On the Use of Uncalibrated Digital Phased Arrays for Blind Signal Separation for Interference Removal in Congested Spectral Bands

Lauren O. Lusk

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Joseph D. Gaeddert, Chair

A. A. (Louis) Beex

Steven W. Ellingson

April 21, 2023

Blacksburg, Virginia

Keywords: Beamforming, Digital Phased Arrays, Interference Suppression, Singular Value Decomposition

On the Use of Uncalibrated Digital Phased Arrays for Blind Signal Separation for Interference Removal in Congested Spectral Bands

Lauren O. Lusk

(ABSTRACT)

With usable spectrum becoming increasingly more congested, the need for robust, adaptive communications to take advantage of spatially-separated signal sources is apparent. Traditional phased array beamforming techniques used for interference removal rely on perfect calibration between elements and precise knowledge of the array configuration; however, if the exact array configuration is not known (unknown or imperfect assumption of element locations, unknown mutual coupling between elements, etc.), these traditional beamforming techniques are not viable, so a blind beamforming approach is required.

A novel blind beamforming approach is proposed to address complex narrow-band interference environments where the precise array configuration is unknown. The received signal is decomposed into orthogonal narrow-band partitions using a polyphase filter-bank channelizer, and a rank-reduced version of the received matrix on each sub-channel is computed through reconstruction by retaining a subset of its singular values. The wideband spectrum is synthesized through a near-perfect polyphase reconstruction filter, and a composite wideband spectrum is obtained from the maximum eigenvector of the resulting covariance matrix.The resulting process is shown to suppress numerous interference sources (in special cases even with more than the degrees of freedom of the array), all without any knowledge of the primary signal of interest. Results are validated with both simulation and wireless laboratory over-the-air experimentation.

On the Use of Uncalibrated Digital Phased Arrays for Blind Signal Separation for Interference Removal in Congested Spectral Bands

Lauren O. Lusk

(GENERAL AUDIENCE ABSTRACT)

As the number of devices using wireless communications increase, the amount of usable radio frequency spectrum becomes increasingly congested. As a result, the need for robust, adaptive communications to improve spectral efficiency and ensure reliable communication in the presence of interference is apparent. One solution is using beamforming techniques on digital phased array receivers to maximize the energy in a desired direction and steer nulls to remove interference. However, traditional phased array beamforming techniques used for interference removal rely on perfect calibration between antenna elements and precise knowledge of the array configuration. Consequently, if the exact array configuration is not known (unknown or imperfect assumption of element locations, unknown mutual coupling between elements, etc.), these traditional beamforming techniques are not viable, so a beamforming approach with relaxed requirements (blind beamforming) is required. This thesis proposes a novel blind beamforming approach to address complex narrow-band interference in spectrally congested environments where the precise array configuration is unknown. The resulting process is shown to suppress numerous interference sources, all without any knowledge of the primary signal of interest. Results are validated with both simulation and wireless laboratory experimentation conducted with a two-element array, verifying that proposed beamforming approach achieves a similar performance to the theoretical performance bound of receiving packets in AWGN with no interference present.

Acknowledgments

There are many people who I would like to thank, without whom this thesis would not exist. First and foremost, I would like to thank my advisor Dr. Gaeddert for taking me on as his student and helping me grow as a researcher. I will forever be thankful your patience and willingness to teach me. I could not ask for a better advisor. Next, I would like to thank Dr. Beex for his encouragement and input especially during the early stages of my graduate career and this thesis. Your depth of knowledge is inspiring, and I'm very thankful that I was able to receive your advice. To Dr. Ellingson, thank you for serving on my committee. I appreciate you taking the time to do so. To my friends, near and far, thank you for supporting me and helping me stay sane. Finally, to my family and my boyfriend, thank you for supporting my decision to pursue graduate school in Virginia. I could not have made it this far in life without your continuous love and support.

Contents

List of Figures

List of Tables

List of Abbreviations

$2L$ The block size for processing SVD, consistent with L for Hann window

h_0 The channelizer prototype filter

M The number of sub-channels in channelizer

N The number of antenna elements ("sensors")

$x_i(n, m)$ i is the element index; n is the time index; m is the channel index

AWGN additive white Gauss noise

CCI co-channel interference

CMA constant modulus algorithm

CNN convolutional neural network

CSV composite steering vector

dB decibels

DFT discrete Fourier transform

DOA direction of arrival

EM electromagnetic

EVM error vector magnitude

FCC Federal Communications Commission

FFT fast Fourier transform

IoT Internet of Things

IQR interquartile range

ISI intersymbol interference

ISM industrial, scientific, and medical

LO local oscillator

LMS least mean-squares

MUSIC multiple signal classification

MVDR minimum variance distortionless response

OFDM orthogonal frequency-division multiplexing

PDF probability density function

PER packet error rate

PFB polyphase filterbank

PSD power spectral density

RQ Rayleigh quotient

QPSK quaternary phase-shift keying

RFFE radio frequency front end

RLS recursive least-squares

SCORE spectral self-coherent restoral

SDR software-defined radio

SIC successive interference cancellation

SINR signal-to-interference-plus-noise ratio

SNR signal-to-noise ratio

SOI signal of interest

STAP space-time adaptive processing

SV singular value

SVD singular value decomposition

SVIRA Singular Value Interference Removal Algorithm

USRP universal software radio peripheral

Chapter 1

Introduction

1.1 Motivation

With the increased demand on robust and reliable wireless communications links, open spectrum bands are becoming more and more scarce, particularly as spectrum sharing solutions become more prevalent. Spectrum sharing solutions are especially of interest in the industrial, scientific, and medical (ISM) bands. Historically, these bands are reserved for industrial, scientific, and medical use (hence the name) for devices such as microwave ovens and shortwave medical heating devices. However, since the Federal Communications Commission (FCC) opened the bands to unlicensed users in 1985 [1], a growing number of both licensed and unlicensed users attempt to share the available spectrum. Many of the unlicensed users are short-ranged, low-power wireless communication systems that take advantage of the unlicensed spectrum to cut costs. These unlicensed devices include Bluetooth, contactless smart cards, wireless microphones, garage door openers, RFID tags, Wi-Fi, and other devices associated with the Internet of Things (IoT). Although they are permitted to transmit in the bands, unlicensed devices are not allowed to interfere with the transmission of licensed devices and must accept a certain amount of interference. As the number of devices using the ISM bands increases, the amount of interference a device must contend with also increases.

Furthermore, as shared spectrum bands incorporate more and more disparate signal types

and waveforms, new solutions need to account not only for the interference they receive, but also the interference they impart on existing users. For the general problem of robust wireless communications in congested spectrum environments, reliability can be achieved in several different ways:

- *spread-spectrum communications* which generally distribute the information rate disproportionately across the bandwidth so as to minimize the statistical impact of interference,
- *successive interference cancellation* which seeks to iteratively remove interference through estimation and subtraction,
- *spectral white-space estimation* methods which attempt to locate (and possibly predict) which frequency bands are available, often using non-contiguous spectrum waveforms, and
- *multi-antenna methods* which include a entire array of techniques to provide spatial diversity

Combinations of the above can also be used to improve reliability and mitigate interference; however, the combined techniques are often still restricted by the limitations of the underlying techniques. Spread-spectrum techniques impart a performance degradation on both the legacy and secondary users. Successive interference cancellation methods are computationally intense and rely heavily on the correct estimation of parameters to operate while white space estimation requires waveform re-definition, and multi-antenna methods traditionally require precise calibration to produce accurate beams (e.g. nulls in the direction of interference sources).

The potential benefit in spatial diversity afforded by multi-antenna systems is substantial

in its ability to simultaneously null interference while providing gain in the direction of a desired signal. In particular, digital phased-array systems provide substantial benefits over their analog counterparts, including the ability to process simultaneously occuring distinct beams and apply any number of signal processing techniques for multi-spatial signal separation. Traditional digital phased-array processing, however, is expensive and requires precise calibration of antenna spacing to properly steer beams. The calibration issue becomes increasingly complex when modern platforms are taken into account. Many modern platforms are often already overloaded with existing diverse and possibly heterogeneous antennas, and have little to no space to incorporate a large array. As such, an optimal solution would be to treat the existing antennas as a digital array; however, the calibration between the existing antennas might not be precise enough to utilize traditional beamforming algorithms. Additionally, narrow-band interfering signals may be too numerous for the degrees of freedom afforded by the array to null—this is particularly true for recovering a wide-band carrier in congested frequency bands. The discussion on beamforming methods is continued in Section 2.2.

Furthermore, while the design of new waveforms can partially address this calibration problem, many legacy systems are inflexible to changes to the physical layer outside the radio frequency front end (RFFE). As such, this thesis proposes a method capable of operating on legacy systems operating in new environments, rather than designing a new waveform to accommodate the specific demands of the user.

1.2 Outline and Contributions

With the clear limitations of current approaches for multi-antenna spectrum-sharing solutions in dynamic wireless environments, the work in this thesis provides a solution to this

problem by:

1. providing an assessment of existing beam-forming methods for the purpose of interference suppression,

2. introducing a novel beamforming method "SVIRA" that combines polyphase filterbank (PFB) channelizers with singular value decomposition (SVD) matrix decomposition for semi-blind signal separation,

3. presenting an objective assessment of parameter trade-space for the proposed algorithm in a variety of scenarios and configurations,

4. verifying the design on a laboratory testbed in a controlled wireless environment

The remainder of this thesis is organized as follows. Chapter 2 first provides the necessary background and mathematical foundation for defining the problem space, providing a review of previous work on this topic. Chapter 3 presents a novel application of almost-perfect reconstruction polyphase filter-bank channelizers with matrix decomposition methods for applying sub-band specific beamforming. Chapter 4 provides an overview of parameter selection and develops an end-to-end simulation for assessing performance of the proposed algorithm enhancements in both AWGN and interference environments. Chapter 5 describes verification of the algorithm in a laboratory environment using software-defined radio hardware. Finally Chapter 6 provides concluding remarks and a discussion of future work.

1.3 Publications

Conference papers

- M. R. Williamson, W. C. Headley, W. H. Clark IV, J. McCollum, T. Krauss, **L. Lusk**, D. Jenkins, T. Villemez, M. O. Moore, D. J. Jakubisin, A. Poetter, A. J. Michaels, A. A. Beex, and J. D. Gaeddert, "Multi-Antenna Pre-processing for Improved RFML in Congested Spectral Environments." *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)* [2].

- **L. Lusk**, and J. D. Gaeddert, "Blind Interference Suppression using Polyphase Filterbank Channelizers and Rank-Reduced Array Processing." *In preparation.*

Chapter 2

Background

This chapter includes a general problem statement and literature survey of potential solutions. Traditional phased array processing is discussed a well as blind beamforming and a few non-beamforming techniques.

2.1 Interference Suppression

The general problem of multi-antenna interference suppression involves detecting the presence of one or more nuisance signals and suppressing their energy while amplifying an intended signal of interest using the spatial separation of sensor elements. Let us first consider a wide-band signal of interest (SOI), represented as a discrete set of K band-limited linear modulated symbols at a baud rate $1/T$, viz.

$$s(t) = \sum_{k=0}^{K-1} a_k g(t - kT) \tag{2.1}$$

where $a_k \in \mathcal{A}$ are a set of uncorrelated and unknown data symbols and $g(t)$ is a band-limited square-root Nyquist pulse-shaping filter. For this problem, it is assumed that all values K, a_k, $\mathcal{A}$, $g(t)$, and T are unknown; however, the power spectral density of $s(t)$ can be assumed to be reasonably flat, and T is assumed to be reasonably wide.

The receiver is a collection of N sensing antennas with an unknown spatial configuration.

The received signal is a delayed version of $s(t)$ with a carrier frequency and phase offset, channel gain (due to path loss), and distorted by a set of interfering signals. The continuous-time representation of the received complex baseband signal on sensor i can be represented as

$$r_i(t) = \gamma s(t-\tau)e^{j\alpha_i} + \sum_p \zeta_p(t)e^{j\beta_{i,p}} + w_i(t), \tag{2.2}$$

where $\zeta_p(t)$ is one of P unknown interfering signals, γ is an unknown but fixed channel gain, and $w_i(t)$ is a wide-sense stationary random Gaussian process with a power spectral density (PSD) $N_0/2$. Note that all of the interfering signals are assumed to be narrow-band and may be spatially separated. Also, SOI is assumed to be uncorrelated with both the interfering signals and noise. Furthermore, while the exact noise PSD is not known, it can be estimated. Finally, the received signal from each sensor then passes through a discrete-time sampler at a rate $F_s = 1/T_s$. The final discrete-sampled time series on sensor i at time index n is thus

$$r_i(n) = r_i(nT_s) \tag{2.3}$$

2.2 Beamforming Methods

The array itself is considered as a set of N digitized sensing elements in three dimensions, each with x, y, and z coordinates in terms of the carrier wavelength λ, viz. $\mathbf{a}_n = [a_n^{(x)}, a_n^{(y)}, a_n^{(z)}]^T$. Therefore, the array configuration can be conceptualized as a $N \times 3$ matrix $\mathbf{a}$ representing the positions of each element's x, y, and z coordinates, respectively, in terms of the carrier wavelength λ. The complex gain w on sensor n of a signal with an incoming azimuthal angle θ off the x/z plane and an elevation angle ϕ off the x/y plane is the projection of $\mathbf{a}_n$ on the plane defined by the pointing vector: a real-valued vector indicating the direction in x, y,

and z for the direction in which the array has maximum gain, viz.

$$\mathbf{p}(\theta, \phi) = \left[\cos(\theta)\cos(\phi),\, \cos(\theta)\sin(\phi),\, \sin(\phi)\right]^T \tag{2.4}$$

The weighting vector which provides maximum gain in this direction is simply the complex phase of the vector projection of the pointing vector on each element's position in 3-D space, evaluated as

$$\mathbf{w}(\theta, \phi) = \exp\Big\{-j2\pi\mathbf{a}^T\mathbf{p}(\theta, \phi)\Big\} \tag{2.5}$$

Traditional phased-array processing would treat the combination of SOI and interfering signals in (2.2) as a linear combination of vectors of length N (which corresponds to the number of sensors in the array) across time and attempts to maximize the output signal-to-interference-plus-noise ratio (SINR) by steering a beam in the direction of SOI as it points nulls in the direction of interferers. An array with N sensors has $N-1$ degrees of freedom, so it is only able to guarantee controlling at most $N-1$ nulls, which means the array can only guarantee to remove at most $N-1$ interfering signals. Additionally, since the goal is to maximize the output SINR, traditional beamformers are unable to distinguish between SOI and interferers coming from the same or similar direction of arrival (DOA). Thus, creating situations in which SOI is either suppressed or interference is let through.

The following sections discuss traditional digital beamforming techniques as well as their limitations. Analogue and hybrid beamforming methods are not considered as the array is presumed to be a digital phased array.

2.2.1 MVDR (Capon)

The minimum variance distortionless response (MVDR) beamformer aims to maximize the SINR in the direction of SOI. Assuming the precise array configuration and DOA of SOI are known a priori, MVDR is considered an optimal solution [3]. This type of beamforming creates weights that minimize the amount of interference and noise while maintaining a distortionless response in the DOA of SOI; thus maximizing the SINR of the output. However, if the knowledge of SOI's DOA and the array configuration is incomplete or imprecise, the performance will degrade as SOI is treated as interference instead of as the desired signal. Similarly, if either the DOA of SOI or the array configuration is unknown, MVDR cannot be used.

The algorithm to implement MVDR, as described by [4], is as follows: At any instant k, the output of the beamformer $y(k)$ can be represented as

$$y(k) = \mathbf{w}^H \mathbf{r}(k) \tag{2.6}$$

where $\mathbf{r}(k)$ is the received signal at instance k, $(\cdot)^H$ indicates the Hermitian transpose, and $\mathbf{w}_{\text{MVDR}}$ is a vector of weights determined by the algorithm. As stated previously, the purpose of MVDR is to maximize the SINR of the input signal, which is

$$\text{SINR} \triangleq \frac{\text{E}\left[|\mathbf{w}_{\text{MVDR}}^H \mathbf{s}|^2\right]}{\text{E}[|\mathbf{w}_{\text{MVDR}}^H (\mathbf{i}+\mathbf{n})|^2]} = \frac{\sigma_s^2 |\mathbf{w}_{\text{MVDR}}^H \mathbf{p}(\theta,\phi)|^2}{\mathbf{w}_{\text{MVDR}}^H \mathbf{R}_{i+n} \mathbf{w}_{\text{MVDR}}} \tag{2.7}$$

where $\mathbf{p}(\theta,\phi)$ is the pointing (steering) vector defined in (2.4), σ_s^2 is the power in SOI, $\mathbf{i}$ and $\mathbf{n}$ are the interference and noise vectors respectively, and $E[\cdot]$ signifies the expectation. $\mathbf{R}_{i+n}$

is a matrix of the expectation of the covariance of the interference-plus-noise.

$$\mathbf{R}_{i+n} \triangleq \mathrm{E}\Big[(\mathbf{i}(k) + \mathbf{n}(k))(\mathbf{i}(k) + \mathbf{n}(k))^H\Big] \tag{2.8}$$

With the SINR as defined, the maximization problem is such that the interference and noise is minimized and the input signal is fixed. This can be expressed as the following optimization problem

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w} \quad \text{such that} \quad \mathbf{w}^H \mathbf{p}(\theta, \phi) = 1 \tag{2.9}$$

whose solution is

$$\mathbf{w}_{\mathrm{MVDR}} = \alpha \mathbf{R}_{i+n}^{-1} \mathbf{p}(\theta, \phi) \tag{2.10}$$

where $\alpha = 1/\mathbf{p}^H(\theta, \phi)\mathbf{R}_{i+n}^{-1}\mathbf{p}(\theta, \phi)$ and $(\cdot)^{-1}$ indicates matrix inversion.

While this beamforming method is optimal under certain conditions, it has two main issues: first, the success of the beamformer is heavily determined by the accuracy of the assumed array configuration and DOA of SOI; secondly, this method is limited to $N - 1$ degrees of freedom with which to successfully null interference. Therefore, if there are more than $N - 1$ interfering sources, additional processing must be used.

To date, most of the literature focuses on mitigating the first issue and suggests different approaches in order to obtain a more accurate DOA. An approach using Bayesian statistics is proposed in [5]. This approach derives the posterior probability density function (PDF) of the DOA of SOI by knowing a priori the PDF of the desired DOA. Afterwards, the weights are determined using the posterior PDF of the DOA. While this approach is able to produce better results than the classic MVDR, it is still dependent upon some a priori knowledge of SOI's DOA.

On the other hand, some approaches, such as the one proposed by [6], do not require any knowledge of the DOA. This approach uses fractional Fourier transforms to estimate the time-frequency curve of the received signal then derives the steering vector from the estimated curve. After obtaining the steering vector, traditional MVDR beamforming is implemented. One of the most popular DOA estimation techniques is multiple signal classification (MUSIC).

MUSIC

The MUSIC beamformer is a subspace based algorithm that relies on the eigenvectors of the covariance matrix of the input signals to separate the signal from the noise subspace and offers better angular resolution than MVDR. Since beamforming relies on the eigenspace of the covariance matrix, the performance degrades when any of the received signals are correlated [7][8][9]. As with MVDR, MUSIC assumes knowledge of the array configuration and is limited by its degrees of freedom, but unlike MVDR, MUSIC requires knowing the number of signal sources (including both SOI and interference). The primary drawbacks of MUSIC are that precise knowledge of the array configuration is required as well as knowledge or an accurate estimation of the noise-plus-interference covariance matrix. If the array geometry changes or the assumed knowledge is not accurate enough, the performance degrades. Similarly, if the statistical properties of the noise or interference changes during transmission, MUSIC will not be able to form an optimal beam. As such, this method is not considered in this thesis. Instead, blind beamforming methods are primarily investigated.

2.2.2 Blind Beamforming Methods

Blind beamforming methods assume no knowledge of the DOA of the signals (both the SOI as well as the interfering signals). Blind methods seek to adapt the beam weights to isolate the signal of interest whose characteristics may still be unknown; however, the array configuration and phase alignment of the elements are still known *a priori*.

Assuming the general direction of SOI is known, a method utilizing a network of frequency-invariant beamformers is proposed in [10] for receiving wideband signals in multipath environments. Each frequency-invariant beamformer points its main beam in a different direction in such a way that all of the directions of interest are covered by the network. Afterwards, the output of the network undergoes blind source extraction by applying weights and summing to get SOI, which is a simplified version of blind source separation in which only SOI is extracted. While this method does not rely on knowing the exact DOA of SOI, it does require a general knowledge of SOI's DOA. Also, in order to successfully extract SOI using blind source extraction, identifying statistical characteristics of SOI must be known to calculate the appropriate weights. As such, the computational complexity of this algorithm is variable as it is determined by how narrow of an area must be covered by the frequency-invariant beamforming network.

A different blind approach leverages the non-Gaussian distribution of SOI to approximate its DOA [11]. Afterwards, the resulting weight vector is applied to a minimum power distortionless response beamformer, which extracts SOI. While effective, this approach will not work if SOI does have a Gaussian distribution or if the interference has a non-Gaussian distribution, and it also still relies on perfect knowledge of the array configuration in order to beamform.

CMA

The constant modulus algorithm (CMA) beamforming process exploits the constant-modulus properties of SOI that possess modulation schemes such as MSK, and FSK. If SOI does not have a constant modulus, this method is not as effective. This technique is considered an adaptive beamformer, which means the weights are iteratively updated and convergence speed is a concern. The weights are determined by the cost function

$$\min_{\mathbf{w}} \mathbf{J}_{p,q}(\mathbf{w}) = \mathcal{E}\left[\left|\mathbf{r}_m^H \mathbf{w}\right|^p \iota\right]^p \tag{2.11}$$

where $\mathbf{w}$ is a vector of beamforming weights, $\mathbf{r}$ is the vector of the input signal, $p > 0$, $q > 0$, and ι is the modulus of SOI. Typically, the cost function is minimized through the use of stochastic gradient descent and the mean squared error, and $p = 1$ and $q = 2$ for optimal convergence [12]. However, while convergence is optimized for classic CMA, the convergence speed is still relatively slow.

A normalized CMA approach is proposed in [13]. The modified weight update equation coupled with initializing the weight vector improves the convergence speed as demonstrated by the simulated results performed by [14] of [13]'s algorithm. Another way to improve the convergence is to reduce the computational complexity. By employing frequency domain interpolation and spatial domain clustering, [15] is able to reduce the complexity of a recursive least-squares CMA for orthogonal frequency-division multiplexing (OFDM) signals. A hybrid approach that combines recursive least-squares (RLS) CMA and least mean-squares (LMS) CMA is proposed in [16]. This algorithm is optimized for use on a uniform linear array and attempts to balance the robustness against steering angle error, associated with LMS-CMA, with the convergence speed and good interference cancellation of the RLS-CMA.

SCORE

The spectral self-coherent restoral (SCORE) beamformer uses cyclostationary properties of SOI to create a weight vector that is used to beamform a reference signal. Like CMA, SCORE is an iterative process, so the initialization of the weight vector is important for fast convergence. This algorithm is well-liked because the only required knowledge is the cyclic frequency of SOI, which can be obtained from the symbol rate or carrier frequency. However, just like CMA, the classic SCORE algorithm is slow to converge. One of the commonly used methods with a faster convergence is the Cross-SCORE algorithm, which trades speedy convergence for higher complexity by solving for the weight vector using eigenvalue decomposition [17]. Another method, proposed by [18], reduced the complexity of the classic least-squared SCORE by guaranteeing a real-valued weight vector. In a different approach, the authors of [19] decided to initialize the weight vector using the dominant mode prediction (DMP) algorithm. By doing so, they were able to increase the convergence speed and use the multi-target SCORE to extract multiple SOIs in parallel. Although SCORE is a useful beamforming method, it is not viable if SOI possesses no cyclosationary features or the features are otherwise removed during transmission.

