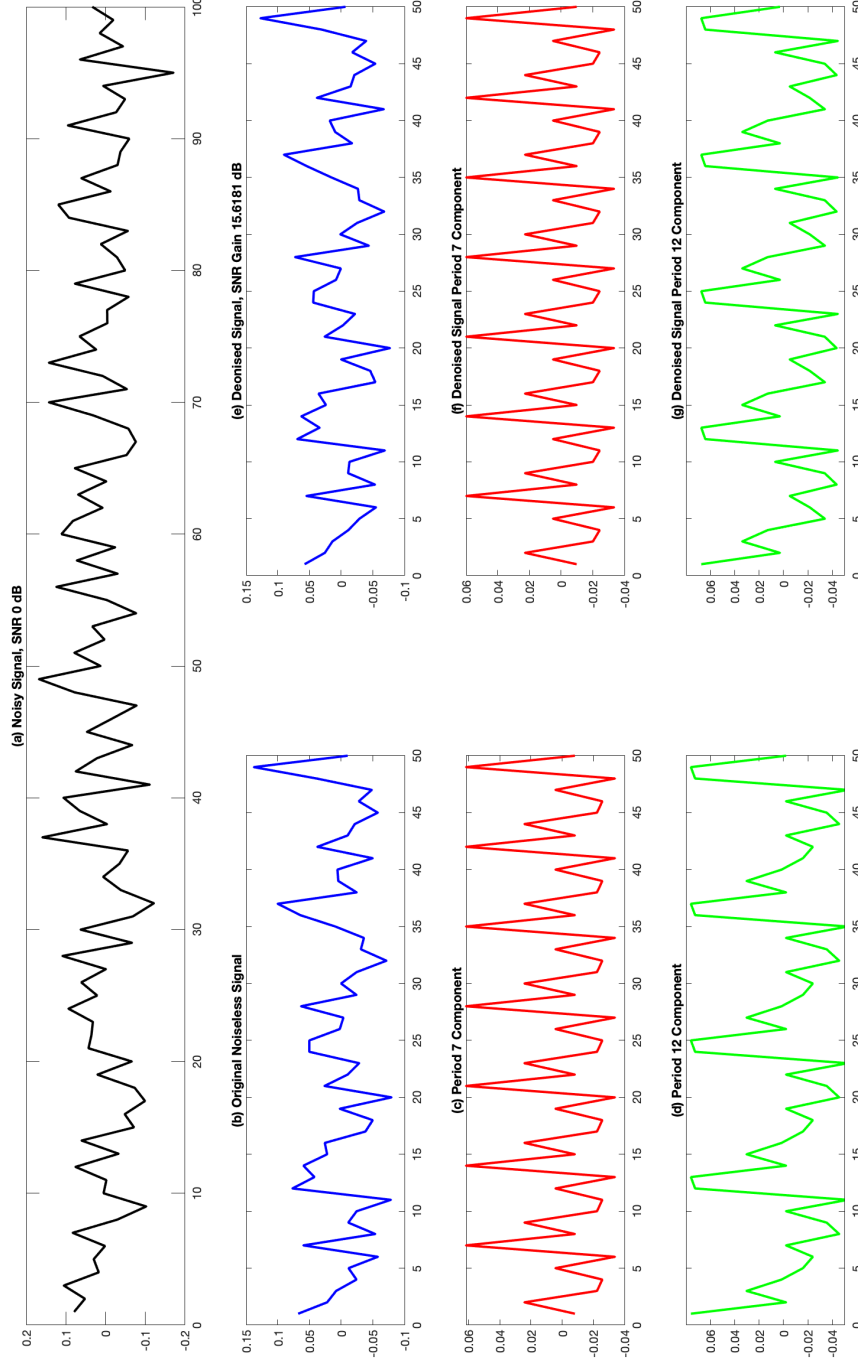


Figure 7.19: Separating period-7 and period-12 components from a zero-mean period-84 signal with the proposed denoising framework that uses the Capon-optimized bank with the smallest coprime decimation M_q^s and ‘with-divisors’ dictionary. (a) Noisy signal, (b)-(d) noiseless signal and its components, (e)-(g) denoised signal and its separated components according to Eq. (7.48).

where P_1, \dots, P_L are divisors of P and $\widehat{\mathbf{b}}_P$ contains coefficient from the estimated vector $\widehat{\mathbf{b}}$ (Eq. (7.12)) that correspond to subspaces $\mathbf{H}_{P_1}, \dots, \mathbf{H}_{P_L}$. We should note that the accuracy of component separation depends crucially on the accuracy of the component period estimation. Fig. 7.19 shows an example of a period-84 signal with SNR 0 dB having zero-mean components with periods 7 and 12. The signal length is 400, but we only plot 50 or 100 samples for better visual comparison. Fig. 7.19(b)-(d) show the noiseless signal and its components, whereas Fig. 7.19(e)-(g) show the denoised signal and its components obtained as per Eq. (7.48). As the component period estimation was correct in this case, we see that the separated components from the denoised signal match quite well with the components from the original signal.

7.7.3 Denoising an ECG Signal

As an example of a real-life signal that can be approximately periodic, we consider the task of denoising a heart monitoring ECG recording. We use the ECG recordings from the MIT-BIH normal sinus rhythm database [202], [203]. We first downsample the signal by a factor of three and add a 0 dB random white Gaussian noise to it. Fig. 7.20 shows the resampled ECG signal, its noisy version, and the output produced by the proposed denoising framework that uses Capon-optimized bank and ‘with-divisors’ dictionary. The periodicity in the ECG peaks (R region) is captured correctly in the denoised output, but the denoising is not so great in other parts (P, Q, S, and T regions). This is because of multiple reasons. The primary reason is that the original ECG signal is not exactly integer-periodic, whereas our method is particularly designed for integer-periodic signals. Secondly, the hyperparameters were not optimized for such types of approximately periodic signals. Furthermore, note that our denoising method is designed to produce outputs with integer-periodic components only. However, this suppresses the abnormalities or the non-periodic components in the ECG, and this may not be desirable for a clinical diagnosis.

7.8 Concluding Remarks

In this chapter, we proposed a hybrid periodicity-aware denoising framework. The analysis part consists of Ramanujan filter banks, and the synthesis is done via a pruned Ramanujan dictionary. The proposed method is guaranteed to produce a denoised output composed of integer-periodic components, unlike the other denoising methods from the literature. We further proposed to optimize the analysis filters using a multi-band Capon formulation, which further improved denoising perfor-

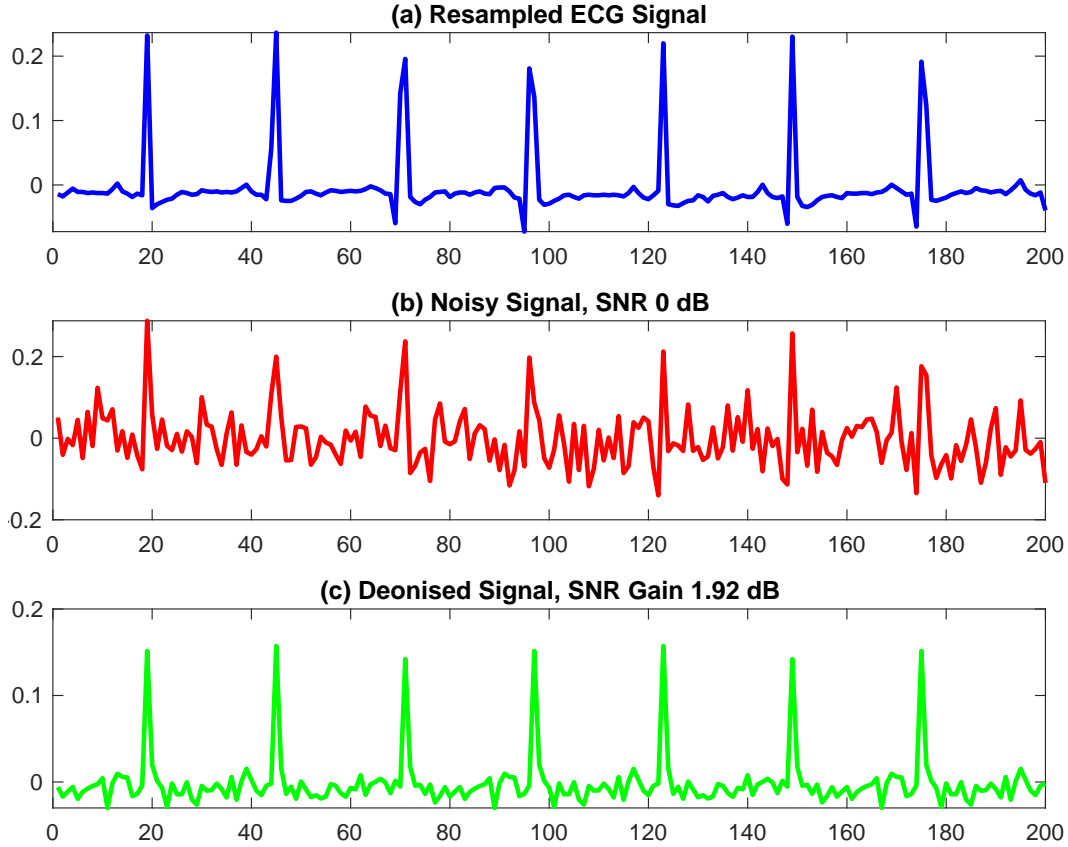


Figure 7.20: Denoising an ECG recording of a normal sinus rhythm. (a) Original ECG (resampled), (b) noisy ECG, and (c) denoised ECG.

mance, at the cost of increased computations. Some ways to reduce computations based on a recursive computation of the inverses of Toeplitz autocorrelation matrices were discussed. We also derived several multirate properties of Ramanujan subspace signals. In particular, the coprime decimation properties further allowed us to reduce the computations required in the analysis stage of the proposed denoising framework. Extensive Monte-Carlo simulations demonstrated that the proposed denoising framework provides better SNR gains than the other denoising methods from the literature.

Out of the four variants of the proposed methods, denoising using **Capon-optimized** filter bank and **‘with-divisors’** dictionary was found to perform best. It has the following features:

- Works well with small coprime decimation in the analysis stage, and for signals with a range of different periods.

- Best hyperparameters depend on signal length, range of periods, etc. (For length 400 signals with periods from 6 to 15: $l = 3$, threshold = 0.5, and $\lambda = 0.01$ are optimal.)
- Performance does not degrade as P_{max} is increased.
- Can estimate the component periods better than RFB.
- Can be used for component separation, when applicable.
- Calculating filter coefficients is computationally expensive.

Although the proposed method works well when denoising synthetically generated integer-periodic signals, it has some limitations. It is inadequate for denoising non-integer periodic signals or approximately periodic signals such as ECG. Furthermore, the method is currently not suitable for signals with time-varying periods. Adapting the method for these challenging scenarios would require several careful modifications that vary depending on the nature of how the periodicity structure changes over time.

In the future, penalties can be introduced for using dictionary columns corresponding to the q -th Ramanujan subspace in the l_1 optimization problem based on (some decreasing function of) the energy of the q -th filter output. As the proposed method is based on filtering around ‘coprime frequencies’ corresponding to several integers, we expect the denoising performance to degrade if the noise is concentrated around these frequencies and is not white (as was considered in this chapter). The effect of such noise models on the proposed denoising framework should be explored in the future.

THEORETICAL PROPERTIES OF IDEAL RAMANUJAN FILTERS

8.1 Introduction

In the previous chapter, we introduced a hybrid analysis-synthesis denoising framework for periodic signals, utilizing Ramanujan analysis filter banks and synthesis dictionaries. The initial stage of this denoising process employed the Ramanujan filter bank (RFB), which is effective in detecting periodicity in streaming data. However, the synthesis component of our proposed denoising framework relied on solving an optimization problem with Ramanujan dictionaries. This approach may not be ideal, especially when the periodic structure of the signal varies over time. Instead of the hybrid analysis-synthesis framework, a perfect reconstruction analysis-synthesis bank [188] using Ramanujan filters would be of great interest for denoising periodic signals. Implementing such a filter bank requires decimating the outputs from the RFB during the analysis stage, as well as incorporating an interpolation filter model for Ramanujan subspace signals in the synthesis stage. Due to the multiband frequency response of the Ramanujan filters (see Fig. 7.2), a standard maximally decimated filter bank does not yield a beneficial structure.

We also proved several multirate properties of Ramanujan subspace signals related to decimation in the previous chapter. This was valuable for reducing the computational complexity of the hybrid analysis-synthesis denoising framework by decimating the outputs of the Ramanujan filters. To make progress towards developing a synthesis filter bank counterpart for the RFB, we now investigate interpolation properties of Ramanujan filters in this chapter. In the first part, we propose an interpolation filter model for Ramanujan subspace signals, wherein an expander $\uparrow M$ is followed by a q -th ideal Ramanujan filter. The expander $\uparrow M$, in general, reduces the output space of the filter to a proper subspace of the q -th Ramanujan subspace \mathcal{S}_q . However, we show that the output space of the interpolation filter is the entire \mathcal{S}_q when M and q are coprime. We also extend this model to the case of more general periodic signals by using a multirate synthesis bank model. These models have potential application in the denoising of periodic signals using a complete analysis-synthesis filter bank structure. This latter aspect requires a detailed study and is beyond the scope of this

chapter. We discuss the general case when M and q may not be coprime. In this case, we provide a bound on the dimension of the output space of the interpolation filter, and also conjecture that the bound is in fact the actual dimension of the output space. The developments in this chapter are largely motivated by the theoretical interest in understanding the properties of Ramanujan subspace signals and Ramanujan filters.

In the second part of this chapter, we analyze the locations of zeros of the transfer functions of FIR Ramanujan filters. FIR Ramanujan filters have a few periods of the Ramanujan sums as their filter coefficients. Understanding these locations is important for developing a corresponding synthesis filter bank. We discover an interesting structure regarding the positions of the zeros in Ramanujan filters. We prove that all zeros of these filters lie either on or inside the unit circle in the z -plane. This finding indicates that a simple ‘inverse synthesis filter’ cannot be used in the synthesis filter bank, because such inverse filters would be unstable because of having some poles on the unit circle in the z -plane. Additionally, we derive a factorization formula for the filter response that helps identify the common factors among different Ramanujan filters. A key application of these results is the potential to obtain efficient implementations of the RFB by sharing the common factors present in various filters, based on the set of shared zeros among different Ramanujan filters.

8.1.1 Scope and Outline

In Sec. 8.2, we introduce the interpolation filter model for Ramanujan subspace signals. The expander $\uparrow M$ generally reduces the output space of the filter to a proper subspace of the q -th Ramanujan subspace S_q . However, we demonstrate that when M and q are coprime, the output space of the interpolation filter encompasses the entire S_q . Additionally, we discuss a more general multirate synthesis bank model for periodic signals. In Sec. 8.3, we examine the scenario where M and q may not be coprime. In this case, we provide a bound on the dimension of the output space of the interpolation filter and conjecture that this bound represents the actual dimension of the output space. In Sec. 8.4 we prove a lemma that characterizes the locations of zeros of FIR Ramanujan filters in terms of critical points of cyclotomic polynomials by using Lucas’s theorem. As a corollary, a factorization formula is obtained for the filters. In Sec. 8.5 we show that the plots of zeros are indeed in accordance with the derived result, and identify further structure in the locations of zeros for some specific families of Ramanujan filters. Zeros of Ramanujan filters in which multiple periods are used as filter coefficients are also characterized. In

Sec. 8.6, we illustrate a possible application of derived results in designing efficient implementation structures for Ramanujan filter banks. Sec. 8.7 concludes the chapter. The content of this chapter is based on papers [195], [204].

8.2 Interpolation Filter Model for Ramanujan Subspace Signals

Recall that the q -th Ramanujan subspace \mathcal{S}_q [52] is the $\phi(q)$ -dimensional space of signals of the form

$$x(n) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \alpha_k W_q^{kn} = \sum_{l=0}^{\phi(q)-1} \beta_l c_q(n-l), \quad (8.1)$$

where α_k and β_l are complex coefficients. The two representations in (8.1) are equivalent because the ‘coprime DFT columns’ W_q^{kn} where $(k, q) = 1, 0 \leq k \leq q-1$ and the first $\phi(q)$ shifted versions of $c_q(n)$ both span the same space of signals \mathcal{S}_q [52]. \mathcal{S}_q contains those period- q signals that can have non-zero DFT value only at the ‘coprime frequencies’ $2\pi k/q$ where $0 \leq k \leq q-1$ and k is coprime to q . Thus, signals in \mathcal{S}_q have an infinite length in the time domain and infinitesimally small bandwidth in the frequency domain. The ideal q -th Ramanujan filter has $c_q(n)$ as its filter coefficients. Thus, it has a frequency response

$$C_q(e^{j\omega}) = \sum_n c_q(n) e^{-j\omega n} = 2\pi \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} \delta(\omega - 2\pi k/q), \quad (8.2)$$

where $\delta(\cdot)$ is the Dirac-delta function. The frequency response $C_q(e^{j\omega})$ is nonzero only at $\phi(q)$ coprime frequencies.

When an input signal $s(n)$ passes through an ideal Ramanujan filter with impulse response $c_q(n)$, the output of this digital filter has the form

$$\sum_k s(k) c_q(n-k), \quad (8.3)$$

which is a linear combination of shifted versions of $c_q(n)$. The collection of all outputs is therefore the set of all signals spanned by $\{c_q(n-k)\}$ and is called the **output space** of the filter $c_q(n)$. It can be shown [52] that this is nothing but the $\phi(q)$ -dimensional Ramanujan subspace \mathcal{S}_q which can also be spanned by the $\phi(q)$ shifted versions $c_q(n-k), 0 \leq k \leq \phi(q)-1$ of the q -th Ramanujan sum $c_q(n)$.

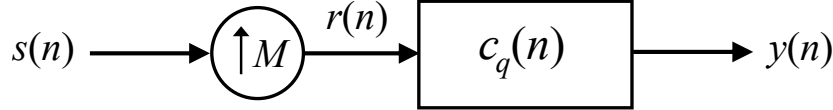


Figure 8.1: An interpolation filter. Here the filter has impulse response $c_q(n)$ which is the q th Ramanujan sum.

Now consider Fig. 8.1 where the ideal Ramanujan filter is preceded by an expander $\uparrow M$. The expander has the input-output description [188]

$$r(n) = \begin{cases} s(n/M) & \text{if } n \text{ is multiple of } M, \\ 0 & \text{otherwise.} \end{cases} \quad (8.4)$$

This system shown in the Fig. 8.1 is an *interpolation filter*. Outputs of interpolation filters often arise in the context of filter bank bases (Sec. 11.4, of [188]), and are also referred to as shift invariant subspaces because if $y(n)$ is in this space then so is $y(n - iM)$ for any integer i . Let the output space of the Ramanujan interpolation filter from Fig. 8.1 be denoted by \mathcal{V}_M . Now, since the filter output is a linear combination of the form

$$\sum_k s(k) c_q(n - kM), \quad (8.5)$$

the output space is usually only a subspace of \mathcal{S}_q , that is, $\mathcal{V}_M \subseteq \mathcal{S}_q$. We will now show that under certain conditions, the output space \mathcal{V}_M of Fig. 8.1 is the entire Ramanujan subspace \mathcal{S}_q .

Lemma 11 (Interpolation using ideal Ramanujan filter). The output space \mathcal{V}_M of the interpolation filter shown in Fig. 8.1 is the entire Ramanujan space \mathcal{S}_q when $(M, q) = 1$.

Proof. To show that Fig. 8.1 spans the q -th Ramanujan subspace \mathcal{S}_q , we need to show that any $x(n) \in \mathcal{S}_q$ can be produced as the output of the interpolation filter for some input signal $s(n)$. Let the $x(n) \in \mathcal{S}_q$ be

$$x(n) = \sum_{l=0}^{\phi(q)-1} \beta_l c_q(n - l). \quad (8.6)$$

We want to find an input $s(n)$ such that the output $y(n)$ given by Eq. (8.5) equals $x(n)$ in (8.6). When $(M, q) = 1$, by Euclid's lemma [61] there exists $\alpha, \beta \in \mathbb{Z}$ such that

$$\alpha M + \beta q = 1. \quad (8.7)$$

Now consider the input $s(n)$ to be

$$s(n) = \begin{cases} \beta_l & \text{if } n = \alpha l, 0 \leq l \leq \phi(q) - 1, \\ 0 & \text{otherwise.} \end{cases} \quad (8.8)$$

The filter output can then be expressed as

$$y(n) = \sum_k s(k) c_q(n - kM) = \sum_{l=0}^{\phi(q)-1} \beta_l c_q(n - \alpha l M). \quad (8.9)$$

But since $c_q(n)$ is periodic with period q , we have

$$c_q(n - \alpha l M) = c_q(n - (\alpha l M \bmod q)) \quad (8.10)$$

$$= c_q(n - l(\alpha M \bmod q)) = c_q(n - l). \quad (8.11)$$

The last equality follows from Eq. (8.7). Thus,

$$y(n) = \sum_{l=0}^{\phi(q)-1} \beta_l c_q(n - l) = x(n). \quad (8.12)$$

This proves that $\mathcal{V}_M = \mathcal{S}_q$ when $(M, q) = 1$. \square

A natural question to ask is whether $\mathcal{V}_M = \mathcal{S}_q$ can also happen when $(M, q) \neq 1$, or whether $(M, q) \neq 1$ guarantees that \mathcal{V}_M is a proper subspace of \mathcal{S}_q . This will be explored in Sec. 8.3. We first show how the interpolation filter model for signals in \mathcal{S}_q can be extended for all signals satisfying $y(n) = y(n + q)$ by constructing a synthesis filter bank model.

