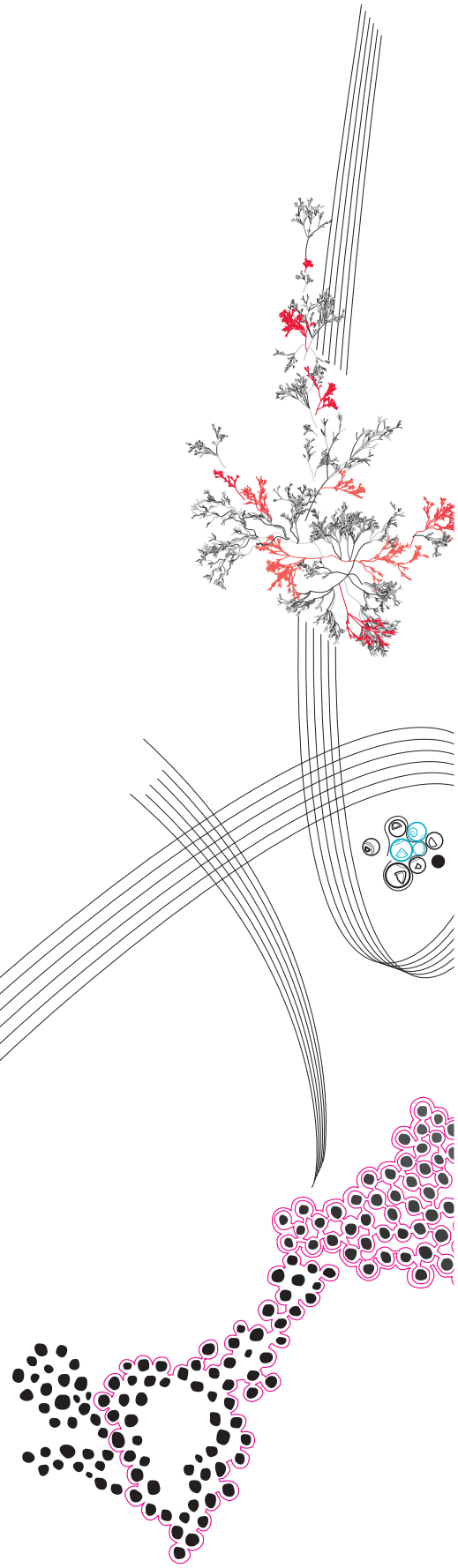


MSc Embedded Systems
Final Project

Performance and Computational Complexity of Wideband Sound Source Localization Algorithms: A Comprehensive Analysis

Joshua Aäron de Bie

Supervisors:
Prof. Dr. Ir. André Kokkeler
Dr. Ing. Anastasia Lavrenko
Dr. Ir. Nikolaos Alachiotis
Dr. Ir. Chris Hellenthal
Ir. Marco Kruijswijk
Ir. Gerard Bos

May , 2024

Department of Radio Systems
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Acknowledgements

I would like to express my gratitude to my supervisors André Kokkeler, Anastasia Lavrenko, Nikolaos Alachiotis, Chris Hellenthal, Marco Kruijswijk, and Gerard Bos for their time, wisdom, and guidance, which were instrumental in making this thesis possible. Additionally, I extend my heartfelt thanks to my family for their unwavering support throughout my entire academic career and in all my endeavors.

Contents

Abstract

Microphone arrays are arrangements of microphones that are used to capture auditory signals within an environment. Due to the difference in the microphone's positions the signals from a signal source present in the environment will arrive at different points in time depending on the microphone's position within the array. Sound source localization (SSL) utilizes the time difference of arrival (TDOA) to extract useful information, with the primary goal being to determine the direction-of-arrival (DOA) of signal sources present in the environment. SSL algorithms are either narrowband or wideband. Narrowband algorithms focus on a narrow frequency band and are therefore only able to locate signal sources that fall within this frequency band. Wideband algorithms observe a much larger frequency band, which allows them to detect multiband signal sources or several narrowband signals emitting signals at different frequencies. Furthermore, wideband algorithms can be used to estimate the frequencies of these signal sources. Although, these wideband algorithms show additional capabilities when compared to narrowband algorithms, their usages are limited due to their increased computational complexity. This increase can however be mitigated by employing parallel processing techniques.