Eigen-Based Approaches

Another group of blind beamformers is the eigen-based beamformers, which typically assume that interference is uncorrelated. These types of beamformers decompose a covariance matrix into its eigenvalues and eigenvectors to identify a composite steering vector (CSV), which with to steer the beamformer. A basic eigen-based approach is proposed by [20], where the interference covariance matrix (made of only the interference and noise) is decomposed using eigenvalue decomposition. The eigenvector corresponding to the largest eigenvalue is then

assigned as the CSV. Afterwards, any traditional beamforming method, such as MVDR, can be used to form the beam. An eigen-based beamformer is also considered to improve beamforming for multipath environments in [21].

Instead of decomposing a single interference covariance matrix, this approach categorizes the interference into strong and mild. Spatial spectrum techniques are used to estimate the covariance matrix of the mild interference, and the covariance matrix of the strong interference is estimated using the eigenvalues of the sample covariance matrix. These estimations are used to reconstruct the interference-plus-noise covariance matrix, which is then subtracted from the sample covariance matrix. With the interference and noise removed, the principal eigenvector of the sample covariance matrix is used as the CSV.

The blind beamformer proposed by [22] performs eigenvalue decomposition on a truncated version of SVD of the received signal, and using the decomposed eigenvalues and eigenvectors, all of the signals can be separated. By truncating the SVD, this algorithm is able to reduce the noise and problem space, which is a principle that the SVIRA proposed in this thesis attempts as well. Nevertheless, despite not relying upon knowledge of the DOA, most of the eigen-based beamformers still require knowledge or computation of the interference covariance matrix. It is also important to note that if coherent signals do exist, this will reduce the rank of the source covariance matrix, which in turn renders the majority of the eigen-based beamformers ineffective. In those cases, which can be typical of multipath environments, additional processing must be employed such as spatial smoothing as seen in [23]. Finally, all of these beamforming algorithms are still limited to being able to only null as many interfering signals as there are degrees of freedom.

Deep Learning Approach

An additional approach with gaining popularity is using deep learning methods to determine the optimal beamforming weight vector. A few reasons to account for the growing popularity includes the ability to run parallel processing to suppress wideband interference, and removing the reliance on DOA. A convolutional neural network (CNN) is used to compute the optimal weight vectors in [24] in order to reduce the computational complexity when faced with a large number of antennas and filter taps. The CNN is able to estimate nearly optimal weight vectors even with and signal-to-noise ratio (SNR) as low as -5 dB and as low as 400 snapshots of SOI. Another approach bypasses the traditional calculation of MVDR weights and has a deep neural network determine them instead [25][26]. While these approaches offer solutions for many issues associated with traditional and blind beamfroming (precise knowledge of DOA, knowledge of the interference covariance matrix, slow convergence, etc.), deep learning introduces new concerns such as the development of training data. As such, deep learning approaches are not considered in this thesis.

2.3 Related Work

One non-beamforming method for interference suppression is successive interference cancellation (SIC) as described in [27]. Generally speaking, if two or more signals are received concurrently, the strongest (highest SINR) is demodulated and decoded then subtracted from the aggregate received signal. The resulting residual only contains the weaker signal or signals. If there were more than two signals in the aggregate received signal, then the process is repeated with the residual until only the weakest signal is left. Since this process is successive, it has the potential to perfectly remove interfering signals; however if one of the signals is estimated incorrectly, all of the subsequently estimated signals will also be

incorrect and the error will grow with each iteration. This means that if the interference cannot be estimated (as is the case with uncorrelated noise-like interference sources), SIC methods are not effective and can actually degrade performance. As such, these techniques are not considered in this thesis.

Another non-beamforming method used for interference suppression is space-time adaptive processing (STAP). This type of processing is particularly useful for frequency-selective channels, can combat multipath fading, and suppress both intersymbol interference (ISI) and co-channel interference (CCI) [28]. As defined by [29], the primary goal of STAP is to increase the dimensions of the covariance matrix while reducing the spatial correlation between the steering vectors of the received signals. In the paper, the authors propose a STAP approach (based on MVDR) to estimate the DOA of a particular signal by introducing time delay taps at each sensor, and thus expanding the dimensions of the received data matrix. This method is proven in simulation to have higher spatial resolution than MVDR for waveforms with a low SNR (-5 dB) and an array with few degrees of freedom (8 elements). However, since STAP is a type of adaptive processing, the time it takes to converge can be a concern. Additionally, as the data rate, delay profile, or number of sensors increases, the complexity of STAP increases sharply. One method to combat the increasing complexity is to implement STAP on sub-bands. By introducing sub-bands, STAP can be implemented in parallel, converge rapidly, and have a reduced processing complexity [30]. While the issue of increased complexity and slow convergence can be mitigated by using sub-bands, the method provided by [30] is only proven under the assumption that all of the received signals (including both SOI and interferers) are widesense stationary and that that the number of signals present is less than the number of sensors in the array.

2.4 Proposed Approach

In order to address the issue of having too few degrees of freedom, a method of dividing the spectrum into sub-bands is considered. As long as there are not more interfering signals than there are degrees of freedom within each sub-band, any of the previously mentioned beamforming methods can appropriately steer the beam and null out the interference. A possible method of dividing the band is through the use of a fast Fourier transform (FFT). FFTs allow for perfect reconstruction and the frequency resolution can be adjusted depending on the number of bins allowed. However, because FFTs use rectangular windows, they offer poor frequency isolation. Due to its shape, the window only provides about 13 dB of suppression of the neighboring bins, which can cause aliasing from bin to bin. To ensure ample suppression of neighboring sub-bands, a channelizer with perfect reconstruction capabilities is used in the proposed SVIRA.

2.5 Summary

This chapter has provided an assessment of different interference-suppression techniques, many of which rely on more knowledge about the array configuration, calibration, or element interaction than can be reasonably assumed for this problem space. Furthermore, existing techniques make assumptions about knowledge of the SOI's DOA, the number and characteristics of interference sources, and assume no correlation between the signals present in the band. Finally, deep-learning methods, while promising, rely on substantial training data and, to date, are not generally applicable to problem spaces to which they have not been trained. Chapter 3 introduces the methodology for the proposed SVIRA processing.

Chapter 3

Proposed Methodology

In this chapter, we propose the overall approach to SVIRA presented in this work to suppress numerous interfering signals impacting a desired SOI. This chapter is organized as follows: In Section 3.1, the PFB channelizer pair (analysis and synthesis) are described to decompose a wide-band spectrum into near-orthogonal components with the almost-perfect reconstruction property. Section 3.2 describes matrix decomposition using SVD with a discussion for its application to signal separation. Section 3.3 provides a discussion of SOI enhancement using the Rayleigh quotient (RQ) and its application to phased-array processing. Section 3.4 contains a description of the various order of operations possible for this processing in addition to a comparative discussion of how MVDR can be applied to the SVIRA process.

3.1 Polyphase Filterbank Channelizer

3.1.1 Overview

An array with N sensors has $N-1$ degrees of freedom for suppressing interference signals in the band. For environments where the number of nuisance signals exceeds N, the array will not be able to effectively suppress this interference and will perform poorly; however by taking advantage of the fact that nuisance signals might not overlap in frequency (see assumptions in Section 2.1), the received signals may be partitioned into orthogonal spectral

sub-bands for further analysis. This reduces the number of nuisance signals seen by any one sub-band, allowing for better spectral discrimination. Once the interference removal has been completed, the wide-band signal must be reconstructed from its interference-mitigated constituents. Perfect reconstruction for a function $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ are for the output to match its input, outside of a scale and time shift, viz.

$$\hat{x}(n) = \mathcal{F}^{-1}\Big(\mathcal{F}\big(x(n)\big)\Big) \tag{3.1}$$

$$= cx(n-k) \tag{3.2}$$

for a constant c and fixed delay k [31, p. 132]. There are several possibilities for this processing. Perhaps the first that comes to mind is the FFT and its inverse which easily satisfies 3.2; however the spectral performance of a FFT is very poor, exhibiting a $1/f$ rolloff and only a 13 dB side-lobe suppression with neighboring channels.

We propose the use of a PFB channelizer as a method to decompose the spectrum into narrow-band orthogonal components. PFB channelizers can perform this computation efficiently when certain constraints are met (e.g. even spacing between the channels, identical resampling rate for each of the channels) through the use of a single prototype filter and FFTs. By allowing the output rate to exceed the maximally decimated rate, and with care taken in designing the prototype filter, the output of the analysis channelizer can be fed into its complementary synthesis channelizer to provide an almost-perfect[1] reconstruction of the input. Over-sampling by a factor of two allows for a simple architecture for processing signals. To make the channelizer oversample by a factor of 2, the analysis filter bank takes in $M/2$ samples instead of the usual M. [32] With sufficient care to the selection of channelizer parameters, both the number of nuisance signals observed within any one sub-band is within

[1] In general, true perfect reconstruction is not possible with finite impulse response channelizers using a prototype filter; however, a near perfect reconstruction with a slight delay and amplitude variation can be achieved by oversampling.

the degrees of freedom of the sensor array to suppress, and the number of samples on the output in each sub-band provides sufficient information for SVD factorization.

3.1.2 Analysis Polyphase Filterbank Channelizers

The analysis channelizer takes the received signal from the ith antenna element $r_i(n)$ and outputs M signals $R_i(n,m)$, where m is the channel band number ($m = 0, 1, ..., M-1$), using an analysis polyphase filter bank. It is so named because it "analyzes" the spectral components of $r_i(n)$. Say a polyphase filter's impulse response is as defined in [33, (11.10.10)]:

$$h_i(n,m) = h_0(nD - m, m), \quad m = 0, 1, ..., M-1 \tag{3.3}$$

where D is the decimation factor and h_0 is the impulse response of the prototype filter. In this thesis, the prototype filter is a sinc function with a Kaiser-Bessel window with a semi-length of 12 and sidelobe-suppression of 60 dB. Based on the polyphase filter impulse response, the decimated input sequence can be expressed as

$$r_i(n,m) = r_i(nD + m, m) \quad m = 0, 1, ..., M-1 \tag{3.4}$$

Recall that the output of a general uniform discrete Fourier transform (DFT) filter bank is

$$R_i(k,m) = \left[\sum_n r_i(n,m) h_0(kD - n, m) e^{j2\pi m(kD-n)/M} \right] e^{-j2\pi kmD/M} \tag{3.5}$$

where k is the index of the decimated samples and $R_i(\cdot)$ denotes the frequency domain. Notice that summation inside the bracket computes the DFT of the filtered $r_i(n,m)$, and the exponential shifts the resulting DFT down to complex baseband. Using (3.5), the output

of the polyphase filter is

$$R_i(k,m) = \sum_{m=0}^{M-1} \left[\sum_l h_i(l,m) r_i(k-l,m) \right] e^{-j2\pi nmD/M} \tag{3.6}$$

for $m = 0, 1, ..., M-1$. Observe how the summation inside of the brackets represents the convolution of $r_i(l,m)$ with the impulse response $h_i(l,m)$, and the summation outside of the brackets computes the DFT of the filtered $r_i(l,m)$. As seen by the bounds of the outer summation, this does indeed result in the desired M samples. It is important to note that for SVIRA proposed in this paper, the analysis filter bank outputs at $2/M$ the input rate or twice that of a maximally-decimated channelizer ($D = M/2$).

3.1.3 Synthesis Polyphase Filterbank Channelizers

Designing a synthesis polyphase filter is a similar process to designing an analysis polyphase filter but with all of the processing done in reverse. The synthesis filter is so named because it is combining the individual components and "synthesizing" a wide-band signal. First, the inverse DFT must be taken. Then, the signals are up-sampled through an interpolation filter. Finally, all of the signals are combined back into one. This process can be expressed mathematically in a uniform DFT case as given by [33, (11.10.9)]

$$v_i(n) = \frac{1}{M} \sum_{k=0}^{M-1} \left\{ \sum_m \left[Y_i(k,m) e^{j2\pi kmI/M} \right] g(n-mI,m) \right\} \tag{3.7}$$

where I is the interpolation factor, which is equal to D, and $g(n,m)$ is impulse response of the interpolation filter

$$g(n,m) = g_0(n) e^{j2\pi nm/M} \tag{3.8}$$

where $g_0(n)$ is the impulse response of the prototype filter. For the polyphase case, a similar impulse response for the interpolation filter is determined

$$q_i(n,m) = g_0(nI + m) \quad m = 0, 1, ..., M-1 \tag{3.9}$$

noting that the impulse response of the synthesis prototype filter $g_0(n)$ does not necessarily relate to that of the analysis prototype filter $h_0(n)$. Since $D = I = M/2$ in this paper, (3.7) can be rewritten as

$$v_i(n,l) = \sum_m q_i(n-k,l) \left[\frac{1}{M} \sum_{k=0}^{M-1} Y_i(k,m) e^{j2\pi ml/M} \right] \tag{3.10}$$

$$= \sum_k q_i(n-k,l) y_i(k,l) \tag{3.11}$$

for $l = 0, 1, ..., M-1$. The simplification between (3.10) and (3.11) is possible by recognizing that the portion within the brackets in (3.10) is the inverse DFT of $Y_i(k,m)$, and thus everything in the brackets can be replaced by $y_i(k,l)$, which is the inverse DFT of $Y_i(k,m)$. For more information on polyphase filters see [33].

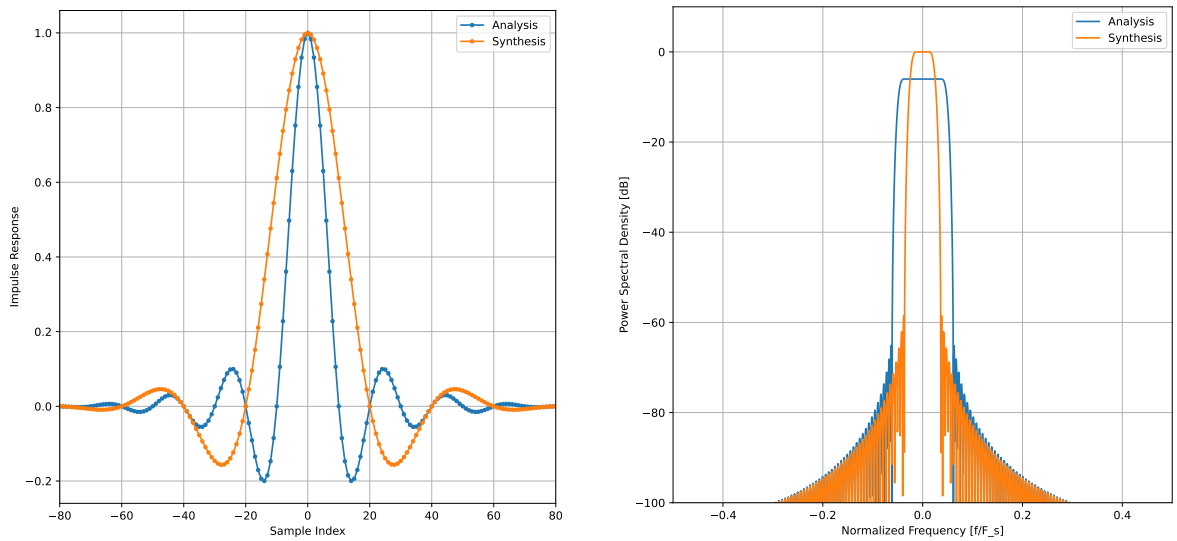
3.1.4 Almost-Perfect Reconstruction

Satisfying the requirement in (3.2) is possible with PFB channelizers with $M > 2$, but only when individual band-pass filters are used for each channel [31]. Perfect reconstruction is not possible with DFT-based PFB channelizers with $M > 2$; however *almost*-perfect reconstruction is possible by eliminating aliasing between sub-channels by

1. suppressing non-adjacent channels, and

2. cancelling aliasing between the overlapping pass-bands of adjacent channels.

The first requirement can be achieved with any number of low-pass filters. The second requirement, while perhaps not as obvious, can be achieved by designing a Nyquist filter and over-sampling the channelizer output [32]. This yields $\hat{x}(n) \approx cx(n-k)$ from (3.2) where the error can be controlled by the filter design.



(a) Impulse response of $h_0(n)$ and $g_0(n)$ (b) Spectral response of $h_0(n)$ and $g_0(n)$

Figure 3.1: PFB channelizer response with $M = 20$ channels and a prototype filter semi-length of 4, and a stop-band suppression of 60 dB

An example of a channelizer prototype filters' impulse and spectral responses are shown in Figure 3.1, above, for $M = 20$ channels. Notice that because the analysis channelizer is over-sampled by a facter of 2 on its output, its prototype filter has twice the bandwidth of that of the synthesis filter. Specifically, the pass-band cut-off frequency for h_0 is F_s/M while the pass-band cut-off for g_0 is $F_s/2M$.

Once the signal is channelized, the outputs $R_i(k,m)$ undergo digital processing—either SVD reconstruction or MVDR—to remove the interference. Afterwards, the next step is to synthesize the N newly processed signals $Y_i(k,m)$ back into a single signal $v_i(n)$.

3.2 SVD Reconstruction

One way to remove the nuisance signals is to take advantage of their relatively high power spectral density in comparison to SOI. This is achieved by taking the SVD of a block of samples from the ith sensor and m^{th} sub-band, removing the high powered singular values, and reconstructing the sample block.

The SVD of any matrix $\mathbf{A}$ can be represented as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H \tag{3.12}$$

where $(\cdot)^H$ denotes the Hermitian transpose operation. If $\mathbf{A}$ is an $n \times m$ matrix with rank r, then $\mathbf{U}$ is a $n \times r$ matrix, and $\mathbf{V}$ is a $m \times r$ matrix. The diagonal matrix Σ is an $r \times r$ matrix, whose diagonals are the singular values (σ). The singular values are strictly positive and are in descending order of energy (e.g. $\sigma_1 > \sigma_2 > ... > \sigma_{r-1} > \sigma_r$). Another way of viewing the singular values is as the positive square roots of the eigenvalues of $\mathbf{A}^H\mathbf{A}$.

The SVD of $\mathbf{A}$ can also be expressed as

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^H \tag{3.13}$$

where $\mathbf{u}_i$ is the ith column vector of $\mathbf{U}$, and $\mathbf{v}_i$ is the ith column vector of $\mathbf{V}$ [34].

Another way to consider $\mathbf{A}$ is as a linear combination of basis functions. According to [35], the first r columns of $\mathbf{U}$ are an orthonormal basis for the column space of $\mathbf{A}$. Likewise, the first r columns of $\mathbf{V}$ are an orthonormal basis for the row space of $\mathbf{A}$ while the last $m - r$ columns of $\mathbf{V}$ are an orthonormal basis for the null space of $\mathbf{A}$.[2]

[2] In the case of digital phased arrays, the null space is really represented as the noise space of the observation across antenna elements.

After decomposition, the high powered nuisance signals are associated with the low ordered (high-powered) singular values, and noise is related to the highest order (lowest power) singular values. For the proposed SVIRA, two thresholds are used to remove noise and interference. As only the singular values that fall between the two thresholds are kept, the upper threshold is chosen to remove the interfering signals while the lower threshold is chosen to remove the noise. The recomposed matrix $\tilde{\mathbf{A}}$ is then

$$\tilde{\mathbf{A}} = \sum_{i \in \mathcal{R}} \sigma_i \mathbf{u}_i \mathbf{v}_i^H \tag{3.14}$$

where $\mathcal{R}$ is the set of singular value indices to be kept. If selected appropriately, $\tilde{\mathbf{A}}$ should contain SOI, minimal noise, and no interference. Therefore, choosing appropriate threshold values is important in order to maximize SOI and limit interference. The threshold selection process is discussed in greater detail in Section 4.

3.3 Signal of Interest Enhancement

To increase the chance of decoding SOI, SOI is enhanced by combining the sensors in such a way as to maximize the coherent gain in matrix $\mathbf{A}$. This can be achieved using the following optimization:

$$\mathbf{c} = \arg\max_{\tilde{\mathbf{c}}} \left(\tilde{\mathbf{c}}^H \mathbf{A}^H \mathbf{A} \tilde{\mathbf{c}} \right) \tag{3.15}$$

where $\mathbf{c}$ represents the columns of $\mathbf{A}$, which contains the data for each sensor. To reach a nontrivial solution (i.e. prevent $\|\mathbf{c}\| \to \infty$), c is constrained such that $\|\mathbf{c}\| = 1$. With this constraint in place, (3.15) can be solved using the Rayleigh quotient as follows:

$$\lambda_{\min}(\mathbf{A}^H \mathbf{A}) \leq \frac{\mathbf{c}^H \mathbf{A}^H \mathbf{A} \mathbf{c}}{\mathbf{c}^H \mathbf{c}} \leq \lambda_{\max}(\mathbf{A}^H \mathbf{A}) \tag{3.16}$$

where λ_k represents the kth eigenvalue of the square matrix $\mathbf{A}^H\mathbf{A}$. The lower bound of (3.16) is achieved by the eigenvector corresponding to the minimum eigenvalue of $\mathbf{A}^H\mathbf{A}$. Likewise, the upper bound is achieved by the eigenvector corresponding to the maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$ [36].

In this way, the coherent gain of the columns of $\mathbf{A}$ is maximized by using the coefficients that maximize the Rayleigh quotient as weights to combine the columns of $\mathbf{A}$, which relates to the sensors. To ensure that the gain is continuous from frame to frame, the phase of each sensor is adjusted before combination. That is to say, the elements of the weighting vector $\mathbf{c}$ are allowed to be complex, but the solution to (3.15) is therefore unique only to a phase ambiguity. In essence, if $\hat{\mathbf{c}}$ is a solution to (3.15), then so is $\hat{\mathbf{c}}e^{j\theta}$ for all angles θ. Computation of the eigenvalues for (3.16) using computer systems often results in a phase ambiguity that can be problematic when combining finite-length observations. As such, the ambiguity is resolved by rotating all elements of $\mathbf{c}$ by the same phase such that $\arg\{c(0)\} = 0$.

Although the Rayleigh combination is shown in Fig. 3.3 as being performed before synthesis, it can also be performed after synthesis. The merits of performing this type of combination before or after synthesis is discussed in Chapter 4.5.

3.4 Configuration Order

The three main components—channelizer, SVD reconstruction, and combination—can be arranged in different configurations to make up SVIRA. This section details two configurations—combination after synthesis and combination before synthesis—as well as a channelizer version of MVDR, which is used later as a point of comparison in Chapter 4.

3.4.1 Combination After Synthesis

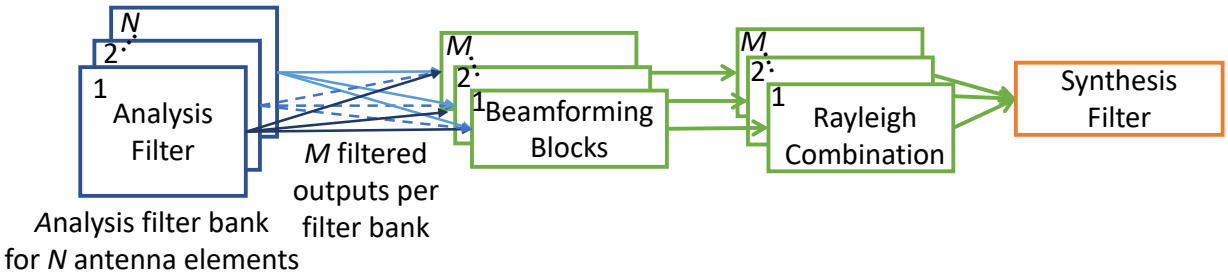
This configuration is characterized by combining the sensor data after synthesis and interference removal. The general procedure for an N-sensor array is as follows:

1. The data are passed through N identical PFB analysis filter banks with M channels.
2. Each of the M filtered outputs across the N sensors are then processed using SVD reconstruction to remove the interference.
3. The resulting data are synthesized using a bank of M synthesis filters.
4. Finally, the data are combined across sensors using the Rayleigh quotient.