A signal satisfying $y(n) = y(n + q)$ is said to have a repetition index of q . Its period (smallest repetition interval) can be any divisor of q . We can represent such a $y(n)$ as [62]

$$y(n) = \sum_{q_i | q} y_{q_i}(n), \quad (8.13)$$

where q_i are divisors of q and $y_{q_i}(n) \in \mathcal{S}_{q_i}$. As a result, such a signal can be represented as in Fig. 8.2, where $(M_i, q_i) = 1$. In fact the output space of Fig. 8.2 is precisely the space \mathcal{R}_q of all periodic signals satisfying $y(n) = y(n + q)$ because $\mathcal{R}_q = \bigoplus_{q_i | q} \mathcal{S}_{q_i}$, where \oplus denotes the outer sum. This gives a useful synthesis filter-bank representation for any periodic signal satisfying $y(n) = y(n + q)$.

In the future, if we can design a multirate analysis bank in conjunction with the above synthesis filter bank to project a noisy periodic signal onto the subspace

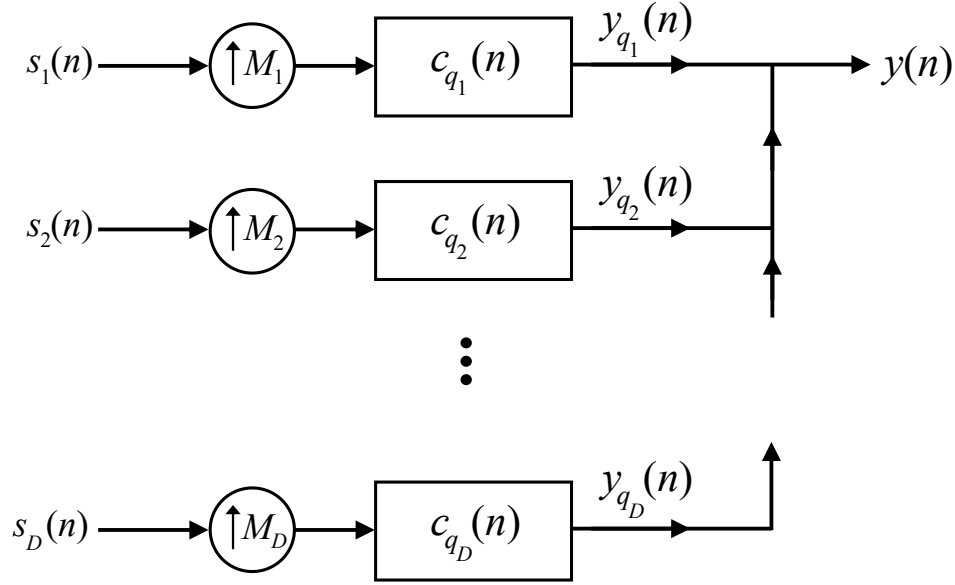


Figure 8.2: Ramanujan multirate synthesis filter bank for signals satisfying $y(n) = y(n + q)$. Here q_i are the D divisors of the integer q , and M_i are coprime to q_i , that is, $(M_i, q_i) = 1$.

\mathcal{R}_q , it will be useful for denoising. One way to achieve this is by using Fig. 2 preceded by an analysis filter bank consisting of ‘biorthogonal partner’ filters [205], [206]. Practically, we need to replace the Ramanujan filters $c_q(n)$ with their FIR approximations, and the output space of the filter bank would only be an approximation of \mathcal{R}_q . An important point to note is that *when the interpolation ratio $M > 1$, it is possible to find FIR biorthogonal filter partners corresponding to FIR approximations of Ramanujan filters*. This cannot be achieved when $c_q(n)$ is used as a digital filter without an expander $\uparrow M$. The filter design problems here are quite involved. We, therefore, leave this for future work and only briefly mention this possible application in the current chapter.

8.3 Further Results on the Output Space

In this section, we consider further theoretical results regarding the interpolation filter model from Fig. 8.1. Firstly, note that the converse of Lemma 11 is not true in general. That is, if $(M, q) \neq 1$, the output space of the interpolation filter can still be \mathcal{S}_q .

For example, take $q = 2$ and M to be any even integer. Clearly $(M, q) = 2$. As the subspace \mathcal{S}_2 has dimension one and is spanned by $c_2(n)$, it just contains all signals for the form $\gamma c_2(n)$ where γ can be any complex scalar. Now note that when $q = 2$,

the interpolation scheme in Fig. 8.1 can produce $y(n) = \gamma c_2(n)$ irrespective of the value of M , for instance, by choosing $s(n) = \gamma \delta(n)$. Thus, it is possible for the interpolation filter to span \mathcal{S}_2 even when $(M, 2) \neq 1$. We come back to this later in Sec. 8.3.2 to discuss whether $\mathcal{V}_M = \mathcal{S}_q$ can happen more generally when $q \neq 2$ and $(M, q) \neq 1$.

Since \mathcal{V}_M is a subspace of \mathcal{S}_q , it is equal to \mathcal{S}_q if and only if its dimension is $\phi(q)$, because \mathcal{S}_q has dimension $\phi(q)$. To examine the dimension of the output space \mathcal{V}_M , we first prove the following result.

Lemma 12. If $(M, q) = g$, the expander $\uparrow M$ in Fig. 8.1 can be changed to $\uparrow g$ without affecting the output space spanned by the interpolation filter.

In other words, we can i) divide M by any of its factors that are coprime to q , or ii) multiply M by any integers that are coprime to q , without altering the output space of the interpolation filter.

Proof. Let the output subspaces spanned by the interpolation filters with expanders $\uparrow M$ and $\uparrow g$ be \mathcal{V}_M and \mathcal{V}_g respectively. These interpolation filters are shown in Fig. 8.3 with correspondingly labeled input and output signals. To show that the two output spaces are the same, we will show that for any given input $s_M(n)$ there exists a corresponding $s_g(n)$ such that $y_M(n) = y_g(n)$ and vice versa.

Since g divides M , let $M = M'g$. Now for any given input $s_M(n)$, choose $s_g(M'n) = s_M(n)$ and 0 otherwise. It can be checked that both outputs $y_M(n)$ and $y_g(n)$ are the same in this case. Thus, $\mathcal{V}_M \subseteq \mathcal{V}_g$.

Next, since $(M, q) = g$, by Euclid's Lemma (or Bežout identity) [61], $\exists \alpha, \beta \in \mathbb{Z}$ such that

$$\alpha M + \beta q = g. \quad (8.14)$$

Now for any given input $s_g(n)$, choose

$$s_M(n) = \begin{cases} s_g(l) & \text{if } n = \alpha l, \\ 0 & \text{otherwise.} \end{cases} \quad (8.15)$$

Then the output $y_M(n)$ is

$$y_M(n) = \sum_{k=-\infty}^{\infty} s_M(k) c_q(n - Mk) = \sum_{l=-\infty}^{\infty} s_g(l) c_q(n - M\alpha l). \quad (8.16)$$

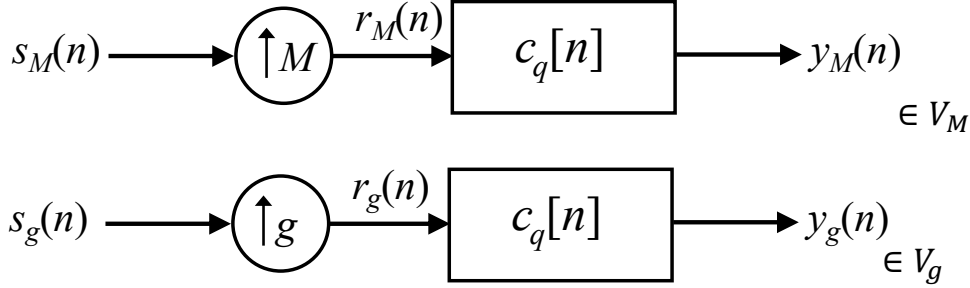


Figure 8.3: Figure for Lemma 12. When $(M, q) = g$, the output spaces \mathcal{V}_M and \mathcal{V}_g are identical.

Since $c_q(n)$ is periodic with period q , we get

$$y_M(n) = \sum_{l=-\infty}^{\infty} s_g(l) c_q(n - (M\alpha \bmod q)l) \quad (8.17)$$

$$= \sum_{l=-\infty}^{\infty} s_g(l) c_q(n - gl) = y_g(n). \quad (8.18)$$

The second last equality follows from Eq. (8.14). Thus, $\mathcal{V}_g \subseteq \mathcal{V}_M$. Combined with $\mathcal{V}_M \subseteq \mathcal{V}_g$, we get $\mathcal{V}_g = \mathcal{V}_M$. \square

8.3.1 A Bound on the Dimension of \mathcal{V}_M

In this section, we prove a bound on the dimension of \mathcal{V}_M , or equivalently \mathcal{V}_g , when $(M, q) = g$. Note that

$$\mathcal{V}_M = \mathcal{V}_g = \text{span}\{c_q(n), c_q(n \pm g), c_q(n \pm 2g), \dots\} \quad (8.19)$$

Since $c_q(n)$ is periodic with period q , it is enough to consider

$$\text{span}\{c_q(n), c_q(n - g), c_q(n - 2g), \dots, c_q(n - (q' - 1)g)\} \quad (8.20)$$

where $q' = q/g$. Now clearly, $\dim \mathcal{V}_g = \text{rank}(\mathbf{B}_q^{(g)})$ where $\mathbf{B}_q^{(g)}$ is a $q \times q'$ matrix

$$\mathbf{B}_q^{(g)} = \begin{bmatrix} | & & | & & | \\ c_q(n) & c_q(n - g) & \dots & c_q(n - (q' - 1)g) \\ | & & | & & | \end{bmatrix}. \quad (8.21)$$

For every $0 \leq l \leq q' - 1$, the $(l + 1)$ -th column of $\mathbf{B}_q^{(g)}$ has the entries $c_q(n - lg)$ for $n = 0, 1, \dots, q - 1$. Note that matrix $\mathbf{B}_q^{(g)}$ retains every g -th column of the circulant matrix \mathbf{B}_q defined in [52], starting from the first column. For $g = 1$, $\mathbf{B}_q^{(g)} = \mathbf{B}_q$, and it has rank $\phi(q)$ [52]. Here we want to more generally find the rank of $\mathbf{B}_q^{(g)}$, where g can be any divisor of q . We now prove the following.

Lemma 13. $\dim \mathcal{V}_M = \text{rank}(\mathbf{B}_q^{(g)}) \leq \phi(q')$.

Proof. Consider the columns of $\mathbf{B}_q^{(g)}$:

$$c_q(n - lg) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} W_q^{(n-lg)k} = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} W_q^{kn} W_q^{-lgk}. \quad (8.22)$$

Since g divides q , we get

$$c_q(n - lg) = \sum_{\substack{k=0 \\ (k,q)=1}}^{q-1} W_q^{kn} W_{q'}^{-lk} \quad (\text{where } q' = q/g). \quad (8.23)$$

Thus, the matrix $\mathbf{B}_q^{(g)}$ can be factorized as

$$\mathbf{B}_q^{(g)} = \mathbf{V}_q \cdot \left(\mathbf{C}_{q'}^{(q)} \right)^H, \quad (8.24)$$

where \mathbf{V}_q is a $q \times \phi(q)$ matrix formed by retaining the ‘coprime index columns’ of the $q \times q$ DFT matrix

$$\mathbf{V}_q = \left[\begin{array}{c|c|c|c|c} | & & | & & | \\ W_q^n & \dots & W_q^{kn} & \dots & W_q^{(q-1)n} \\ | & & | & & | \end{array} \right], \quad \begin{array}{l} 0 \leq k \leq q-1 \\ \text{s.t. } (k,q)=1, \\ 0 \leq n \leq q-1 \end{array} \quad (8.25)$$

and $\mathbf{C}_{q'}^{(q)}$ is a $q' \times \phi(q)$ matrix

$$\mathbf{C}_{q'}^{(q)} = \left[\begin{array}{c|c|c|c|c} | & & | & & | \\ W_{q'}^l & \dots & W_{q'}^{kl} & \dots & W_{q'}^{(q-1)l} \\ | & & | & & | \end{array} \right], \quad \begin{array}{l} 0 \leq k \leq q-1 \\ \text{s.t. } (k,q)=1, \\ 0 \leq l \leq q'-1. \end{array} \quad (8.26)$$

Note that even though the q' -th root of unity is involved in $\mathbf{C}_{q'}^{(q)}$, the index k still takes values in the range $1 \leq k \leq q$ satisfying $(k, q) = 1$. As \mathbf{V}_q is a $q \times \phi(q)$ submatrix of the $q \times q$ DFT matrix, it has rank $\phi(q)$. To find the rank of $\mathbf{C}_{q'}^{(q)}$, consider the set of its columns:

$$\text{rank}(\mathbf{C}_{q'}^{(q)}) = \dim \left\{ W_{q'}^{kl}, 0 \leq k \leq q-1, (k, q) = 1 \right\} \quad (8.27)$$

$$= \dim \left\{ W_{q'}^{(k \bmod q')l}, 0 \leq k \leq q-1, (k, q) = 1 \right\}. \quad (8.28)$$

Note that since q' is a divisor of q , $(k, q) = 1$ implies that $(k, q') = 1$. Then by Euclid’s lemma [61], $(k - dq', q') = 1$ for all $d \in \mathbb{Z}$. So, in particular,

$((k \bmod q'), q') = 1$ whenever $(k, q) = 1$. Combining this with the fact that $0 \leq (k \bmod q') \leq q' - 1$, we see that there can only be $\phi(q')$ distinct values of $(k \bmod q')$ in Eq. (8.28). Thus, $\text{rank}(\mathbf{C}_{q'}^{(q)}) \leq \phi(q')$. Now since

$$\text{rank}(\mathbf{B}_q^{(g)}) \leq \min\{\text{rank}(\mathbf{V}_q), \text{rank}(\mathbf{C}_{q'}^{(q)})\}, \quad (8.29)$$

$\text{rank}(\mathbf{B}_q^{(g)})$ is also upper bounded by $\phi(q')$. This completes the proof. \square

We end this section by presenting a conjecture that is motivated by the fact that $\text{rank}(\mathbf{B}_q^{(g)})$ was found to be equal to $\phi(q')$ in all examples that we numerically checked:

Conjecture 2. $\dim \mathcal{V}_M = \text{rank}(\mathbf{B}_q^{(g)}) = \phi(q')$.

In other words, we observed that the bound proved in Lemma 13 is in fact the dimension of \mathcal{V}_M . The validity of this conjecture depends on another conjecture:

Conjecture 3. Let $q, q' \in \mathbb{N}$ be such that $q'|q$. Then the following two sets are identical:

$$\mathcal{A} = \{k \bmod q' \mid 0 \leq k \leq q - 1, (k, q) = 1\}, \quad (8.30)$$

$$\mathcal{B} = \{k' \mid 0 \leq k' \leq q' - 1, (k', q') = 1\}. \quad (8.31)$$

Furthermore, for any $0 \leq k' \leq q' - 1$ such that $(k', q') = 1$, there are exactly $\phi(q)/\phi(q')$ values of k satisfying $(k, q) = 1$, $0 \leq k \leq q - 1$ such that $k \bmod q' = k'$.

Conjecture 3 was numerically checked to be true for many different values of q and q' , however, a proof is not yet known. Furthermore, it can be readily verified that the Conjecture 1 from Chapter 7 about the decimation of Ramanujan sums

$$c_q(Mn) = \frac{\phi(q)}{\phi(q')} c_{q'}(n), \quad (8.32)$$

where $(M, q) = g$ and $q' = q/g$, is also a direct consequence of Conjecture 3.

8.3.2 Can $\mathcal{V}_M = \mathcal{S}_q$ When $(M, q) \neq 1$?

As stated before, $\mathcal{V}_M = \mathcal{S}_q$ can happen if and only if $\dim \mathcal{V}_M = \phi(q)$. Since $\dim \mathcal{V}_M$ is upper bounded by $\phi(q')$, where $q' = q/(M, q)$ (Lemma 13), let's examine the ratio $\phi(q)/\phi(q')$. Let q and q' have prime factorizations [61]

$$q = \prod_i p_i^{m_i} \quad \text{and} \quad q' = \prod_i p_i^{n_i}, \quad (8.33)$$

where p_1, p_2, \dots are all the primes in increasing order and the exponents $m_i, n_i \geq 0$. Since $q' | q$, we have $n_i \leq m_i \forall i$. Then, using the formula for the Euler totient function [61], we have

$$\frac{\phi(q)}{\phi(q')} = \frac{\prod_{i:m_i \geq 1} p_i^{m_i-1} (p_i - 1)}{\prod_{i:n_i \geq 1} p_i^{n_i-1} (p_i - 1)} \quad (8.34)$$

$$= \prod_{i:n_i \geq 1} p_i^{m_i-n_i} \prod_{\substack{i:n_i=0, \\ m_i \geq 1}} p_i^{m_i-1} (p_i - 1). \quad (8.35)$$

The first product is over the primes that divide both q and q' , and the second product is over the primes that divide q but not q' . With this, it can be verified that the ratio $\phi(q)/\phi(q')$ is equal to 1 if and only if

$$\begin{cases} m_i = n_i & \forall i \text{ s.t. } n_i \geq 1, \\ m_i = 1, p_i = 2 & \forall i \text{ s.t. } n_i = 0, m_i \geq 1. \end{cases} \quad (8.36)$$

It can be verified that this condition is satisfied by q and q' if and only if either i) $q = q'$, or ii) $q = 2q'$ for an odd integer q' .

Now given that these are the only possibilities for which $\phi(q) = \phi(q')$, as a consequence of Conjecture 2, we can state the following stronger version of Lemma 11: If the Conjecture 2 is true, $\mathcal{V}_M = \mathcal{S}_q$ **if and only if** either i) $(M, q) = 1$, or ii) $(M, q) = 2$, and $q/2$ is odd.

8.4 Locations of Zeros of FIR Ramanujan Filters

In this section, we study the locations of zeros of Ramanujan filters. For the rest of the chapter, we consider the FIR Ramanujan filters, which have the first few periods of the Ramanujan sums as their filter coefficients. Note that this is different from the ideal Ramanujan filters considered in previous sections.

The transfer function of the q -th FIR Ramanujan filter having one period of the Ramanujan sum as its filter coefficient is given by

$$C_q(z) = \sum_{n=0}^{q-1} c_q(n) z^{-n}. \quad (8.37)$$

Note that later in Sec. 8.5.4 will also define extended filters based on l successive periods of $c_q(n)$ [63][64], and its transfer function will be denoted by $C_q^{(l)}(z)$:

$$C_q^{(l)}(z) = \sum_{n=0}^{ql-1} c_q(n) z^{-n}. \quad (8.38)$$

In the light of this definition from Eq. (8.38), the transfer function of the filter with a single period of Ramanujan sum as its filter coefficient should be denoted as $C_q^{(1)}(z)$. We drop the superscript (l) when $l = 1$ to avoid the clutter.

In order to make $C_q(z)$ a polynomial in z with positive powers, we multiply it by z^{q-1} to get $\widehat{C}_q(z)$:

$$\widehat{C}_q(z) = z^{q-1}C_q(z) = \sum_{n=0}^{q-1} c_q(n)z^{q-1-n}. \quad (8.39)$$

Note that the zeros of $\widehat{C}_q(z)$ and $C_q(z)$, which are not at the origin, are the same. Also, $C_q(z)$ has no zero at $z = 0$ for any positive integer q . Hence, for simplicity, we study the location of zeros of $\widehat{C}_q(z)$ and discard the zeros at the origin to get the zeros of $C_q(z)$.

As $c_q(0) \neq 0$ for any q , $\widehat{C}_q(z)$ is a degree $q - 1$ polynomial in z . Hence, it has $q - 1$ (possibly complex) zeros. In this section, we will prove the following Lemma. The terms ‘cyclotomic polynomial’ and ‘critical points’ are explained after the lemma.

Lemma 14 (Zeros of Ramanujan Filters). Let $C_q(z) = \sum_{n=0}^{q-1} c_q(n)z^{-n}$ and let $\widehat{C}_q(z) = z^{q-1}C_q(z)$. Out of the $q - 1$ zeros of $\widehat{C}_q(z)$:

1. $\phi(q) - 1$ zeros are located strictly inside the unit circle in z -plane and correspond to the critical points of the q -th cyclotomic polynomial $F_q(z)$ defined in (8.40).
2. The remaining $q - \phi(q)$ zeros are located on the unit circle in z -plane at $W_q^k, (k, q) \neq 1$.

The zeros of $C_q(z)$ are the non-zero zeros of $\widehat{C}_q(z)$.

The q -th cyclotomic polynomial $F_q(z)$ mentioned above is defined as

$$F_q(z) \triangleq \prod_{\substack{k=1 \\ (k,q)=1}}^q (z - W_q^k). \quad (8.40)$$

The term ‘critical points’ of $F_q(z)$ refers to the zeros of $dF_q(z)/dz$.

Note that $F_q(z)$ is a degree $\phi(q)$ polynomial. Its zeros are located on the unit circle in the z -plane at $z = W_q^k$, where $(k, q) = 1, 0 \leq k \leq q - 1$. Also note

$$\dot{F}_q(z) \triangleq \frac{dF_q(z)}{dz} = F_q(z) \left(\sum_{\substack{k=1 \\ (k,q)=1}}^q \frac{1}{z - W_q^k} \right). \quad (8.41)$$

We will appeal to the following theorem by Lucas in the proof.

Theorem 8 (Lucas, theorem (6,1) of [207]). All the critical points of any non-constant polynomial $f(z)$ lie in the convex hull \mathcal{H} of zeros of $f(z)$. If the zeros of $f(z)$ are not collinear, no critical point of $f(z)$ lies on the boundary of \mathcal{H} unless it is a multiple zero of $f(z)$.