In this paper, four different wideband SSL algorithms will be explored to determine their accuracy, precision and computational complexity. These algorithms are CLEAN, SRP-PHAT, 2D-MUSIC and 2D Unitary ESPRIT. First each algorithm's implementation is described. Next, the computational complexity is derived for both the sequential and a parallelized implementation of each algorithm. Lastly, simulations are performed to evaluate the performance of each algorithm for both single and multiple coherent narrowband signal sources. The results show that the SRP-PHAT and CLEAN algorithm perform well when subjected to a single narrowband signal source even when subjected to noise and reverberation. However, when comparing the algorithms' performance when subjected to multiple coherent sources, only SRP-PHAT maintains consistent performance. Even when subjected to high noise levels and reverberation. Additionally, its parallelized computational complexity belongs to one of the lowest of all the algorithms considered in this paper.

Keywords: sound source localization (SSL), wideband signals, data dependency graphs (DDG), parallel processing techniques, computational complexity, real-time processing

Nomenclature

λ	Loop gain
λ_i	Eigenvalue of the i-th source incident on the array
$\mathbf{G}$	Steering matrix
$\mathbf{g}$	Steering vector
ω_k	Center frequency of the k-th frequency bin
C	Cross-spectral matrix/Covariance matrix
c	Speed of sound in dry air at $20°C$
d_x	Distance between adjacent elements in microphone array in the x-direction
d_y	Distance between adjacent elements in microphone array in the y-direction
f_s	Sampling frequency of each element in the array
K	Number of frequency bins processed by the SSL algorithms
M	Number of microphones in the array
M_x	Number of columns of elements in microphone array in the x-direction
M_y	Number of rows of elements in microphone array in the y-direction
N	Number of snapshots within a time-frequency bin
P	Number of signal source to be found by the SSL algorithsm
u_i	Direction cosine of the i-th source incident on the array in the direction of u
v_i	Direction cosine of the i-th source incident on the array in the direction of v

Chapter 1

Introduction

In the realm of audio signal processing, sound source localization (SSL) stands as a pivotal technique akin to the human sense of hearing, enabling machines to perceive and analyze auditory scenes. At its essence, SSL refers to the process of determining the spatial coordinates of sound-emitting sources within an environment. Just as our ears triangulate the direction of sound, SSL devices utilize multiple microphones or sensors to pinpoint the origin of acoustic signals. These sensors are often organized in a geometric pattern, including rectangular, circular, spherical and spiral patterns. Such arrangements of microphones are also referred to as *microphone arrays*.

The applications of SSL span a wide spectrum of domains, ranging from robotics [69, 41, 53], structural analysis [38, 27, 50, 28], surveillance systems, [77, 45, 64, 74], teleconferencing scenarios [35, 2], hearing aids [46, 34, 52, 21], and leak detection [83, 8] to mention a few. In robotics, for instance, accurate localization of sound sources enables robots to respond intelligently to auditory cues, enhancing their interaction capabilities with humans and their environment. In surveillance systems, SSL aids in identifying and tracking potential threats or anomalous activities in crowded or noisy environments. In teleconferencing scenarios and hearing aids, precise localization of speakers facilitates clearer audio communication by focusing on the speaker of interest and suppressing background noise. In industrial environments that work with compressed air, vacuum, or gas systems leaks can occur which can damage machinery and create hazardous working environments. Such leaks exhibit acoustic characteristics that can be detected with specially designed devices which employ SSL techniques to locate and analyze the characteristics of such leaks.

Despite being a longstanding and extensively researched topic, SSL remains a formidable challenge in the field. The propagation of sound is influenced by several factors, such as noise and reverberation. These environmental elements add complexity to the task of distinguishing between authentic sound sources and other auditory phenomena. Noise poses a significant challenge as it can mask the presence of genuine sound sources, while reverberation can influence the location at which the signal source is estimated to be located at. Additionally, in scenarios where multiple sound sources are present, these sources can interfere with each other, adding additional complexity to the spatial characteristics in the environment. When dealing with coherent sources, which are sources that exhibit similar characteristics in their intensity and frequencies, the task of isolating the desired source locations becomes particularly challenging.