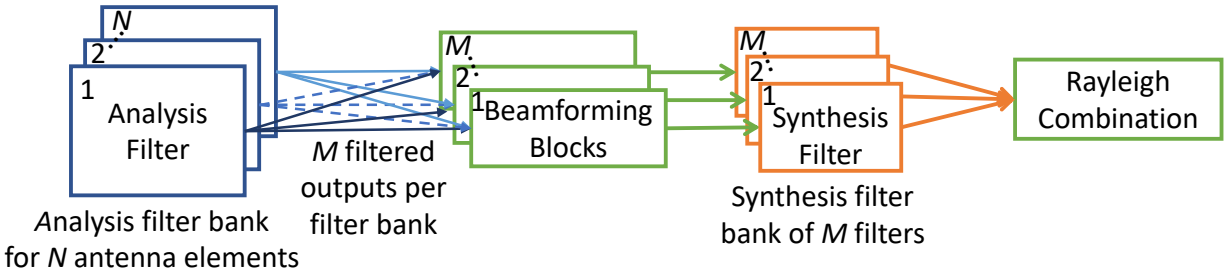
A block diagram of the general procedure is outlined in Fig. 3.2b, where the beamforming block contains the SVD Reconstruction Block from Fig. 3.3.

A more detailed explanation of the process is as follows:

1. The complex valued matrix $\mathbf{R}$ has the dimensions $N \times k$, where N is the number of antenna elements (or sensors), and k is the number of samples in the buffer.
2. The matrix $\mathbf{R}$ is filtered through an analysis filter bank defined by a M-band channelizer, whose output is oversampled by a factor of 2. This process outputs a sequence of matrices $\mathbf{C}_m$ for $m = 0, 1, 2, ..., M-1$, where m corresponds to the band number. Each matrix $\mathbf{C}_m$ has the dimensions $N \times 2L$, where $2L$ is the block size after decimation.
3. Each matrix $\mathbf{C}_m$ is windowed by a periodic, L-point Hann window into framed segments, which have a segment overlap of 50% and sample length L. That is to say, for every sensor c_i with the indices $1 : k$, the newly framed counterpart is $c_{f,i} = c_i[fL : (f+1)L - 1]$, where f is the frame index, which continues to iterate until it segments



(a) Combination After Synthesis



(b) Combination Before Synthesis

Figure 3.2: Block diagram of SVIRA Combination Before and After Synthesis Configurations

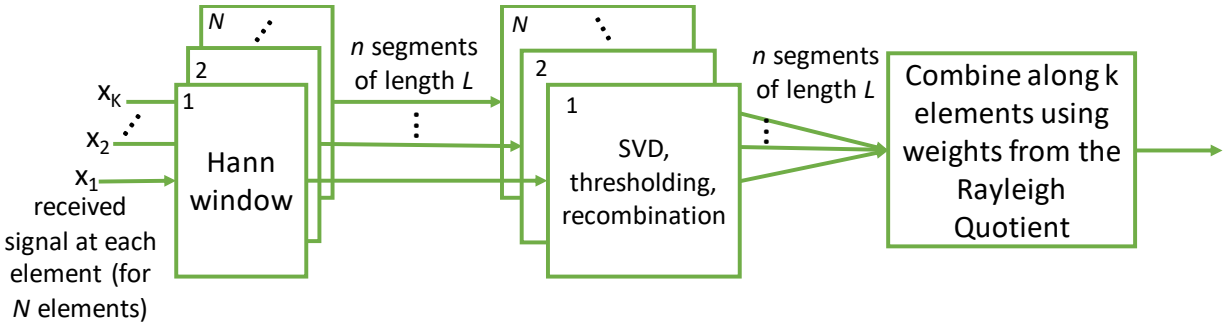


Figure 3.3: SVD Reconstruction beamforming block

the entire matrix $\mathbf{C}_m$. The total number of frames is heavily dependent upon L and the amount of overlap.

4. For each frame, the resulting matrix $\mathbf{A}_m$ – for band number $m = 0, 1, 2, ..., M-1$ – has the size $L \times N$. As noted by its size, $\mathbf{A}_m$ contains the framed data from all N sensors.

5. The matrix $\mathbf{A}_m$ is then processed – and interfering signals are removed – using SVD reconstruction.

6. The frames are linearly combined to reconstruct each $\hat{\mathbf{C}}_m$, which has the same shape as $\mathbf{C}_m$ ($N \times 2L$). Given the type of window and the amount of overlap, $\mathbf{C}_m$ would be perfectly reconstructed in the time domain if singular values had not been removed from $\mathbf{A}_m$. As such, the newly reconstructed $\hat{\mathbf{C}}_m$ is $\mathbf{C}_m$ with the nuisance signals mostly, if not completely, removed.

7. Each $\hat{\mathbf{C}}_m$ is synthesised through its own synthesis filter (M synthesis filters in total) to produce the matrix $\hat{\mathbf{R}}$, which has the shape $N \times k$.

8. The data from all of the sensors in $\hat{\mathbf{R}}$ is combined, using weights defined by the Rayleigh Quotient, into a single vector $\tilde{\mathbf{r}}$, which is k samples long.

9. Analyze the probability of packet error by attempting to decode the packet in $\tilde{\mathbf{r}}$.

3.4.2 Combination Before Synthesis

Another configuration is Combination Before Synthesis, which combines across sensors before using the synthesis filter to recombine across frequency. This configuration follows the same general procedure as Combination After Synthesis until step 6. Instead, the procedure—as detailed in Fig. 3.2a—is as follows:

6. The frames are linearly combined to reconstruct each $\hat{\mathbf{c}}_m$, which is now a vector. Given the type of window and the amount of overlap, $\mathbf{C}_m$ would be perfectly reconstructed if $\mathbf{C}_m$ had not been processed. As such, the newly reconstructed $\hat{\mathbf{c}}_m$ is $\mathbf{C}_m$ linearly combined across sensors with the nuisance signals mostly, if not completely, suppressed.

7. The vectors $\hat{\mathbf{c}}_m$ are synthesised through the channelizer's synthesis filter to produce the vector $\tilde{\mathbf{r}}$, which should be $2L$ samples long.

8. Analyze the probability of packet error by attempting to decode the packet in $\tilde{\mathbf{r}}$.

3.5 MVDR

Assuming the array configuration and DOA of SOI are known, MVDR beamforming can be used to enhance SOI and remove interference. This beamformer creates weights that minimize the amount of interference and noise while maintaining a distortionless response in the DOA of SOI; thus maximizing the SINR of the input. A more detailed description of MVDR can be found in Section 2.2.1. However, since MVDR requires knowledge of SOI's DOA and the array configuration, it is not always feasible to implement.

As a point of comparison, MVDR is used in conjunction with the channelizer. The block diagram for this process can be seen in Fig. 3.4. MVDR is used in place of SVD reconstruction, and since MVDR inherently combines across antenna elements, Rayleigh combination is not needed. Therefore, the modified process, starting at step 5 is as follows:

5. Perform MVDR on the matrix $\mathbf{A}_m$. The result is the vector $\hat{\mathbf{a}}_m$ of length $2L$.

6. The frames are then linearly combined to reconstruct $\tilde{\mathbf{r}}$.

7. Analyze the probability of packet error by attempting to decode the packet in $\tilde{\mathbf{r}}$.

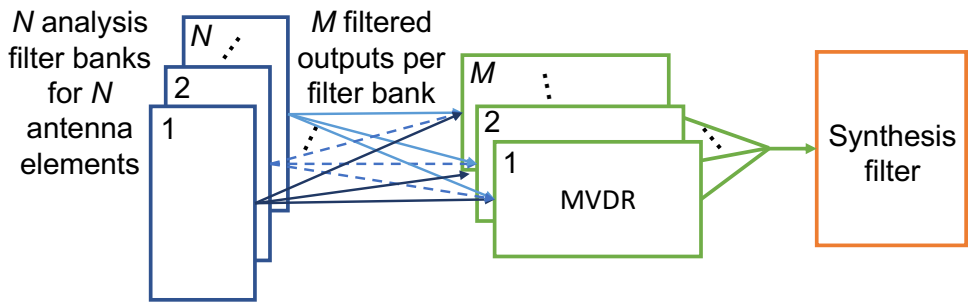


Figure 3.4: Block diagram of the MVDR process with a PFB channelizer

3.6 Summary

This chapter discusses the three main components of SVIRA—channelizer, SVD reconstruction, and SOI enhancement—in detail and presents two configuration methods: Combination Before and Combination After Synthesis. It also details a process combining MVDR with a channelizer as a point of comparison to evaluate the performance of SVIRA. While this chapter identifies the different parameters for the SVIRA, parameter selection is the subject of the next chapter (Chapter 4).

Chapter 4

Parameter Selection and Simulation

In this chapter, the selection of appropriate parameters for processing signals is discussed, and an overview of a representative waveform suitable for both simulations as well as wireless experimentation is provided. Additionally, an analysis of the trade-space with parameters of the SVIRA algorithm is discussed.

4.1 Representative Waveform

In order to provide a qualitative performance comparison of different algorithms, parameters, and channel environments, it is desirable to use a waveform which has as a flat PSD to decode. While bit error rate measurements for linearly-modulated signals with known timing and carrier offsets are sufficient for simulation, it is the intent of this work to demonstrate the efficacy of the proposed approach in a wireless environment on laboratory commercial off-the-shelf hardware. This implies that the performance of a waveform's ability to synchronize in the presence of interference is of critical importance to the analysis of algorithms.

The intent of the work in this thesis is to provide a general interference suppression capability to existing receivers and not to define a custom waveform for which the proposed front-end processing would be required to function properly. This would be counterproductive to a generalized approach and would imply that the process put forth in this work would have only a narrow application; however, the definition of a *representative* waveform for which to

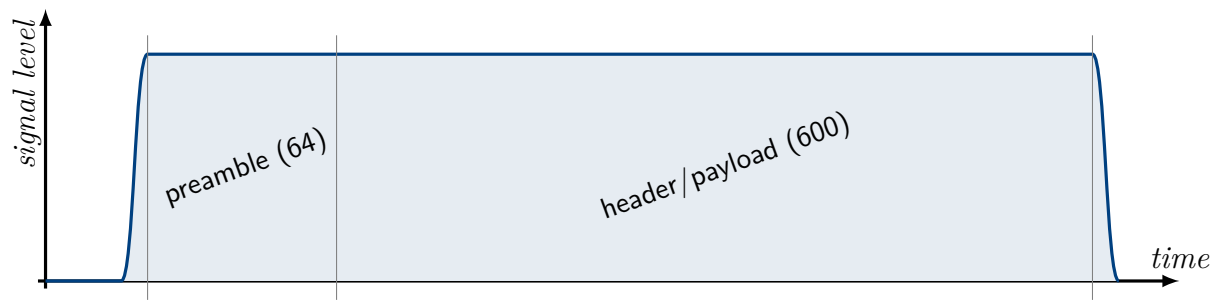


Figure 4.1: Framing structure used for evaluating proposed interference-suppression approach

test this approach proves useful and is described below.

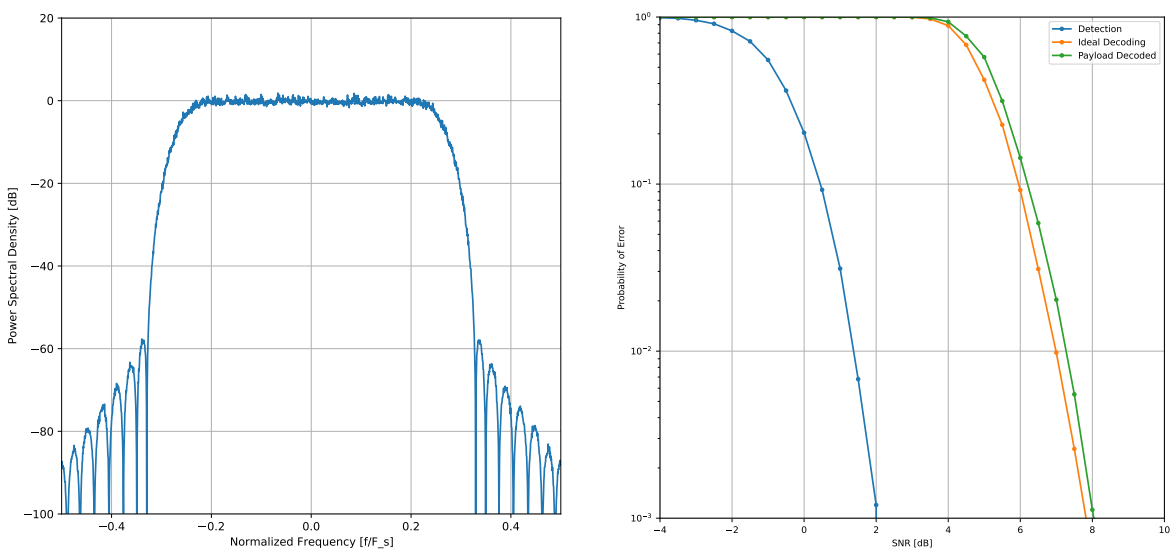
4.1.1 Waveform Overview

The chosen waveform has fixed parameters and its data are protected with r1/2 Golay(24,12) coding and a 24-bit cyclic redundancy check. The preamble is a 64-symbol P/N sequence with good auto-correlation properties, and the header (or payload) consists of 72 bytes of user-defined data, which is modulated to 600 QPSK symbols (e.g. $\mathcal{A} = \exp(j\pi\{\frac{1}{4}, \frac{3}{4}, -\frac{1}{4}, -\frac{3}{4}\})$ as used in (2.1)) with 30 pilot symbols added for precise carrier phase and frequency recovery. Additionally, a square-root Nyquist filter is used for both the transmit and receive sides. With padding to account for filter transients, each frame is 720 symbols, yielding a spectral efficiency of exactly 576 bits / 720 symbols = 0.8 b/s/Hz. The waveform receiver correlates against the preamble and provides a coarse carrier and timing offset estimate, applies a matched filter, recovers the pilot symbols, performs maximum-likelihood carrier phase and gain recovery on the payload, demodulates, decodes, and passes results to the program asynchronously.

The waveform and its receiver are implemented in C with Python bindings; the source code can be found in [37]. It is important to note that the pre-processor does *not* have *a priori*

knowledge of the waveform (e.g. the preamble or pilots); only knowledge that its PSD is flat. As such, the waveform could be substituted with any waveform with a flat PSD, and the pre-processor would provide similar results. A graphical depiction of the waveform can be seen in Figure 4.1.

4.1.2 Waveform Performance



(a) Power Spectral Density of framing structure used for experimentation

(b) Frame detection and decoding performance in AWGN

Figure 4.2: Performance of framing structure used for experimentation in AWGN channels

The spectral representation of the waveform and its performance in AWGN can be seen in Figure 4.2. Note that the power spectral density is reasonably flat over the passband and that the side-lobes are suppressed by nearly 60 dB from the main signal's passband, indicated in Figure 4.2a. Furthermore, Figure 4.2b indicates that the frame can be reliably detected (10^{-3} missed detect rate) at just 2 dB SNR. With ideal decoding (no channel impairments such as timing, carrier frequency/phase offsets, and perfect gain correction), packets can be reliably decoded at more than 8 dB SNR. When the frame synchronizer is required to

estimate and correct for channel impairments, the performance loss is minimal: less than 0.05 dB worse than the ideal case.

4.2 Simulation Setup

In this section, the data, parameters, and performance metrics used in simulation are detailed. The array is a 2-dimensional, uniform circular array with a spacing of 0.3 λ between elements, where λ is wavelengths. It is important to note that the array does not have to be in this specific geometry for the SVD reconstruction process to work. Theoretically, the proposed SVD reconstruction method could work with any array configuration as it does not use the configuration information.

Interference sources are represented as tones; each interferer is generated at a random frequency and transmitted from a random angle in azimuth. For the purpose of simplicity, interference from various points in elevation is not considered. All signals are transmitted from an elevation of 0°. It is important to note that this SVD based approach is general enough that a different location in the plane or in elevation would have the same type of effect on the various sensor signals.

Since the goal of this process is to improve an array's ability to successfully receive and decode a packet, packet validation—whether or not the packet is successfully decoded—is the basic performance metric. Determining the number of valid packets transmitted at various SNR values provides a basis for packet error rate (PER) curves. These PER curves can then be compared in decibels (dB) against the limiting ideal case, in which there is no interference. Additionally, the required SNR for a 1% PER is used to compare the performance of different parameters such as the number of channels M in the channelizer, block size $2L$, etc.

4.3 Relative Thresholds

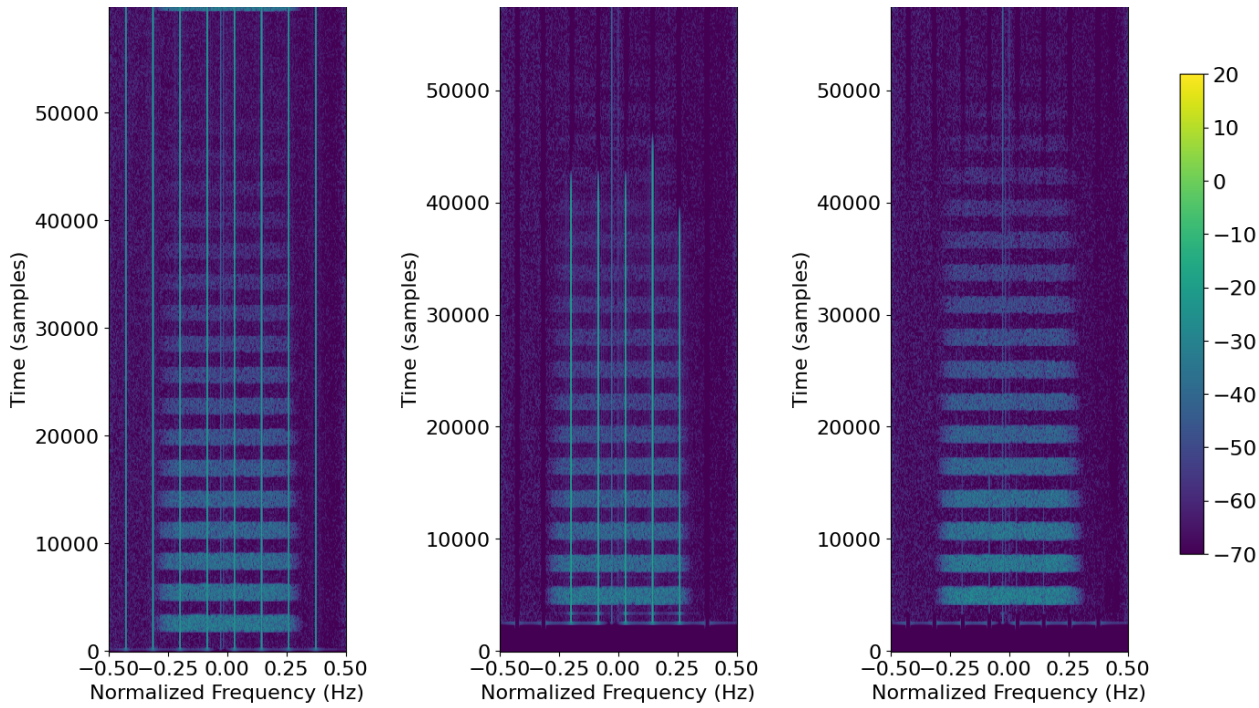
While absolute thresholds work when the range of singular value (SV)s is known, it is inconvenient and inefficient to calculate them all before setting the thresholds. This calculation must be performed not only for every new packet and interference combination, but it is also necessary for every change of channelizer parameters (i.e. number of channels and block size). Therefore, a relative threshold selection process is preferred. Since the lowest—weakest powered—SV is indicative of the noise floor, it can be used as a reference point for the upper threshold. For example, if the absolute thresholds are 10 dB and 30 dB, with a noise floor of 9 dB, then the relative thresholds could be 0 dB and 20 dB. Since the thresholds are relative, they are more robust to changes in channelizer parameters, and the range is mainly dependent on the SNR of the packet.

However, an issue with relative thresholds occurs when there are more signals present in a sub-band than degrees of freedom. In this case, the lowest SV represents the signal as well as the noise subspace, which can cause the SV to fluctuated wildly. As such, observing singular values across frequency is introduced to combat this type of fluctuation.

4.4 Observation across Frequency

In order to compensate for fluctuations in the lowest singular value, SVIRA observes SV magnitudes across frequency in order to set its thresholds. All of the singular values are computed for a given block of samples. Then the tenth percentile of the lowest singular value across all frequencies is taken as the reference point. This results in flat thresholds across frequency, which in turn allow for proper suppression of the nuisance signals.

For example, examine Fig. 4.3b. The high SNR packets at the output are heavily notched—



(a) Input (b) Output (c) Output Using Averaging

Figure 4.3: Captured data from File 19; $M = 100$, $2L = 64$, SVD thresholds from -1 dB to 15 dB

to the extent that they cannot be decoded. Notice that the five notched bands correspond to five high powered, interfering tones from the input. Thus, it can be surmised that the algorithm is taking an overly aggressive approach to removing the interference. It cannot successfully distinguish between the packet and interference in those bands.

The reason for the aggressive removal becomes evident when looking at the singular values from Fig. 4.4. Observe the SVs for channel 0 and channel 2. Channel 0 only contains SOI and noise whereas channel 2 has SOI, noise, and an interfering tone. Looking at Fig. 4.4, channel 0's SV 0 is indicative of SOI, and its SV 1 is representative of the noise subspace. However, channel 2's SV 0 is indicative of the high powered, interfering tone, and its SV 1 is representative of SOI. Since the thresholds are relative to lowest singular value, notice how the high powered portion of the SOI subspace is removed from channel 0 and part of

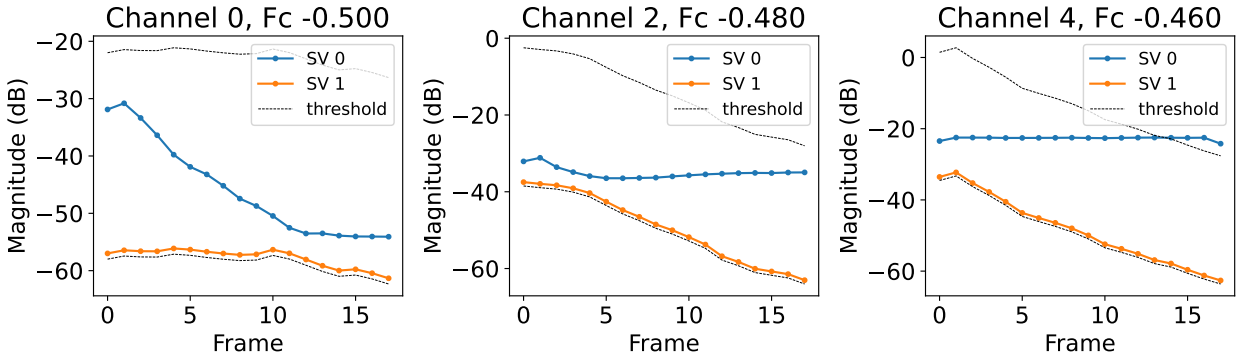


Figure 4.4: Singular values plotted across time (frame number)

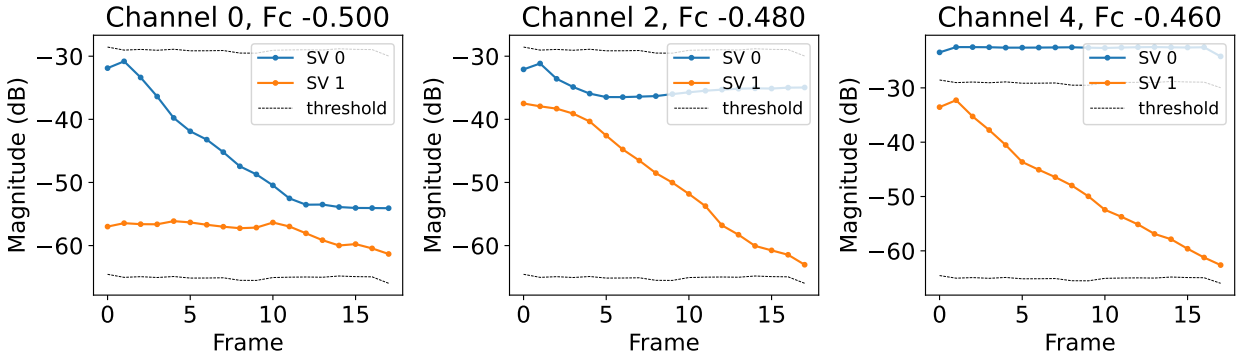


Figure 4.5: Singular values averaged over frequency plotted across time (frame number)

the interfering signal sneaks through the first four frames of channel 2. Both of these issues combined explain why the high SNR packets cannot be decoded.