Proof of Lemma 14. We can rewrite $C_q(z)$ as follows:

$$C_q(z) = \sum_{n=0}^{q-1} c_q(n) z^{-n} = \sum_{n=0}^{q-1} \sum_{\substack{k=1 \\ (k,q)=1}}^q W_q^{kn} z^{-n}. \quad (8.42)$$

$$\begin{aligned} \text{Therefore, } \widehat{C}_q(z) &= z^{q-1} \sum_{\substack{k=1 \\ (k,q)=1}}^q \sum_{n=0}^{q-1} (W_q^k z^{-1})^n \\ &= z^{q-1} \sum_{\substack{k=1 \\ (k,q)=1}}^q \frac{1 - (W_q^k z^{-1})^q}{1 - W_q^k z^{-1}} \\ &= z^{q-1} (1 - z^{-q}) \sum_{\substack{k=1 \\ (k,q)=1}}^q \frac{1}{1 - W_q^k z^{-1}} \\ &= (z^q - 1) \sum_{\substack{k=1 \\ (k,q)=1}}^q \frac{1}{z - W_q^k} \\ &= \frac{z^q - 1}{F_q(z)} \dot{F}_q(z). \end{aligned} \quad (8.43)$$

A related expression for $\dot{F}_q(z)/F_q(z)$ was obtained in a different context for one sided z -transform of Ramanujan sums in [208]:

$$\sum_{n=0}^{\infty} c_q(n) z^{-n} = \frac{z dF_q(z)/dz}{F_q(z)} \quad (8.44)$$

The infinite sum in (8.44) converges to the RHS only when $|z| > 1$, whereas no such assumption is required in the derivation (8.43).

Now, from (8.43), we can see that either of the two factors in (8.43) can contribute to zeros of $\widehat{C}_q(z)$. For the first factor, the zeros of $F_q(z)$ are located at $z = W_q^k$, where $(k, q) = 1, 0 \leq k \leq q-1$ and the zeros of the numerator $(z^q - 1)$ are at $z = W_q^k, 0 \leq k \leq q-1$. Hence, after cancellation we are left with $q - \phi(q)$ zeros for $\widehat{C}_q(z)$ which are located at $W_q^k, (k, q) \neq 1$.

The remaining $\phi(q) - 1$ zeros of $\widehat{C}_q(z)$ are contributed by the other factor $\dot{F}_q(z)$. Since $F_q(z)$ is a non-constant polynomial, by Lucas's theorem we have that the critical points of $F_q(z)$, namely the zeros of $\dot{F}_q(z)$, lie in the convex hull of the zeros of $F_q(z)$. Since the zeros of $F_q(z)$ are on the unit circle in the z -plane, the convex hull of the zeros is strictly inside the unit circle, except at the zeros itself. Now note that $\dot{F}_q(z)$ cannot have a zero where $F_q(z)$ is zero, since none of the zeros of $F_q(z)$ are repeated. Hence, all the zeros of $\widehat{C}_q(z)$ contributed by the term $\dot{F}_q(z)$ lie strictly inside the unit circle in the z -plane. This completes the proof of Lemma 14. \square

Now we obtain an expression for $C_q(z)$ which enables efficient implementation of Ramanujan filter banks as explained in Sec. 8.6 later. We have

$$C_q(z) = \frac{\widehat{C}_q(z)}{z^{q-1}} = \frac{z^q - 1}{z^{q-\phi(q)} F_q(z)} \cdot \frac{\dot{F}_q(z)}{z^{\phi(q)-1}}. \quad (8.45)$$

Applying an identity for cyclotomic polynomials [209], [210]

$$z^q - 1 = \prod_{q_k | q} F_{q_k}(z) \quad (8.46)$$

gives us

$$C_q(z) = \left(\prod_{\substack{q_k | q \\ q_k < q}} \widehat{F}_{q_k}(z) \right) \cdot \frac{\dot{F}_q(z)}{z^{\phi(q)-1}}, \quad (8.47)$$

where

$$\widehat{F}_q(z) = z^{-\phi(q)} F_q(z) \quad (8.48)$$

is a causal version of the cyclotomic polynomial.

8.5 Identifying Further Structure in the Locations of Zeros

Fig. 8.4 shows plots of zeros of $C_q(z)$ for some selected values of q . In each of these plots, the zeros are indeed in accordance with Lemma 14. We can see that the locations of zeros seem to exhibit a lot more structure than what is stated in Lemma 14. For example, notice that $z = 1$ is always a zero of $C_q(z)$ for any $q > 1$. This follows from the fact that $\sum_{n=0}^{q-1} c_q(n) = 0$ for any q . In the following subsections, we consider special cases of q and further derive properties for zeros of corresponding families of Ramanujan filters. We believe that these are of sufficient academic interest to merit inclusion here. We also obtain zeros for generalized filters with l periods of Ramanujan sums as filter coefficients instead of just one.

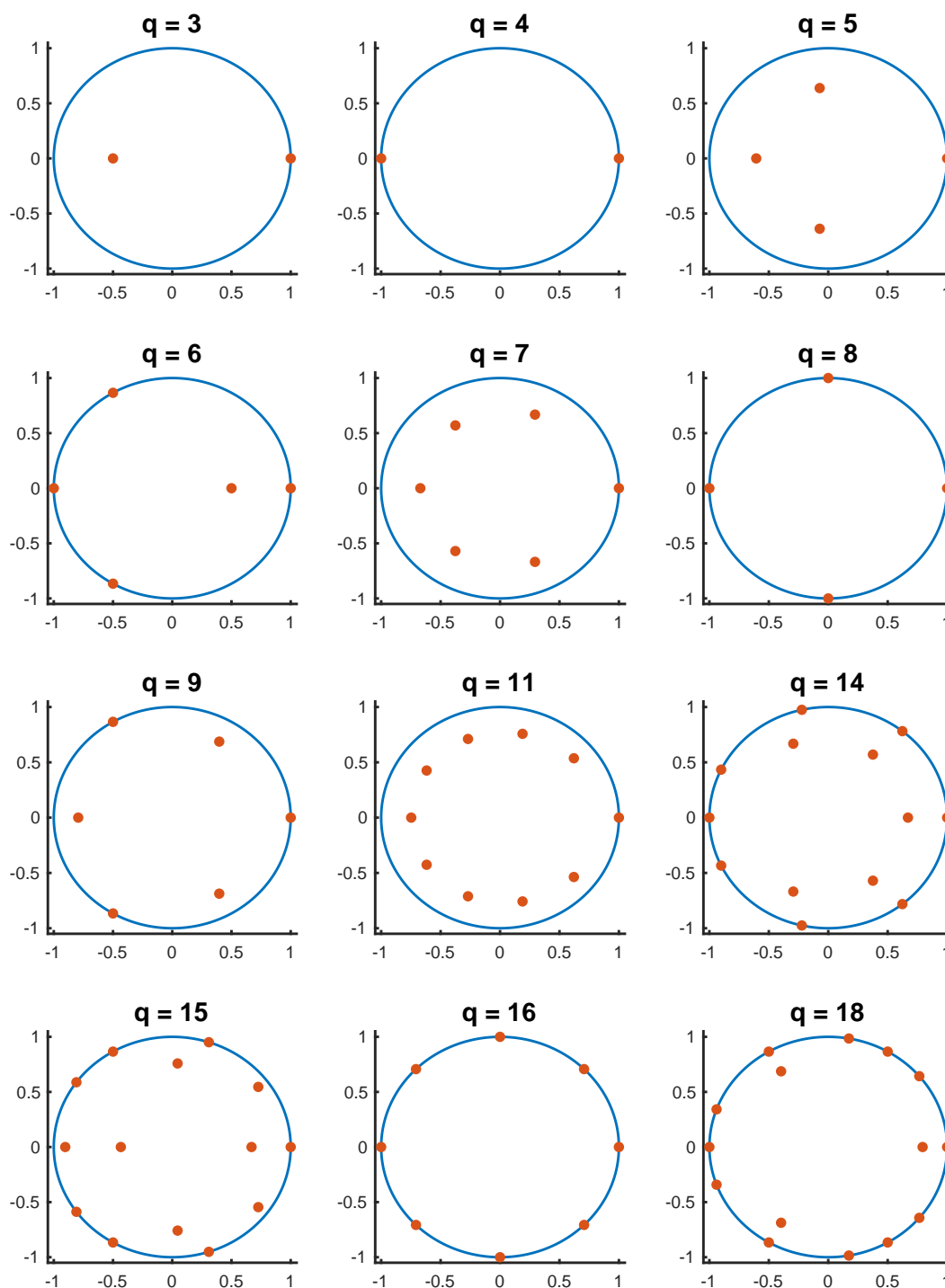


Figure 8.4: Zeros of $C_q(z)$ for some selected values of q plotted on the z -plane.

8.5.1 Case When q is a Power of Two

For $q = 2^m$ where m is a natural number, we have [52]

$$c_q(n) = \begin{cases} 0 & \text{if } 2^{m-1} \text{ does not divide } n, \\ -2^{m-1} & \text{if } 2^{m-1} \text{ divides } n \text{ but } 2^m \text{ does not divide } n, \\ 2^{m-1} & \text{if } 2^m \text{ divides } n. \end{cases} \quad (8.49)$$

Hence we have

$$\begin{aligned} \widehat{C}_q(z) &= 2^{m-1} z^{(2^m-1)} - 2^{m-1} z^{(2^{m-1}-1)} \\ &= 2^{m-1} z^{(2^{m-1}-1)} (z^{(2^{m-1})} - 1). \end{aligned} \quad (8.50)$$

Discarding the zeros at $z = 0$, the zeros of $C_q(z)$ are the 2^{m-1} -th roots of unity, and lie on the unit circle in the z -plane. No zeros lie inside the unit circle. We can indeed verify this result from Fig. 8.4. The zeros of $C_4(z)$ are the square roots of unity. Similarly, the zeros of $C_q(z)$ for $q = 8$ and $q = 16$ are the fourth and the eighth roots of unity, respectively.

8.5.2 Case When q is a Prime Number

For $q = p$, a prime, we have [52]

$$c_q(n) = \begin{cases} q-1 & \text{if } n \text{ is multiple of } q, \\ -1 & \text{otherwise.} \end{cases} \quad (8.51)$$

So we have

$$\begin{aligned} \widehat{C}_q(z) &= (q-1)z^{q-1} - z^{q-2} - z^{q-3} - \dots - 1 \\ &= (z-1) [(q-1)z^{q-2} + (q-2)z^{q-3} + \dots + 2z + 1]. \end{aligned} \quad (8.52)$$

Hence, we have one zero located at $z = 1$, and the other $q-2$ zeros are the zeros of the special integer coefficient polynomial

$$g_q(z) = (q-1)z^{q-2} + (q-2)z^{q-3} + \dots + 2z + 1. \quad (8.53)$$

Since q is prime, we know from Lemma 14 that the zeros of $g_q(z)$ are strictly inside the unit circle. We can verify this result for prime values of $q = 3, 5, 7, 11$ from the Fig. 8.4. Except at $z = 1$, all the zeros of $C_q(z)$ lie strictly inside the unit circle.

8.5.3 Case When q is a Power of a Prime

For $q = p^m$ where p is a prime, we have [52]

$$c_q(n) = \begin{cases} 0 & \text{if } p^{m-1} \text{ does not divide } n, \\ -p^{m-1} & \text{if } p^{m-1} \text{ divides } n \text{ but } p^m \text{ does not divide } n, \\ (p-1)p^{m-1} & \text{if } p^m \text{ divides } n. \end{cases} \quad (8.54)$$

Hence we have

$$\begin{aligned}\widehat{C}_q(z) &= (p-1)p^{m-1}z^{(p^{m-1})} + \sum_{k=1}^{p-1} -p^{m-1}z^{(p^{m-1}-kp^{m-1})} \\ &= p^{m-1}z^{(p^{m-1}-1)} \left[(p-1)z^{((p-1)p^{m-1})} - \sum_{k=1}^{p-1} z^{(p-k-1)p^{m-1}} \right].\end{aligned}\quad (8.55)$$

On further simplification, we get

$$\widehat{C}_q(z) = p^{m-1}z^{(p^{m-1}-1)}\widehat{C}_p(z^{p^{m-1}}). \quad (8.56)$$

Hence, discarding the zeros at $z = 0$, we have that the zeros of $\widehat{C}_q(z)$ are the p^{m-1} -th roots of zeros of $\widehat{C}_p(z)$.

To visualize the relation (8.56), compare the locations of zeros for $C_3(z)$ and $C_9(z)$ from Fig. 8.4. Here we have $p = 3$, $m = 2$, and $q = 3^2 = 9$. The zero of $C_3(z)$ at $z = 1$ gives rise to three zeros of $C_9(z)$ on the unit circle. These three zeros are the p^{m-1} -th i.e. 3^{rd} roots of $z = 1$. Similarly, the zero of $C_3(z)$ at $z = -1/2$ gives rise to three zeros of $C_9(z)$ that lie inside the unit circle, and correspond to cube roots of $z = -1/2$.

8.5.4 Case When Multiple Periods of Ramanujan Sums are Used in the Filter Definition

When the filter impulse response has l periods of $c_q(n)$ as filter coefficients, we have

$$C_q^{(l)}(z) = \sum_{n=0}^{ql-1} c_q(n)z^{-n} = \sum_{n=0}^{q-1} c_q(n)z^{-n} [1 + z^{-q} + \dots + z^{-(l-1)q}] \quad (8.57)$$

$$= \left(\sum_{n=0}^{q-1} c_q(n)z^{-n} \right) \left(\sum_{k=0}^{l-1} z^{-qk} \right) = C_q(z) \left(\frac{1 - z^{-ql}}{1 - z^{-q}} \right). \quad (8.58)$$

From this we can see that the $ql - 1$ zeros of $C_q^{(l)}(z)$ are of two categories: (a) the zeros of $C_q(z)$, and (b) zeros that are ql -th roots of unity which are not also q -th roots of unity. There are $q(l-1)$ such zeros of the second category. Fig. 8.5 shows the locations of zeros for $C_7^{(l)}$ for $1 \leq l \leq 5$.

8.5.5 A Special Case of q as a Product of Two Numbers

One can ask whether we can characterize zeros of $C_q(z)$ where $q = p_1 p_2$, where p_1 and p_2 are some integers. We were not able to characterize this for general integers

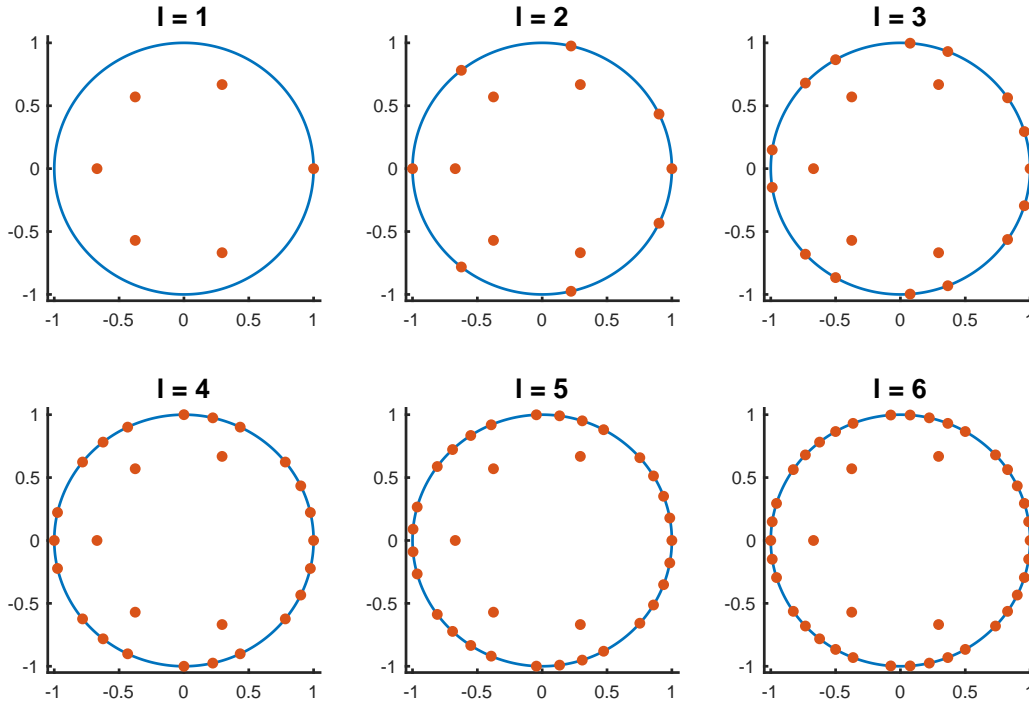


Figure 8.5: Zeros of $C_7^{(l)}(z)$ for $1 \leq l \leq 6$. As l increases, only the number of zeros on the unit circle increases, whereas the zeros inside the unit circle stay the same.

p_1 and p_2 . However, in this section we consider a very specific case of this, where $p_1 = 2$ and p_2 is odd. Here we use the multiplicative property of the Ramanujan sums [52]:

$$c_{p_1 p_2}(n) = c_{p_1}(n) c_{p_2}(n) \text{ if } (p_1, p_2) = 1. \quad (8.59)$$

When $p_1 = 2$ and p_2 is odd, the coprime condition is satisfied. Now note that $c_2(n) = \{1, -1\}$ in its first period. Hence for $q = 2p_2$ we have

$$C_q(z) = \sum_{n=0}^{q-1} c_2(n) c_{p_2}(n) z^{-n} = \sum_{n=0}^{q-1} (-1)^n c_{p_2}(n) z^{-n} \quad (8.60)$$

$$= \sum_{n=0}^{q-1} c_{p_2}(n) (-z)^{-n} = C_{p_2}^{(2)}(-z). \quad (8.61)$$

Hence the zeros of $C_q(z)$ for $q = 2p_2$, where p_2 is odd, are the negatives of the zeros of $C_{p_2}^{(2)}(z)$. As an example the zeros of $C_{14}(z)$ ($q = 14$ from Fig. 8.4) are the negatives of the zeros of $C_7^{(2)}(z)$ ($l = 2$ from Fig. 8.5).

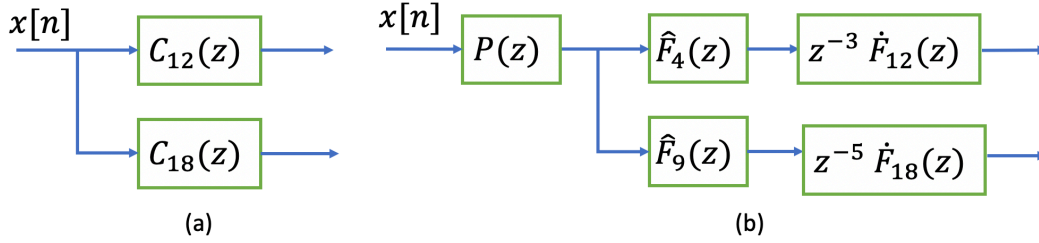


Figure 8.6: (a) Standard implementation of two filters from RFB, (b) Equivalent efficient implementation, by extracting a common factor $P(z)$.

8.6 Efficient Structure for Ramanujan Filter Banks

In this section, we show how the factorization (8.47) leads to a possible way of implementing the Ramanujan filter banks more efficiently. Firstly, note that cyclotomic polynomials have integer coefficients. The coefficients of $F_q(z)$ remain small even when q is large. In particular, the first 104 cyclotomic polynomials have no coefficients other than 1, 0, or -1 [209], [210]. Hence, the first 104 cyclotomic polynomials can be implemented without any multipliers!

Now we present an example of an efficient implementation of a 2-filter RFB consisting of $C_{12}(z)$ and $C_{18}(z)$. We pull out a common filter factor from the first terms of the expression (8.47) for $q = 12$ and $q = 18$. Note that

$$\begin{aligned} C_{12}(z) &= P(z) \hat{F}_4(z) \cdot \left(z^{-3} \dot{F}_{12}(z) \right) \text{ and} \\ C_{18}(z) &= P(z) \hat{F}_9(z) \cdot \left(z^{-5} \dot{F}_{18}(z) \right), \end{aligned} \quad (8.62)$$

where, $P(z)$ is the common factor given by

$$P(z) = \hat{F}_1(z) \hat{F}_2(z) \hat{F}_3(z) \hat{F}_6(z). \quad (8.63)$$

Hence, instead of standard implementation as in Fig. 8.6(a), we can implement the two filters as in Fig. 8.6(b). Since the filter $\hat{F}_q(z)$ has order of $\phi(q)$, $P(z)$ is a sixth order filter (since $\sum_{q_k|q} \phi(q_k) = q$ [61]). Therefore, the implementation as in Fig. 8.6(b) saves computations corresponding to a sixth order filter, by reusing the common filter factor $P(z)$.

This saves a lot of repetitive convolutions of signals with impulse responses. This particular example was chosen as it well illustrates the point of filters having common factors, owing to many common divisors of 12 and 18. In practice, an RFB usually has all filters from $C_1(z)$ to $C_N(z)$ for some integer N . When the filter bank has many filters, there are multiple ways in which common factors can be shared, and

some are more efficient than others. An interesting problem for the future would be to identify the most efficient way to exploit such common factors.

8.7 Concluding Remarks

In this chapter, we considered the interpolation filter model for Ramanujan subspace signals. In general, when the ideal q -th Ramanujan filter $C_q(e^{j\omega})$ is preceded by an expander $\uparrow M$, the output space of the filter is a subspace of the q -th Ramanujan subspace \mathcal{S}_q . But under the coprimality of M and q , the output space is the full Ramanujan subspace \mathcal{S}_q . For the general case when $(M, q) = g$, we proved that the dimension of the output space of the interpolation filter is bounded by $\phi(q/g)$, and furthermore conjectured it to be exactly $\phi(q/g)$. In the future, we hope to prove this by proving Conjecture 2 and Conjecture 3, for which we haven't been able to find counterexamples. The interpolation filter model (in particular Lemma 11) has potential application in developing a complete analysis-synthesis filter bank structure which is useful for denoising periodic signals. However, this requires several careful practical considerations, which will be explored in future work.