At the core of SSL devices lies an audio processing algorithm which takes the received microphone signals from the microphone array and processes these to provide the spatial coordinates of the sound sources. Depending on the algorithm, more information such as

Algorithm	**References**
CLEAN/CLEAN-PSF	[73, 18, 10, 79, 5]
CLEAN-SC	[73, 18, 10, 79]
High-Resolution CLEAN-SC	[73]
GCC-PHAT	[68, 7, 51, 16, 84, 12]
SRP-PHAT	[16, 84, 85, 12]
MUSIC	[83, 7, 51, 66, 59, 58]
GEVD-MUSIC	[51]
GSVD-MUSIC	[51]
SEVD-MUSIC	[51]
ESPRIT	[66, 59, 58]
DAMAS	[49, 5, 18, 76, 10, 79]
DAMAS2	[49, 18, 76, 10]
DAMAS3	[18]
NNLS	[10, 79]
FFT-NNLS	[49, 18, 10, 79]
TBD	[20]
SF-MCA	[9]
BSI	[68]
NMCFLMS	[68]
NSM	[7]
TC-NSM	[7]
FPL	[16]
SRPD	[85]
FISTA	[49, 79]
FBM	[46]
SBM	[46]
MWCM	[46]
BSS	[47, 55]
Least-squares estimators	[31, 47]
Bayes-based Richardson–Lucy (RL)	[79]

TABLE 1.1: Subset of available SSL algorithms

the signal source's frequency and intensity can also be extracted. Some SSL algorithms utilize the TDOA originating from the difference in positions of each microphone in the microphone array to triangulate the DOAs of the incoming sound waves, allowing them to determine the location in both the azimuth and elevation direction relative to the microphone array. Other algorithms employ beamforming techniques by constructing spatially selective beams in the direction of the sound sources, enhancing the signal-to-noise ratio (SNR) of desired sources while suppressing interference from other directions. More recently, machine learning-based methods have been explored which employ deep learning and neural networks to learn the complex patterns and relationships from large datasets, enabling robust and accurate localization in diverse environments. Table 1.1 provides only a subset of the many algorithms developed for SSL.

When locating sound sources, it is important to determine what type of signal sources need to be analyzed. Signal sources can either be *narrowband* or *wideband*. Narrowband auditory signals have a small bandwidth, meaning that they occupy a narrow range of frequencies in the audio spectrum. These signals typically consist of a single frequency or

a narrow band of frequencies around a specific carrier frequency. Wideband signals, as the name suggests, have a larger bandwidth, and encompass a broader range of frequencies compared to narrowband signals.

Narrowband SSL algorithms are designed to locate narrowband signals. For this purpose, the algorithm will require knowledge on the carrier frequency of the signal which it attempts to locate. This requires knowledge of the signal sources before performing the actual localization, which might not always be available. Wideband SSL algorithms analyze a wider frequency spectrum of the received audio signals and consider multiple frequency components present in the wideband signals to estimate the spatial coordinates of these sound sources. However, wideband algorithms feature a higher computational complexity than narrowband algorithms.

Narrowband SSL algorithms can sometimes be adapted to implement wideband characteristics. This adaptation involves applying the narrowband version of the algorithm with a window function. The window function is applied to the received microphone signals, creating separate segments of signal data. To reduce artifacts at the segment boundaries, some overlap is included between these segments. Each of these segments is Fourier transformed and the resulting magnitudes and phases for each point in time and frequency, which will be referred to as a *frequency bin* in this thesis, is stored in a complex-valued matrix. This is the basic principle behind the Short-Time Fourier Transform (STFT).

By applying the narrowband SSL algorithm to each of the frequency-bins obtained through STFT, the algorithms will yield spatial location estimates for each of the frequency bins. Depending on the algorithm, these estimates are used to find multiple wideband sources or narrowband sources. Although such an approach can lead to a better spatial resolution and additional capabilities such as signal source frequency estimation, it will come at the price of a high computational complexity.