In order to successfully remove all of the SVs related to the interference, thresholds that are constant from frame to frame are desired. Using the tenth percentile of the lowest singular values across all frequencies as a reference point, flat thresholds are achieved as shown in Fig. 4.5. Unlike before, none of the SVs associated with the interference are within the threshold bounds, and thus will be removed with recombination. The resulting spectrum is shown in Fig. 4.3c, where the interfering tones are removed.

4.5 Synthesis before/after SVD combining

As discussed in Section 3.4, PFB synthesis can occur either before or after SVD combination. The question is whether the application of the Rayleigh quotient in (3.16) is more effective at recovering the SOI before or after synthesis. If the combination takes place before synthesis, the process is similar to MVDR and only requires one synthesis filter. However, there is a coherence loss associated with this configuration. Although the channelizer is designed for near perfect reconstruction, it does not take combining across antenna elements into account. Therefore, if a bank of synthesis filters is used to first reconstruct the received data on each element, and the data is combined across sensors after synthesis, then the coherence of the signals is better maintained; thus providing a better result.

The difference in performance between the two configurations is seen in Fig. 4.6. The performance bound for 2 antennas is created by receiving simulated packets over an AWGN channel without any interference present at a single receiver. Then the packet error rate is improved by an additional 3 dB due to the performance gain from using a two-sensor array. Note that while the Combination Before Synthesis case has a PER that performs

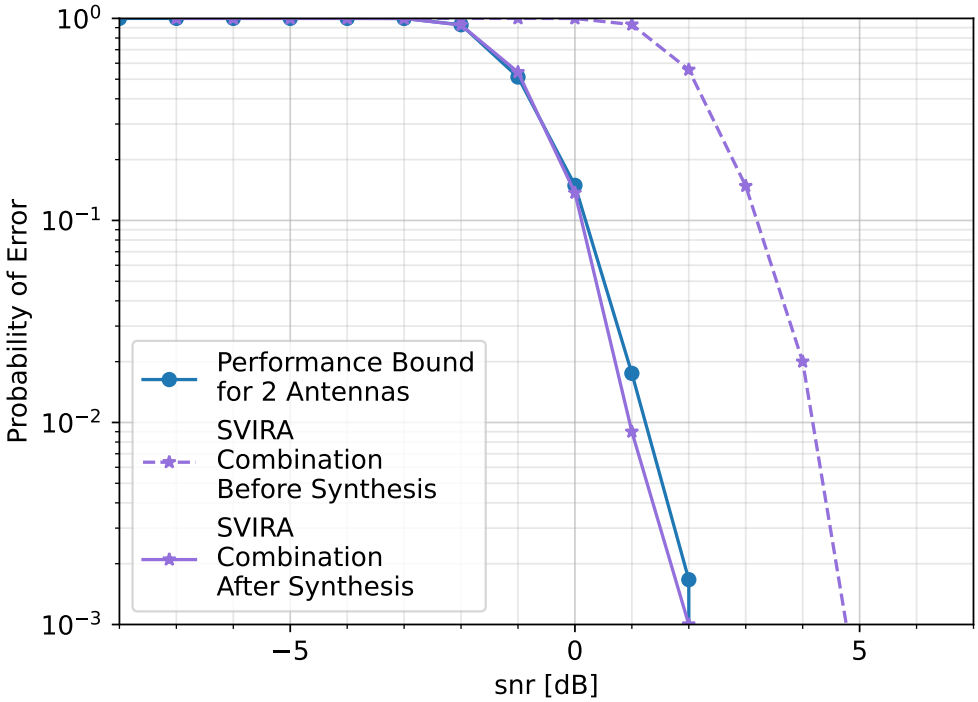


Figure 4.6: Combination Before vs. After Synthesis SVIRA with a 2-sensor array

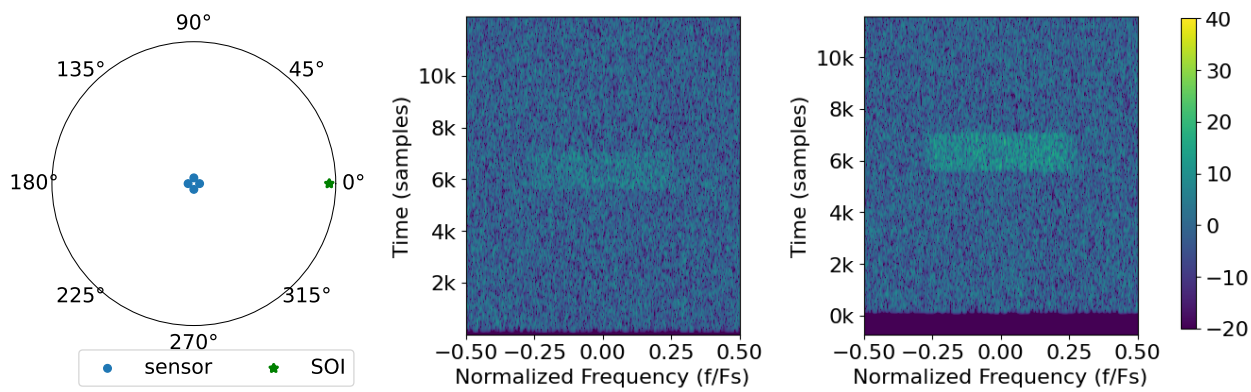
3 dB worse than the performance bound, the Combination After Synthesis case lies directly on the performance bound itself. Clearly, combining the sensor data after the synthesis filter is the better performer out of the two configurations.

The Combination After Synthesis method can outperform the Combination Before Synthesis method because it can make decisions based on more information. In the combination before synthesis case, the Rayleigh quotient only has $2L$ samples to make a decision on, which means it is able to maximize the energy for that particular $2L$ segment. However, since SOI spans multiple blocks, the enhanced blocks might not be fully maximized as the weights are found using fewer samples than are actually in the frame. Therefore, if sensor combination takes place after synthesis, the Rayleigh quotient can be found for the entirety of SOI all at once. In this way, the energy of SOI is ensured to be maximized and thus provide the best chance for decoding.

4.6 Single Packet Examples

This section investigates the performance of SVIRA on a single packet in five scenarios: (1) no interference, (2) one interference source, (3) a different single interference source, (4) two interference sources, and (5) three interference sources.

4.6.1 Scenario 1: No Interference



(a) Array Configuration (b) Input (c) Output

Figure 4.7: Scenario 1: example of the Combination After Synthesis process operating on a single frame without interference; SNR = 3 dB, $M = 100$, $2L = 64$, and SVD thresholds set as -10 dB to 80 dB

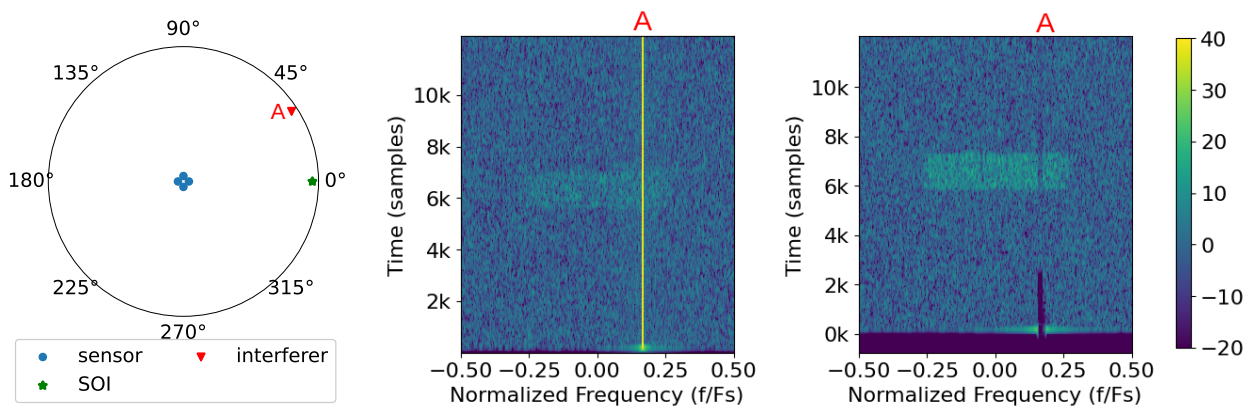
As an example, Fig. 4.7a shows the configuration of a circular array with four sensors. The SOI is set to arrive from 0° in azimuth. Fig. 4.7b shows a spectrum waterfall plot of the data received at the first of the four elements with an SNR of 3 dB. Notice that this SNR is not high enough as to decode a frame, according to the PER performance in Fig. 4.2b. After undergoing channelization, SVD reconstruction, and combining across sensor elements, SOI is clearly visible in Fig. 4.7c and successfully decoded. The following observations can be made from Fig. 4.7:

1. the SOI is enhanced by about 6 dB by coherently combining the antenna elements as

made evident by its higher power level in the output plot;

2. there is a processing delay in the output incurred by both the PFB channelizer as well as the windowing function, visible in the output plot.

4.6.2 Scenario 2: One Interference Source



(a) Array Configuration (b) Input (c) Output

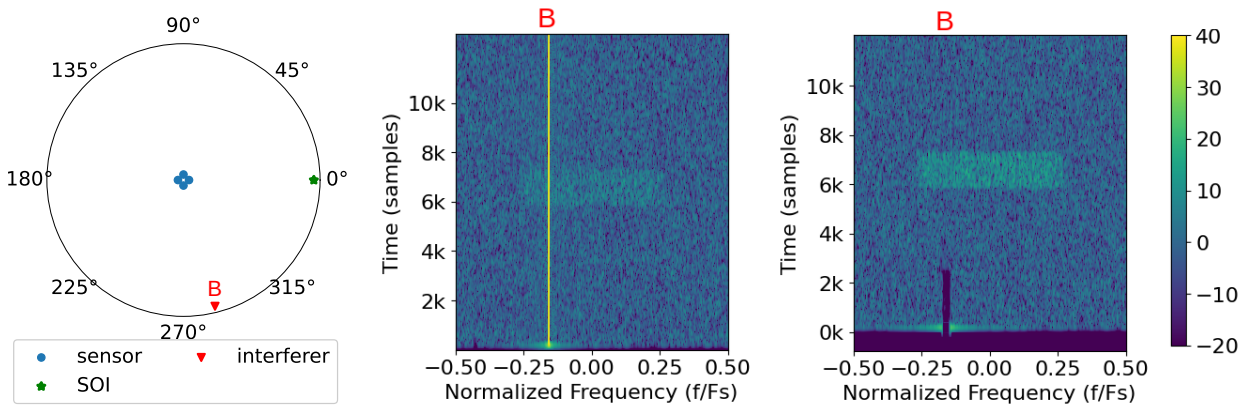
Figure 4.8: Scenario 2: example of the Combination After Synthesis process operating on a single frame with one interferer with an SNR of 10 dB; SNR of the packet is 3 dB, $M = 100$, $2L = 64$, and SVD thresholds set as -1 dB to 15 dB

Using the same array configuration as in Scenario 1 (Section 4.6.1), an interfering signal A is introduced. Along with SOI, a tone with an SNR of 10 dB is set to arrive at an azimuth of 33° as seen in Fig. 4.8a. The spectrum as received by the first antenna element is show in Fig. 4.8b, where the interferer is shown to have a center frequency of 0.166 F_s. Note that since the interference is high powered, SOI cannot be decoded until the interference is removed. After undergoing the SVIRA processing, the interference has been removed, as shown in Fig. 4.8c, and the packet is successfully decoded. The following observations can be made from Fig. 4.8:

1. SOI experiences similar a 6 dB enhancement as in Scenario 1 (Section 4.6.1);

2. due to how close the DOA of the interference is to the DOA of SOI, the power of SOI is lowered in the particular sub-band where the interference is removed.

4.6.3 Scenario 3: One Interference Source



(a) Array Configuration (b) Input (c) Output

Figure 4.9: Scenario 3: example of the Combination After Synthesis process operating on a single frame with one interferer with an SNR of 10 dB; SNR of the packet is 3 dB, $M = 100$, $2L = 64$, and SVD thresholds set as -1 dB to 15 dB

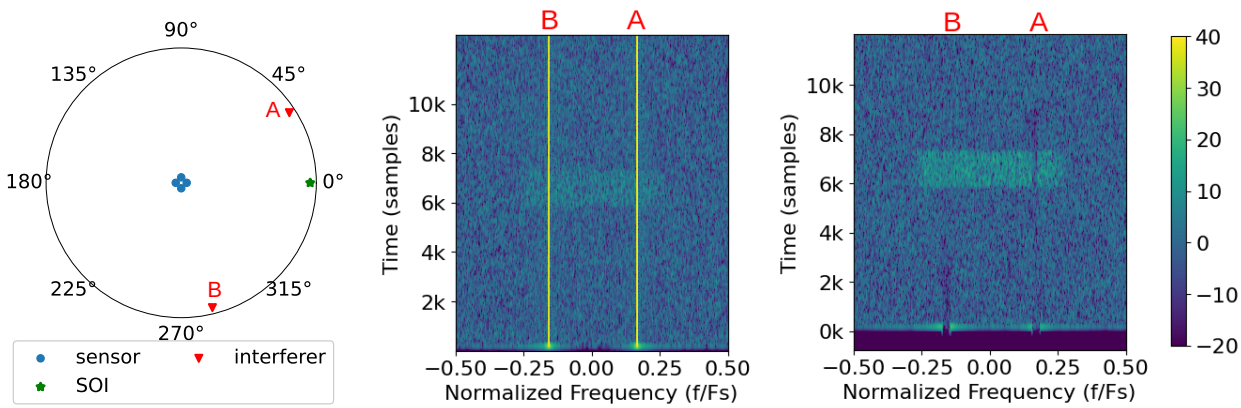
This time, an interfering signal impinges the circular array at an angle of 284° as seen in Fig. 4.9a. The spectrum as seen by the first sensor is displayed in Fig. 4.7b, which shows the interfering signal with a center frequency of -0.157 F_s and SNR of 10 dB. Once again, the interference must be removed in order to successfully decode SOI. After undergoing the Combination After Synthesis configuration of the SVIRA, the interference is cleanly removed from the spectrum as depicted in Fig. 4.8c. The following observations can be made from Fig. 4.9:

1. once again, SOI experiences the same 6 dB enhancement as when there was no interference (see Scenario 1), and SOI is successfully decoded.

2. unlike in Scenario 2 (Section 4.6.2), SOI does not show signs of reduced power in

the same band where the interference was removed. This is due to the DOA of the interference being sufficiently removed from that of SOI, so they do not share singular values (or similar eigenspace). Thus, the interference can be removed without visibly impairing the power of SOI.

4.6.4 Scenario 4: Two Interference Sources



(a) Array Configuration (b) Input (c) Output

Figure 4.10: Scenario 4: example of the Combination After Synthesis process operating on a single frame with two interferers with an SNR of 10 dB; SNR of the packet is 3 dB, $M = 100$, $2L = 64$, and SVD thresholds set as -1 dB to 15 dB

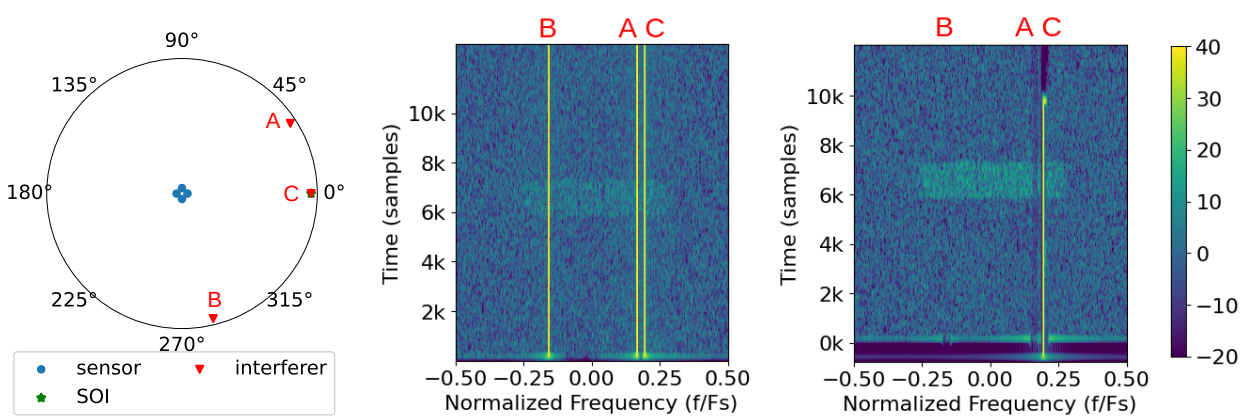
Consider the scenario in which the same array encounters both interference A and B as described in Scenarios 2 and 3 (Sections 4.6.2 and 4.6.3). The array configuration for this scenario is shown in Fig. 4.10a, in which SOI arrives from 0° and interference A and B arrive from 33° and 284° respectively. As seen in the spectrum from the first sensor in Fig. 4.10b, both high powered interfering tones impede the decoding of SOI. After applying the SVIRA, the output spectrum in Fig. 4.10c shows all of the interference is successfully removed, and SOI can be properly decoded.

The following observations can be made from Fig. 4.10:

1. like before, SOI is enhanced by about 6 dB and is successfully decoded;

2. both interference A and B are successfully removed because they are frequency separable and appear in different sub-bands of the channelizer. Notice that like before, interference A degrades some of the power of SOI while interference B does not. This indicates that the algorithm is able to successfully operate independently on each sub-band, and though both interference A and B are present in this scenario, they are treated the same as they are when each is the only interfering signal.

4.6.5 Scenario 5: three interference sources



(a) Array Configuration (b) Input (c) Output

Figure 4.11: Scenario 5: example of the Combination After Synthesis process operating on a single frame with three interferers with an SNR of 10 dB; SNR of the packet is 3 dB, $M = 100$, $2L = 64$, and SVD thresholds set as -1 dB to 15 dB

Next, consider a scenario when there are 3 interfering signals: A, B and C. As shown in Fig. 4.11a, the array configuration is the same as before as well as the DOAs of SOI and interference A and B. The new interfering signal C arrives a the array from the same direction as SOI (0° in azimuth). Like the other interfering signals, it also has an SNR of 10 dB. Interference C also has a normalized center frequency of 0.195 F_s, which can be seen

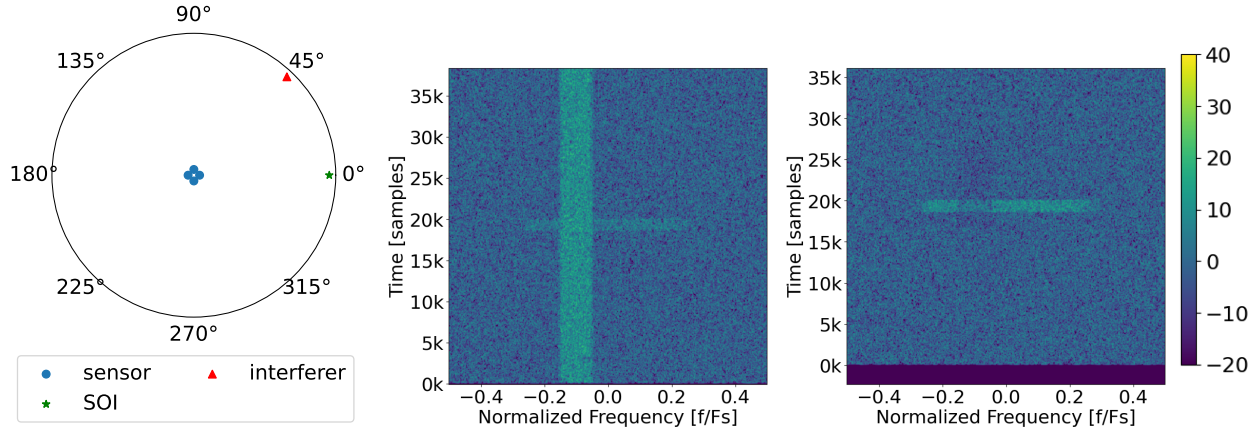
along with the other interferences and SOI in the spectrum received at the first sensor in Fig. 4.11b. After undergoing the SVIRA, the output spectrum in Fig. 4.10c shows interfering signals A and B are successfully removed while interfering signal C is reduced in power but not removed. As such, SOI cannot be decoded. The following observations can be made from Fig. 4.11:

1. the majority of SOI experiences similar levels of enhancement as in the previous scenarios; however, because of interference C, it cannot be decoded;

2. as before, interference A and B can be successfully removed with minimal impact on SOI, but interference C cannot. While all of the interfering signals are frequency separable and thus can be processed relatively independently, because interference C comes from the same DOA as SOI, the interference cannot be separated from SOI. This is due to them both sharing the same eigenspace and thus singular values. The only way in which to potentially decode SOI is to allow the sub-band containing interference C to be notched out, which will introduce some ISI and performance degradation.

4.6.6 Scenario 6: narrow-band noise interference sources

In addition to removing tones, SVIRA is able to remove narrowband AWGN. Consider a scenario where there is an AWGN signal that occupies 10% of the band and arrives at the previously established array from 45° as shown in Fig. 4.12a. As in previous scenarios, SOI has an SNR of 3 dB and a DOA of 0°, and the interference has an SNR of 10 dB. The narrowband noise has a normalized center frequency of -0.100 F_s, which can be seen in the spectrum from the first sensor, Fig. 4.12b, along with SOI. After using SVIRA, the narrowband noise is successfully removed, and SOI can be decoded. The following observations can be made from Fig. 4.12:



(a) Array Configuration (b) Input (c) Output

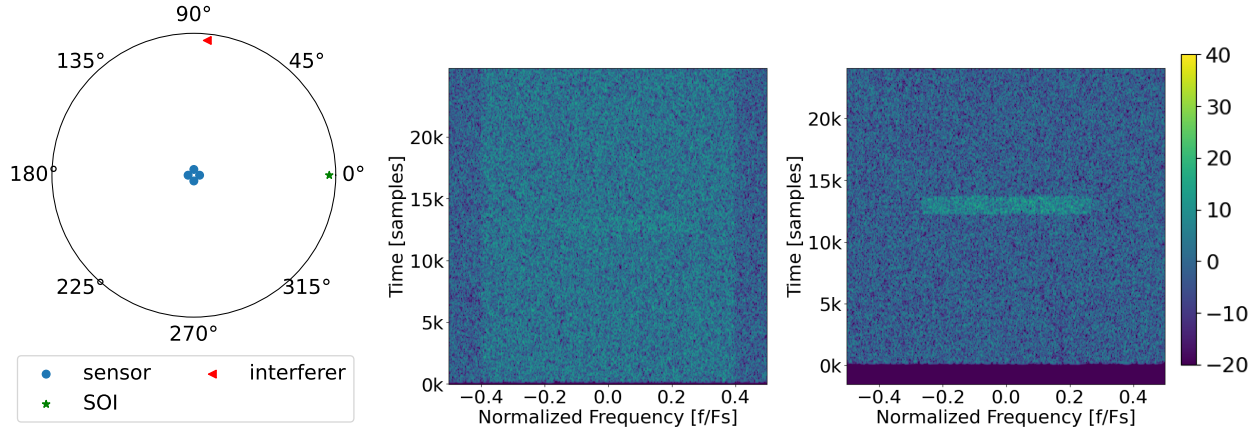
Figure 4.12: Scenario 6: example of the Combination After Synthesis process operating on a single frame with a narrowband interferer with an SNR of 4 dB; SNR of the packet is 3 dB, $M = 300$, $2L = 64$, and SVD thresholds set as -1 dB to 7 dB

1. overall, the majority of the SOI has a 6 dB improvement;

2. the narrowband noise interference can be successfully removed, but like in Scenario 2 (Section 4.6.2), in the sub-bands where the narrowband noise interference used to be, SOI is slightly reduced in power because of the DOA of the interference is close enough to SOI's DOA that SOI and the interference share singular values and have similar eigenspaces.