In this chapter, we also proved that all the zeros of FIR Ramanujan filters lie on or inside the unit circle in the z -plane. A general proof was based on Lucas's theorem. We considered different special cases of q , such as primes and powers of primes, and discovered interrelations between them. We also characterized the zeros of Ramanujan filters having l periods of $c_q(n)$ instead of just one period. It was shown with an illustrative example that the factorization of the filter transfer function in terms of cyclotomic polynomials and their derivatives opens up a possibility of efficient implementation of RFB.

LEARNING-BASED METHODS FOR REPRESENTING PERIODIC SIGNALS AND DOA ESTIMATION

9.1 Introduction

In this chapter, we explore the use of two popular learning-based methods for periodicity estimation and DOA estimation. In the first part, we examine whether a popular dictionary learning algorithm called K-SVD [74] can effectively learn periodicity dictionaries from data. In the second part, we explore the use of a convolutional neural network for DOA estimation.

It is common in many signal processing applications to use overcomplete dictionaries to represent data [70]–[75]. A good review of different dictionary types can be found in [70]. One way to construct representation dictionaries is to design them based on a-priori knowledge or a mathematical model regarding the signals that are to be represented. Such dictionaries are called *analytical dictionaries*. These dictionaries often have the advantage of fast implicit implementation. For example, a *tight frame* is an $m \times n$ dictionary \mathbf{D} that satisfies

$$\mathbf{D}\mathbf{D}^T\mathbf{x} = \frac{n}{m}\mathbf{x} \quad (9.1)$$

for all vectors \mathbf{x} . For this dictionary \mathbf{D} , \mathbf{D}^T is a possible analysis operator that produces analysis coefficients. This means that for a given signal \mathbf{x} , a valid coefficient vector corresponding to the dictionary \mathbf{D} can be computed easily as $\frac{m}{n}\mathbf{D}^T\mathbf{x}$. Some examples of analytical dictionaries are curvelets, contourlets, and bandlets.

Another way to develop dictionaries is to use a set of realizations of data for training. The advantage of such trained dictionaries that are directly learned from data is that they adapt to the non-idealities of the data well instead of relying on inaccurate modeling of these non-idealities. There are many dictionary learning algorithms, such as K-SVD [74], method of optimal directions (MOD) [211], union of orthonormal bases [212], and so on. Overcomplete dictionaries are widely used along with sparse vector recovery formulation for many applications such as denoising [75], DOA estimation [73], and data compression [72], to name a few. In addition to analytical and learned dictionaries, there is also a class of parametric

dictionaries [213]. Atoms of these dictionaries are structured but have a few free parameters that can be optimized by the training algorithm according to the data.

We have seen in Chapter 7 that Ramanujan sums are useful in identifying periodic structures in signals. The dictionaries used for this purpose are analytical dictionaries such as Ramanujan and Farey dictionaries [53], [62]. In the first part of this chapter, we explore the use of dictionary learning methods in the context of period estimation and periodic signal representation using dictionaries. We investigate whether K-SVD can learn the known dictionaries for period estimation from sparse coefficient periodic data generated from Ramanujan and Farey dictionaries. We also experimentally analyze the dependence of the similarity of the learned dictionary and the known dictionaries for period estimation with different parameters.

The second part of this chapter focuses on DOA estimation. Most of the classical DOA estimation methods are based on the standard data model used in array signal processing (Eq. (2.1)). Classical subspace-based methods like MUSIC, ESPRIT, and their variants (Sec. 2.2) estimate DOAs using an estimate of the covariance matrix from finite snapshots. Dictionary-based methods and associated sparse signal recovery approaches [214] are also popular for DOA estimation. Atomic norm minimization based methods [17] have also been developed for DOA estimation. These methods exploit the mathematical framework of the data model and hence work very well when the received data is indeed in accordance with the model. However, the widely used data model in array signal processing (Eq. (2.1)) is based on many simplifying assumptions such as far-field signals, isotropic sensors, and so on. Furthermore, in the presence of non-idealities or imperfections such as gain and phase errors, mutual coupling, and sensor position errors, the received data is not in accordance with the simplified data model (Eq. (2.1)). Such deviation of received data from the assumed data model deteriorates the performance of classical DOA estimation methods.

To overcome this, data-adaptive techniques such as machine learning and deep neural networks are used. Although there have been a few attempts in the past with radial basis functions [215] and support vector regression [216], the renewed interest is due to improved computational resources and the success of neural networks in many real-world problems [77]. In [148], Liu et. al. successfully show that a network based on auto-encoder and parallel multi-layer classifiers outperforms classical methods in the presence of array imperfections. Following that, many networks such as CNN [149] and ResNet [217] have been proposed for DOA estimation, and

demonstrate improved performance. Papers like [218], [219] also propose deep learning based methods to perform the traditional MUSIC algorithm.

In the second part of this chapter, we work with the CNN-based method proposed in [149]. The reason for this is twofold. First, [149] uses a simple four-layer convolutional network. Thus, it has fewer parameters and a smaller training time compared to many of the other networks for DOA estimation. And second, the method combines model-based and data-driven approaches. The data model is incorporated in the method through the so-called ‘proxy spectrum’, which is the input provided to the network. The network learns to refine this proxy spectrum based on the training data provided.

We show how the above method can be applied to sparse arrays such as nested arrays. In Sec. 2.3 we saw that well-designed sparse arrays such as nested arrays, coprime arrays, and MRAs can estimate $O(N^2)$ sources using N physical sensors in the array. By modifying the CNN-based method for sparse arrays, we demonstrate how to augment the machine learning methods with the ability to identify more sources than the number of sensors by the use of coarray principles. Furthermore, we also propose many possible alternative inputs (proxy spectra) to the network without changing the network structure. The experimental simulations demonstrate that the proposed proxy spectra can lead to improved DOA estimation performance, especially for the sparse arrays.

Chapter Outline. In Sec. 9.2 we review the known dictionaries for period estimation and a popular dictionary learning algorithm, the K-SVD. In Sec. 9.3 we formalize the dictionary-learning goals of this chapter and describe the experimental setting. Sec. 9.4 presents experimental results and discussions. Next, in Sec. 9.5 we review the CNN-based method from [149]. In Sec. 9.6 we explain how to adapt the CNN-based method for sparse arrays. In Sec. 9.7 we consider several other possible proxy spectra that can be used as input to the CNN. Sec. 9.8 presents experimental simulations demonstrating the ability to identify more sources. We also compare the DOA estimation errors for the various proposed proxy spectra. Sec. 9.9 concludes the chapter. The content of this chapter is based on papers [220], [221].

9.2 Review of Periodicity Dictionaries and K-SVD Algorithm

In this section, we review the known dictionaries for period estimation and the K-SVD algorithm used for dictionary learning.

9.2.1 Nested Periodic Dictionaries for Period Estimation

Any length N periodic signal \mathbf{x} with period $P \leq N$ can be represented as [52] $\mathbf{x} = \mathbf{F}_N \mathbf{b}$, where

$$\mathbf{F}_N = \begin{bmatrix} \mathbf{G}_{q_1} & \mathbf{G}_{q_2} & \dots & \mathbf{G}_{q_k} \end{bmatrix}. \quad (9.2)$$

Here, q_1, q_2, \dots, q_k are the divisors of signal length N . The matrices \mathbf{G}_{q_i} have the $\phi(q_i)$ circularly shifted versions of the q_i -th Ramanujan sums as the columns and hence span the corresponding periodic subspaces S_{q_i} . The columns are periodically repeated to make the number of rows equal to N . This representation is called the Ramanujan Periodic Transform (RPT). The component or hidden periods can be found by identifying the set of divisors $\{q_i\}$ that have corresponding non-zero entries in the coefficient vector \mathbf{b} .

Although the RPT is shown to have some applications, it is useful only when the component periods of the signal are the divisors of the signal length N . Generalizing the transform matrix for RPT, a framework of nested periodic matrices and dictionaries was proposed in [53]. Namely, the above matrix \mathbf{F}_N was replaced with full rank matrix \mathbf{B} defined as

$$\mathbf{B} = \begin{bmatrix} \mathbf{H}_{q_1} & \mathbf{H}_{q_2} & \dots & \mathbf{H}_{q_k} \end{bmatrix}, \quad (9.3)$$

where \mathbf{H}_{q_i} can be any $N \times \phi(q_i)$ matrix containing columns of period q_i . Examples of nested periodic matrices are natural basis matrices, DFT matrices, and Ramanujan matrices. In order to identify periods that may not be divisors of signal length, the use of overcomplete Farey dictionaries was proposed in [191] and was extended to other periodicity matrices in [53]. Here, the matrix \mathbf{B} in the above formulation is replaced with a dictionary

$$\mathbf{D} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_{P_{max}}]. \quad (9.4)$$

Here, each \mathbf{H}_i contains $\phi(i)$ columns of period i , hence all the periods from 1 to P_{max} are represented in this dictionary. The last periods of \mathbf{H}_i are truncated if needed in order to make their length equal to the signal length N . The period estimation problem is formulated as a sparse vector recovery problem in this setting. Examples of nested periodic dictionaries include the Farey Dictionary, which has columns from multiple DFT matrices, the Ramanujan dictionary, which has shifted Ramanujan sums as columns, and the natural periodic dictionary, which has columns from different identity matrices.

9.2.2 Learning Dictionary via K-SVD

Over the last two decades, many researchers have developed methods to learn dictionaries from the given data samples. The objective is to represent the data as sparse linear combinations of the columns of a dictionary. In this chapter, we use K-SVD [74], which is one such popular dictionary learning algorithm. K-SVD is shown to have numerous applications, including denoising [75], compression [74], and face recognition [222]. It is a generalization of the k -means clustering algorithm. The objective function of K-SVD is

$$\min_{\mathbf{D}, \mathbf{\Gamma}} \|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{subject to } \|\boldsymbol{\gamma}_i\|_0 \leq T_0 \quad \forall i. \quad (9.5)$$

Here, \mathbf{Y} is the training data matrix, \mathbf{D} is a dictionary, and $\mathbf{\Gamma}$ is the coefficient matrix of the data with respect to the dictionary \mathbf{D} , and $\boldsymbol{\gamma}_i$ denotes the i -th column of $\mathbf{\Gamma}$. In this sparsity-constrained formulation, each coefficient vector $\boldsymbol{\gamma}_i$ is constrained to have at most T_0 non-zero entries.

The K-SVD algorithm alternates between two steps. In the first stage of sparse coding, the dictionary \mathbf{D} is kept fixed and the coefficients $\boldsymbol{\gamma}_i$ are updated with the solution of l_0 norm constrained error minimization problem

$$\hat{\boldsymbol{\gamma}}_i = \min_{\boldsymbol{\gamma}_i} \|\mathbf{Y}_i - \mathbf{D}\boldsymbol{\gamma}_i\|_2^2 \quad \text{subject to } \|\boldsymbol{\gamma}_i\|_0 \leq T_0. \quad (9.6)$$

As the l_0 norm is NP-hard to deal with, approximate sparse coding algorithms like orthogonal matching pursuit (OMP) [223] or FOCUSS [224] can be used at this stage. The optimal solution is recovered with a high probability by OMP if the sparsity T_0 is low [225].

In the second stage, columns of the dictionary \mathbf{D} are updated successively. To update the i -th column, the representation error matrix without the i -th column is evaluated as follows:

$$\mathbf{E}_i = \mathbf{Y} - \sum_{j \neq i} \mathbf{d}_j \boldsymbol{\gamma}_j^T. \quad (9.7)$$

Here, $\boldsymbol{\gamma}_j^T$ denotes the j -th row of $\mathbf{\Gamma}$, and \mathbf{d}_j is the j -th column of the dictionary \mathbf{D} . To preserve sparsity constraint, a reduced error matrix \mathbf{E}_i^R is formed by keeping those columns of \mathbf{E}_i with indices from the set $\mathcal{H}_i = \{j \mid \Gamma_{ij} \neq 0\}$. These indices correspond to examples that use the i -th column in their current representation. With this, the problem reduces to finding a rank one approximation of the restricted error matrix \mathbf{E}_i^R , which can be conveniently found from the SVD: $\mathbf{E}_i^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$. The superscript R is used to denote matrices or vectors restricted according to the

set \mathcal{H}_i , whereas the superscript H denotes the conjugate transpose of the matrix. The dictionary column is updated by the first left singular vector $\mathbf{d}_i = \mathbf{u}_1$, and the corresponding coefficient row is updated with the first right singular vector scaled with the largest singular value $\gamma^i = \sigma_1 \mathbf{v}_1^H$.

The K-SVD can also be formulated as an error-constrained objective minimizing the sparsity as follows:

$$\min_{\mathbf{D}, \Gamma} \|\gamma_i\|_0 \quad \text{subject to } \|\mathbf{y}_i - \mathbf{D}\gamma_i\|_2 \leq \epsilon \forall i. \quad (9.8)$$

Fast implementation of K-SVD has been developed in [226] using batch-OMP. The K-SVD can also be used for complex dictionary learning as in [227].

K-SVD is susceptible to local minima, and its performance depends on initialization. Thus, as we will see in Sec. 9.4, we average the performance over many Monte-Carlo runs and also study the effect of random initialization as against knowledge-based initialization.

9.3 Goals and Experimental Setting

Now we use the dictionary learning methods in the context of period estimation and periodic signal representation using dictionaries. In particular, we seek answers to two questions.

1. Can K-SVD learn the periodicity dictionaries like Ramanujan and Farey from the data without explicit periodicity constraints? If so, under what conditions?
2. What are the advantages of K-SVD learned dictionaries over known periodicity dictionaries? Does it offer better representation in case of noisy data?

In the remainder of this section, we describe the evaluation metrics and experimental setting.

9.3.1 Evaluation Metrics

In order to compare the learned dictionaries under different settings, we use two evaluation metrics. The dictionary similarity metric $\rho(\mathbf{D}, \tilde{\mathbf{D}})$ measures how well the columns from the original dictionary \mathbf{D} are recovered by the learned dictionary $\tilde{\mathbf{D}}$. To compute this, for each column in the original dictionary, we find a column from the learned dictionary that has the maximum normalized inner product with

it. We then take an average of this maximum inner product over all columns of the dictionary. Thus,

$$\rho(\mathbf{D}, \tilde{\mathbf{D}}) = \frac{1}{M} \sum_{i=1}^M \max_k |\mathbf{d}_i^H \tilde{\mathbf{d}}_k|. \quad (9.9)$$

A value of $\rho(\mathbf{D}, \tilde{\mathbf{D}})$ close to 1 indicates that most of the columns from the original dictionary are close to some column of the reconstructed dictionary.

We also define the following two error metrics $e_1(\mathbf{X}, \mathbf{Y}, \mathbf{D})$ and $e_2(\mathbf{Y}, \mathbf{D})$ that capture how well the learned dictionary can represent the training data:

$$e_1(\mathbf{X}, \mathbf{Y}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^N \frac{\|\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{x}_i\|_2^2}{\|\mathbf{x}_i\|_2^2}, \quad (9.10)$$

$$e_2(\mathbf{Y}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^N \frac{\|\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{y}_i\|_2^2}{\|\mathbf{y}_i\|_2^2}. \quad (9.11)$$

Here, \mathbf{Y} is the noisy data matrix, \mathbf{X} is the noiseless data matrix, and \mathbf{D} is the learned dictionary. Note that in order to calculate these metrics, we first need to compute the coefficient vectors $\boldsymbol{\gamma}_i$ from the given noisy data \mathbf{Y} adhering to sparsity or error constraints. With the coefficients $\boldsymbol{\gamma}_i$ calculated, e_1 is the relative mean squared error (MSE) of the representation $\mathbf{D}\boldsymbol{\Gamma}$ with respect to the noiseless data \mathbf{X} . Note that even if \mathbf{Y} does not appear directly in Eq. (9.10), it is required to find the coefficients $\boldsymbol{\gamma}_i$. Since error e_1 captures how well the representation based on the learned dictionary is able to match the noiseless data, it can be regarded as the denoising error. Similarly, e_2 is relative MSE with respect to noisy data \mathbf{Y} . This indicates how well the dictionary can represent noisy data. Note that if the training data does not have any noise, then $\mathbf{X} = \mathbf{Y}$, and thus $e_1 = e_2$.

9.3.2 Experimental Setting

For all our experiments, we first create either a Ramanujan or a Farey dictionary that can represent component periods up to some integer P_{max} . We then generate T_0 -sparse data from the dictionary and learn a dictionary using K-SVD from this data. The standard (real-valued) K-SVD is used for the real data generated from the Ramanujan dictionary, whereas the complex-valued K-SVD is used for the complex data generated from the Farey dictionary.

With the following default values of parameters, we vary one or two parameters at a time and see how the performance varies. The maximum component period P_{max} is 15, and the signal length or the number of rows of the dictionary is 60. Thus, the

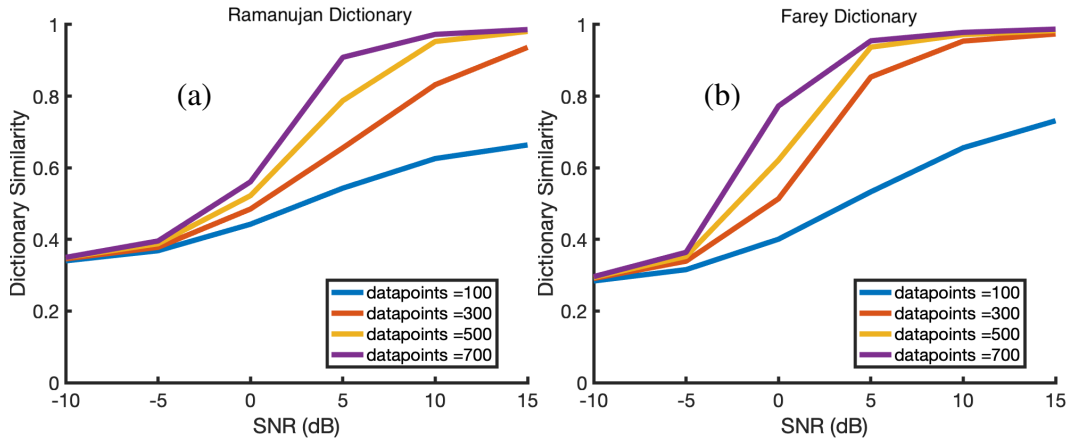


Figure 9.1: Variation of dictionary similarities with SNR and datapoints. (a) Ramanujan Dictionary, and (b) Farey dictionary.

number of dictionary columns is $\Phi(15) = 72$, so that the true underlying dictionary is overcomplete. We set the sparsity to 5, and 500 data points are used for dictionary learning. 100 Monte-Carlo simulations are run for each parameter set, each with 100 K-SVD iterations.

9.4 Experimental Results

Our experiments show that the K-SVD algorithm is indeed able to learn the well-known periodicity dictionaries from data. The accuracy with which this learning takes place (i.e., dictionary similarity and representation errors) depends on a number of parameters in the experiments, as explained through the simulations next.

Fig. 9.1(a) shows the variation of dictionary similarity with SNR of the training data and the number of datapoints for the Ramanujan dictionary. We see that the K-SVD is able to recover the underlying dictionary when SNR is high, and more data helps recover the dictionary columns even at relatively lower SNR values. Fig. 9.1(b) shows similar trends observed for the Farey dictionary. Thus, we see that the complex K-SVD can also learn dictionaries with similar performance.

Next, Fig. 9.2 shows representation errors e_1 and e_2 for the case of learning the Ramanujan dictionary. Note that both representation errors reduce with higher SNR and more data. However, it is interesting to note that for very few data points, e_2 in fact does not reduce with increasing SNR. Exactly similar trends were also observed for complex K-SVD, hence we have skipped those graphs here.

To see the effect of sparsity, we perform two kinds of experiments. In one experiment, all the training datapoints have fixed sparsity, and in another experiment, different

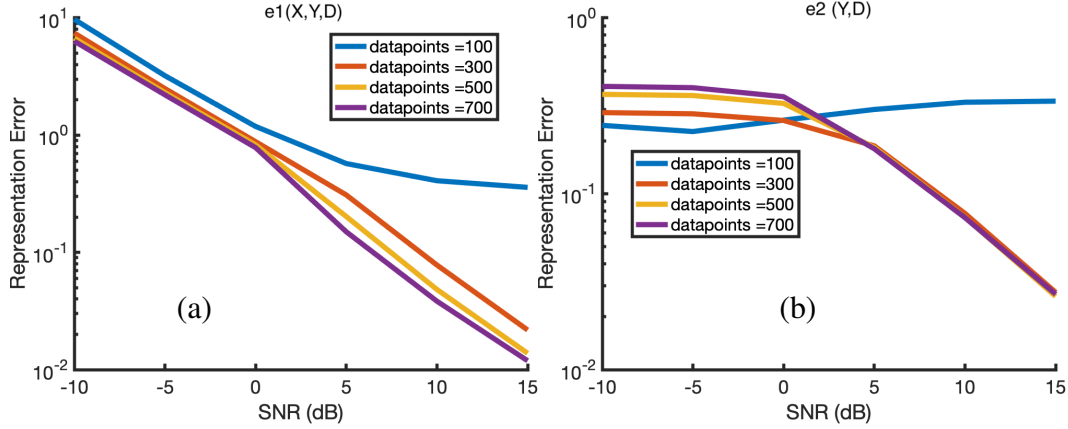


Figure 9.2: Variation of representation errors with SNR and datapoints for Ramanujan dictionary. (a) Error e_1 , and (b) error e_2 .