Algorithms that feature a high computational complexity might become impractical to use with real-time systems. For such systems it is crucial that sound signals are processed within strict time constraints to ensure timely response and decision-making. However, achieving this in the context of SSL, where algorithms must analyze incoming audio data to determine the direction of sound sources, presents a significant challenge. In response, researchers have developed algorithms such as ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) and FPL (Fast Phased-Locked Loop) to address this challenge. These algorithms prioritize computational efficiency without compromising localization accuracy. By leveraging innovative techniques such as rotational invariance and efficient signal processing methods, these algorithms reduce the computational burden associated with SSL, making them well-suited for real-time applications.

To mitigate the computational complexity one can make use of parallel processing techniques to perform equivalent operations in parallel. This requires determining which processes exist within the algorithm and how these are dependent on each other. To help identify opportunities for parallelization, it is possible to make use of data dependency graphs (DDG). These graphs are constructed of smaller elements called processing elements (PE) which describe a single task. PEs that are not connected are independent of each other and can be performed in parallel. Therefore, these graphs illustrate how an operation can be performed in parallel by visualising the data dependencies present in the operation. These DDGs can be transformed into signal flow graphs (SFGs) and eventually into systolic array designs to develop dedicated hardware. Additionally, the DDGs can assist in creating software implementations of the algorithms. These software implementations can prove to be effective in exploiting the capabilities of platforms with parallel processing capabilities such as GPUs, FPGAs and multi-core CPUs.

Research goal

This thesis has received backing from Benchmark which provided the research topic of this thesis. Benchmark stands as a multinational entity specializing in engineering services, offering a diverse array of solutions to its clientele. Its offerings include design engineering, precision machining, comprehensive electronic assembly services, and lifecycle management support. Such provisions play a pivotal role in aiding customers across various sectors in the advancement of technologies pertinent to defense, medical, aerospace, complex industrial, communication, advanced computing, and semiconductor capital equipment domains.

A real-life example of products that utilize SSL techniques are the ii900-Series of acoustic imagers developed by the Fluke Corporation and Benchmark. These devices leverage SSL techniques to enable its users to visually inspect environments in real-time for leaks, partial discharges (PD), and mechanical issues in a non-intrusive and safe manner.

At the core of such products lies an algorithm which enables it to perform SSL. As shown in Table 1.1, many SSL algorithms have been developed over the years. However, this makes it a time-consuming process to find the optimal algorithm for the desired application. Therefore, the goal of this research will be to perform a comparison of four different SSL algorithms to determine how they compare to one another. The algorithms are compared based on their accuracy, precision, and computational complexity. These three performance metrics are described in the list below.

1. **Accuracy** accuracy is determined by the distance between the estimation signal source position given by each algorithm and the actual source positions.

2. **Precision** is determined by the algorithm's ability to estimate the signal source position consistently which is represented by the standard deviation in its estimated signal source positions.

3. **Computational complexity** indicates how well the algorithms total number of operations that need to performed scales with the size of the input.

The CLEAN algorithm is the first SSL algorithm that will be investigated in this study and serves as the reference for the other algorithms. The second algorithm is SRP-PHAT. This algorithm has proven to be a powerful technique for SSL using microphone arrays. It stands out for its ability to accurately localize sound sources even when subjected to noisy and reverberant environments. Additionally, the SRP-PHAT algorithm offers a high spatial resolution, enabling the precise localization of sound sources. Although the algorithm has a high computational complexity, it can be implemented efficiently which makes it suitable for real-time processing applications. The third algorithm is 2D-MUSIC. This algorithm has been widely acclaimed for its ability to accurately determine the parameters of multiple signal sources even when subjected to noise. It provides a high resolution in identifying and localizing signals that are closely spaced in frequency. However, the algorithm has a high computational complexity which makes it less suitable for real-time applications. Hence the last algorithm that will be considered is the 2D Unitary ESPRIT algorithm. The 2D Unitary ESPRIT algorithm is a computationally efficient alternative to MUSIC for estimating signal parameters in array signal processing applications, making it particularly attractive for real-time or resource-constrained systems. Furthermore, the ESPRIT algorithm has been stated to have a high estimation accuracy and resolution comparable to MUSIC. It achieves this by effectively utilizing the inherent structure of the microphone array geometry to estimate the signal parameters. Lastly, the algorithm robustness allows it to provide accurate estimates even in noisy environments.