4.6.7 Scenario 7: wideband noise interference sources

Finally, consider a scenario when there is wideband interference. As shown in Fig. 4.13a, the array configuration is the same as before as well the DOA of SOI, and the wideband interference arrives at 84° azimuth. Like in the previous examples, SOI has an SNR of 3 dB, but unlike previous examples, the interference has as an SNR of 5 dB. From the spectrum as seen by the first sensor in Fig. 4.13b, the interference is shown to be centered in frequency and occupy 80% of the band. After applying SVIRA to the samples, the spectrum output



(a) Array Configuration (b) Input (c) Output

Figure 4.13: Scenario 7: example of the Combination After Synthesis process operating on a single frame with a wideband interferer with an SNR of 5 dB; SNR of the packet is 3 dB, $M = 200$, $2L = 64$, and SVD thresholds set as -1 dB to 8 dB

displayed in Fig. 4.13c reveals SOI, which can be decoded, without any of the wideband interference. The following observations can be made from Fig. 4.13:

1. once again, the SOI is shown to have the same 6 dB enhancement as in previous scenarios;

2. the wideband interference can be successfully removed with minimal disturbance to SOI as it arrives from an entirely different direction than SOI.

4.6.8 Discussion

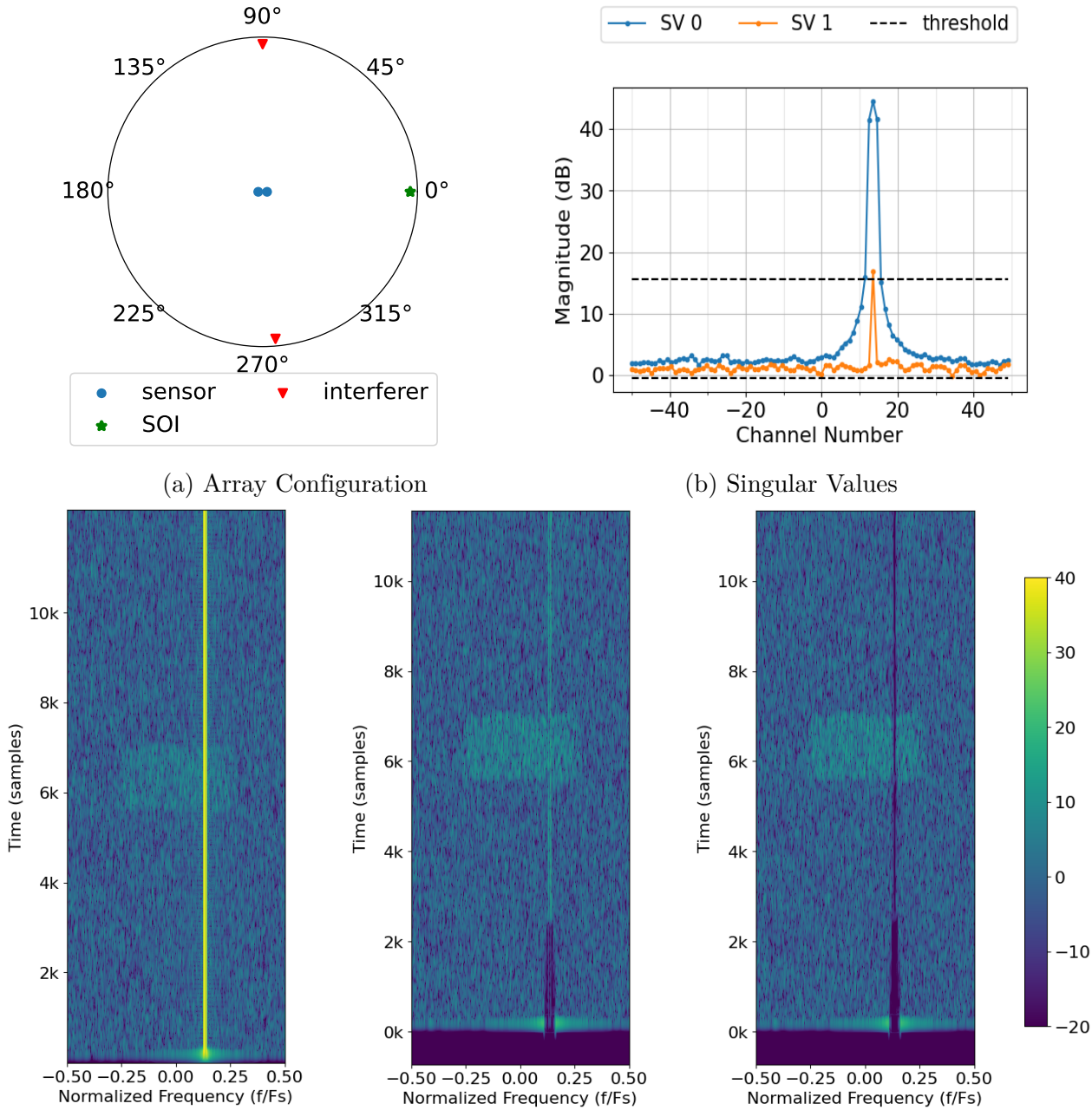
It is clear from the previous examples that there are several inherent limitations with the system as defined. In the event that the source of interference falls in the same eigenspace as the SOI, the beam at that sub-channel cannot simultaneously suppress the interference without suppressing SOI. As such, a different mitigation strategy must be employed which would minimize loss to the receiver. In this case, the spectral content within this narrow sub-

channel can be removed, effectively notching the band which is impacted by the interference. This is discussed in Section 4.7, below. Furthermore, the choice of SV selection is investigated in Sections 4.3 and 4.4. Finally, the choice for PFB synthesis before or after SVD combining is discussed in Section 4.5.

4.7 Notching

As exemplified by Scenario 5 in the previous section, one issue to consider is when interference comes from the same direction as SOI. In this case, SOI and the interference share eigenspace, and due to the nature of the SVD, the interference will appear in the same singular value as SOI. Therefore, a trade-off between introducing a notch in SOI or allowing a certain amount of interference to remain must be considered. In the case of Scenario 5, it is better to notch SOI and accept a small amount of ISI; otherwise, SOI cannot be decoded. Furthermore, the amount of ISI introduced by notching one sub-band in that particular case is not enough to prevent successful decoding from happening.

Another case to consider for notching is when multiple interfering signals are not frequency separable, come from different DOAs, and when there are not enough degrees of freedom to remove them. In this case, a two-sensor array is considered as shown in Fig. 4.14a. SOI has a DOA of 0°; interference A arrives from 90°, and interference B arrives from 275°. The spectrum as observed by the first sensor is displayed in Fig. 4.14c. As in previous scenarios, SOI has an SNR of 3 dB, and both interfering signals have an SNR of 10 dB. Interference A has a normalized center frequency of 0.14 f/Fs, and interference B has a center frequency of 0.13 f/Fs. Since the two interfering signals are so close in frequency, they are indistinguishable from each other in the spectrum plot. Being so close in frequency, the two interfering signals are not separable using a channelizer with 96 sub-bands.



(a) Array Configuration

(b) Singular Values

(c) Input

(d) Output of SVIRA with no notching

(e) Output of SVIRA with notching

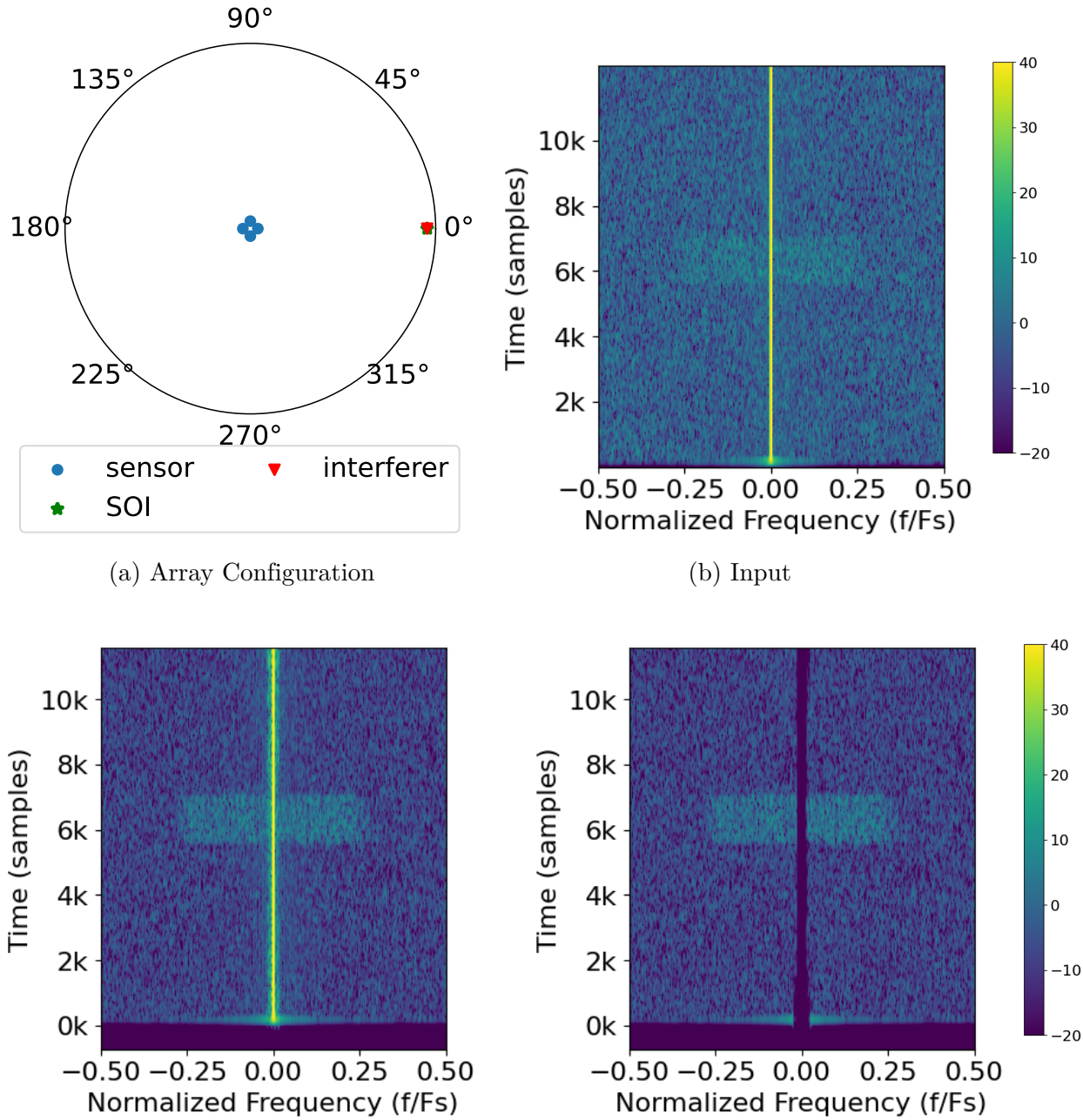
Figure 4.14: Notching Example: Combination After Synthesis SVIRA operating on a single frame with two interferers coming from nearly opposite directions; the SNR of SOI is 3 dB; the SNR of the interference is 10 dB; $M = 96$, $2L = 64$, and the SVD thresholds are -1 dB to 15 dB.

Observe the singular values for samples 3k to 6k in Fig. 4.14b. Notice that there is a relatively flat bump in SV 0 from channel -25 to 25. This bump represents SOI. The spike in SV 0 at channel 14 is due to the two interfering tones. SV 1 is mainly indicative of the noise space; however, in channel 14, it also has a spike. Since both interfering signals come from completely different directions, their eigen-spaces are also different, so interference shows up in both SVs at channel 14. Therefore, if notching is not allowed, the lowest powered SV (SV 1) is kept when both SVs are outside the thresholds, and the spectral output of the algorithm is Fig. 4.14d, where some of the interference is allowed to "sneak" through. Even with this amount of interference, the SOI cannot be decoded, so notching the spectrum is considered. By notching out the sub-band (Fig. 4.14e), a small amount of ISI is introduced, but SOI is successfully decoded.

Unsuccessful decoding of SOI is not limited to the SVIRA when SOI and interference share a DOA, so a notching technique is also proposed for the channelized MVDR process used in this thesis. Consider the scenario with a four sensor array when interference and SOI both arrive from the same direction (0°) as shown in Fig. 4.15a. The spectrum from the first sensor shows the interfering signal has a normalized center frequency of 0 F_s and hinders decoding of SOI. The received spectrum is processed using a channelizer with 96 sub-bands and a classic MVDR beamformer. Since MVDR maximizes the SINR coming from a particular direction – in this case, the direction of SOI – and the interference comes from the same direction as SOI, the SINR of the interference is maximized along with SOI as seen in Fig. 4.15c. With such a strong interfering signal still present, SOI cannot be decoded.

Therefore, a method to notch the particular sub-band containing the interference is proposed. The proposed notching method is as follows:

- The minimum eigenvalue of the interference-plus-noise covariance matrix $\mathbf{R}_{i+n}$ is selected as the reference point.



(a) Array Configuration

(b) Input

(c) Output of SVIRA without threshold

(d) Output of SVIRA with threshold of 5 dB

Figure 4.15: Notching Example: MVDR operating on a single frame with one interferer coming from the same direction as SOI; SNR of SOI is 3 dB; SNR of the interference is 10 dB; $M = 96$, $2L = 64$, threshold applied in (d) is 5 dB

- A relative threshold is added to the reference point, and the sum is used as the threshold. If the power of the values in $\mathbf{R}_{i+n}$ exceed the threshold, then that value is notched.

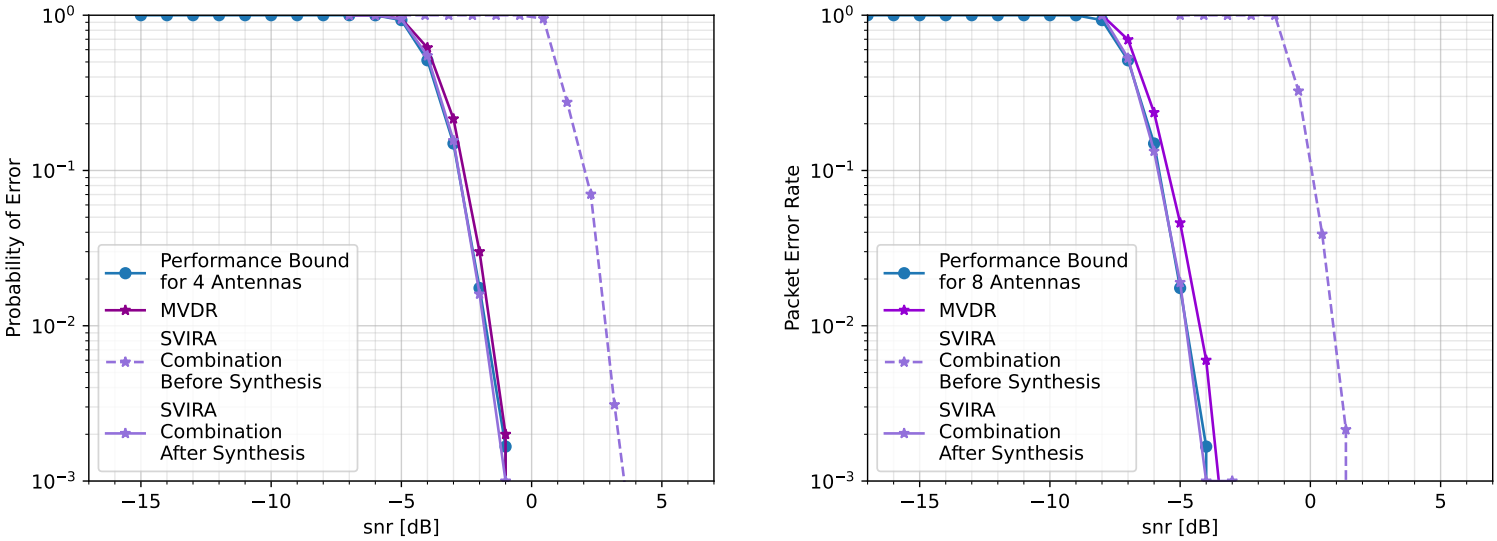
In this way, the high powered interference signals that have a similar DOA to SOI are removed, creating a notch similar to when a notch filter is applied. With this notching method applied to MVDR, the resulting output spectrum does indeed contain a notch where the interference was as seen in Fig. 4.15d, and SOI can be decoded.

4.8 Simulation Performance in AWGN Channels

To evaluate the efficacy of the proposed SVIRA, first consider its performance in an AWGN environment with no interference. Fig. 4.16a shows the PER curves for MVDR and SVIRA for both the Combination Before and After Synthesis configurations on a four-sensor array. Note that both MVDR and Combination After Synthesis SVIRA perform almost just as well as the performance bound, with SVIRA barely outperforming MVDR. However, the Combination Before Synthesis SVIRA is not able to perform nearly as well as the other two beamformers. Similarly, observing the PER curves for an eight-sensor array operating in AWGN with no interference in Fig. 4.16b, MVDR performs within 0.5 dB of the performance bound while Combination After Synthesis SVIRA aligns perfectly with the performance bound.

As discussed in Section 2, MVDR maximizes the signal-to-interference-plus-noise ratio in a particular direction, subject to knowledge of the array element arrangement, orientation, and phase offsets. This additional information enhances the ability of an array to suppress interference while pointing beams in the direction of a signal of interest. Additionally, as previously discussed in Section 4.5, in the Combination After Synthesis configuration, the Rayleigh Quotient is able to make a decision based on all of SOI's samples rather than

just $2L$ of them as it would in the Combination Before Synthesis configuration. Therefore, it is expected for both MVDR and Combination After Synthesis SVIRA to outperform Combination Before Synthesis.



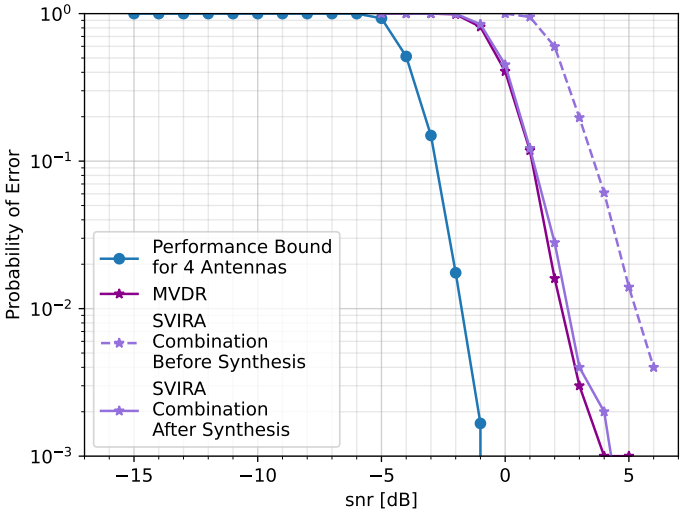
(a) 4 Sensors

(b) 8 Sensors

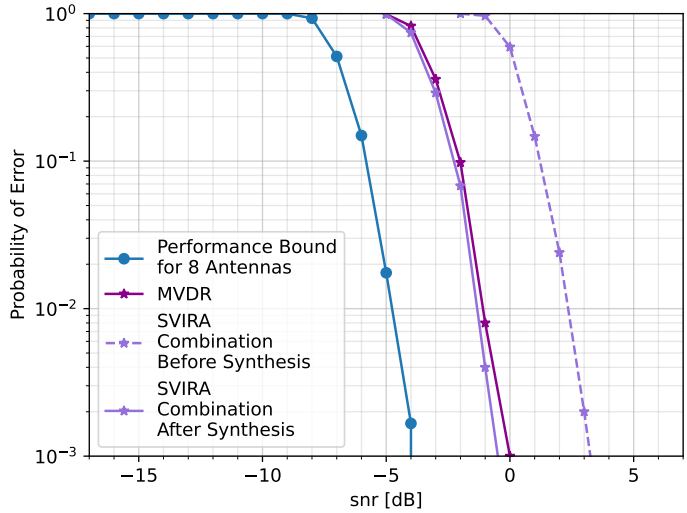
Figure 4.16: Comparison of MVDR and SVIRA Combination Before and After Synthesis with no interference; $M = 96$, $2L = 64$, SVD thresholds are -10 dB to 80 dB, and the MVDR threshold is 5 dB

4.9 Performance in AWGN with Narrow-band Interference

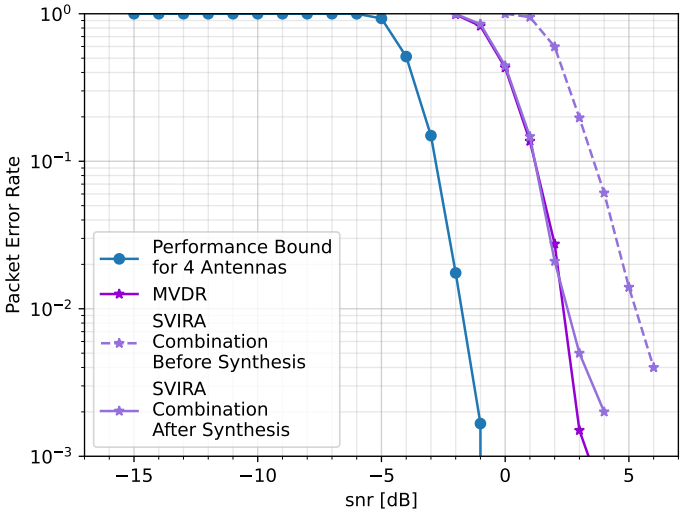
Next, consider how the SVIRA compares to MVDR beamforming when there is interference present. First, a simplified case is considered where the DOA of the interference is limited to $\pm 10°$ of SOI's DOA. Since SOI arrives from 0° in azimuth, the DOAs of the interferers are restricted to the inclusive range of 10° to 350°. By restricting the DOA of the interfering signals, we are guaranteeing that no interference will have the exact DOA as SOI, which will



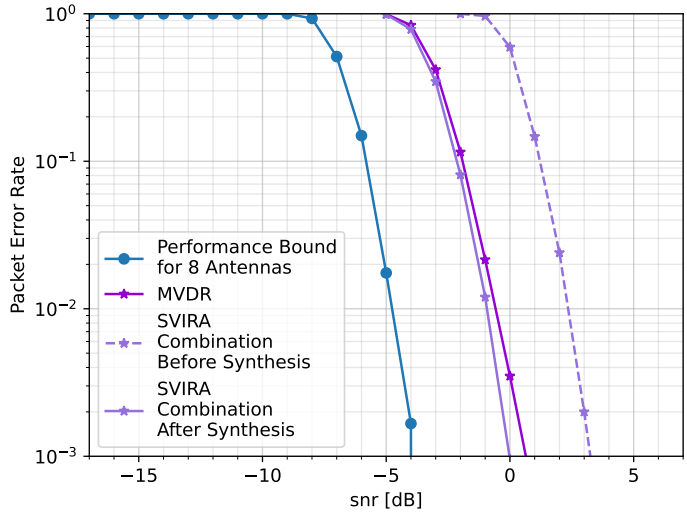
(a) 4 Sensors, interference DOA restricted (10° to 350°)



(b) 8 Sensors, interference DOA restricted (10° to 350°)



(c) 4 Sensors, interference DOA unrestricted



(d) 8 Sensors, interference DOA unrestricted

Figure 4.17: Comparison of MVDR and SVIRA Combination Before and After Synthesis with 8 interference sources; $M = 200$, $2L = 64$, SVD thresholds are -1 dB to 15 dB, and the MVDR threshold is 5 dB

in turn limit the need for notching. This was investigated to assess just how detrimental coincidental interference is on system performance.

The PER curves for a four-sensor array when 8 interfering tones are present are shown in Fig. 4.17a. Notice that while MVDR and Combination After Synthesis SVIRA are not able to achieve the performance bound in this case, they still have a similar PER performance in comparison to each other. However, the Combination Before Synthesis case still performs worse by at least least 3 dB. When the array has eight sensors instead of four, the PER curves are as displayed in Fig. 4.17b. Once again, MVDR and Combination After Synthesis SVIRA have roughly the same performance, but the Combination Before Synthesis case is not able to perform nearly as well.

Now, consider the full scenario in which the DOA of the interference is unrestricted, and the interferers may come from any angle in azimuth including the same DOA as SOI. Despite increasing the probability of introducing more nulls, the PER curves for the 4 sensor unrestricted DOA case (Fig. 4.17c) are very similar to the curves from the 4 sensor restricted DOA case (Fig. 4.17a). Similarly, 8 sensor unrestricted DOA PER curves (Fig. 4.17d) are similar to the 8 sensor restricted DOA curves (Fig. 4.17b). This suggests that for this particular array configuration, even when the interference is allowed to arrive at the same or similar DOA as SOI, both MVDR and SVIRA are able to achieve comparable PER performances to the case when the interference DOA is restricted to never be the same as the DOA of SOI.