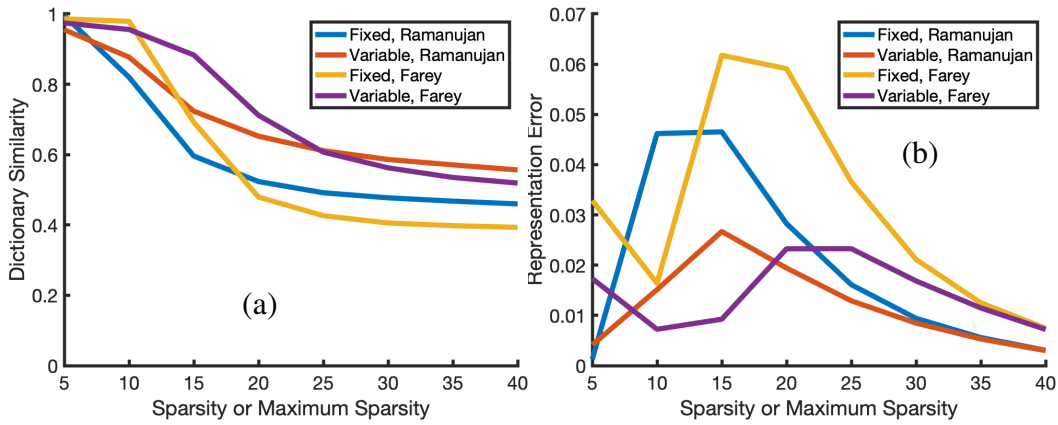


Figure 9.3: Variation of (a) dictionary similarity and (b) representation error with sparsity and maximum sparsity.

datapoints can have different sparsity, which we call *variable sparse* data. For example, 20 variable sparse data means that the sparsity of different datapoints can be anything between 1 and 20. Based on Fig. 9.3, for both real and complex K-SVD, we see that the K-SVD learned dictionary has a lower representation error when sparsity is very low or high. On the other hand, the dictionary similarity decreases monotonically as sparsity increases. This suggests that even when we have low similarity to the underlying dictionary, the representation error can be small. Also, note that K-SVD is able to learn a good dictionary even when sparsity is not fixed. In fact, the representation error and dictionary similarity improve in this case. Intuitively, this can be attributed to the fraction of the data that has lesser sparsity, as it reveals more information about the columns of the underlying dictionary.

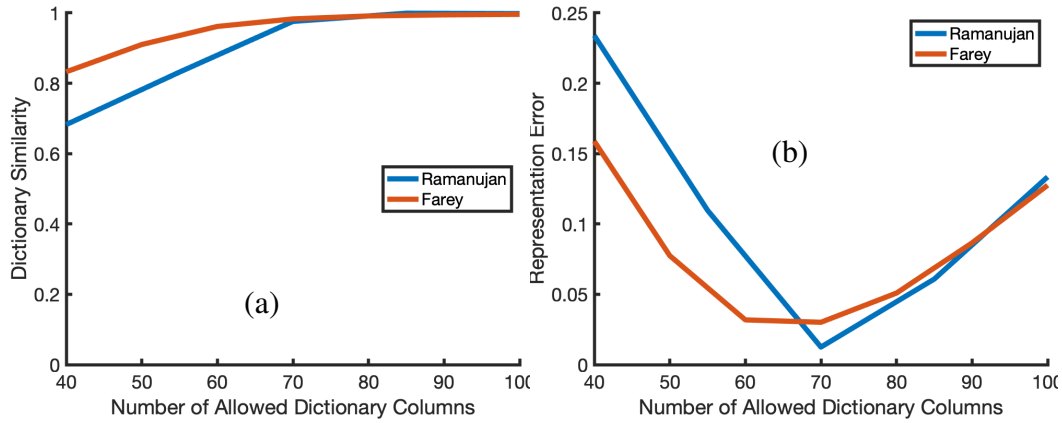


Figure 9.4: Variation of (a) dictionary similarity and (b) representation error with the allowed number of dictionary columns.

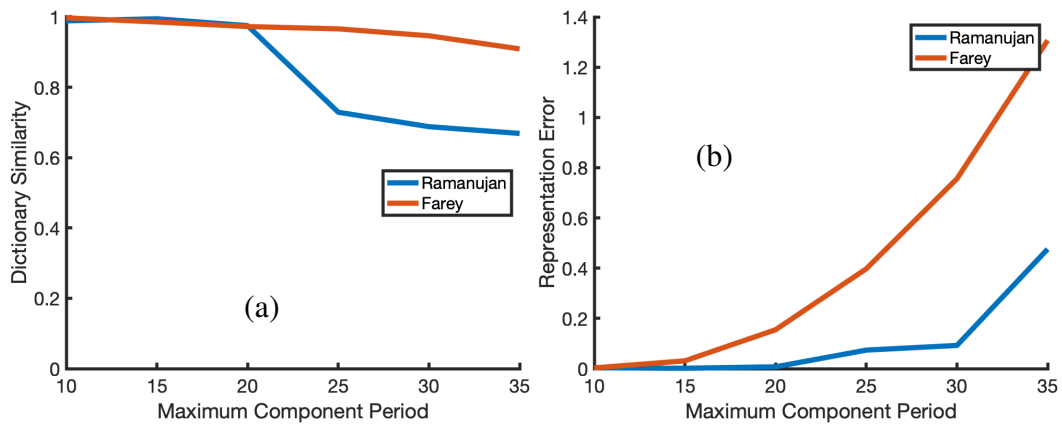


Figure 9.5: Variation of (a) dictionary similarity and (b) representation error with the maximum period.

Fig. 9.4 shows the variation with the number of columns the learned dictionary is allowed to have. Note that when the allowed number of columns exceeds 72, we recover all atoms of the underlying dictionary. Also interesting is that if more than 72 columns are allowed, the representation error does not reduce, but in fact increases.

From Fig. 9.5 we note that with the length fixed, a better dictionary is learned when the data has a lower component period. This means that there needs to be at least a certain number of periods of the periodic components included in the data length. Also, note that the complex-valued K-SVD is much more robust to larger component periods than the real-valued K-SVD. The similarity is relatively higher even for larger periods.

Fig. 9.6 demonstrates results of a similar nature as the signal length is varied. Here,

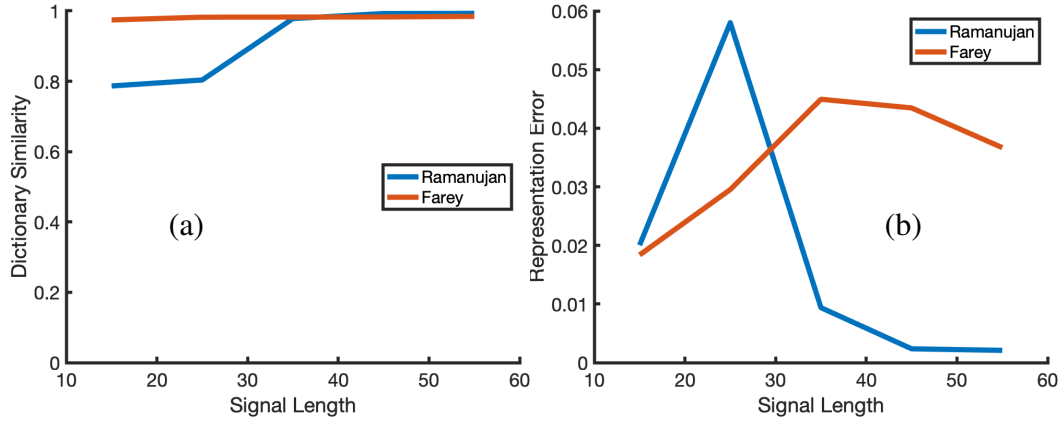


Figure 9.6: Variation of (a) dictionary similarity and (b) representation error with signal length.

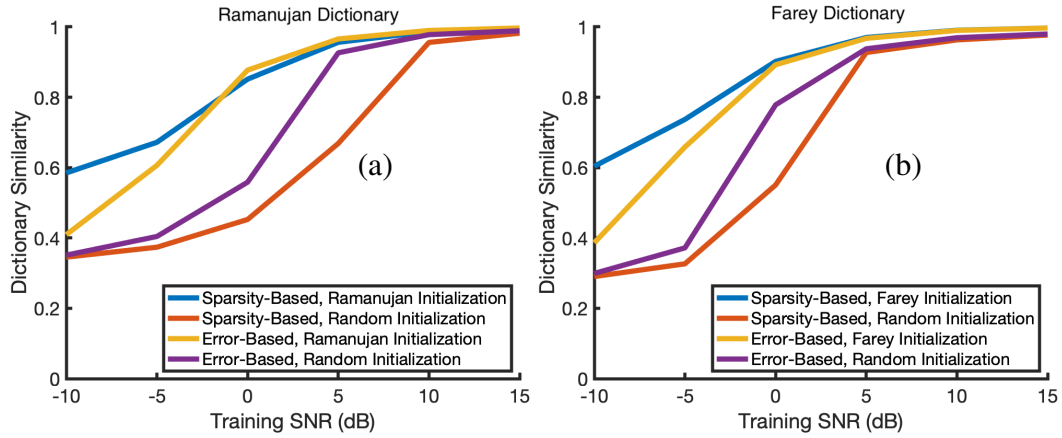


Figure 9.7: Variation of dictionary similarities for different K-SVD versions and initialization. (a) Ramanujan dictionary, and (b) Farey dictionary.

with the maximum period fixed, a longer signal length is preferred, which again emphasizes that there needs to be at least a certain number of periods of the periodic components included in the data length. Here too, the complex K-SVD is more robust to smaller signal lengths.

Fig. 9.7 compares two different versions of K-SVD, namely, sparsity-constrained formulation (Eq. (9.5)) and error-constrained formulation (Eq. (9.8)). For both versions, we consider two initializations: random initialization and Ramanujan initialization. At high SNR, both versions and random as well as knowledge-based initialization of K-SVD perform equally well. Also, notice that for random initialization, error based optimization gives marginally better dictionary similarity than sparsity-based optimization.

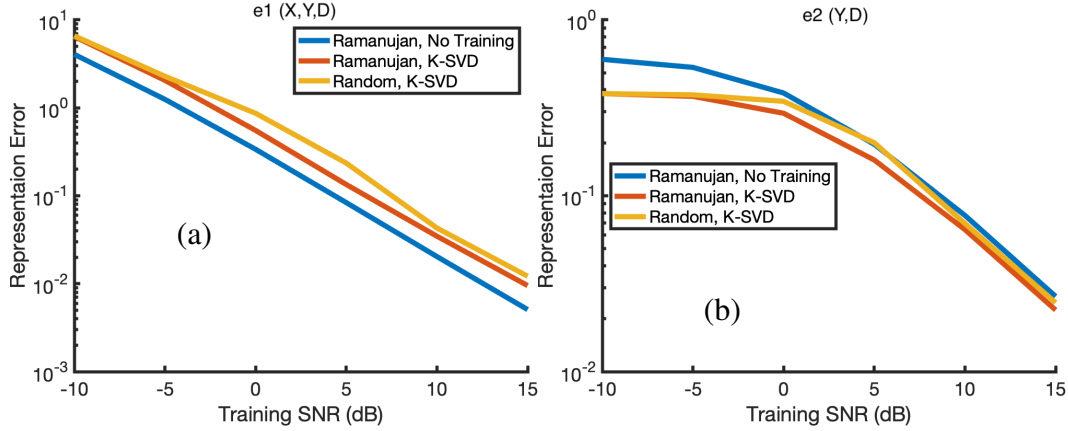


Figure 9.8: Effect of initialization on representation error e_2 for Ramanujan dictionary.

Now in pursuit of seeking whether there is any advantage of learned dictionaries over the known analytic dictionaries, we study the variations of representation error for known and trained dictionaries, and with different initializations. From Fig. 9.8(a), we see that for denoising type error e_1 , the Ramanujan dictionary without any further optimization gives better performance than the learned dictionaries. However, with regards to the representation error e_2 (Fig. 9.8(b)), the learned dictionaries adapt better to the noisy data. Here too, all the trends were similar for the Farey dictionary as well, and hence we have skipped those graphs.

Within the learned dictionaries with two different initializations, we see that the Ramanujan initialization has both errors smaller than the random initialization. Thus, the knowledge-based initialization is better than the random initialization for representing noisy data through a learned dictionary.

9.5 Review of CNN-Based Method from [149]

In the remainder of the chapter, we focus our attention on the recent CNN-based method [149] proposed for DOA estimation. Here, our primary contribution is demonstrating how to adapt such machine learning methods for sparse arrays. This is done by proposing a modification of the input proxy spectrum used as input to the recent CNN-based method for sparse arrays. The second contribution is proposing alternative proxy spectra that may be used as input to the CNN.

The method in [149] first generates a proxy spectrum based on a dictionary formulation of the DOA estimation problem, as explained below. The proxy spectrum is then refined further by a four-layer convolutional neural network. Assume that the DOAs

are impinging from a spatial scope of $\Theta = [-60^\circ, 60^\circ]$. Consider the discretized set of directions $\phi_1, \phi_2, \dots, \phi_L$ sampled with 1° angular separation within the spatial scope Θ . With this, the array output signal (Eq. (2.1)) can also be expressed as:

$$\mathbf{x}[k] = \sum_{l=1}^L \mathbf{a}(\phi_l) \bar{s}_l[k] + \bar{\mathbf{n}}[k], \quad (9.12)$$

$$\bar{s}_l[k] = \mathbf{s}_m[k] \text{ if } \phi_l \approx \theta_m, \text{ zero otherwise.} \quad (9.13)$$

Here, the steering vectors $\mathbf{a}(\cdot)$ are defined in terms of the physical direction angles $\phi_1, \phi_2, \dots, \phi_L$ instead of their ω -domain equivalents as in Eq. (2.1). $\bar{\mathbf{n}}[k]$ now consists of error terms due to angle discretization in addition to the noise. Let the manifold matrix corresponding to the discretized set of directions be

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{a}(\phi_1) & \mathbf{a}(\phi_2) & \dots & \mathbf{a}(\phi_L) \end{bmatrix} \in \mathbb{C}^{N \times L}. \quad (9.14)$$

Now, the vectorized form of the covariance matrix can be expressed as:

$$\hat{\mathbf{y}} = \text{vec}(\hat{\mathbf{R}}_{\mathbf{xx}}) = \tilde{\mathbf{A}}\eta + \epsilon. \quad (9.15)$$

Here, $\eta \in \mathbb{R}^L$ is the spatial spectrum vector of the signal with non-zero entries at locations corresponding to the DOAs $\theta_1, \dots, \theta_D$ and ϵ is the error term. The matrix $\tilde{\mathbf{A}}$ is defined as

$$\tilde{\mathbf{A}} = \bar{\mathbf{A}}^* \odot \bar{\mathbf{A}}, \quad (9.16)$$

where \odot denotes the Khatir-Rao (KR) product (column-wise Kronecker product). Eq. (9.15) is the motivation for the proxy used as input to the network in [149]:

$$\hat{\eta} = \tilde{\mathbf{A}}^H \text{vec}(\hat{\mathbf{R}}_{\mathbf{xx}}). \quad (9.17)$$

Since $\eta \in \mathbb{R}^L$, $\hat{\eta}$ typically has a very small imaginary component under moderate noise conditions.

Next, the real and imaginary parts of this proxy spectrum are provided as input to a four-layer convolutional network. The network structure used in [149] is given in Fig. 9.9. The network is trained with ideal spectrum η as labels and hence learns to refine the input proxy spectrum $\hat{\eta}$ to produce an output spectrum that is close to the true spectrum η .

Notice that the method combines model-based and data-driven approaches. The data model is incorporated through the input proxy spectrum $\hat{\eta}$, which is derived based on Eq. (9.15). The method also incorporates the data-driven approach as the proxy spectrum is refined by the CNN, which is trained based on the data.

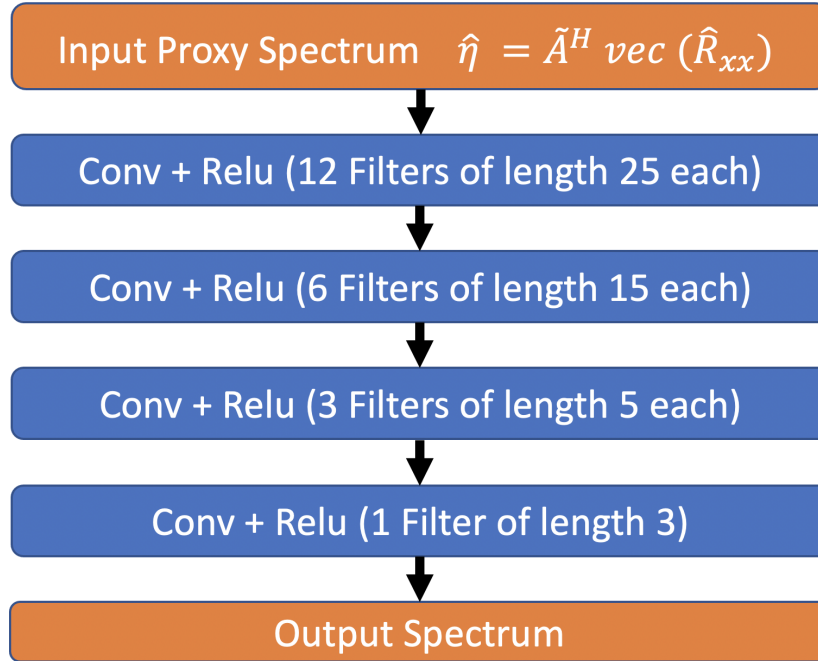


Figure 9.9: The CNN network structure used in [149]. The output size of each layer is maintained to be L by appropriate zero padding before performing convolutions, where L is the number of discretized directions considered (see Eq. (9.12)).

9.6 Proposed Modifications for Sparse Arrays

In this section, we adapt the above CNN-based method for sparse arrays. We have seen in Sec. 2.3 that sparse arrays such as nested arrays, coprime arrays, and MRAs [33] can identify $O(N^2)$ DOAs using N sensors, and the key is estimating correlation values corresponding to the different lags determined by the difference coarray.

Here, we consider a nested array with stx sensors:

$$\mathbf{z} = [1 \quad 2 \quad 3 \quad 4 \quad 8 \quad 12]. \quad (9.18)$$

The difference coarray of this array is a uniform linear array (ULA) with 23 locations from -11 to 11, and the correlations $\hat{R}(l)$ can be estimated for all lags from $l = -11$ to $l = 11$ as per Eq. (2.22). Let $\bar{\mathbf{r}}$ denote the 23-dimensional vector consisting of these $\hat{R}(l)$ values. It can be shown that

$$\bar{\mathbf{r}} = \mathbf{B}\boldsymbol{\eta} + \mathbf{e}, \quad (9.19)$$

where \mathbf{B} is the coarray manifold matrix

$$\mathbf{B} = [\mathbf{b}(\phi_1) \quad \mathbf{b}(\phi_2) \quad \dots \quad \mathbf{b}(\phi_L)], \quad (9.20)$$

with $\mathbf{b}(\phi)$ as the coarray steering vector

$$\mathbf{b}(\phi) = \begin{bmatrix} e^{-11j\pi\sin\phi} & e^{-10j\pi\sin\phi} & \dots & e^{11j\pi\sin\phi} \end{bmatrix}^T, \quad (9.21)$$

and \mathbf{e} is the error term, which is typically small.

To adapt the CNN method for sparse arrays, we form the spatially smoothed matrix $\widehat{\mathbf{R}}_{SS}$ according to Eq. (2.31). The manifold matrix corresponding to the nonnegative part of the ULA coarray is

$$\bar{\mathbf{A}}_c = \begin{bmatrix} \mathbf{a}_c(\phi_1) & \mathbf{a}_c(\phi_2) & \dots & \mathbf{a}_c(\phi_L) \end{bmatrix}, \quad (9.22)$$

where

$$\mathbf{a}_c(\theta) = \begin{bmatrix} 1 & e^{j\pi\sin\theta} & \dots & e^{11j\pi\sin\theta} \end{bmatrix}^T \quad (9.23)$$

is the corresponding coarray steering vector. To adapt the CNN method for sparse arrays, we use $\widehat{\mathbf{R}}_{SS}$ in place of $\widehat{\mathbf{R}}_{xx}$, and $\bar{\mathbf{A}}_c$ in place of $\bar{\mathbf{A}}$. Note that this adaptation can also be used for all machine learning based methods that use the covariance matrix $\widehat{\mathbf{R}}_{xx}$ as the input to the network. This coarray adaptation augments the machine learning methods with the capability to identify more sources than the number of sensors.

In particular, for the method under consideration, we use

$$\tilde{\mathbf{A}}_c = \bar{\mathbf{A}}_c^* \odot \bar{\mathbf{A}}_c \quad (9.24)$$

in place of $\tilde{\mathbf{A}}$ in Eq. (9.14). This gives a possible proxy spectrum (called Hermitian proxy in the next subsection) that is derived based on the coarray:

$$\widehat{\eta} = \tilde{\mathbf{A}}_c^H \text{vec}(\widehat{\mathbf{R}}_{SS}). \quad (9.25)$$

In the next subsection, we also propose several other proxy spectrum alternatives.

9.7 Proposed Proxy Spectra

In addition to the above adaptation for sparse arrays, we also propose several other ways to construct the proxy spectrum at the input of the CNN. We enumerate them here, providing reasons for choosing these proxies.