Using the three performance metrics three research questions are defined which concretely state the goal of this research.

1. Which of the selected algorithms achieves the highest accuracy and precision for a single signal source when subjected to noise and reverberation?

2. Which of the algorithms can estimate the direction of multiple coherent signal sources?

3. What are the sequential and parallelized computational complexities of each algorithm?

This study abides by the following structure. Each of the selected SSL algorithms will be described in more detail in Chapter 2. Their computational complexities are explored in Chapter 3 and parallel processing techniques are applied to determine how each algorithm can be parallelized. Simulations are performed to determine the accuracy and precision of each algorithm when locating a single and two coherent signal sources. The results from these simulations are presented individually for each algorithm in Chapter 4. In Chapter 5 the obtained computational complexities and simulation results for each algorithm are compared to each other to determine what the strengths and weaknesses of each algorithm are. Finally, the study is concluded in Chapter 6 and additional research directions are presented for future research.

Chapter 2

Sound Source Localization Algorithms

SSL algorithms seek to extract useful information from the signals received by a set of microphones. Additionally to estimate the DOA information, algorithms can be used to estimate the intensity of the signal, as well as its frequency. CLEAN, SRP-PHAT and 2D-MUSIC are algorithms that construct a spectrum by computing the received signal strengths for a predetermined set of scan locations. By selecting the maxima from the spectrum, the DOA's are found. 2D Unitary ESPRIT however, does not use a spectrum to locate its peaks and instead directly derives the DOA's from eigenvectors obtained through eigendecomposition. In this chapter, the common dependencies for each algorithm will be discussed first, followed by a separate and detailed exploration of each algorithm.

2.1 Microphone array

The microphone array is a collection of microphones arranged in a specific configuration to capture audio signals from various directions. Many two-dimensional as well as three-dimensional configurations exist for microphone arrays each with different properties. Some common configurations include, linear [82, 56, 61], rectangular [54], circular [32], spiral [57] and spherical [42] arrays. This paper will only consider an uniform rectangular array (URA) containing 25 elements with inter-element spacing of five centimeters in both the x- and y-direction, which are denoted as d_x and d_y respectively. The exact arrangement is given in Figure 2.1.

The inter-element spacing and its sampling rate are two important properties of the microphone array as it determines the frequencies that the microphone can effectively capture [17]. The Nyquist-Shannon criterion states that to accurately capture and reconstruct a signal, the sampling rate must be at least twice the highest frequency component present in the signal. Sampling at a lower frequency results in **time aliasing**. The frequency that the microphone array can effectively capture is known as the Nyquist frequency. For the URA in Figure 2.2 a sample rate of 20 kHz was utilized, meaning that its Nyquist frequency is equal to 10 kHz.

The inter-element distance determines the cutoff frequency at which **spatial aliasing** [6] occurs. The cutoff frequency, f_c, can be calculated according to Equation (2.1).

$$f_c = \frac{c}{2\Delta d} \tag{2.1}$$

Here, c is the speed of sound and Δd is the inter-element distance. For the URA in Figure

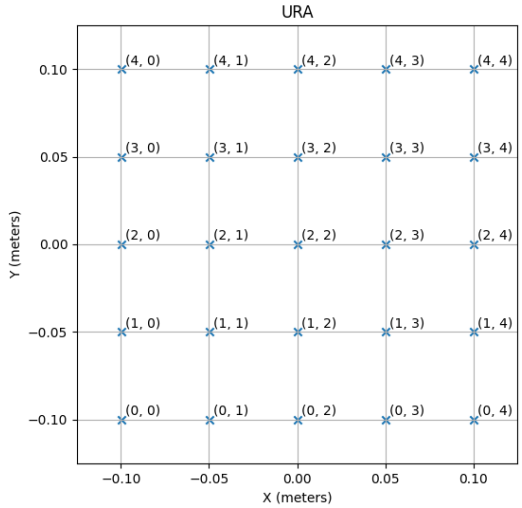


FIGURE 2.1: Arrangement of the URA

2.2, Δd is equal to 0.05 meters. Hence, the cutoff frequency f_c is equal to $\frac{343.0}{2 \cdot 0.05} = 3430$ Hz.