To illustrate the importance of having a perfect knowledge of the array configuration on MVDR beamforming, an experiment is run with the simulated packets and the channelized MVDR in two modes: (1) classic MVDR for which there is perfect knowledge of the array configuration, and (2) invalid MVDR for which the assumed array configuration is not perfect. The actual array configuration for these tests is a uniform circular array with a distance

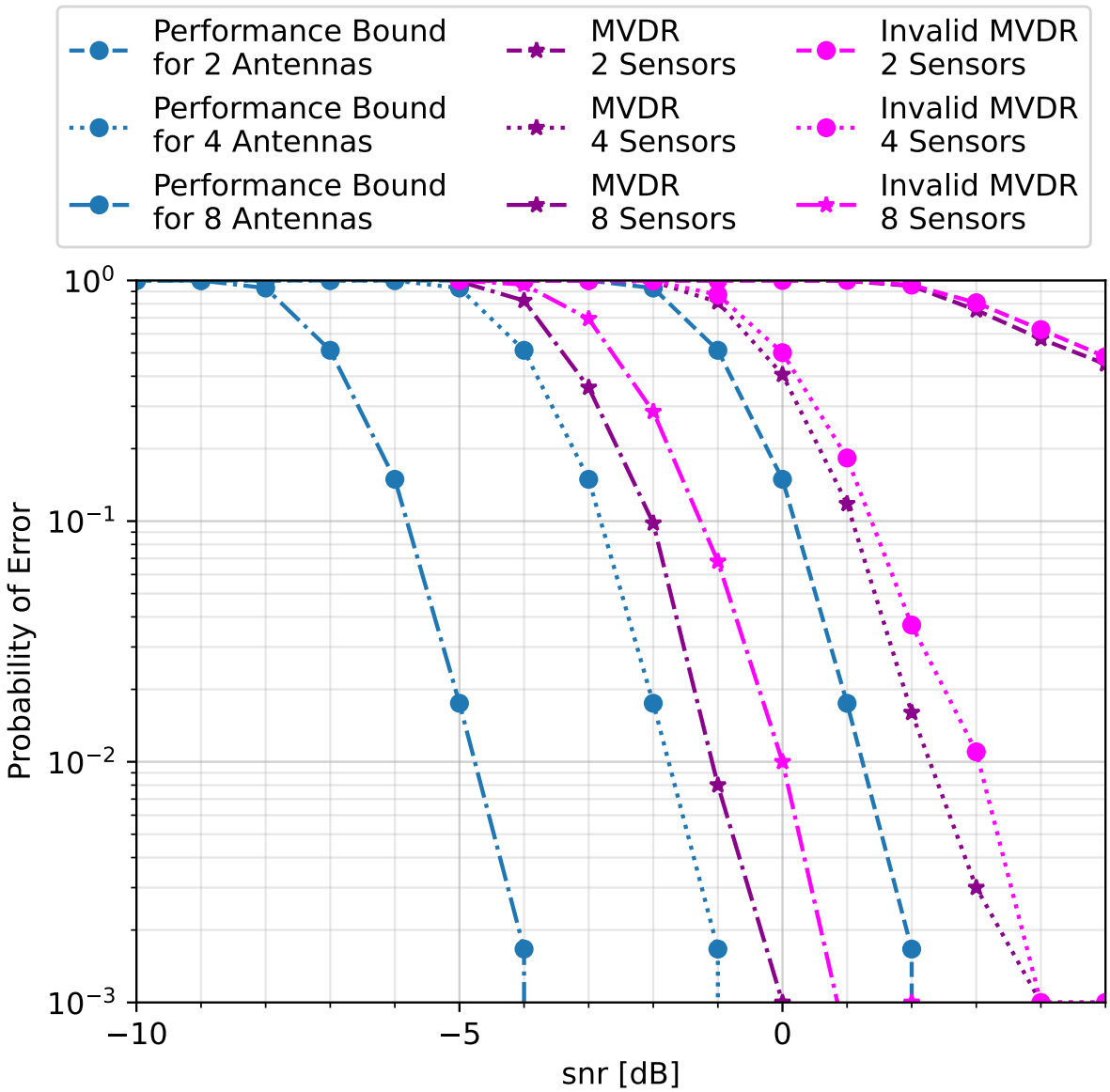


Figure 4.18: Comparison of classic MVDR and invalid MVDR performance for 2-sensor, 3-sensor, and 4-sensor arrays with 8 interference sources

of 0.3λ between adjacent sensors, where λ is wavelengths. In the classic MVDR case, perfect knowledge of the array configuration is assumed. However, for the invalid MVDR case, the distance between adjacent sensors is 0.25λ. This process is repeated for an array with two, four, and eight sensors. Although the eror in the assumed distance is only 5% of a wavelength, the incorrect assumption is enough to significantly impact the PER performance as seen in Fig. 4.18.

The expected PER performance difference for two antennas and four antennas is about 3 dB because the enhancement between them is a factor of two, which can be seen between the performance bound of two and four antennas. Likewise, there is a factor of two between four and eight antennas, which translates to an improvement of 3 dB—again shown by their respective performance bounds. This factor of 3 dB can be seen between the PER of MVDR for 4 sensors and 8 sensors and again in the PER of the invalid MVDR for 4 sensors and 8 sensors. It is more difficult to see it for the 2 sensors and 4 sensors curves because both the classic MVDR and invalid MVDR 2 sensor curves display poor performance.

MVDR is not able to perform well when there are only two sensors in the array due to the randomized nature of the interference placement. While the interferers are purposefully not allowed to have a DOA within 10° of SOI, they can still be close together in frequency. With 8 interfering tones being placed in the band, there is a high likelihood that multiple tones would have similar frequencies and be place in the same sub-band. In this case, the number of interfering signals would be greater than the one degree of freedom available within the sub-band, so MVDR would not be able to remove the interference without adding too much distortion to the packet.

It is important to note that the invalid MVDR estimates consistently perform worse than their classic, valid counterparts. There is roughly a 1 dB performance difference between the classic MVDR and invalid MVDR for the 8 sensors case; an average performance difference

of 0.5 dB for the 4 sensors case, and a performance difference of less than 0.05 dB for the 2 sensors case. This seems to imply that the larger the array, the worse the performance degradation will be if the wrong array configuration is assumed.

4.10 Results and Discussion

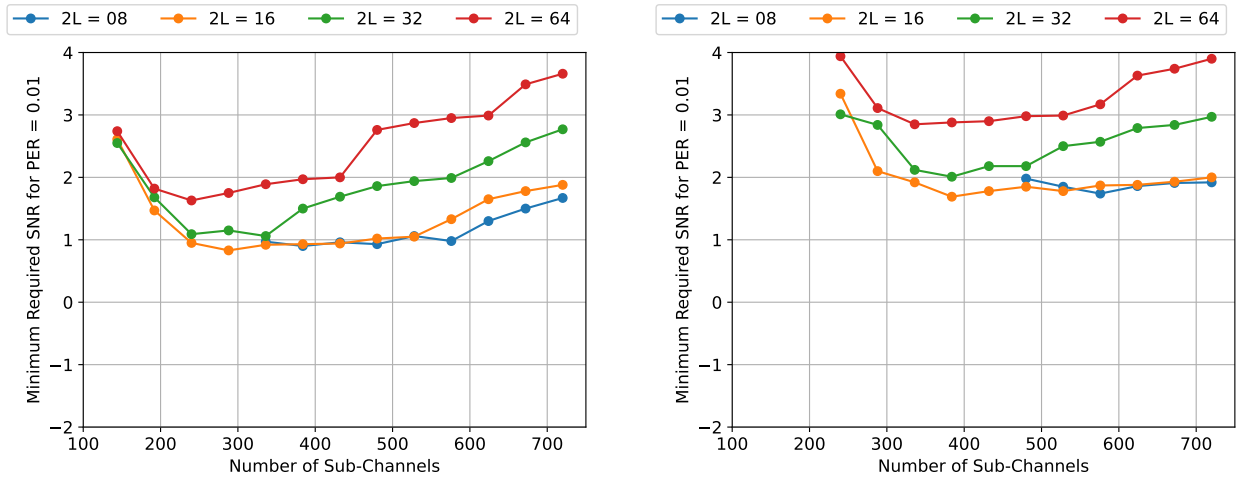
Now that the general performance of SVIRA is determined, some parameter sweeps are used to determine the optimal number of sub-bands and block size. Packets of varying SNR are run through the Combination Before Synthesis SVIRA with thresholds from 1 dB to 15 dB for different block sizes ($2L$) and number of sub-bands M. The sweeps were run for two different cases: with 6 interfering tones present and 8 interfering tones. After determining the PER curves, the minimum SNR needed to achieve a PER of 1% is determined. The minimum SNR required versus the number of sub-channels is shown in Fig. 4.19.

In the case of 6 interfering tones (see Fig. 4.19a), the optimal number of sub-bands seems to be between 200 and 350. A smaller channelizer size (fewer sub-bands) means that the sub-bands are wider in frequency and allow for more interferers to occupy a single sub-band. On the other hand, a larger channelizer size reduces the signal observation. This is especially apparent for channelizers with 500 or more sub-bands. Furthermore, it seems that the smaller the block size, the better the algorithm tends to perform. A block size of 16 samples outperforms that of 32 and 64 samples no matter the number of sub-channels in the channelizer. However, when the block size reaches 8 samples, it becomes too small to make a marked improvement over 16 samples. A similar phenomenon happens when the parameters are swept in the presence of 12 interfering tones.

In Fig. 4.19b, once again, generally, using a smaller block size produces better performance than using larger ones. A block size of 16 samples generally produces the best performance,

and a block size of 8 samples is not able to perform consistently better. Furthermore, this time, the optimal number of sub-channels is between 300 to 400. This is likely due to the increased number of interferers and the need for a greater ability to separate them in frequency; hence, an increase in the optimal number of sub-channels.

In general, there appears to be an optimal range for the size of the channelizer that includes roughly 100 sub-channels. If the channelizer is too small (fewer sub-bands), more interference is let into each sub-band, and the chance of removal is reduced. If the channelizer is too large (more sub-bands), then the observation of the signal is reduced. Similarly, there is also in an optimal range for the block size. Although a smaller block size is generally preferred, a block size that is too small results in poor estimates of the SVs. Too large a block size and time-varying effects must be taken into consideration.



(a) Parameter sweep for 6 Tones

(b) Parameter sweep for 12 Tones

Figure 4.19: Parameter sweep for SVIRA Combination Before Synthesis

In summary, great care must be taken in the parameter selection of the processing as it has a direct impact on the algorithm's ability to perform properly. Selecting an appropriate threshold, choosing an appropriately sized channelizer, and setting a suitable block size are all important considerations. When chosen appropriately, SVIRA has been shown to

have a similar PER performance as MVDR. Even with imperfect knowledge of the antenna spacing, the performance of the proposed solution greatly out-performs MVDR under the wrong assumptions (or miscalibration) of the array, and under certain circumstances the proposed SVIRA can even match the performance of an appropriately calibrated MVDR.

Chapter 5

Verification on Hardware

An important aspect to digital phased array research is to carry out experimentation on representative hardware in a laboratory environment. While theoretical modeling can achieve performance bounds in an ideal scenario, the accuracy of such is subject to imperfections in assumptions (mismatched antenna patterns, cross-polarization, timing offsets between sample streams), non-ideal hardware (non-linear power amplifier distortion, I/Q imbalance, local oscillator leakage, etc.), and channel effects (multi-path, independent gains per transmitter and per sensor). Consequently, the value of testing algorithms in a laboratory environment's impairments is paramount to their validity.

In this chapter, the laboratory experimentation to validate the processing and assumptions on readily-available software-defined radio (SDR) hardware and antennas is outlined.

5.1 Hardware Setup and Description

The laboratory used to conduct the experiment is small and contains metal shelving and other electronics that contribute to electromagnetic (EM) interference thus creating a multi-path environment. Additionally, as the lab is a shared environment, during certain data captures, other researchers are present in the room and may have been moving about.

A picture of the actual lab setup is shown in Fig. 5.1b, and a diagram of the antenna

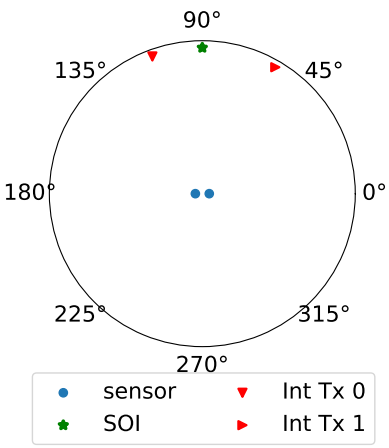
locations is shown in Fig. 5.1c. As seen in Fig. 5.1b, the main transmitter is comprised of a universal software radio peripheral (USRP) E-320 [38] and a dipole antenna. SOI is transmitted at a center frequency of 2.205 GHz and sample rate of 100 kHz. However, since there is considerable local oscillator (LO) leakage, the antenna is off-tuned to compensate.

The receiver array is placed approximately 154 cm away from the SOI transmitter as displayed in Fig. 5.1c. There are two antennas that make up the array, and each antenna is connected to one of the receiver ports on a USRP E-320. The centers of each antenna element are approximately 1.6 wavelength apart. Additionally, each element of the array is approximately equidistant from the transmitter of SOI in order to achieve as close to equal receiver gain at each antenna element.

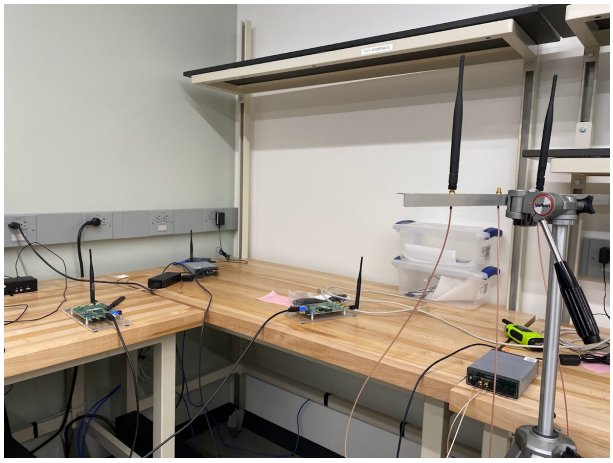
The interference signals are generated using antennas connected to two USRP B-210s. The first interference transmitter (Int. Tx. 0) is located 131 cm away, and the other (Int. Tx. 1) is 100 cm away. During testing, either one or both of the B-210s are used to transmit uniformly spaced tones of a given number. Although only tonal interference is verified using hardware, other types of interference could also be verified using the same hardware in the same configuration. Since other interference types have been shown to be removed in simulation, it is likely that they can also be removed from captured data from a lab experiment.

5.2 Data Collection Setup

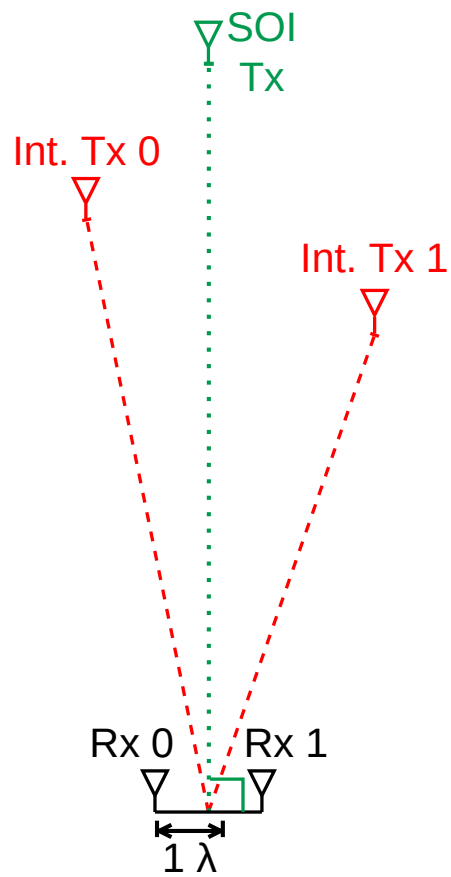
For each data set, there are two distinct transmissions that are transmitted consecutively: one capture contains additionally transmitted interference while the other does not. Transmission A only concerns the SOI packets, whose transmission gain sweeps from a high to low gain in 1 dB steps. During this transmission, the receiver is turned on first, then the



(a) Array configuration



(b) Picture of setup in the lab



(c) Diagram of relative positions of transmitters and receivers

Figure 5.1: Lab setup

SOI transmitter is activated to send packets. Neither of the interference emitters are on during this transmission sequence, so the resulting data should just be SOI on top of the background spectral environment. This should allow for an accurate depiction of the current spectral environment at the time of the transmission. Once transmission A is completed, transmission B immediately follows. Transmission B transmits SOI packets at the same gain values as Transmission A; however, this time intentional interference in the form of tones is added. The transmission sequence is a follows: The interference emitter or emitters are turned on and transmit continuously throughout the data capture. Next, the receiver is switched on, and finally, the SOI transmitter is allowed to transmit packets.

Since the spectral environment could change in a given moment due to the other equipment in the lab, it is best to transmit the second transmission sequence in quick succession. Additionally, as the lab is shared and data captures are collected at different times on different days, any of the antennas could have moved between transmissions resulting in different DOAs and distances between the transmitters and the receiver array. Therefore, it is important to always transmit both transmission A and B consecutively.

Transmissions A and B allow for a direct comparison of the proposed process with and without relatively high SNR interference in a real world environment. Simple PER curves are once again used as the preferred method to analyze the efficacy of the proposed algorithm on the captured data. By embedding the transmission gain value into the header of the packet, it is simple to generate PER curves based on the known transmission gain. However, when comparing the captured data to simulation, using transmission gain as a basis is no longer a viable option since it cannot be directly compared to the known SNR values of the simulated packets. Therefore, a process is needed to transform the known transmitted gain values into approximate SNR values, so the captured and simulated data can be compared side-by-side.

5.3 SNR and Gain Calibration

In order to directly compare the captured hardware results with those from simulation, a method of estimating SNR must be devised. The transmission gain of each packet is used as a basis to estimate the SNR because it is known. Since the transmission gain and SNR of the packet should have a linear relationship, only one gain-SNR relationship pair needs to be found in order to calculate the SNR for the rest of the transmission gains. Once a gain-SNR reference pair is determined, the ith SNR associated with the ith gain can be calculated using

$$\text{SNR}_i[\text{dB}] = \text{gain}_i + (\text{SNR}_{\text{ref}} - \text{gain}_{\text{ref}}) \tag{5.1}$$

As such, it is important that the estimated reference SNR is as accurate as possible.

5.3.1 Calibration Method and Example

In order to illustrate how the SNR is estimated, examine the example presented by Fig. 5.2. In this example, the noise floor is 0 dB, and the SNR of the packet is 15 dB. To estimate the SNR of a single packet, first, the index of the beginning of the packet is found, which is used to create a snippet of the data file that isolates the packet. Then the PSD of the snippet is taken and a bounding box is drawn around the packet, which is represented in red in Fig. 5.2a. Using the principle of exclusion, a smaller bounding box containing the signal-plus-noise PSD is drawn in gray, and the two bounding boxes in black are created to solely contain the PSD of the noise. The signal-plus-noise power and noise power can both be determined using ordered statistics on the PSD values in the appropriate bounding boxes. Once the signal-plus-noise power and noise power are estimated, the SNR can be

approximated using

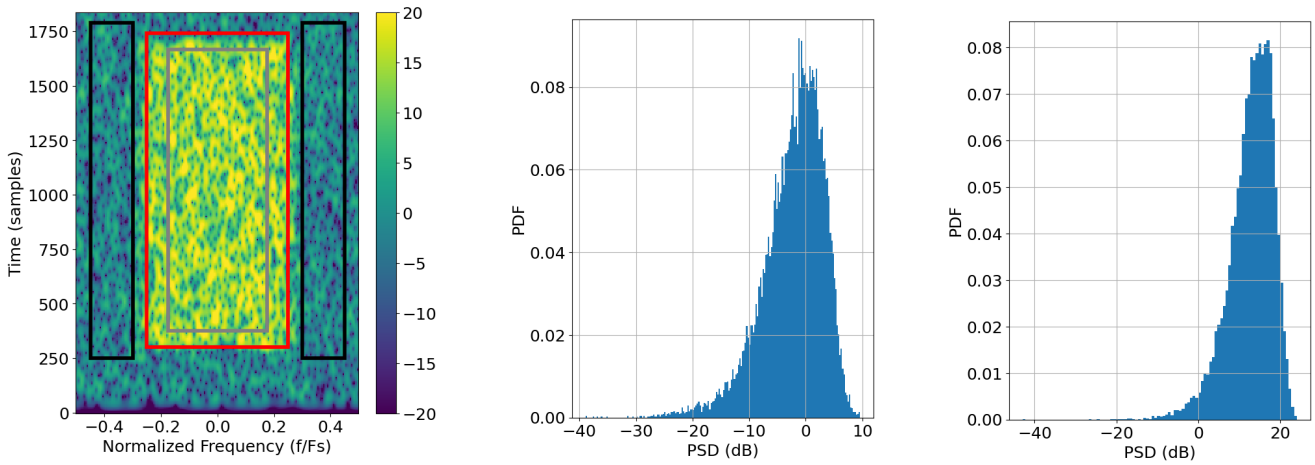
$$\text{SNR[linear]} = \frac{\text{signal power - noise power}}{\text{noise power}} \tag{5.2}$$

Thus, it is important to achieve a reasonably accurate estimation (< 0.1 dB) of both the signal-plus-noise power and the noise power.

The use of ordered statistics attempts to mitigate the influence of unknown signals in the background of the snippet. In this example, the median noise power is -1.5 dB, and the median signal-plus-noise power is 13.7 dB, which gives an SNR estimate of 15.0 dB. On the other hand, the mean of the noise power is 0.0 dB, and the mean of the signal-plus-noise power is 15.2 dB, which gives an SNR estimate of 15.0 dB. Despite the estimated SNR values being the same, the approximated signal-plus-noise powers and noise powers are different by 1.5 dB. By estimating at different SNR and noise floor values, a 1.5 dB bias is found when using the median as the ordered statistic. Therefore, the mean is used as the ordered statistic of estimation. This choice is confirmed by looking at the probability density of the PSD values of the noise (Fig. 5.2b) and signal-plus-noise (Fig. 5.2c). Neither of their distributions is Gaussian, and both distributions skew to the right. As such, the mean provides a more accurate estimation of the power than the median, which falls to the left of the peaks.

5.3.2 Validation in Simulation

This method is validated in simulation by generating 1,000 packets with random AWGN for each SNR from -5 dB to 15 dB in 1 dB steps. For each value of the true SNR, the median, standard deviation, and variance of the estimated SNRs is found. Fig. 5.3b shows the median estimated SNR versus the true SNR, and the error bars represent the interquartile range (one standard deviation above and below the median). If the estimated SNR matches



(a) PSD of the entire snippet

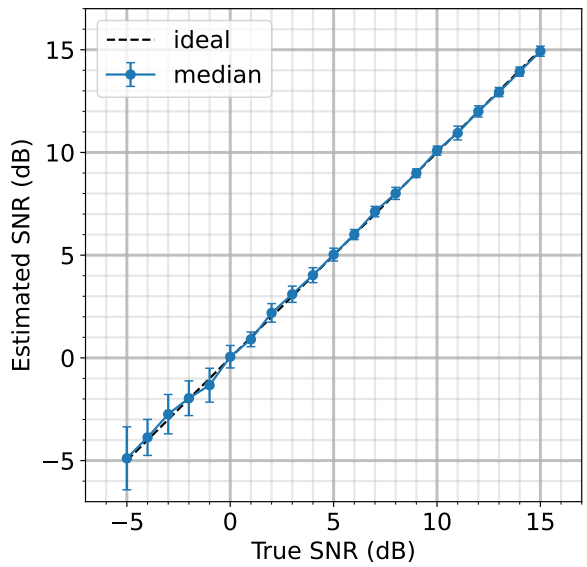
(b) Probability density of the noise PSD (from the black bounding boxes in (a)).