1. **Original proxy** (from [149]): See Eq. (9.15).

$$\widehat{\eta} = \tilde{\mathbf{A}}^H \text{vec}(\widehat{\mathbf{R}}_{xx}) \quad (9.26)$$

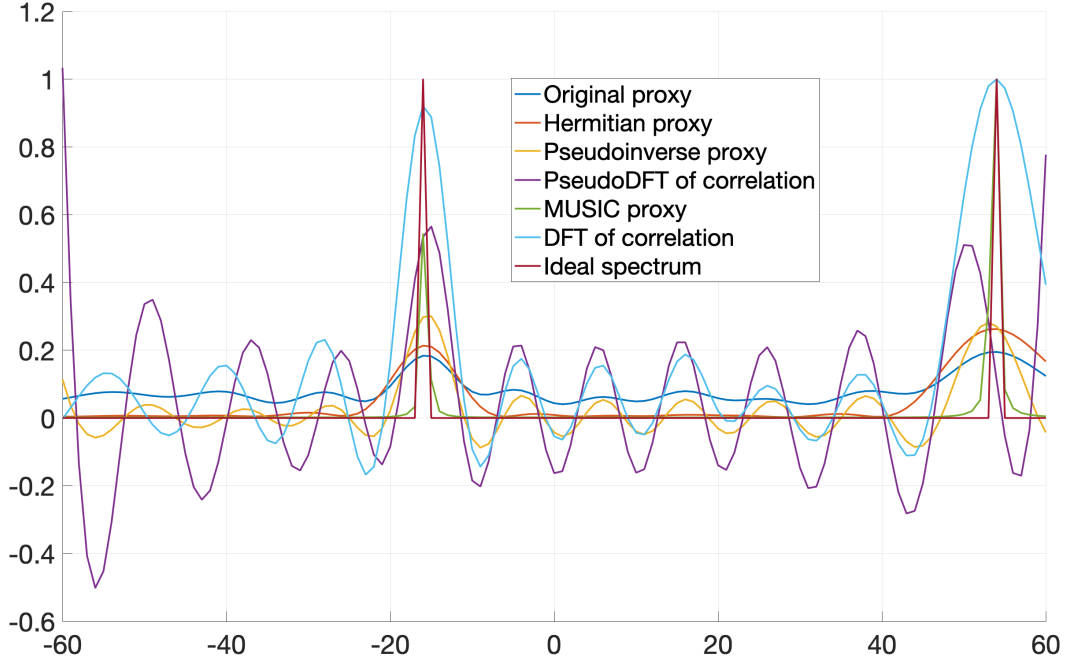


Figure 9.10: Comparison of different proxy spectra for a two-source scenario from training data. The original proxy has a non-zero ‘floor’, whereas pseudoinverse, DFT, and pseudo-DFT proxies have ringing patterns.

2. **Hermitian proxy:** This is the adaptation of the original proxy for the case of sparse arrays (see Eq. (9.25)).

$$\hat{\eta} = \tilde{\mathbf{A}}_c^H \text{vec}(\hat{\mathbf{R}}_{SS}) \quad (9.27)$$

We call it Hermitian proxy as it premultiplies the vectorized $\hat{\mathbf{R}}_{SS}$ with the Hermitian of the matrix $\tilde{\mathbf{A}}_c$.

3. **Pseudoinverse proxy:** Here, the idea is to use pseudoinverse $\tilde{\mathbf{A}}_c^\dagger$ instead of $\tilde{\mathbf{A}}_c^H$ used in the Hermitian proxy. This is expected to lead to a closer approximation of the ideal spectrum in Euclidean distance when noise is small (See Eq. (9.15)).

$$\hat{\eta} = \tilde{\mathbf{A}}_c^\dagger \text{vec}(\hat{\mathbf{R}}_{SS}) \quad (9.28)$$

4. **DFT of correlation:** Here we take L samples of the DFT of $\bar{\mathbf{r}}$ at non-uniform frequency locations $\pi \sin \phi_l$, $1 \leq l \leq L$. In view of (9.20) and (9.21), this is mathematically equivalent to

$$\hat{\eta} = \mathbf{B}^H \bar{\mathbf{r}}, \quad (9.29)$$

where \mathbf{B} is as defined in (9.20).

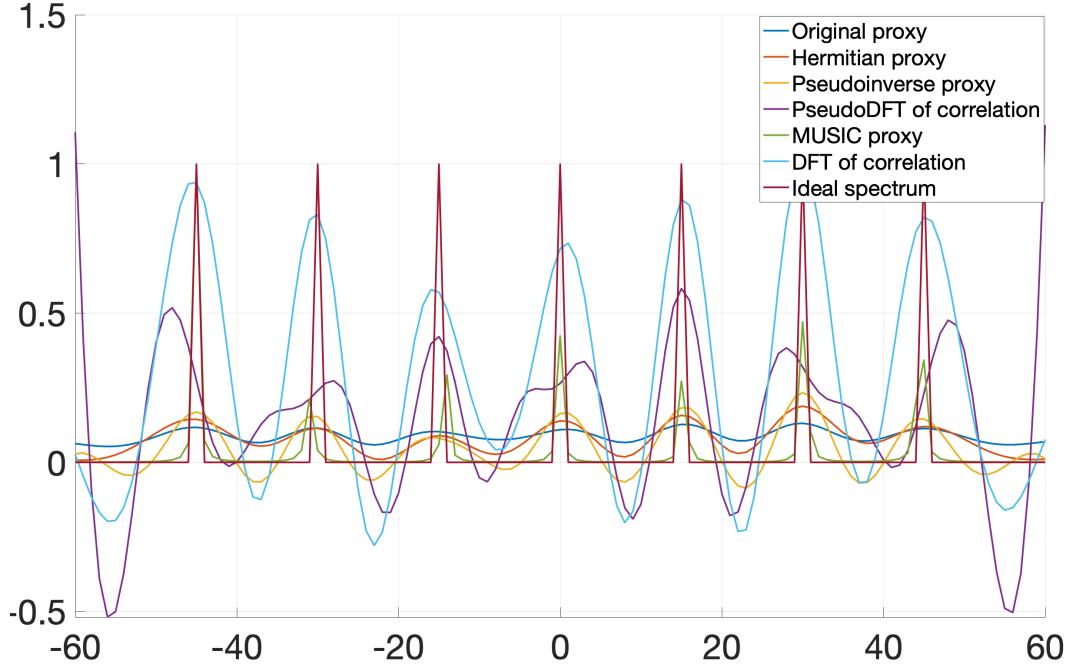


Figure 9.11: Comparison of different proxy spectra for a seven-source scenario used for testing (where the number of sensors is only 6). The peaks in original proxy (from [149]) are much harder to identify in this case.

5. **‘Pseudo-DFT’ of correlation:** Here we use the pseudoinverse \mathbf{B}^\dagger in place of \mathbf{B}^H used in the DFT of correlation proxy. The motivation is similar to the pseudoinverse proxy above.

$$\hat{\eta} = \mathbf{B}^\dagger \bar{\mathbf{r}} \quad (9.30)$$

6. **Normalized MUSIC proxy:** Here, input to the network is a normalized MUSIC spectrum. The idea is to explore if a collaborative approach between MUSIC and neural network based methods leads to improved results.

In Fig. 9.10, we qualitatively compare the different proxies obtained for a two-source scenario from the training data. The real parts of different proxies are plotted against the discretized directions ϕ_1, \dots, ϕ_L on the X -axis. We also plot the ideal spectrum for comparison, which takes the value unity at DOA locations and is zero everywhere else. Notice that the original proxy (9.26) has a nonzero ‘floor’ for all angles. Also, the peaks in the original proxy spectrum are not as sharp as the other proxies. The Hermitian proxy (9.27) adapted for sparse arrays has its ‘floor’ much closer to zero. The Pseudoinverse proxy (9.28), non-uniform DFT of correlation (9.29), and pseudo-DFT of correlation (9.30) are observed to have oscillating behavior in the

portions where the ideal spectrum is zero. These proxies also take negative values in their oscillating pattern. The MUSIC proxy is closest to the ideal spectrum, as expected.

Fig. 9.11 demonstrates the different proxies obtained for a seven-source scenario, which is considered later for testing. The number of sources is greater than the number of sensors in this case. Note that the original proxy (9.26) has peaks that are practically very hard to distinguish because of the non-zero ‘floor’. This deteriorates the performance of the original proxy when more sources are present, as we will see in the simulations next.

9.8 Simulation Results

Now we compare the performance of the original and proposed input proxy spectra through simulation examples. We compare their training and test losses, ability to resolve more sources than sensors, and DOA estimation errors.

9.8.1 Training Data and Methodology

With the above different proposed input proxy spectra, separate CNNs with the same 4-layer structure shown in Fig. 9.9 are trained. We follow a similar procedure to that in [149]. $K = 256$ snapshots are used. Signals in the generated dataset set have $D = 2$ on-grid sources with integer angular separations between 1° and 40° . All such possible direction settings within the spatial scope $\Theta = [-60^\circ, 60^\circ]$ are considered. For each direction setting, we generate 10 random signals with randomly chosen SNR in the range 0 to 10 dB. The labels consist of the ideal spectrum vector, with a value of 1 at the locations corresponding to signal directions and 0 otherwise. All networks are trained on the same training data for 150 epochs with mean squared error loss with Adam optimizer [77]. 80% of the data is used for training while the remaining 20% is used for testing. Although the training data here has only 2 sources, we consider more sources later for testing.

9.8.2 Training and Test Losses

Fig. 9.12 and Fig. 9.13 show training and test losses, respectively. It is observed that all the proposed proxy spectra, except the Hermitian proxy (9.27), provide an improvement over the original method [149] in terms of the training and test error. It is observed that the pseudoinverse proxy spectrum (9.28) works better than the Hermitian proxy (9.27). Similarly, pseudo-DFT of correlation (9.30) works better than its DFT (9.29). Note that although the MUSIC proxy is closest to the

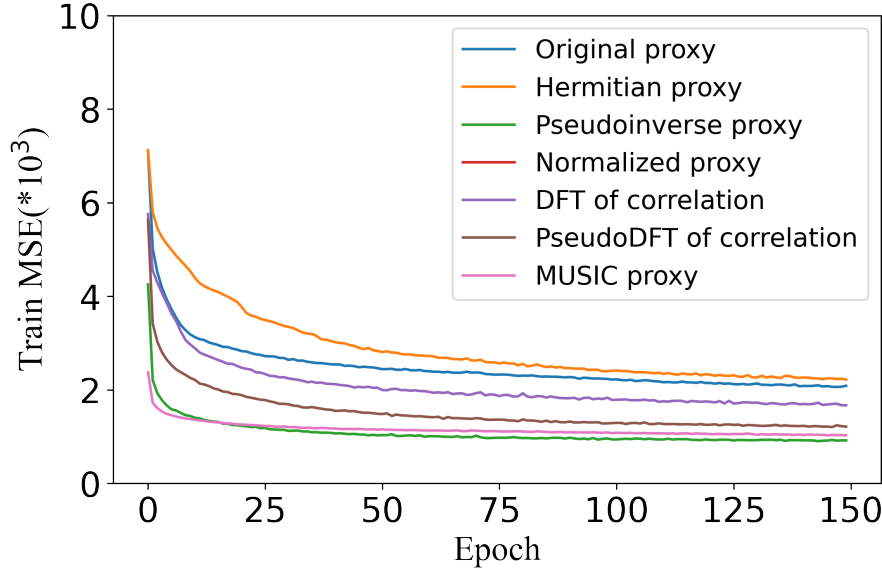


Figure 9.12: Training mean squared losses for different input proxy spectra

ideal spectrum, the pseudoinverse proxy provides marginally better training and test errors.

9.8.3 Estimating More Number of Sources

To demonstrate the ability of the sparse array to detect more sources than sensors, we next evaluate the models for the case when seven sources ($\geq N = 6$) with 5 dB SNR are present. Note that a ULA with six sensors could have detected only five sources based on second-order statistics. Although the seven DOAs considered in the example are well separated, they do not lie on the integer angle grid $[\phi_1, \dots, \phi_L]$. Also, notice that the separation between the two extreme sources is larger than 40° , which was the maximum angular separation between sources in the training examples. Thus, this testing scenario is challenging for the models in many aspects. Fig. 9.14 shows the estimated spectra by CNNs trained with different input proxies.

We see from the figure that the original proxy (9.26) and the Hermitian proxy (9.27) are not able to identify all seven sources. All the other proposed proxies provide seven clear peaks in the output spectrum that are close to the true DOA angles (denoted with red diamonds). This shows that the nested arrays can be used in conjunction with neural network models for estimating more DOAs than the number of sensors.

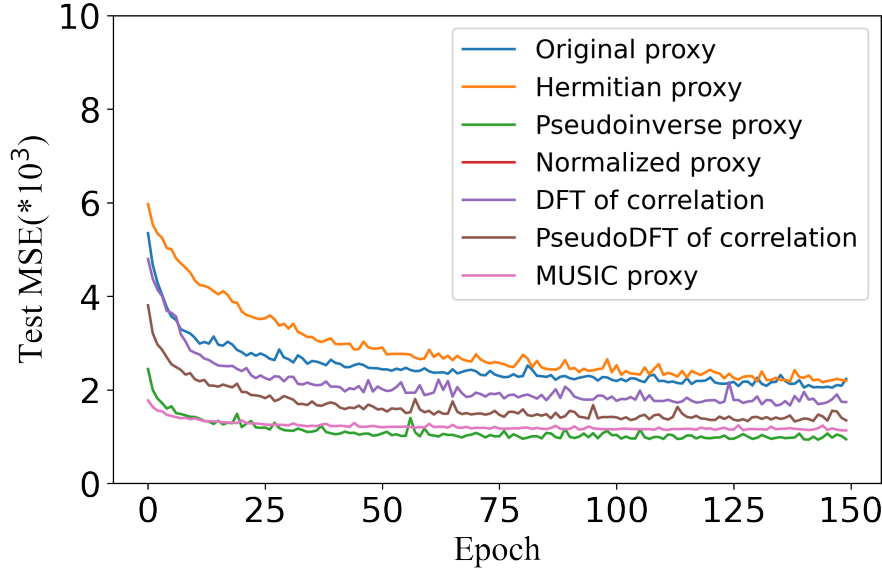


Figure 9.13: Test mean squared losses for different input proxy spectra

9.8.4 Mean Squared Error Analysis

Next, we evaluate the mean squared errors in the DOA estimate obtained by different models in two-signal scenarios. In each of the plots in Fig. 9.15, there are six different segments. The two sources present in those segments are separated by approximately 5.5° , 13.5° , 20.67° , 50° , 60° , and 70° . The sources do not necessarily lie on the discretized angular grid. In each segment, the source locations are changed so as to span the spatial scope from -60° to 60° , keeping the separation between the two sources approximately the same. The solid orange and green lines show how the true DOA values are varied, and the dots represent the estimated DOAs by different models. We see that all the models are able to track the DOAs well. Fig. 9.16 shows the plots of DOA estimation errors obtained from the above plots. It can be seen that most estimation errors are between -1° and 1° . However, there are also some large outliers in certain cases, which are not seen here because of the restricted Y-axis range.

Table 9.1 summarizes average absolute errors and mean squared errors from the previous plot. We also compare the errors with the sparse Bayesian learning (SBL) based DOA estimation method [228]. Implementation for SBL provided with [149] was used. In terms of average absolute error, the pseudoinverse proxy has the best performance, followed by the pseudo-DFT proxy. To calculate the mean squared error for DOA estimates, we have excluded the outlier examples whose estimates

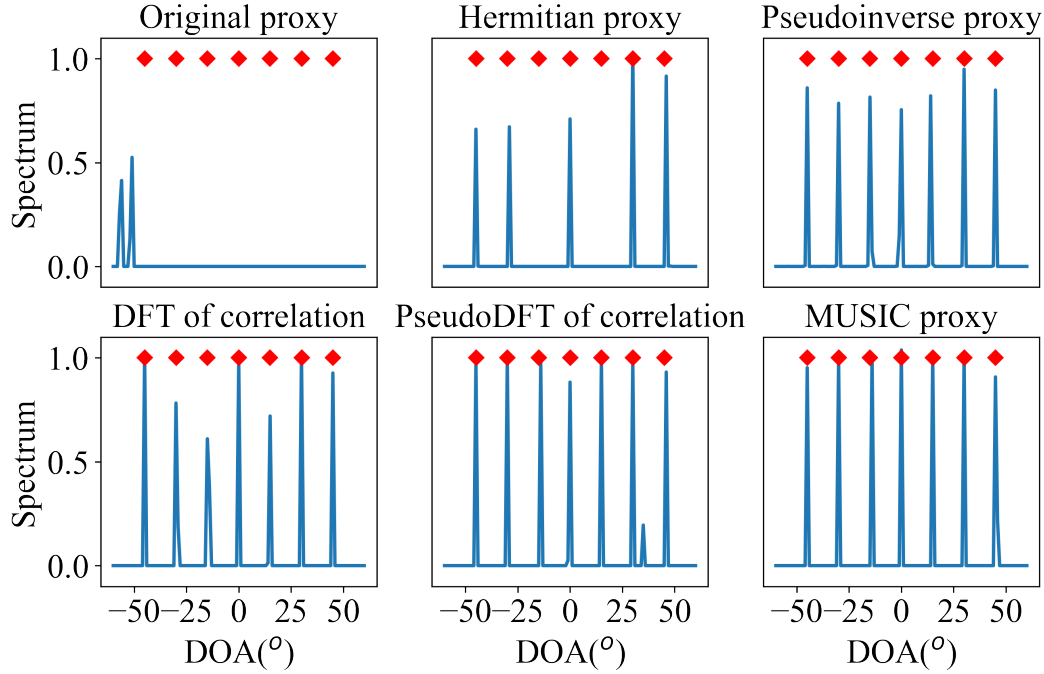


Figure 9.14: Estimated spectra for a seven-source scenario, with a 6-sensor nested array. True DOAs are denoted in red diamonds. Original proxy (9.26) and Hermitian proxy (9.27) do not identify all the sources, whereas all other proposed proxies correctly identify the sources.

Table 9.1: Average absolute errors and mean squared errors obtained with different input proxies

Proxy Spectrum	Avg. absolute error	MSE
Original proxy	0.2501	0.1486
Hermitian proxy	0.2517	0.1356
Pseudoinverse proxy	0.2201	0.0868
DFT proxy	0.2395	0.1151
Pseudo-DFT proxy	0.2219	0.0909
MUSIC proxy	0.2421	0.0938
SBL	0.2988	0.0724

are more than 10 degrees off from the true value. We compute the MSE for the remaining examples only. With this metric, we see that the SBL method provides slightly better performance, followed closely by pseudoinverse and pseudo-DFT proxy.

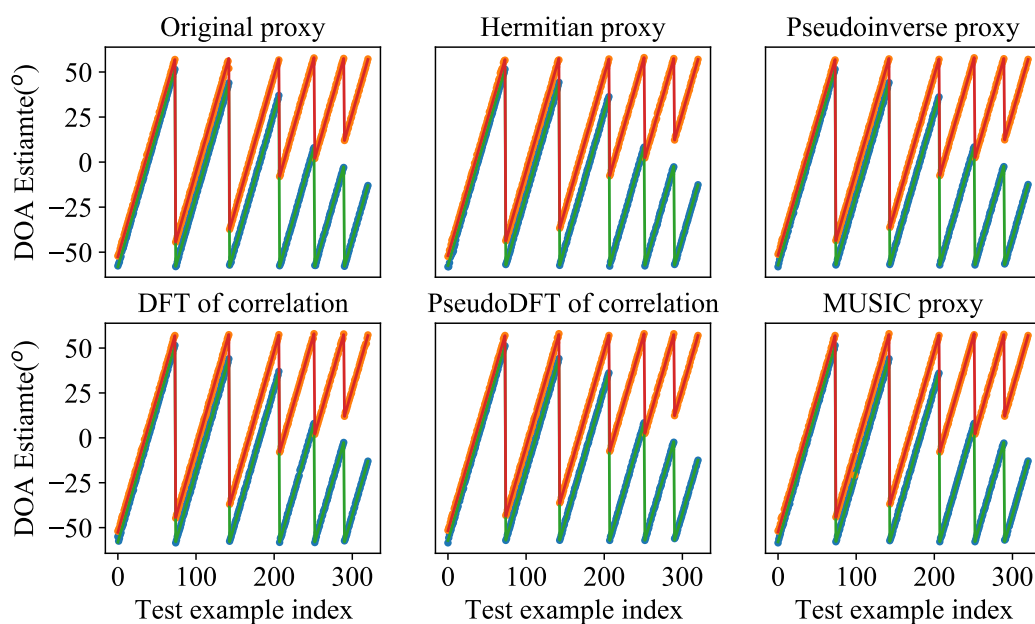


Figure 9.15: Estimated DOAs (shown in dots) with all proxies track the true DOAs (solid lines) well. The two sources present in the six segments are separated by approximately 5.5° , 13.5° , 20.67° , 50° , 60° , and 70° .

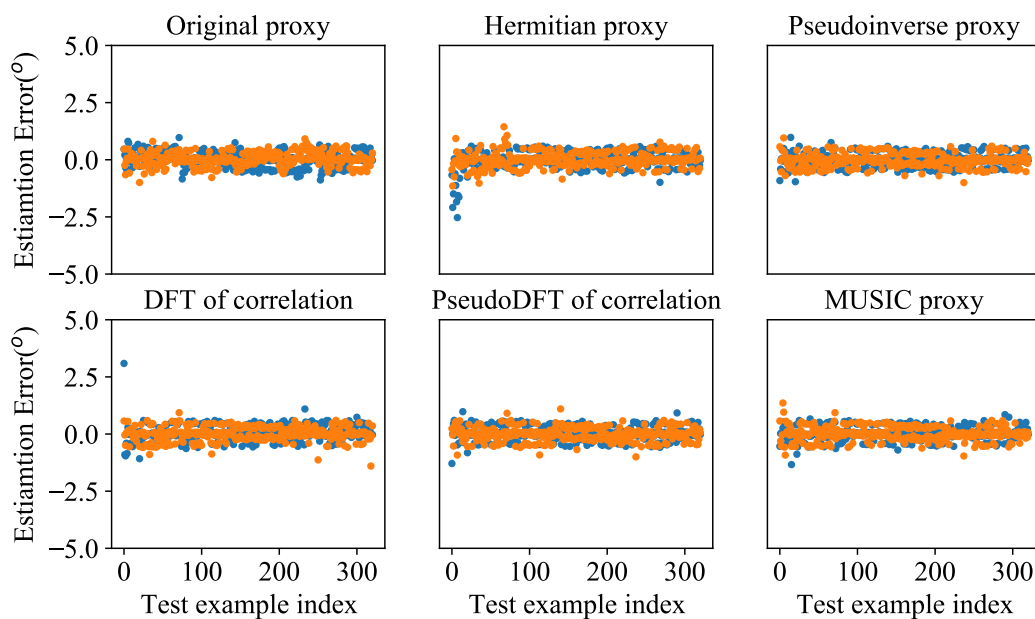


Figure 9.16: DOA estimation errors with different proxies obtained from the above plot. Some large outliers are not seen due to the restricted Y-axis range.