Due to the geometric positioning of each element and depending on the direction of the incoming signals, the time-of-arrival (TOA) of a signal will differ at each microphone. SSL algorithms exploit this TDOA to determine the DOAs of the signals incident on the microphone array. The size of a microphone array is important as it determines the array's minimum resolvable separation angle between two sound sources. This can be formally described by what is known as the Rayleigh criterion [62]. Mathematically, the Rayleigh criterion can be presented as shown in Equation (2.2).

$$\theta_{min} \approx 1.22 \frac{\lambda}{d} \tag{2.2}$$

In the context of SSL, the criterion defines the minimum resolvable angle θ in radians, at which two sound sources, each producing sound waves with a wavelength λ, can be resolved by a microphone array with diameter d.

2.2 Steering vectors

The algebraic representation of the received signal $\mathbf{x}$ at time t for each of the M microphones present in the microphone array is given in Equation (2.3).

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \in \mathbb{C}^M \tag{2.3}$$

The source signals emitted from P different signal sources is represented by the signal vector $\mathbf{s}$. The additive noise present at each microphone is considered to be uncorrelated and is represented by $\mathbf{n}$. Lastly, $\mathbf{A} \in \mathbb{C}^{M \times P}$ represents the steering matrix, containing the M-dimensional steering vectors for each of the P sources. Each steering vector describes how the signal for a source propagates to each microphone in the array. Steering vectors are crucial to many beamforming methods and are used to focus the microphone array to a specific direction by steering the main lobe of the beamformer. The steering vector represents the phase relationship between the elements in the array and different DOAs.

The steering vectors essentially hold the phase shifts that will maximize the signal strength in the specified direction while minimizing interference from other directions. These phase shifts are derived from the geometry of the array, the DOA of interest and the frequency of interest. For a URA where its elements are uniformly distributed along the x- and y-axis, the phase shifts for each microphone in the array in the direction of (θ, ϕ) is given by Equation (2.4).

$$\mathbf{g}(\theta, \phi, \omega_k) = \exp\left(\frac{-j2\pi\mathbf{m}_x d_x \omega_k}{c}\sin(\theta)\cos(\phi) + \frac{-j2\pi\mathbf{m}_y d_y \omega_k}{c}\sin(\theta)\sin(\phi)\right) \tag{2.4}$$

In this equation, d_x and d_y represent the inter-element spacing of the microphones in the x- and y-direction respectively and $\mathbf{m}_x$ and $\mathbf{m}_y$ are the vectors containing the x- and y-indices of each microphone in the array, respectively. Lastly, ω_k, θ, and ϕ represent the frequency, azimuth angle and elevation angle that focus the beamformer towards the desired direction. There exist many different methods to define the steering vector. However, a distinction is made between far-field and near-field sources. Far-field sources are best represented by assuming plane waves, whereas near-field sources are better represented by assuming spherical waves. In this paper, the steering vectors defined in Equation (2.4) assume far-field sources.

Instead of computing a single steering vector, it is also possible to construct the *steering matrix* containing the steering vectors for each of the L possible scan directions. For this purpose, let us define the vectors

$$\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_{L-1}]$$

$$\boldsymbol{\phi} = [\phi_0, \phi_1, \ldots, \phi_{L-1}]$$

which contain the azimuth and elevation angles of each of the L scan directions respectively. Then the steering matrix $\mathbf{G}(\omega_k) \in \mathbb{C}^{L\times M}$ is obtained by substituting the vectors into Equation (2.4).

$$\mathbf{G}(\omega_k) = \exp\left(\frac{-j2\pi\mathbf{m}_x d_x \omega_k}{c}\sin(\boldsymbol{\theta})\cos(\boldsymbol{\phi}) + \frac{-j2\pi\mathbf{m}_y d_y \omega_k}{c}\sin(\boldsymbol{\theta})\sin(\boldsymbol{\phi})\right) \tag{2.5}$$