(c) Probability density of the signal-plus-noise PSD values (from the grey bounding box in (a))

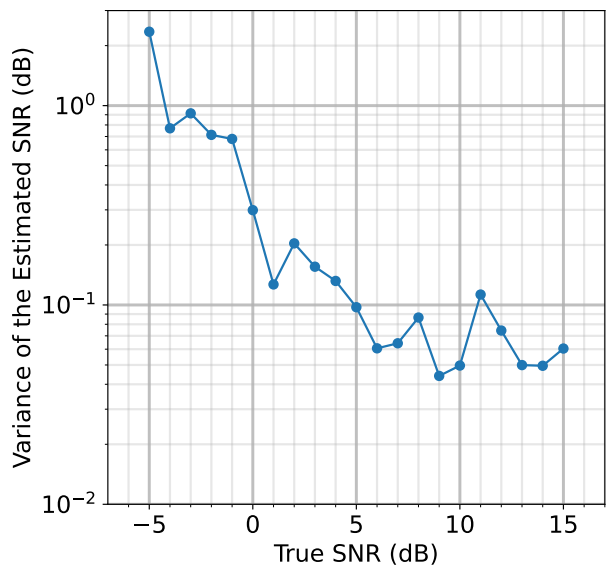
Figure 5.2: SNR Calibration Example: SNR of the packet is 15 dB, and the noise floor is 0 dB

the true SNR exactly, the blue median line will completely cover the black ideal. Since the median of the estimated SNRs is so close to the true SNR, the blue line barely deviates from the ideal, which indicates that this is a viable method accurately estimating the SNR and contains no bias. However, notice that the error bars increase as the SNR decreases. This indicates that the estimate is less robust at lower SNRs. Similarly, Fig. 5.3d illustrates that the variance of the estimated SNR is higher at low SNRs. At -5 dB, the variance is a little over 1 dB, but anything higher than 6 dB has a variance of 0.1 dB. Therefore, so long as the true SNR of the snippet is above 6 dB, it can be estimated accurately within 0.1 dB. While it is good that the SNR can be estimated accurately, it would be beneficial if the estimation process could maintain its accuracy but with a reduced number of estimated SNRs.

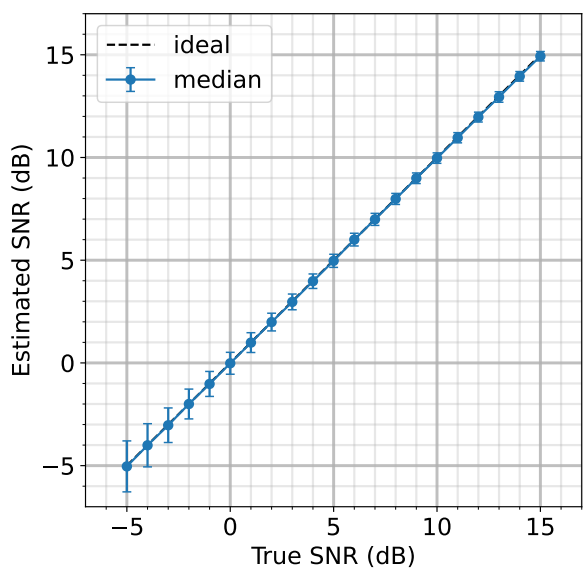
Consider Fig. 5.3a, which contains the same statistical information as Fig. 5.3b but for 20 trials. Note that the medians of the high SNRs (>5 dB) show results comparable to the 1,000 trials case and fall on the ideal line (Fig. 5.3a), and the variance of these estimates is



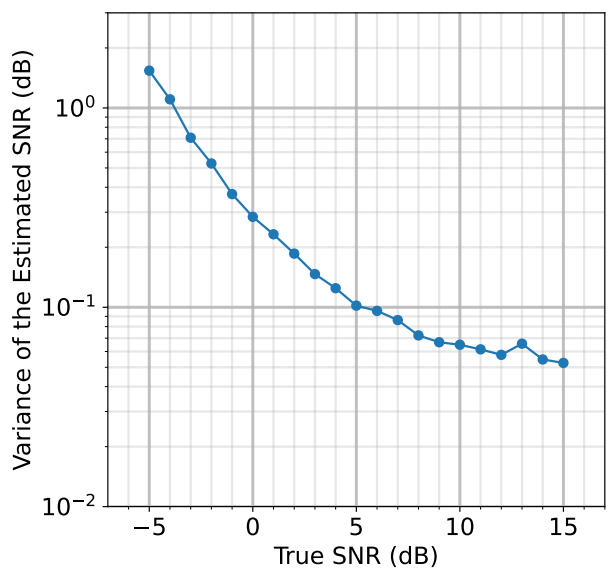
(a) Estimated SNR vs. true SNR for 20 trials, where the error bars are the interquartile range (IQR)



(c) Variance of the estimated SNR for 20 Trials



(b) Estimated SNR vs. true SNR for 1,000 trials, where the error bars are the IQR



(d) Variance of the estimated SNR for 1000 Trials

Figure 5.3: Statistics of the estimated SNR for simulated packets

generally below 0.1 dB (Fig. 5.3c). However, at SNRs below 5 dB—particularly at negative SNRs—the estimates no longer perfectly follow the ideal line as outlined in Fig. 5.3a. It is difficult for this method to accurately estimate negative SNRs. Although the median of the 20 estimates for each negative SNR is within a dB of the true SNR, the variance and standard deviation of the estimates are far greater than when 1,000 packets are estimated. Compare Fig. 5.3c to Fig. 5.3d. At an SNR of -5 dB, estimating only 20 packets yields a variance of 2.3 dB while there is only a variance of 1.5 dB when 1,000 packets are considered. Additionally, for all negative SNRs, the variance of the estimated SNRs for 20 packets is no less than 0.7 dB whereas it decreases to about 0.3 dB for 1,000 packets. This is to be expected as when there are fewer packets to estimate the SNR from, the resulting variance (and thus standard deviation) of the estimates will generally be greater. Nevertheless, since this estimation technique is only going to be used on packets of a particular reference gain, so long as the gain is set relatively high, the robustness of the estimate of lower SNRs is not of concern. Therefore, as SNRs of 5 dB and above can be just as accurately estimated within 0.1 dB with 20 trials as with 1,000, it is reasonable to estimate the SNR of captured data using only 20 packets.

5.3.3 Bias Correction

Although this method is able to accurately estimate the SNR of simulated packets, it does have some bias when estimating the SNR of captured packets. Observe how Rx 1 and 0 without the bias correction fall about 1 dB away from the simulated performance bound in Fig. 5.4. Since the difference between the two is consistent, an inherent bias of 1 dB is surmised. This could in part be due to the background noise not being flat across the band as seen in Fig. 5.5. Notice that the noise floor on the left edge of the band is higher than that on the right edge. The average power on the left edge is -64.5 dB while it is -67.9 dB on

the right edge, which means there is around a 3.4 dB discrepancy in the noise floor from the left to the right side of the band. It is possible that the front end of the receiver has a small frequency-dependent response, and the signal itself could have a small frequency-dependent slope. Despite averaging the two edges, the resulting SNR still has a small bias as seen in Fig. 5.4.

The bias is corrected using the error vector magnitude (EVM) of the estimated packets. Since the EVM is a measurement of the overall distortion of the symbols, it not only takes noise distortion into account, it also considers IQ mismatch, nonlinearity, thermal noise, channel distortion, carrier leakage, etc. Thus EVM should not be used to accurately estimate SNR, but it can be useful to correct the bias. The bias removal procedure is as follows:

1. Find the EVM of each packet.
2. Estimate the SNR of each packet.
3. For each packet, determine the difference between the EVM and estimated SNR.
4. Take the median of the differences as the bias.
5. Add the bias to the estimated SNR.

Once the bias is removed, the Rx 0 and Rx 1 align with the performance bound (as displayed in Fig. 5.4) as expected.

5.4 Results and Discussion

In this section, three hardware captures are compared to simulated results. The first capture (File 8) contains no interference, so its results are used as a baseline performance for the

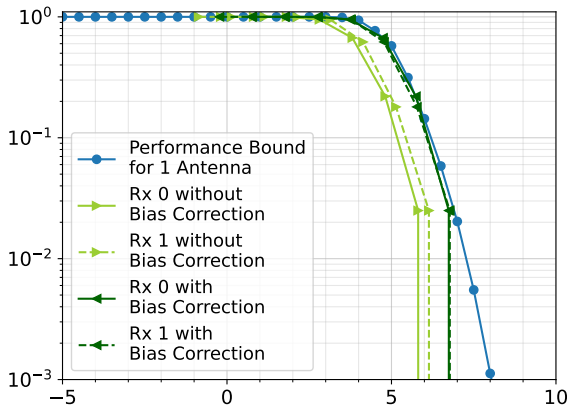


Figure 5.4: PER curves for Rx 0 and Rx 1 from File 27 with and without bias correction

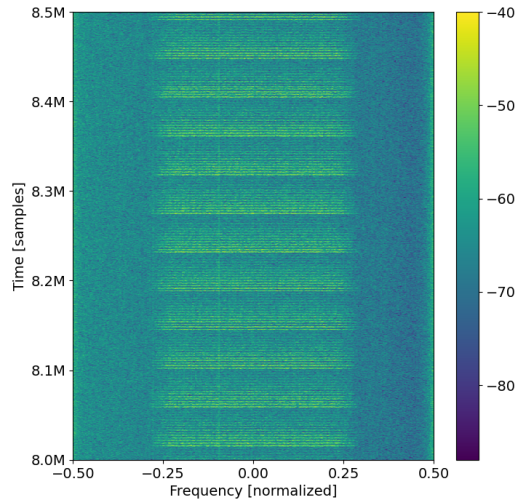


Figure 5.5: Snippet from captured data File 27, which shows the uneven noise floor

other two captures. The second capture (File 27) includes 8 interfering tones from Emitter 0. Finally, the last capture (File 41) has 13 interfering tones: 6 from Emitter 0 and 7 from Emitter 1. Only Transmission A is considered for File 8, which contains 800 sweeps. Likewise, File 41 contains 800 sweeps for Transmission A, but it also has 1,000 sweeps for Transmission B. Each sweep for both Files 8 and 41 contains one packet per transmitted gain level, which ranges from -8 dB to -22 dB, inclusive, in 1 dB steps (total of 15 steps). Thus, Files 8 and 41 each contain 12,000 packets worth of data for Transmission A, and File 41 has 15,000 packets worth of data for Transmission B. File 27 contains 200 sweeps for Transmission A and 500 sweeps for Transmission B. Each sweep also has 15 steps total, but its transmission gains range from -10 dB to -24 dB, inclusive, in 1 dB steps. As such, there are 3,000 packets in Transmission A and 7,500 packets in Transmission B.

In order to ensure the simulated data used to compare with is analogous to the captured data, three simulated datasets are created to emulate the captured transmissions. However, the channel for each simulated transmission is AWGN so as to establish a baseline ideal performance. Simulated File 4 is created to simulate Transmission A. File 4 receives 1,000

sweeps of 14 packets, whose SNR varies from -5 dB to 8 dB, inclusive, by 1 dB. Simulated File 5 is generated to emulate the circumstance of File 27's Transmission B. There are 8 interfering tones set to come from one emitter (roughly the same DOA as Emitter 0), and 1000 sweeps containing packets whose SNR ranges from -5 db to 9 dB, inclusive, are received through an AWGN channel. Finally, simulated File 6 is constructed to emulate the data received in File 41. There are 1,000 sweeps of 14 packets, whose SNR starts at 8 dB and sweeps down to -5 dB in 1 dB steps. Additionally, there are 13 interfering tones generated across the band: 6 coming from a similar DOA as Emitter 0, and 7 coming from a similar DOA as Emitter 2.

A summary of the parameters used to process each captured file is shown in Tab. 5.1. Likewise, Tab. 5.2 shows the parameters used to process the generated files. Note that each of the simulated files are processed using the same parameters as their captured counterparts. The parameters used for each file are deemed the optimal parameters for that particular file based on performance sweeps conducted on the captured files.

	Number of Sweeps	Transmission Gain Range per Sweep (dB)	Number of Tones	Number of Interference Sources	M	B	Thresholds (dB)
File 8 (Trans. A)	800	-8 to -22	0	0	96	64	-1, 80
File 27 (Trans. A)	200	-10 to -24	0	0	96	64	-1, 80
File 27 (Trans. B)	500	-10 to -24	8	1	370	128	-1, 25
File 41 (Trans. A)	800	-8 to -22	0	0	96	64	-1, 80
File 41 (Trans. B)	1000	-8 to -22	13	2	800	320	-1, 15

Table 5.1: Parameters used for the captured data files

	Number of Sweeps	SNR Range per Sweep (dB)	Number of Tones	Number of Interference Sources	M	B	Thresholds (dB)
File 4	1000	-5 to 8	0	0	96	64	-1, 80
File 5	1000	-5 to 9	8	1	370	128	-1, 25
File 6	1000	-5 to 8	13	2	800	320	-1, 15

Table 5.2: Parameters used for the generated data files

5.4.1 No Interference Source

Fig. 5.6 depicts the data from File 8 before, during, and after it is run through the proposed algorithm. A snapshot of the spectrum as captured by Rx 0 can be seen in Fig. 5.6a. Since File 8 is only concerned about Transmission A, approximately 3 full sweeps can be see in the spectrogram without any interference. Therefore, the raw data from sensors Rx 0 and Rx 1 can be combined without the need to threshold for interference removal. This is achieved by setting the thresholds wide (-1 dB to 80 dB) and inputting the raw data into the algorithm. By setting the thresholds wide, most of the SVs should be kept as they all fall within the thresholds (see Fig. 5.6c), so there should be a 3 dB performance gain – associated with combining two antenna elements – at the output of the algorithm. A snapshot of the spectrum of the output is shown in Fig. 5.6b, where there is indeed an increase in the power of each packet. While it is difficult to determine if there is indeed a 3 dB improvement by looking at the spectrograms, it becomes clear when looking at the PER.

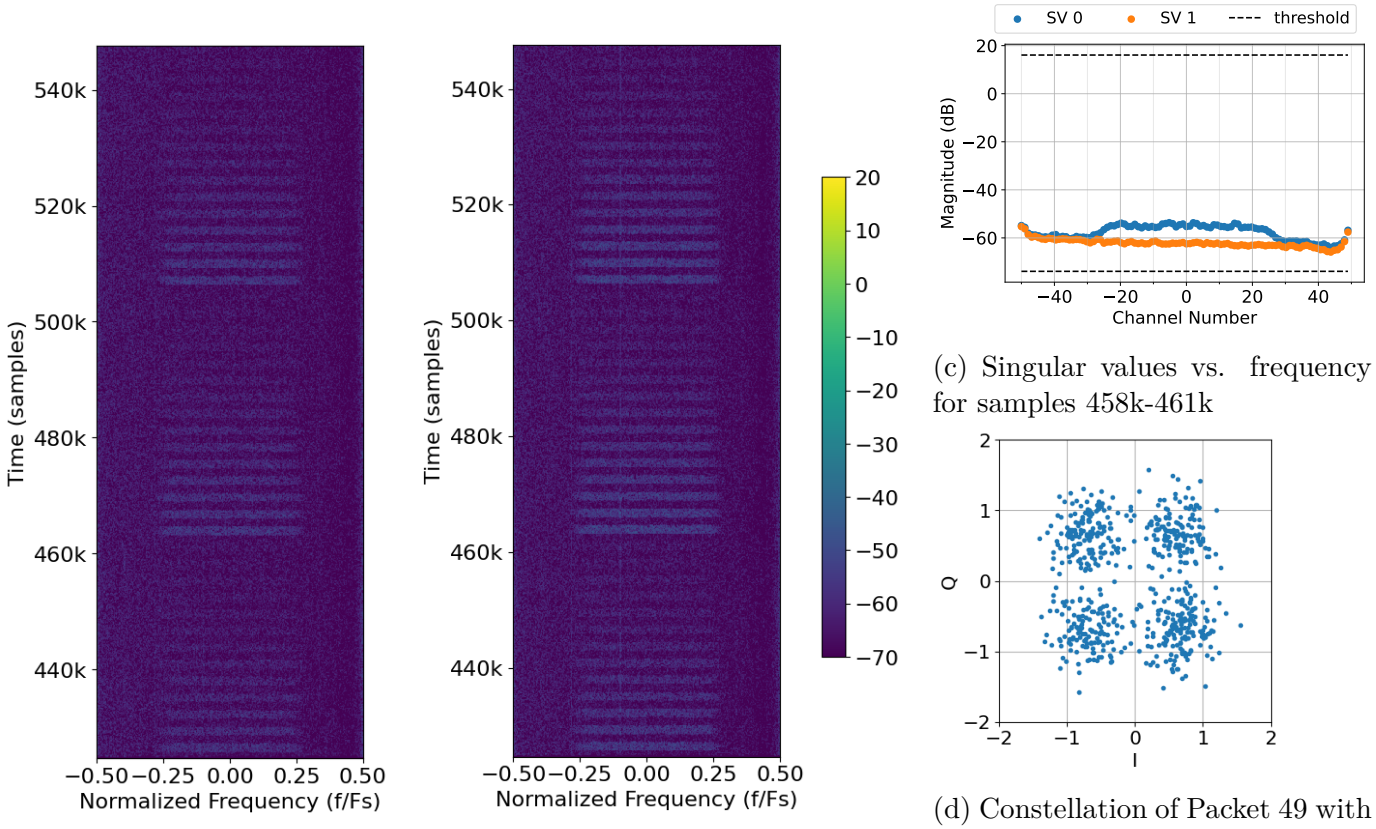
Fig. 5.7a compares the performance of the proposed algorithm on simulated data and the captured data from File 8. Note that the decoded raw data from Rx 0 and Rx 1 for both captured and simulated align with the expected performance bound for a single antenna. This indicates that the SNR of captured data is properly estimated and there are no major issues with the hardware. When the two simulated raw sensors are combined, there is a 3 dB improvement from the raw data, which can be seen when comparing the PER of the combined

sensors (orange diamonds) to the raw (orange squares). Similarly, there is a roughly 3 dB improvement when the raw, captured sensors Rx 0 and Rx 1 are combined. This improvement is made more evident when the simulated and captured combined sensor curves are directly compared to the performance bound of an array with 2 antennas in Fig. 5.7b. Note that the combined simulated sensors follow the performance bound exactly, as expected. However, there is a small performance difference between the combined captured Rx 0 and Rx 1 and the performance bound, which can be attributed to imperfections introduced by the hardware such as channel distortion, LO leakage, etc. Additionally, there are a few signals in the spectral background (seen in Fig. 5.6b, when zooming in) that are also enhanced with process and prevent successful decoding at lower SNRs.

5.4.2 1 Interference Source Case

Next, consider a case when interference is introduced. Interference Tx. 0 transmits 8, evenly spaced tones across the band as seen in Fig. 5.8a. All eight tones can be clearly see in the singular values (Fig. 5.8c) and are removed by virtue of the upper threshold. The resulting spectrum (Fig. 5.8b) no longer has visible interference. Once the interference is removed, the packets can be decoded. The constellation for successfully decoded Packet 62 is shown in Fig. 5.8d. Notice that while there is some distortion to the symbols overall, all of the symbols are tightly grouped in the appropriate quadrant.

Once again, when no interference is transmitted, both simulated and captured Rx 0 and Rx 1 align with the performance bound for 1 antenna (Fig. 5.9a, which indicates the SNR calilbration for this capture is correct. When the data from the respective arrays is combined, there is a 3 dB improvement. Note that the curve representing the combined captured elements (green diamonds) is very similar to that of the simulated (orange diamonds). The



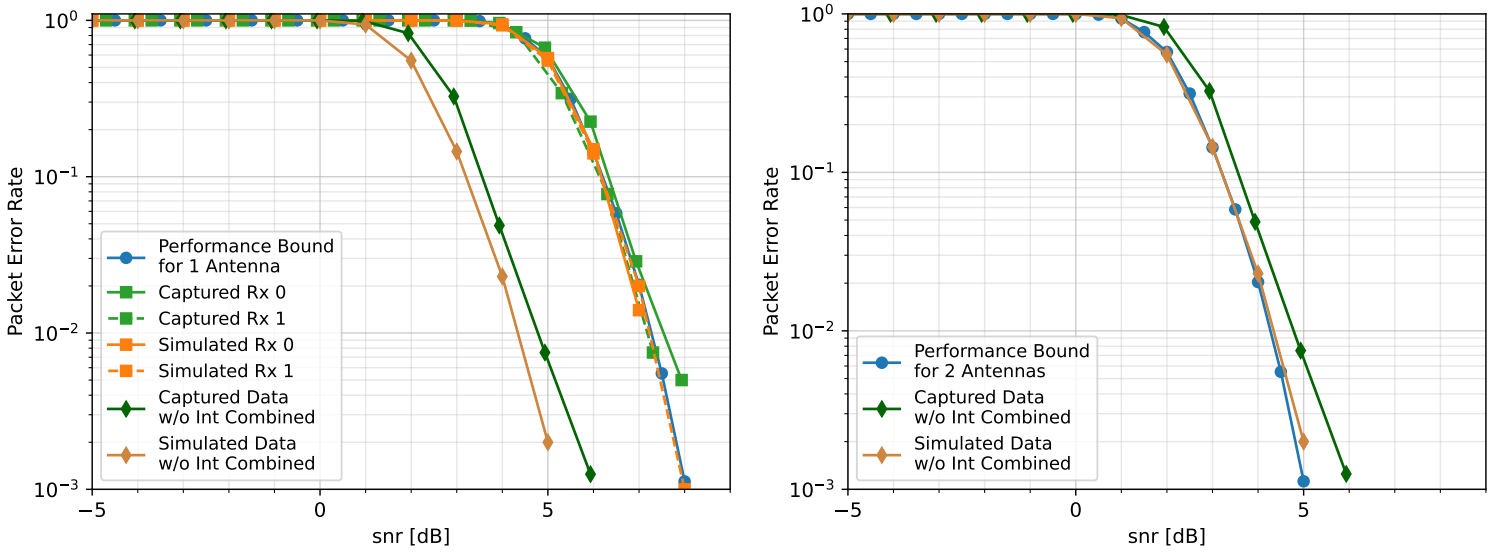
(a) Spectrum as seen by Rx 0

(b) Output of SVIRA

(c) Singular values vs. frequency for samples 458k-461k

(d) Constellation of Packet 49 with a gain of -12 dB

Figure 5.6: Processing Captured Data File 8 (No Interference Source Case) with $M = 96$, $2L = 64$, and thresholds of -1 dB to 80 dB

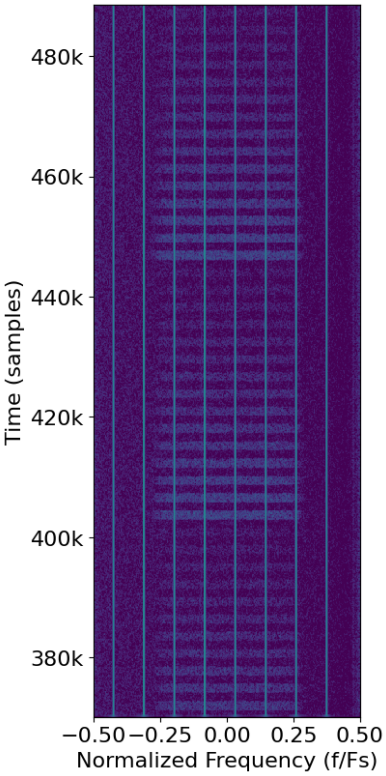


(a) PER compared to performance bound for 1 antenna (b) PER compared to performance bound for 2 antennas

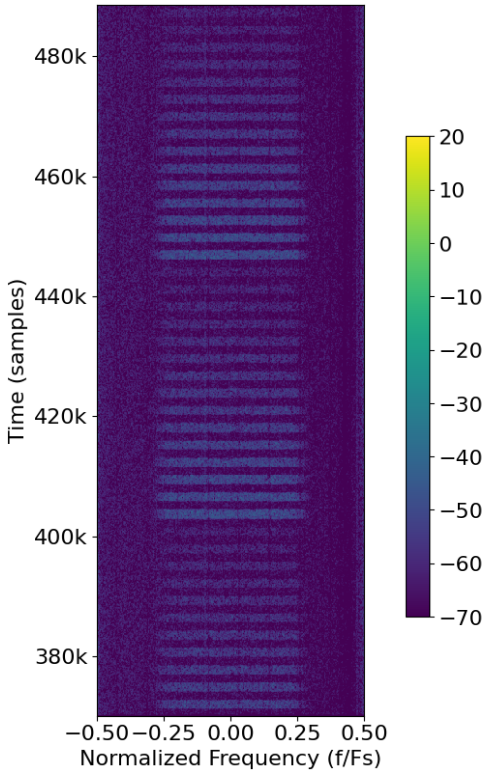
Figure 5.7: PER curves for No Interference Source Case

performance difference between the two is 0.5 dB at most. It is possible that the difference in performance between this case and the one presented in Fig. 5.7a is due to the spectral background on the given day or the position of the antennas.