9.9 Concluding Remarks

In the first part of this chapter, we explored the role of a popular dictionary learning algorithm, K-SVD, in the context of period estimation and periodic signal representation. The experiments demonstrated that K-SVD has the ability to learn dictionaries for specialized applications without explicit application-specific constraints. We did not explicitly constrain the dictionary columns to be periodic with different periods, and still, the dictionary could be recovered under favorable circumstances. In some situations, it was also seen that the learned dictionary not being close to the ground truth may not imply poor representation and vice versa. The learning algorithm may find an equally good and sparse representation of the data through some other dictionary.

In the second part of this chapter, we demonstrated how an existing CNN-based DOA estimation method can be adapted to sparse arrays. This augments the machine learning methods with the capability of identifying more sources than the number of sensors present in the array. We further showed that simple modifications of the input proxy spectra lead to improved DOA performance. In the future, it would be of interest to study the performance of different proxy spectra for sparse arrays in the presence of array imperfections and non-idealities. Larger networks can be trained with large labeled data and high compute power. This will be required to obtain DOA estimates on a finer grid, for example, 0.5° or less. 2D extension of machine learning-based methods for sparse arrays can also be investigated in the future.

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we studied two applications involving line spectrum signals: direction of arrival (DOA) estimation and periodic signal denoising. In Chapter 3 we developed rational arrays for DOA estimation that are suitable under aperture constraint. In Chapter 4, we proposed new array design criteria and developed weight-constrained arrays that can handle strong mutual coupling. In Chapter 5, we proposed improved algorithms for DOA estimation using weight-constrained arrays, and presented a generalized construction of weight-constrained sparse arrays. In Chapter 6, we investigated the efficiency coarray-MUSIC and a Toeplitz covariance matrix construction approach for DOA estimation. Chapter 7 focused on developing a denoising framework for periodicity-aware signal denoising using Capon-optimized Ramanujan filter banks and pruned Ramanujan dictionaries. Chapter 8 developed theoretical results regarding Ramanujan filters. Chapter 9 explored the use of dictionary learning and machine learning for periodic signal representation and DOA estimation.

In addition to the topics covered in this thesis, we believe there are many interesting directions to explore in the future to expand the scope of this research:

1. *Beamforming under aperture constraints and mutual coupling:* In this thesis, we focused on developing new arrays for DOA estimation. However, the challenges related to aperture constraints and high mutual coupling have a similarly detrimental effect on beamforming with sensor arrays. Therefore, it is important to explore the use of the proposed rational arrays and weight-constrained arrays for beamforming applications. One of the known drawbacks of sparse arrays in beamforming is that the sidelobes can be irregular and may even have a higher gain than the main lobe. Consequently, the development of new sparse and rational arrays specifically tailored for beamforming applications will be required. Specialized beamformers, such as the Capon beamformer, optimize filter taps to suppress interference or achieve a desired beampattern. The application of rational arrays and weight-constrained arrays should also be investigated for these purposes.

2. *Two-dimensional arrays under aperture constraint and mutual coupling:* Although this thesis focused exclusively on linear arrays for identifying the DOAs of sources situated in a plane, in practical scenarios, sources are not usually confined to a single plane in space, and their directions are defined by two angles: azimuth (ϕ) and elevation (θ). Linear arrays are insufficient for determining these two parameters; hence, two-dimensional (2D) arrays such as uniform rectangular arrays (URA), uniform circular arrays (UCA), and L-shaped arrays are usually used for this [229]. There is potential to generalize the proposed rational arrays to 2D configurations. This would require extending the rational number-theoretic results to rational matrices. Previous research [230], [231] has laid the theoretical groundwork necessary for adapting integer coprime arrays to two dimensions. Similar developments for rational coprime arrays will be an area of future interest. Additionally, similar to the weight-constrained arrays introduced in this thesis to mitigate the effects of mutual coupling in one-dimensional arrays, the design of two-dimensional sparse arrays with strategically placed ‘central holes’ in the 2D difference coarray warrants further investigation.
3. *Addressing practical challenges of real-time deployment and correlated sources:* Several applications involving sensor arrays, including the increasingly popular field of autonomous driving [232], necessitate real-time DOA estimation and target localization. Such applications cannot depend on collecting a large number of snapshots to construct the sample covariance matrix of the array output. In a rapidly changing environment, the DOAs may not even remain fixed over a long time required to collect adequate snapshots. Moreover, computationally intensive DOA estimation algorithms, such as optimization-based methods, may not be suitable for real-time identification of DOAs and tracking targets. There has been research on DOA estimation using a single snapshot [233]–[235], and these techniques should be further investigated in relation to the rational arrays and weight-constrained arrays proposed in this thesis. Another significant challenge arises when the source amplitudes are correlated, a scenario commonly encountered in environments where multipath propagation occurs. Currently, the theoretical foundation of coarray-domain DOA estimation relies on the assumption that the sources are uncorrelated. Therefore, sparse arrays and coarray-domain DOA estimation algorithms that are robust against source correlations will be of great interest in the future. Advancements in both theoretical and algorithmic approaches are necessary

to further this research area.

4. *Aperture-aware array design and evaluation:* As mentioned in Chapter 4, to the best of our knowledge, the previous literature on sparse arrays has considered constructing and comparing arrays with a fixed number of sensors N . Rather than only comparing different array geometries with a fixed number of sensors, we propose to additionally compare the array geometries under a fixed aperture constraint. This approach can be referred to as an ‘aperture-constrained’ evaluation of sensor arrays. To achieve optimal performance in such an aperture-constrained setting, it is essential to employ an aperture-aware design for sparse arrays. One method to accomplish this is to impose a strict constraint on the array aperture during the design stage. The concept of aperture-aware array design introduced in this work can be further explored to systematically design sparse arrays under aperture constraints. Several intriguing design questions arise in this context. For instance, given a fixed aperture A and a fixed number of sensors $N < A$, it would be of interest to determine the sensor arrangement that minimizes mutual coupling and optimizes DOA estimation performance.
5. *‘Periodic’ signals with non-integer periods:* The theory developed using Ramanujan subspaces, filter banks, and dictionaries is designed for discrete-time periodic signals with integer periods. However, many signals, such as ECGs, are continuous-time signals that are sampled and converted into discrete-time signals. This process introduces two fundamental challenges. Firstly, the original continuous-time signal may only be approximately periodic rather than strictly periodic. Secondly, even if the continuous-time signal is strictly periodic, it may not remain periodic after sampling, or its period may change in the discrete-time representation [236]. This poses a major challenge to applying the theoretical framework developed based on Ramanujan subspaces for representing and analyzing continuous-time approximately periodic signals. It would be of great interest in the future to further this line of research and expand the scope of the Ramanujan-subspaces based development for continuous-time approximately periodic signals.

BIBLIOGRAPHY

- [1] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.
- [2] H. Krim and M. Viberg, “Two decades of array signal processing research: The parametric approach,” *IEEE Signal Processing Magazine*, 1996.
- [3] R. Carriere and R. L. Moses, “High resolution radar target modeling using a modified prony estimator,” *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 1, pp. 13–18, 1992.
- [4] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.
- [5] R. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.
- [6] M. Christensen and A. Jakobsson, *Multi-pitch estimation*. Morgan & Claypool Publishers, 2009.
- [7] P. Stoica, R. L. Moses, *et al.*, *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005, vol. 452.
- [8] Z. Leonowicz, T. Lobos, and J. Rezmer, “Advanced spectrum estimation methods for signal analysis in power electronics,” *IEEE Transactions on Industrial Electronics*, vol. 50, no. 3, pp. 514–519, 2003.
- [9] X. Andrade, J. N. Sanders, and A. Aspuru-Guzik, “Application of compressed sensing to the simulation of atomic systems,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 35, pp. 13 928–13 933, 2012.
- [10] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, 1986. DOI: 10.1109/TAP.1986.1143830.
- [11] 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.
- [12] B. Ottersten, M. Viberg, and T. Kailath, “Analysis of subspace fitting and ml techniques for parameter estimation from sensor array data,” *IEEE Transactions on Signal Processing*, vol. 40, no. 3, pp. 590–600, 1992.
- [13] L. Hu, Z. Shi, J. Zhou, and Q. Fu, “Compressed sensing of complex sinusoids: An approach based on dictionary refinement,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3809–3822, 2012.

- [14] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE transactions on signal processing*, vol. 53, no. 8, pp. 3010–3022, 2005.
- [15] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE transactions on information theory*, vol. 59, no. 11, pp. 7465–7490, 2013.
- [16] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *Journal of Fourier Analysis and Applications*, vol. 19, pp. 1229–1254, 2013.
- [17] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5987–5999, 2013.
- [18] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, "Review of noise removal techniques in ecg signals," *IET Signal Processing*, vol. 14, no. 9, pp. 569–590, 2020.
- [19] D. Jia, K. Yanagisawa, Y. Ono, *et al.*, "Multiwindow nonharmonic analysis method for gravitational waves," *IEEE Access*, vol. 6, pp. 48 645–48 655, 2018.
- [20] J. Li and P. Stoica, *MIMO radar signal processing*. John Wiley & Sons, 2008.
- [21] S. Stergiopoulos, *Advanced signal processing handbook: theory and implementation for radar, sonar, and medical imaging real time systems*. CRC press, 2017.
- [22] F. Le Chevalier, *Principles of radar and sonar signal processing*. Artech house, 2002.
- [23] M. G. Amin, *Sparse Arrays for Radar, Sonar, and Communications*. John Wiley & Sons, 2024.
- [24] R. N. Bracewell, "Radio astronomy techniques," in *Astrophysik V: Verschiedenes/Astrophysics V: Miscellaneous*, Springer, 1962, pp. 42–129.
- [25] S. Rost and C. Thomas, "Array seismology: Methods and applications," *Reviews of geophysics*, vol. 40, no. 3, pp. 2–1, 2002.
- [26] S. U. Pillai, *Array signal processing*. Springer Science & Business Media, 2012.
- [27] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013.
- [28] D. Pavlidi, A. Griffin, M. Puigt, and A. Mouchtaris, "Real-time multiple sound source localization and counting using a circular microphone array," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2193–2206, 2013.

- [29] P. Suetens, *Fundamentals of medical imaging*. Cambridge university press, 2017.
- [30] S. M. Patole, M. Torlak, D. Wang, and M. Ali, “Automotive radars: A review of signal processing techniques,” *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.
- [31] J. Leech, “On the representation of $1, 2, \dots, n$ by differences,” *Journal of the London Mathematical Society*, vol. 1, no. 2, pp. 160–169, 1956.
- [32] P. Erdős and I. Gál, “On the representation of $1, 2, \dots, n$ by differences,” *Indagationes Math*, vol. 10, no. 9, pp. 379–382, 1948.
- [33] A. Moffet, “Minimum-redundancy linear arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 16, no. 2, pp. 172–175, 1968.
- [34] P. Pal and P. P. Vaidyanathan, “Nested arrays: A novel approach to array processing with enhanced degrees of freedom,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010.
- [35] P. P. Vaidyanathan and P. Pal, “Sparse sensing with co-prime samplers and arrays,” *IEEE Transactions on Signal Processing*, 2011. DOI: 10.1109/TSP.2010.2089682.
- [36] T. Svantesson, “Modeling and estimation of mutual coupling in a uniform linear array of dipoles,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.
- [37] T. Svantesson, “Mutual coupling compensation using subspace fitting,” in *Proceedings of the 2000 IEEE Sensor Array and Multichannel Signal Processing Workshop*, IEEE, 2000, pp. 494–498.
- [38] B. Friedlander and A. J. Weiss, “Direction finding in the presence of mutual coupling,” *IEEE transact. on antennas and propagation*, 1991.
- [39] C.-L. Liu and P. P. Vaidyanathan, “Super nested arrays: Linear sparse arrays with reduced mutual coupling—Part I: Fundamentals,” *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3997–4012, 2016.
- [40] R. Kolpakov, G. Bana, and G. Kucherov, “Mreps: Efficient and flexible detection of tandem repeats in dna,” *Nucleic acids research*, vol. 31, no. 13, pp. 3672–3678, 2003.
- [41] G. Benson, “Tandem repeats finder: A program to analyze dna sequences,” *Nucleic acids research*, vol. 27, no. 2, pp. 573–580, 1999.
- [42] S. V. Tenneti and P. P. Vaidyanathan, “Detecting tandem repeats in DNA using Ramanujan filter bank,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 21–24.
- [43] A. Heger and L. Holm, “Rapid automatic detection and alignment of repeats in protein sequences,” *Proteins: Structure, Function, and Bioinformatics*, vol. 41, no. 2, pp. 224–237, 2000.

- [44] M. A. Andrade, C. Perez-Iratxeta, and C. P. Ponting, “Protein repeats: Structures, functions, and evolution,” *Journal of structural biology*, vol. 134, no. 2-3, pp. 117–131, 2001.
- [45] M. R. Schroeder, “Period histogram and product spectrum: New methods for fundamental-frequency measurement,” *The Journal of the Acoustical Society of America*, vol. 43, no. 4, pp. 829–834, 1968.
- [46] D. J. Hermes, “Measurement of pitch by subharmonic summation,” *The journal of the acoustical society of America*, vol. 83, no. 1, pp. 257–264, 1988.
- [47] A. Camacho and J. G. Harris, “A sawtooth waveform inspired pitch estimator for speech and music,” *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.
- [48] T. Drugman and A. Alwan, “Joint robust voicing detection and pitch estimation based on residual harmonics,” *arXiv preprint arXiv:2001.00459*, 2019.
- [49] S. V. Tenneti, *The Nested Periodic Subspaces: Extensions of Ramanujan Sums for Period Estimation*. Doctoral dissertation, California Institute of Technology, 2018.
- [50] M. Planat, “Ramanujan sums for signal processing of low frequency noise,” in *Proceedings of the 2002 IEEE International Frequency Control Symposium and PDA Exhibition (Cat. No.02CH37234)*, May 2002, pp. 715–720.
- [51] S. Pei and K. Lu, “Intrinsic integer-periodic functions for discrete periodicity detection,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1108–1112, Aug. 2015.
- [52] P. P. Vaidyanathan, “Ramanujan sums in the context of signal processing — Part I: Fundamentals,” *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4145–4157, 2014.
- [53] S. V. Tenneti and P. P. Vaidyanathan, “Nested periodic matrices and dictionaries: New signal representations for period estimation,” *IEEE Transactions on Signal Processing*, vol. 63, no. 14, pp. 3736–3750, Jul. 2015.
- [54] P. Saidi, G. Atia, and A. Vosoughi, “Detection of visual evoked potentials using Ramanujan periodicity transform for real time brain computer interfaces,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 959–963.
- [55] P. Saidi, G. Atia, and A. Vosoughi, “On robust detection of brain stimuli with Ramanujan periodicity transforms,” in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct. 2017, pp. 729–733.
- [56] P. Saidi, A. Vosoughi, and G. Atia, “Detection of brain stimuli using Ramanujan periodicity transforms,” *Journal of Neural Engineering*, vol. 16, no. 3, p. 036021, Apr. 2019.

- [57] B. Shah Shaik, V. K. Chakka, S. Goli, and A. S. Reddy, "Removal of narrowband interference (PLI in ECG signal) using Ramanujan periodic transform (RPT)," in *2016 International Conference on Signal Processing and Communication (ICSC)*, Dec. 2016, pp. 233–237.
- [58] B. S. Shaik and V. K. Chakka, "Joint reduction of baseline wander, PLI and its harmonics in ECG signal using Ramanujan periodic transform," in *2016 IEEE Annual India Conference (INDICON)*, Dec. 2016, pp. 1–5.
- [59] B. S. Shaik, V. K. Chakka, and S. Goli, "Ramanujan and DFT mixed basis representation for removal of PLI in ECG signal," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb. 2017, pp. 509–512.
- [60] S. Ramanujan, "On certain trigonometrical sums and their applications in the theory of numbers," *Transactions of the Cambridge Philosophical Society*, vol. XXII, no. 13, pp. 259–276, 1918.
- [61] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, Fourth. Oxford, 1975.
- [62] P. P. Vaidyanathan, "Ramanujan sums in the context of signal processing — Part II: FIR representations and applications," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4158–4172, 2014.
- [63] S. Tenneti and P. P. Vaidyanathan, "Ramanujan filter banks for estimation and tracking of periodicities," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 3851–3855.
- [64] P. P. Vaidyanathan and S. Tenneti, "Properties of Ramanujan filter banks," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug. 2015, pp. 2816–2820.
- [65] P. P. Vaidyanathan and S. Tenneti, "Srinivasa Ramanujan and signal-processing problems," *Philosophical Transactions of the Royal Society A*, vol. 378, Issue 2163, Dec. 2019.
- [66] S. A. Khayam, "The discrete cosine transform (dct): Theory and application," *Michigan State University*, vol. 114, no. 1, p. 31, 2003.
- [67] A. N. Akansu and R. A. Haddad, *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic press, 2001.
- [68] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.
- [69] R. Dony *et al.*, "Karhunen-loeve transform," *The transform and data compression handbook*, vol. 1, no. 1-34, p. 29, 2001.
- [70] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010. DOI: 10.1109/JPROC.2010.2040551.

- [71] I. Tošić and P. Frossard, “Dictionary learning,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011. DOI: 10.1109/MSP.2010.939537.
- [72] N. J. Larsson and A. Moffat, “Off-line dictionary-based compression,” *Proceedings of the IEEE*, vol. 88, no. 11, pp. 1722–1732, 2000. DOI: 10.1109/5.892708.
- [73] D. Malioutov, M. Cetin, and A. S. Willsky, “A sparse signal reconstruction perspective for source localization with sensor arrays,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005. DOI: 10.1109/TSP.2005.850882.
- [74] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [75] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [76] M. Protter and M. Elad, “Image sequence denoising via sparse and redundant representations,” *IEEE Transactions on Image Processing*, 2009.
- [77] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.
- [78] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [79] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [80] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [81] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural networks*, vol. 6, no. 6, pp. 861–867, 1993.
- [82] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [83] S. Qin, Y. D. Zhang, and M. G. Amin, “Generalized coprime array configurations for direction-of-arrival estimation,” *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, 2015.
- [84] M. Ishiguro, “Minimum redundancy linear arrays for a large number of antennas,” *Radio Science*, vol. 15, no. 6, pp. 1163–1170, 1980.

- [85] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1983.
- [86] C.-L. Liu and P. P. Vaidyanathan, "Cramér–Rao bounds for coprime and other sparse arrays, which find more sources than sensors," *Digital Signal Processing*, vol. 61, pp. 43–61, 2017.
- [87] M. Wang and A. Nehorai, "Coarrays, MUSIC, and the Cramér–Rao bound," *IEEE Transactions on Signal Processing*, 2016.
- [88] P. Kulkarni and P. P. Vaidyanathan, "Non-integer arrays for array signal processing," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5457–5472, Nov. 2022. DOI: 10.1109/TSP.2022.3221862.
- [89] Z. Zheng, Y. Huang, W.-Q. Wang, and H. C. So, "Augmented covariance matrix reconstruction for DOA estimation using difference coarray," *IEEE Transactions on Signal Processing*, 2021.
- [90] S. Sedighi, B. S. M. R. Rao, and B. Ottersten, "An asymptotically efficient weighted least squares estimator for co-array-based DoA estimation," *IEEE Transactions on Signal Processing*, 2019.
- [91] A. Raza, W. Liu, and Q. Shen, "Thinned coprime array for second-order difference co-array generation with reduced mutual coupling," *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 2052–2065, 2019.
- [92] A. M. Shaalan, J. Du, and Y.-H. Tu, "Dilated nested arrays with more degrees of freedom (DOFs) and less mutual coupling—Part I: The fundamental geometry," *IEEE Transactions on Signal Processing*, 2022.
- [93] C.-L. Liu and P. P. Vaidyanathan, "Remarks on the spatial smoothing step in coarray MUSIC," *IEEE Signal Processing Letters*, 2015.
- [94] P. P. Vaidyanathan and P. Kulkarni, "Sparse arrays: Fundamentals," *Sparse Arrays for Radar, Sonar, and Communications*, pp. 1–40, 2024. DOI: 10.1002/9781394191048.ch1.
- [95] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2006–2021, 2010.
- [96] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1989.
- [97] P. P. Vaidyanathan and P. Pal, "Why does direct-MUSIC on sparse-arrays work?" In *2013 Asilomar Conference on Signals, Systems and Computers*, IEEE, 2013, pp. 2007–2011.
- [98] B. D. Rao and K. S. Hari, "Performance analysis of root-MUSIC," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1989.