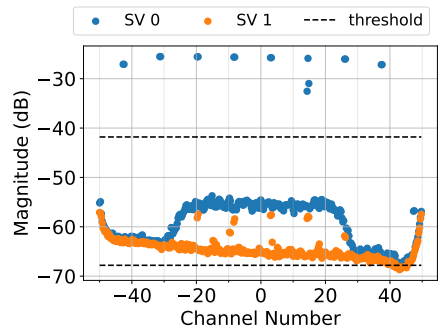
After introducing interference and removing it with the proposed algorithm, the resulting PER curves are closely aligned with the performance bound for 2 antennas (Fig. 5.9b). While there is some performance loss between the simulated combined without interference (orange diamond) and combined with interference removed (orange triangle), it is very small. Therefore, the algorithm is capable of producing results that are nearly as good as receiving packets on a 2 element antenna array without any interference. There exists a similar performance loss between the captured combined without interference (green diamond) and combined with interference removed (green triangle), which indicates the algorithm is capable of achieving a similar level of success with data captured from over the air.



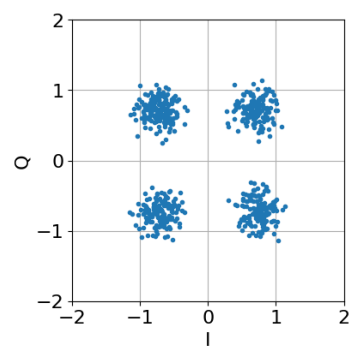
(a) Spectrum as seen by Rx 0



(b) Output of SVIRA

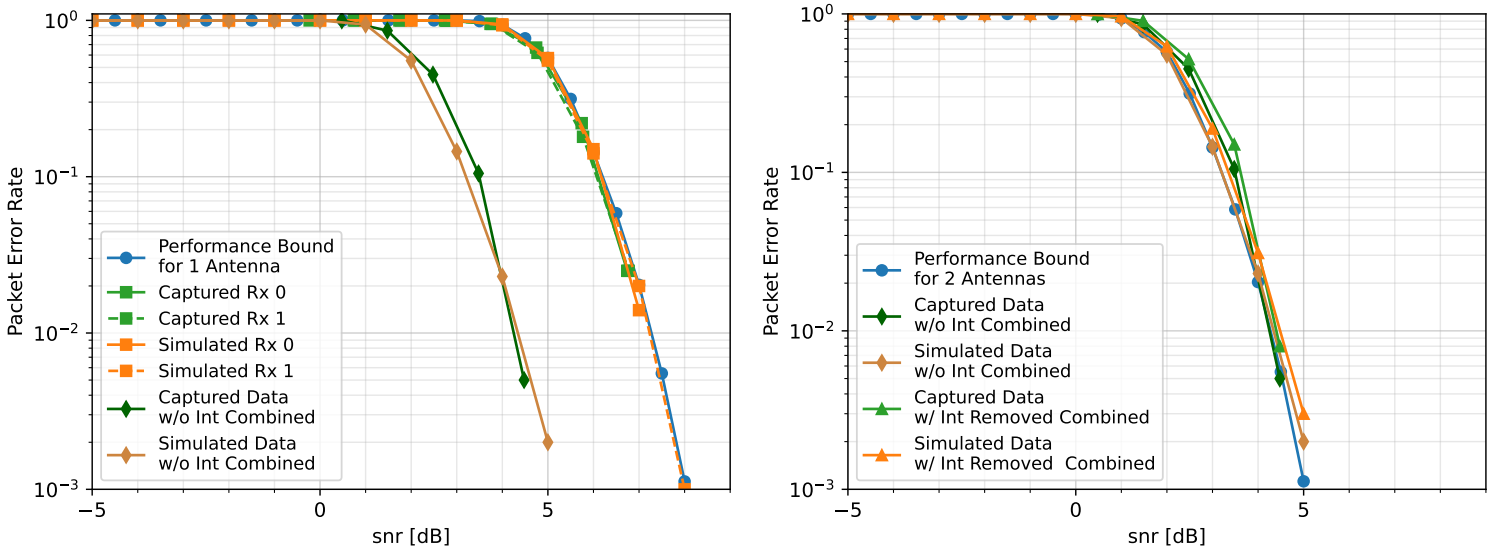


(c) Singular values vs. frequency for samples 464 k to 488 k



(d) Constellation of Packet 62 with a transmission gain of -12 dB

Figure 5.8: Processing Captured Data File 27 (1 Interference Source Case) with $M = 370$, $2L = 128$, and thresholds of -1 dB to 15 dB



(a) PER for no interference

(b) PER when interference is removed

Figure 5.9: PER curves for one interference source emitting 8 tones

5.4.3 2 Interference Sources Case

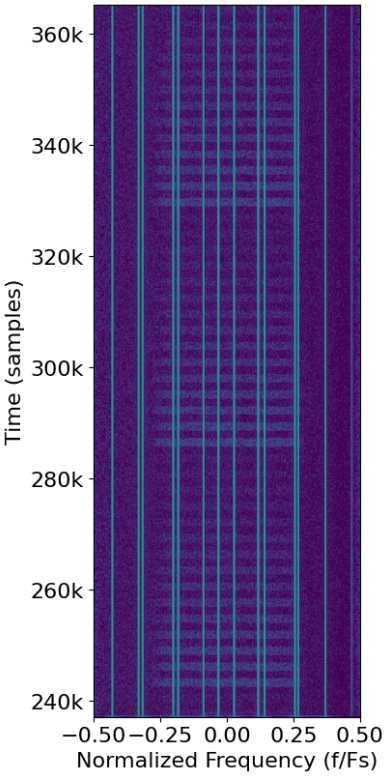
Finally, consider the case when there are 2 interference sources. In this case, Tx 0 transmits 6 evenly-spaced tones, and Tx 1 transmits 7. The spectrum showing the location of the tones and the received packets is shown in Fig. 5.10a. The singular values and thresholds are displayed in Fig. 5.10d. Notice that the upper threshold is chosen to be as tight as possible to the pedestal that represents SOI in order to remove as much interference as possible. While the interference is removed, there is a null at around -0.18 F_s (see Fig. 5.10b). This null, and others like it, causes ISI, which introduces distortion that is visible in the constellation. Observe that the constellation clusters in Fig. 5.10e are not as tight as they were in Fig. 5.8d.

In this capture, the captured Rx 0 has a worse PER performance than the captured Rx 1 (Fig. 5.11a). Captured Rx 0 has an approximate 0.5 dB performance loss in comparison to the performance bound for 1 antenna and Rx 1. This is likely due to the position of the

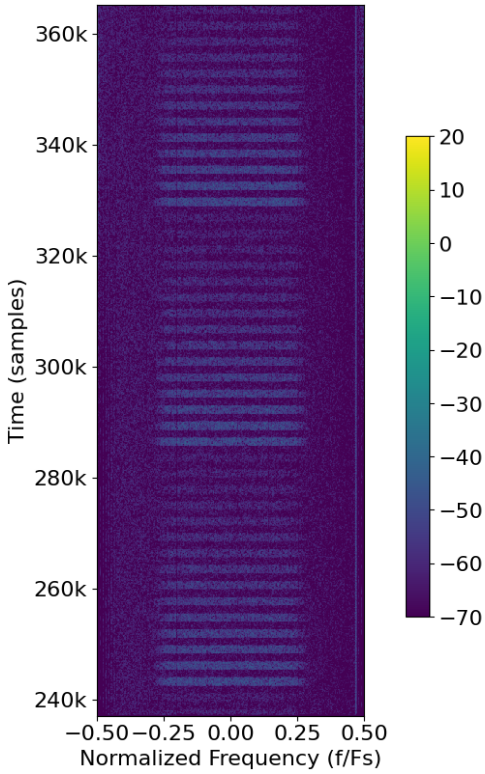
array at the time of data capture, which resulted in the received antennas having different effective gains. Although the PER curves for each captured element are similar to the simulated elements, their combined curves have a performance difference of almost 2 dB.

This performance difference is also seen when the interference is removed (Fig. 5.11b). In the simulated case, the PER curve (orange triangles) is within 0.5 dB of that of the combined without interference curve (orange diamonds), which is aligned with the performance bound for 2 antennas. This suggests the algorithm, with the current parameters, has difficulty removing all of the interference without impacting the integrity of the packet. Similarly, in the hardware case, when the interference is removed, the resulting PER (green triangles) has, at most, a 0.5 dB performance degradation in comparison to the combined case with no interference (green diamonds). Since both the simulated and captured cases involving interference removal perform nearly as well as when the elements are combined when no interference is present, it is likely that the 2 dB performance difference between the simulated and captured cases is due to the difference between the simulated and captured scenarios.

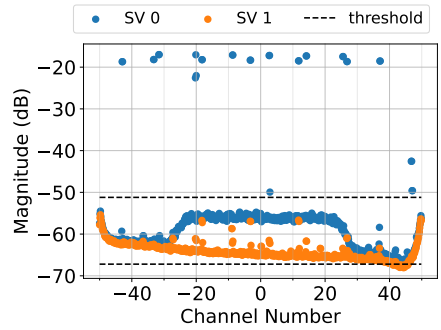
There are several possibilities that could account for the degradation in PER performance. One issue is the uneven noise floor (see Fig. 5.10d). Another could be that SOI is in the same eigenspace as the interference. In this case, there are too few degrees of freedom (too few sensors) to suppress the interference, so the spectrum would need to be notched. Likewise, if SOI is in a similar eigenspace as the interference, then the degrees of freedom available are insufficient. In order to suppress the interference, SOI would also be suppressed. To mitigate this issue, additional elements could be added to the array, and equalization before synthesis could be added to the process. Finally, if the interference lies in the transition band between channels, it gets aliased and potentially "sneaks through" the thresholds. This can be mitigated by using a different number of sub-bands.



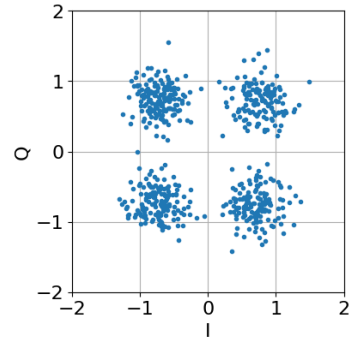
(a) Spectrum as seen by Rx 0



(b) Output of SVIRA

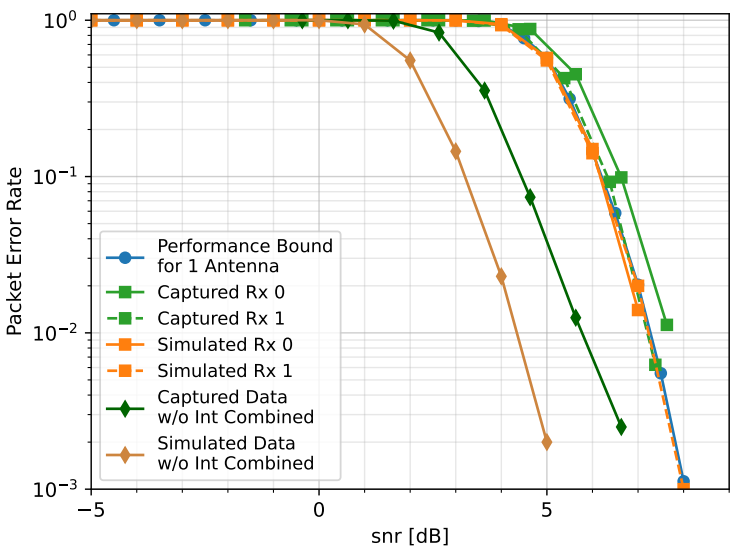


(d) Singular Values, (Block 2, Samples 237k-265k)

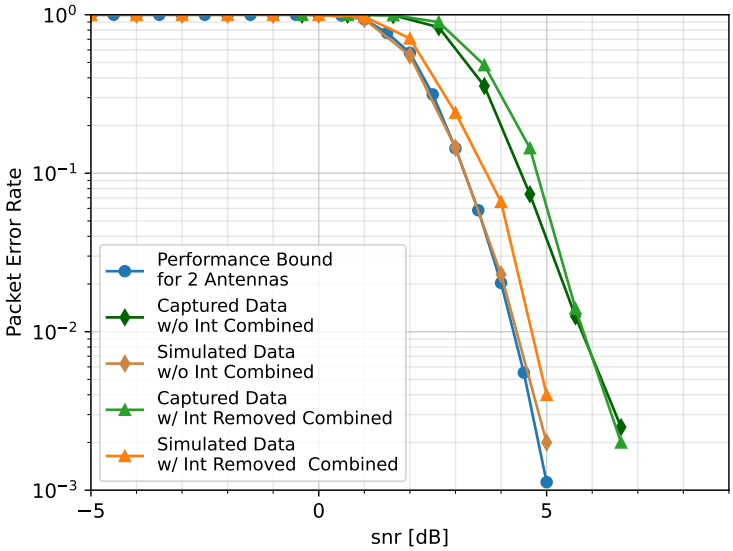


(e) Constellation of Frame 64 with a gain of -12 dB

Figure 5.10: Captured Data from File 41 with $M = 800$, $2L = 320$, and thresholds of -1 dB to 15 dB



(a) PER for no interference



(b) PER when 13 interfering tones are removed

Figure 5.11: PER curves for two interference sources emitting 13 tones

Chapter 6

Summary and Conclusions

In this thesis, a novel beamforming approach to interference removal in spectrally congested environments is proposed. A mathematical description of the problem space along with a survey of traditional and blind beamforming approaches is discussed in Chapter 2. Chapter 3 outlines the proposed Singular Value Interference Removal Algorithm (SVIRA) as a novel sub-band beamformer that combines polyphase filterbank (PFB) channelizers with singular value decomposition (SVD). An overview of parameter selection and evaluation of the simulated performance in relation to a minimum variance distortionless response (MVDR) beamformer is provided in Chapter 4. Finally, Chapter 5 presents the verification of SVIRA in a wireless laboratory over-the-air experiment conducted with a two-element array.

6.1 Conclusions

In simulation, SVIRA can achieve PER performance on par with that of MVDR. When there is no interference, SVIRA is able to combine the sensors in such a way that the packet error rate (PER) perfectly aligns with the theoretical performance bound. Captured data from the two-element array laboratory experiment verifies that the proposed beamforming approach achieves a packet error rate within half a decibel of the theoretical performance bound of receiving packets in AWGN. Although SVIRA performs about 1 dB worse than the performance bound when there are two sources of interference, SVIRA is able to achieve PER

performance similar to the case when no interference is present by removing the narrowband interferers, with a performance loss of at most 0.5 dB. Thus, SVIRA can remove interference in such a way that there is minimal PER performance loss in comparison to when there is no interference present in the band.

6.2 Future Work

Potential future work includes:

- Completely automating the singular value (SV) selection process.
- Adding an equalization to each sub-band to mitigate energy lost from removing interference with a similar direction of arrival (DOA) as signal of interest (SOI), and possibility introduce methods for intersymbol interference (ISI) suppression.
- Extensively investigate the SVIRA's ability to handle wideband noise interference.
- Investigate the algorithm's behavior when there are more sources of interference in a laboratory environment and why it exhibited lower than expected performance in one of the over-the-air experiments when there were only two interference sources.
- Perform tests to see how SVIRA does when the sensors are not calibrated such as the sensors receiving samples at different times, phase offsets, different receiver gain on each sensor, etc.

Bibliography

[1] Federal Communications Commission. Authorization of spread spectrum systems under parts 15 and 90 of the fcc rules and regulations. https://web.archive.org/web/20070314063902/http://www.marcus-spectrum.com/documents/81413R0.txt, 1985. Accessed:2023-03-28.

[2] M. R. Williamson, W. C. Headley, W. H. Clark, J. McCollum, T. Krauss, L. Lusk, D. Jenkins, T. Villemez, M. O. Moore, D. J. Jakubisin, A. Poetter, A. J. Michaels, A. A. Beex, and J. D. Gaeddert. Multi-antenna pre-processing for improved rfml in congested spectral environments. In *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 288–295, 2021. doi: 10.1109/DySPAN53946.2021.9677275.

[3] Jack Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408–1418, 1969.

[4] Sergiy A. Vorobyov. Principles of minimum variance robust adaptive beamforming design. *Signal Processing*, 93(12):3264–3277, 2013. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2012.10.021. URL https://www.sciencedirect.com/science/article/pii/S0165168412003830. Special Issue on Advances in Sensor Array Processing in Memory of Alex B. Gershman.

[5] Xin Song, Jinkuan Wang, Yinghua Han, and Yanbo Xue. A recursive algorithm to robust adaptive beamforming and diagonal loading. In *2006 6th World Congress on Intelligent Control and Automation*, volume 1, pages 2268–2271. IEEE, 2006.

[6] Yipeng Ding and Zhengmin Li. A blind beamforming algorithm based on time-frequency

analysis technology. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 367–371. IEEE, 2021.

[7] Joon-Ho Lee, Yeong-Seok Jeong, Sung-Woo Cho, Woon-Young Yeo, and Kristofer SJ Pister. Application of the newton method to improve the accuracy of toa estimation with the beamforming algorithm and the music algorithm. *Progress In Electromagnetics Research*, 116:475–515, 2011.

[8] TS Ghouse Basha, PV Sridevi, and MN Giri Prasad. Beam forming in smart antenna with precise direction of arrival estimation using improved music. *Wireless Personal Communications*, 71:1353–1364, 2013.

[9] IMustaq Ahmed, BA Anirudh Koushik, GM Charan Kumar, and S Neethu. Simulation of direction of arrival using music algorithm and beamforming using variable step size lms algorithm. In *2022 IEEE Microwaves, Antennas, and Propagation Conference (MAPCON)*, pages 993–997. IEEE, 2022.

[10] Wei Liu. Blind beamforming for multi-path wideband signals based on frequency invariant transformation. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 1–4, 2010. doi: 10.1109/ISCCSP.2010.5463492.

[11] Zbyněk Koldovskỳ, Petr Tichavskỳ, and Václav Kautskỳ. Orthogonally constrained independent component extraction: Blind mpdr beamforming. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1155–1159. IEEE, 2017.

[12] BK Imtiyaz Ahmed and Fathima Jabeen. Implementation adaptive beamforming for qam signals using constant modulus algorithm for smart antenna. *IOSRD Journal of Engineering ISSN No*, pages 1–6, 2016.

[13] Thomas Eireiner, Thomas Muller, J-F Luy, and Frank Owens. Implementation of a

smart antenna system with an improved ncma algorithm. In *IEEE MTT-S International Microwave Symposium Digest, 2003*, volume 3, pages 1529–1532. IEEE, 2003.

[14] BK Imtiyaz Ahmed and Fathima Jabeen. Blind adaptive beamforming simulation using ncma for smart antenna. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 492–495. IEEE, 2017.

[15] Boya Qin, Yunlong Cai, Benoit Champagne, and Minjian Zhao. Complexity reduction techniques for blind adaptive beamforming in ofdm antenna arrays. In *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2014.

[16] Vignesh Ranganathan, G Prabha, and KA Narayanankutty. Constant modulus hybrid recursive and least mean squared algorithm performance comparable to unscented kalman filter for blind beamforming. In *2016 IEEE Annual India Conference (INDICON)*, pages 1–4. IEEE, 2016.

[17] Brain G Agee, Stephan V Schell, and William A Gardner. Spectral self-coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays. *Proceedings of the IEEE*, 78(4):753–767, 1990.

[18] Wei Zhang and Wei Liu. Low-complexity blind beamforming based on cyclostationarity. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2012.

[19] Ji-Hyeon Kim, Young-Kwang Seo, Soon-Young Kwon, Hyoung-Nam Kim, Jin-Oh Park, Hyun Jin Kang, Jae Yun Kim, and Byung Ho Mun. Blind beamforming based on multi-target score with a dmp algorithm. In *International Convference on Electronics, Information and Communication (ICEIC)*, 2019.

[20] Liang Zhang, Bin Liao, Lei Huang, and Chongtao Guo. An eigendecomposition-based approach to blind beamforming in a multipath environment. *IEEE Communications Letters*, 21(2):322–325, 2016.

[21] Zhuang Xie, Chongyi Fan, Jiahua Zhu, Yajing Deng, and Xiaotao Huang. Improved blind beamforming for multipath signals in complex interference environment. In *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, pages 639–643. IEEE, 2020.

[22] Bin Zhao, Jun-An Yang, and Min Zhang. Research on blind source separation and blind beamforming. In *2005 International Conference on Machine Learning and Cybernetics*, volume 7, pages 4389–4393 Vol. 7, 2005. doi: 10.1109/ICMLC.2005.1527711.

[23] Cheng Wang and Jun Tang. Blind beamforming technique for reception of multipath coherent signals. *IEEE Communications Letters*, 20(7):1453–1456, 2016.

[24] Parham Ramezanpour, Mohammad Javad Rezaei, and Mohammad Reza Mosavi. Deep-learning-based beamforming for rejecting interferences. *IET Signal Processing*, 14(7): 467–473, 2020.

[25] Moon Ju Jo, Geon Woo Lee, Jung Min Moon, Choongsang Cho, and Hong Kook Kim. Estimation of mvdr beamforming weights based on deep neural network. In *Audio Engineering Society Convention 145*, Oct 2018. URL http://www.aes.org/e-lib/browse.cfm?elib=19794.

[26] Hansol Kim, Kyeongmuk Kang, and Jong Won Shin. Factorized mvdr deep beamforming for multi-channel speech enhancement. *IEEE Signal Processing Letters*, 29:1898–1902, 2022. doi: 10.1109/LSP.2022.3200581.

[27] Nikolaos I. Miridakis and Dimitrios D. Vergados. A survey on the successive interference

cancellation performance for single-antenna and multiple-antenna ofdm systems. *IEEE Communications Surveys & Tutorials*, 15(1):312–335, 2013. doi: 10.1109/SURV.2012.030512.00103.

[28] Ryuji Kohno. Spatial and temporal communication theory using adaptive antenna array. *IEEE personal communications*, 5(1):28–35, 1998.

[29] Kai Wang, Ling Wang, and Chuang Han. Joint space and time domain processing for doa estimation. In *2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–4. IEEE, 2015.

[30] Yimin Zhang, Kehu Yang, Moeness G Amin, and Yoshio Karasawa. Performance analysis of subband arrays. *IEICE transactions on communications*, 84(9):2507–2515, 2001.

[31] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1998.

[32] fredric j. harris. *Multirate Signal Processing for Communication Systems*. River Publishers, 2 edition, 2021.

[33] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*, chapter 11, pages 794–795. Pearson Education, 4 edition, 2007.

[34] John G. Proakis and Masoud Salehi. *Digital Communications*, page 1087. McGraw Hill Education, 5 edition, 2018.

[35] Bernard Kolman and David R. Hill. *8.2 Spectral Decomposition and Singular Value Decomposition*, page 488–500. Pearson Education, 2008.

[36] Roger A. Horn and Charles R. Johnson. *Matrix analysis*, chapter 1, page 234–239. Cambridge University Press, 2 edition, 2013.

[37] Joseph D. Gaeddert. liquid-dsp, a free and open-source digital signal processing library for software-defined radios. https://liquidsdr.org, 2023. Accessed: 2023-02-16.

[38] National Instruments. USRP E-320. https://www.ettus.com/all-products/usrp-e320/, 2023. Accessed: 2023-04-6.