- [99] C. Chambers, T. C. Tozer, K. C. Sharman, and T. S. Durrani, "Temporal and spatial sampling influence on the estimates of superimposed narrow-band signals: When less can mean more," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3085–3098, 1996.
- [100] F. Schwartau, Y. Schröder, L. Wolf, and J. Schoebel, "Large minimum redundancy linear arrays: Systematic search of perfect and optimal rulers exploiting parallel processing," *IEEE Open Journal of Antennas and Propagation*, vol. 2, pp. 79–85, 2020.
- [101] C.-L. Liu and P. P. Vaidyanathan, "Super nested arrays: Linear sparse arrays with reduced mutual coupling—Part II: High-order extensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4203–4217, 2016.
- [102] M. Yang, L. Sun, X. Yuan, and B. Chen, "Improved nested array with hole-free DCA and more degrees of freedom," *Electronics Letters*, 2016.
- [103] J. Liu, Y. Zhang, Y. Lu, S. Ren, and S. Cao, "Augmented nested arrays with enhanced DOF and reduced mutual coupling," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5549–5563, 2017.
- [104] Z. Zheng, W.-Q. Wang, Y. Kong, and Y. D. Zhang, "MISC array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect," *IEEE Transactions on Signal Processing*, 2019.
- [105] W. Zheng, X. Zhang, Y. Wang, J. Shen, and B. Champagne, "Padded coprime arrays for improved DOA estimation: Exploiting hole representation and filling strategies," *IEEE Transactions on Signal Processing*, 2020.
- [106] E. BouDaher, F. Ahmad, M. G. Amin, and A. Hoorfar, "Mutual coupling effect and compensation in non-uniform arrays for direction-of-arrival estimation," *Digital Signal Processing*, vol. 61, pp. 3–14, 2017.
- [107] F. Sellone and A. Serra, "A novel online mutual coupling compensation algorithm for uniform and linear arrays," *IEEE Transactions on signal processing*, vol. 55, no. 2, pp. 560–573, 2007.
- [108] Z. Ye, J. Dai, X. Xu, and X. Wu, "DOA estimation for uniform linear array with mutual coupling," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 280–288, 2009.
- [109] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.
- [110] H.-S. Lui, H. T. Hui, and M. S. Leong, "A note on the mutual-coupling problems in transmitting and receiving antenna arrays," *IEEE Antennas and Propagation Magazine*, vol. 51, no. 5, pp. 171–176, 2009.
- [111] G. Brown, "Directional antennas," *Proceedings of the institute of radio engineers*, vol. 25, no. 1, pp. 78–145, 1937.

- [112] H. King, "Mutual impedance of unequal length antennas in echelon," *IRE Transactions on Antennas and Propagation*, 1957.
- [113] K. M. Pasala and E. M. Friel, "Mutual coupling effects and their reduction in wideband direction of arrival estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 4, pp. 1116–1122, 1994.
- [114] I. Gupta and A. Ksienski, "Effect of mutual coupling on the performance of adaptive arrays," *IEEE Transact. on Antennas and Propagation*, 1983.
- [115] H. Hui, "Improved compensation for the mutual coupling effect in a dipole array for direction finding," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 9, pp. 2498–2503, 2003.
- [116] H. S. Lui, H. T. Hui, *et al.*, "Mutual coupling compensation for direction-of-arrival estimations using the receiving-mutual-impedance method," *International journal of Antennas and Propagation*, vol. 2010, 2010.
- [117] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*. Basic books, 2011, vol. 1.
- [118] S. Ramo, J. R. Whinnery, and T. Van Duzer, *Fields and waves in communication electronics*. John Wiley & Sons, 1994.
- [119] B. Friedlander, "On the mutual coupling matrix in array signal processing," in *54th Asilomar Conference on Signals, Systems, and Computers*, 2020.
- [120] Z. Yang and K. Wang, "Nonasymptotic performance analysis of direct-augmentation and spatial-smoothing ESPRIT for localization of more sources than sensors using sparse arrays," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.
- [121] Z. Yang, X. Chen, and X. Wu, "A robust and statistically efficient maximum-likelihood method for DOA estimation using sparse linear arrays," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.
- [122] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1990.
- [123] B. D. Rao and K. Hari, "Performance analysis of ESPRIT and TAM in determining the direction of arrival of plane waves in noise," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 12, pp. 1990–1995, 2002.
- [124] P. Sarangi, M. C. Hücümenoğlu, and P. Pal, "Beyond coarray MUSIC: Harnessing the difference sets of nested arrays with limited snapshots," *IEEE Signal Processing Letters*, vol. 28, pp. 2172–2176, 2021.

- [125] P. Sarangi, M. C. Hucumenoglu, R. Rajamaki, and P. Pal, "Super-resolution with sparse arrays: A non-asymptotic analysis of spatio-temporal trade-offs," *arXiv preprint arXiv:2301.01734*, 2023.
- [126] P. P. Vaidyanathan and P. Pal, "Direct-MUSIC on sparse arrays," in *2012 International Conference on Signal Processing and Communications (SP-COM)*, IEEE, 2012, pp. 1–5.
- [127] <https://math.stackexchange.com/questions/151081/gcd-of-rationals>.
- [128] mathworld.wolfram.com/GreatestCommonDivisor.html.
- [129] Y. I. Abramovich, N. K. Spencer, and A. Y. Gorokhov, "Resolving manifold ambiguities in direction-of-arrival estimation for nonuniform linear antenna arrays," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2629–2643, 1999.
- [130] A. Manikas and C. Proukakis, "Modeling and estimation of ambiguities in linear arrays," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2166–2179, 1998.
- [131] B. A. Obeidat, Y. Zhang, and M. G. Amin, "Range and DOA estimation of polarized near-field signals using fourth-order statistics," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 2, 2004, pp. ii–97.
- [132] A. Ahmed, Y. D. Zhang, and B. Himed, "Effective nested array design for fourth-order cumulant-based DOA estimation," in *2017 IEEE Radar Conference (RadarConf)*, IEEE, 2017, pp. 0998–1002.
- [133] P. J. Bevelacqua and C. A. Balanis, "Minimum sidelobe levels for linear arrays," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 12, pp. 3442–3449, 2007.
- [134] R. Harrington, "Sidelobe reduction by nonuniform element spacing," *IRE Transactions on Antennas and Propagation*, vol. 9, no. 2, pp. 187–192, 1961.
- [135] P. J. Bevelacqua and C. A. Balanis, "Optimizing antenna array geometry for interference suppression," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 637–641, 2007.
- [136] B. P. Kumar and G. Branner, "Design of unequally spaced arrays for performance improvement," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 3, pp. 511–523, 1999.
- [137] B. Friedlander, "The root-music algorithm for direction finding with interpolated arrays," *Signal processing*, vol. 30, no. 1, pp. 15–29, 1993.
- [138] F. Belloni, A. Richter, and V. Koivunen, "DoA estimation via manifold separation for arbitrary array structures," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4800–4810, 2007.

- [139] Y. I. Abramovich and N. K. Spencer, "Detection-estimation of more uncorrelated sources than sensors in noninteger sparse linear antenna arrays," in *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No. 00TH8496)*, IEEE, 2000, pp. 99–103.
- [140] Y. I. Abramovich, N. K. Spencer, and A. Y. Gorokhov, "DOA estimation for noninteger linear antenna arrays with more uncorrelated sources than sensors," *IEEE transactions on signal processing*, vol. 48, no. 4, pp. 943–955, 2000.
- [141] M. Rubsamen and A. B. Gershman, "Direction-of-arrival estimation for nonuniform sensor arrays: From manifold separation to fourier domain music methods," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 588–599, 2008.
- [142] M. Wagner, Y. Park, and P. Gerstoft, "Gridless DOA estimation and root-music for non-uniform linear arrays," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2144–2157, 2021.
- [143] P. Kulkarni and P. P. Vaidyanathan, "Rational arrays for DOA estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 5008–5012. DOI: 10.1109/ICASSP43922.2022.9746954.
- [144] P. Kulkarni and P. P. Vaidyanathan, "Rational arrays for DOA estimation: New insights and performance evaluation," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2022, pp. 91–95. DOI: 10.1109/IEEECONF56349.2022.10051963.
- [145] P. Kulkarni and P. P. Vaidyanathan, "Difference coarrays of rational arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095426.
- [146] https://en.wikipedia.org/wiki/Fundamental_theorem_of_arithmetic.
- [147] A. B. Gershman and J. F. Bohme, "A note on most favorable array geometries for doa estimation and array interpolation," *IEEE Signal Processing Letters*, vol. 4, no. 8, pp. 232–235, 1997.
- [148] Z.-M. Liu, C. Zhang, and S. Y. Philip, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–27, 2018.
- [149] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1688–1692, 2019. DOI: 10.1109/LSP.2019.2945115.

- [150] R. Zhang, B. Shim, and W. Wu, "Direction-of-arrival estimation for large antenna arrays with hybrid analog and digital architectures," *IEEE Transactions on Signal Processing*, vol. 70, pp. 72–88, 2021.
- [151] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014.
- [152] J. Shi, G. Hu, X. Zhang, and H. Zhou, "Generalized nested array: Optimization for degrees of freedom and mutual coupling," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1208–1211, 2018.
- [153] P. Kulkarni and P. P. Vaidyanathan, "Weight-constrained sparse arrays for direction of arrival estimation under high mutual coupling," *IEEE Transactions on Signal Processing*, Sep. 2024. DOI: 10.1109/TSP.2024.3461720.
- [154] P. Kulkarni and P. P. Vaidyanathan, "Sparse, weight-constrained arrays with $O(N)$ aperture for reduced mutual coupling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 12 891–12 895. DOI: 10.1109/ICASSP48485.2024.10446770.
- [155] P. Kulkarni and P. P. Vaidyanathan, "Weight-constrained nested arrays with $w(1) = w(2) = 0$ for reduced mutual coupling," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Jul. 2024, pp. 1–5. DOI: 10.1109/SAM60225.2024.10636451.
- [156] Y. D. Zhang, M. G. Amin, and B. Himed, "Sparsity-based DOA estimation using co-prime arrays," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 3967–3971.
- [157] J. Shi, F. Wen, Y. Liu, Z. Liu, and P. Hu, "Enhanced and generalized coprime array for direction of arrival estimation," *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [158] P. Kulkarni and P. P. Vaidyanathan, "On the efficiency of coarray-based direction of arrival estimation," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2023, pp. 1334–1338. DOI: 10.1109/IEEECONF59524.2023.10476980.
- [159] C.-L. Liu, P. P. Vaidyanathan, and P. Pal, "Coprime coarray interpolation for DOA estimation via nuclear norm minimization," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016.
- [160] H. Qiao and P. Pal, "Unified analysis of co-array interpolation for direction-of-arrival estimation," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017.
- [161] C. Zhou, Z. Shi, Y. Gu, and Y. D. Zhang, "Coarray interpolation-based coprime array DOA estimation via covariance matrix reconstruction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

- [162] C.-L. Liu, P. P. Vaidyanathan, and P. Pal, “Coprime coarray interpolation for DOA estimation via nuclear norm minimization,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016.
- [163] X. Wu, W.-P. Zhu, and J. Yan, “A Toeplitz covariance matrix reconstruction approach for direction-of-arrival estimation,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8223–8237, 2017.
- [164] C. Zhou, Y. Gu, Z. Shi, and Y. D. Zhang, “Off-grid direction-of-arrival estimation using coprime array interpolation,” *IEEE Signal Processing Letters*, vol. 25, no. 11, 2018.
- [165] C. Zhou, Y. Gu, Y. D. Zhang, and Z. Shi, “Sparse array interpolation for direction-of-arrival estimation,” *Sparse Arrays for Radar, Sonar, and Communications*, 2024.
- [166] A. Ahmed and Y. D. Zhang, “Generalized non-redundant sparse array designs,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 4580–4594, 2021.
- [167] P. Kulkarni and P. P. Vaidyanathan, “Efficient use of non-central ULA segments in the coarrays of weight-constrained sparse arrays,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2024. DOI: 10.1109/IEEECONF60004.2024.10943027.
- [168] P. Kulkarni and P. P. Vaidyanathan, “Interpolation for weight-constrained nested arrays having non-central ULA segments in the coarray,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2025. DOI: 10.1109/ICASSP49660.2025.10890061.
- [169] P. Kulkarni and P. P. Vaidyanathan, “Generalized constructions of weight-constrained sparse arrays,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2025.
- [170] CVX Research, Inc., *CVX: Matlab software for disciplined convex programming, version 2.0*, <https://cvxr.com/cvx>, Aug. 2012.
- [171] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., Springer-Verlag Limited, 2008, pp. 95–110.
- [172] Z. Yang, X. Chen, and X. Wu, “A robust and statistically efficient maximum-likelihood method for DOA estimation using sparse linear arrays,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 6798–6812, 2023.
- [173] P. Kulkarni and P. P. Vaidyanathan, “On the efficiency of a Toeplitz covariance matrix construction approach for DOA estimation,” in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Oct. 2025, submitted.

- [174] V. Pan, "Sequential and parallel complexity of approximate evaluation of polynomial zeros," *Computers & Mathematics with Applications*, 1987.
- [175] Z.-M. Liu, Z.-T. Huang, and Y.-Y. Zhou, "Sparsity-inducing direction finding for narrowband and wideband signals based on array covariance vectors," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 1–12, 2013.
- [176] K.-L. Chen and B. D. Rao, "A comparative study of invariance-aware loss functions for deep learning-based gridless direction-of-arrival estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. DOI: 10.1109/ICASSP49660.2025.10889620.
- [177] K.-L. Chen and B. D. Rao, "Subspace representation learning for sparse linear arrays to localize more sources than sensors: A deep learning methodology," *IEEE Transactions on Signal Processing*, 2025.
- [178] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.
- [179] G. Chen, W. Xie, and Y. Zhao, "Wavelet-based denoising: A brief review," in *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, 2013, pp. 570–574.
- [180] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.
- [181] T. Yoshizawa, S. Hirobayashi, and T. Misawa, "Noise reduction for periodic signals using high-resolution frequency analysis," *EURASIP Journal on Audio, Speech, and Music Processing*, Jul. 2011.
- [182] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, 2010.
- [183] R. Oektem, L. P. Yaroslavsky, K. O. Egiazarian, and J. T. Astola, "Transform domain approaches for image denoising," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 149–156, 2002.
- [184] D. L. Donoho, I. M. Johnstone, G. Kerkycharian, and D. Picard, "Wavelet shrinkage: Asymptopia?" *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 2, pp. 301–337, 1995.
- [185] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of control*, vol. 50, no. 5, pp. 1873–1896, 1989.
- [186] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, 1996.

- [187] S. V. Tenneti and P. P. Vaidyanathan, "Detection of protein repeats using the Ramanujan filter bank," in *Asilomar Conference on Signals, Systems and Computers*, 2016.
- [188] P. P. Vaidyanathan, "Multirate systems and filter banks," 1992.
- [189] P. Kulkarni and P. P. Vaidyanathan, "Periodicity-aware signal denoising using Capon-optimized Ramanujan filter banks and pruned Ramanujan dictionaries," *IEEE Transactions on Signal Processing*, vol. 71, pp. 494–511, Feb. 2023. DOI: 10.1109/TSP.2023.3244109.
- [190] P. Kulkarni and P. P. Vaidyanathan, "Periodic signal denoising: An analysis-synthesis framework based on Ramanujan filter banks and dictionaries," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 5100–5104. DOI: 10.1109/ICASSP39728.2021.9413689.
- [191] P. P. Vaidyanathan and P. Pal, "The farey-dictionary for sparse representation of periodic signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 360–364.
- [192] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, 2010.
- [193] T. Strohmer and R. W. Heath Jr, "Grassmannian frames with applications to coding and communication," *Applied and computational harmonic analysis*, vol. 14, no. 3, pp. 257–275, 2003.
- [194] P. P. Vaidyanathan and S. Tenneti, "Efficient multiplier-less structures for Ramanujan filter banks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.
- [195] P. Kulkarni and P. P. Vaidyanathan, "On the zeros of Ramanujan filters," *IEEE Signal Processing Letters*, vol. 27, pp. 735–739, Apr. 2020. DOI: 10.1109/LSP.2020.2990610.
- [196] K. Koh, S. Kim, and S. Boyd, *Simple matlab solver for l1-regularized least squares problems*, https://web.stanford.edu/~boyd/l1_ls/, 2008.
- [197] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.
- [198] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.
- [199] J. L. Krolik, "The performance of matched-field beamformers with mediterranean vertical array data," *IEEE Transactions on Signal Processing*, vol. 44, no. 10, pp. 2605–2611, 1996.
- [200] R. G. Lorenz and S. P. Boyd, "Robust minimum variance beamforming," *IEEE transactions on signal processing*, 2005.

- [201] P. P. Vaidyanathan, *The theory of linear prediction*. Morgan & Claypool Publishers, 2008.
- [202] <https://www.physionet.org/content/nsrdb/1.0.0/>.
- [203] A. L. Goldberger, L. A. Amaral, L. Glass, *et al.*, “Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals,” *circulation*, vol. 101, no. 23, e215–e220, 2000.
- [204] P. Kulkarni and P. P. Vaidyanathan, “Interpolation filter model for Ramanujan subspace signals,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095682.
- [205] P. P. Vaidyanathan and B. Vrcelj, “Biorthogonal partners and applications,” *IEEE Transactions on signal processing*, vol. 49, no. 5, pp. 1013–1027, 2001.
- [206] J. R. Treichler, I. Fijalkow, and C. Johnson, “Fractionally spaced equalizers,” *IEEE signal processing magazine*, vol. 13, no. 3, pp. 65–81, 1996.
- [207] M. Marden, *Geometry of Polynomials* (Geometry of Polynomials 3). American Mathematical Society, 1949.
- [208] S. Samadi, M. O. Ahmad, and M. N. S. Swamy, “Ramanujan sums and discrete fourier transforms,” *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 293–296, Apr. 2005.
- [209] T. Nagell, *Introduction to number theory*. Chelsea Publishing Company, 1964.
- [210] J. H. McClellan and C. M. Rader, *Number theory in digital signal processing*. Prentice-Hall, 1979.
- [211] K. Engan, S. O. Aase, and J. Hakon Husoy, “Method of optimal directions for frame design,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 1999, 2443–2446 vol.5. DOI: 10.1109/ICASSP.1999.760624.
- [212] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, “Learning unions of orthonormal bases with thresholded singular value decomposition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2005, v/293–v/296 Vol. 5. DOI: 10.1109/ICASSP.2005.1416298.
- [213] M. Yaghoobi, L. Daudet, and M. E. Davies, “Parametric dictionary design for sparse coding,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4800–4810, 2009. DOI: 10.1109/TSP.2009.2026610.
- [214] Z. Yang, J. Li, P. Stoica, and L. Xie, “Sparse methods for direction-of-arrival estimation,” in *Academic Press Library in Signal Processing, Volume 7*, Elsevier, 2018, pp. 509–581.

- [215] W. Guo, T. Qiu, H. Tang, and W. Zhang, "Performance of RBF neural networks for array processing in impulsive noise environment," *Digital Signal Processing*, vol. 18, no. 2, pp. 168–178, 2008.
- [216] L.-L. Wu and Z.-T. Huang, "Coherent SVR learning for wideband direction-of-arrival estimation," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 642–646, 2019.
- [217] W. Liu, "Super resolution DOA estimation based on deep neural network," *Scientific Reports*, vol. 10, no. 1, pp. 1–9, 2020.
- [218] A. M. Elbir, "Deepmusic: Multiple signal classification via deep learning," *IEEE Sensors Letters*, vol. 4, no. 4, pp. 1–4, 2020.
- [219] J. P. Merkofer, G. Revach, N. Shlezinger, and R. J. van Sloun, "Deep augmented music algorithm for data-driven DoA estimation," *arXiv preprint arXiv:2109.10581*, 2021.
- [220] P. Kulkarni and P. P. Vaidyanathan, "Feature engineering for DOA estimation using a convolutional neural network, for sparse arrays," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2021, pp. 246–250. DOI: 10.1109/IEEECONF53345.2021.9723112.
- [221] P. Kulkarni and P. P. Vaidyanathan, "K-SVD based periodicity dictionary learning," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Nov. 2020, pp. 1333–1337. DOI: 10.1109/IEEECONF51394.2020.9443567.
- [222] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.
- [223] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993, 40–44 vol.1. DOI: 10.1109/ACSSC.1993.342465.
- [224] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997. DOI: 10.1109/78.558475.
- [225] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010. DOI: 10.1109/JPROC.2010.2044010.
- [226] M. Elad, R. Rubinstein, and M. Zibulevsky, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *Technical Report - Computer Science, Technion*, Apr. 2008.

- [227] J. S. Turek, M. Elad, and I. Yavneh, "Clutter mitigation in echocardiography using sparse signal separation," *International Journal on Biomedical Imaging*, Apr. 2015.
- [228] P. Gerstoft, C. F. Mecklenbräuker, A. Xenaki, and S. Nannuru, "Multisnapshot sparse bayesian learning for DOA," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1469–1473, 2016.
- [229] I. Aboumahmoud, A. Muqaibel, M. Alhassoun, and S. Alawsh, "A review of sparse sensor arrays for two-dimensional direction-of-arrival estimation," *IEEE Access*, vol. 9, pp. 92 999–93 017, 2021.
- [230] P. P. Vaidyanathan and P. Pal, "Theory of sparse coprime sensing in multiple dimensions," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3592–3608, 2011.
- [231] P. Pal and P. P. Vaidyanathan, "Coprimalty of certain families of integer matrices," *IEEE transactions on signal processing*, vol. 59, no. 4, pp. 1481–1490, 2010.
- [232] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.
- [233] S. Fortunati, R. Grasso, F. Gini, M. S. Greco, and K. LePage, "Single-snapshot DOA estimation by using compressed sensing," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, pp. 1–17, 2014.
- [234] B. M. Radich and K. M. Buckley, "Single-snapshot DOA estimation and source number detection," *IEEE Signal processing letters*, vol. 4, no. 4, pp. 109–111, 1997.
- [235] P. Häcker and B. Yang, "Single snapshot DOA estimation," *Advances in Radio Science*, vol. 8, pp. 251–256, 2010.
- [236] P. P. Vaidyanathan and S. Tenneti, "When does periodicity in discrete-time imply that in continuous-time?" In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4544–4548.