2.3 Scan space

The algorithms considered in this paper, with the exception of 2D Unitary ESPRIT, make use of a grid-based search in order to determine the DOA of incident signals. Each position in the grid represents the DOA of a signal with an associated intensity value. For this purpose, a scan space needs to be defined which determines the DOA's for which the intensity value will be computed. For each algorithm, the maximum range a scan space can encompass can be defined as $\theta_i \in [-180, 180]$ and $\phi_i \in [0, 90]$. Naturally, it is possible to to choose smaller ranges to reduce the scan space, which can be beneficial in reducing the computational load of the algorithm as will become clear in the next sections where each algorithm's components and their computational complexity are explored in depth. In this paper a reduced scan space is utilized where the elevation angle was limited to $\phi_i \in [0, 50]$ degrees.

2.4 Wideband SSL

Wideband SSL algorithms have a similar goal to narrowband SSL algorithms. Both variants seek to estimate the direction of a sound source relative to the microphone array. However,

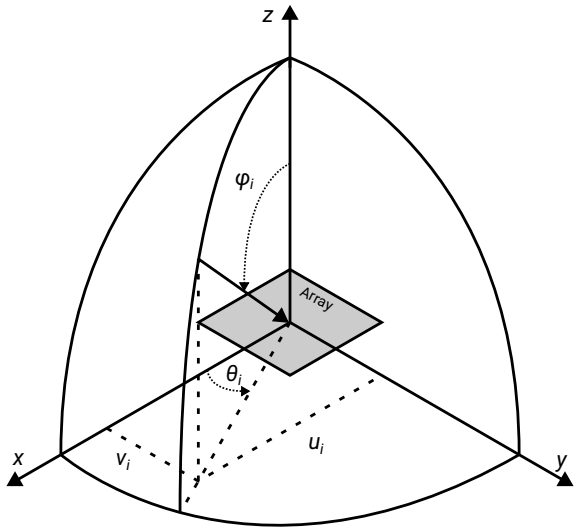


FIGURE 2.2: Illustration of a signal from a source at (θ_i, ϕ_i) incident onto the array.

narrowband algorithms are designed to work with signal sources that emit signals with narrow frequency bands and therefore assumes that the emitted acoustic sound signals fall with this small frequency band. Wideband SSL algorithms are designed to localize sound sources within a wider frequency band, hence the name *wideband*. However, they can also be used to estimate the DOA of narrowband source signals and additionally, their frequency. Wideband algorithms divide the incoming signals in different frequency bins and analyse each frequency bin independently. As mentioned in Chapter 1, this is achieved by pre-processing the received signals by the Short Time Fourier Transform (STFT) to obtain a set of frequency bins. The wideband implementation of each algorithms involves applying the narrowband implementation of the algorithm to each resulting frequency bin.

2.5 CLEAN

CLEAN is an algorithm which was originally developed by Jan Högbom in 1974 to perform deconvolution on images created in radio astronomy [29]. Deconvolution is the process of removing optical distortion from images which enhances the resolution and contrast of these images such that points of interest can more easily be identified. Though it was initially used for radio astronomy, many variations of CLEAN have been developed and applied to a wide range of scenarios, including SSL [13, 72]. In SSL, the CLEAN algorithm operates on an acoustic image obtained through conventional beamforming (CB). The acoustic image initially contains the source signals along with noise. Through an iterative process, the algorithm removes the parts from the acoustic image that correspond with the peak source until no significant peaks are left in the acoustic image. The size of the acoustic image is dependent on the resolution of the scan grid. If this grid is too coarse or out of focus, the error in estimating the peak location will increase [73].

Cross-spectral matrix

The Cross Spectral Matrix (CSM) (also known as the covariance matrix) is a convenient structure which stores all the cross-power densities of each microphone pair along with their complex conjugates for all frequencies of interest. Additionally, the auto-power spectra of each microphone is stored along the diagonal of the CSM. To construct the CSM let us

define the matrix $\mathbf{X}(\omega_k, t)$ which contains the power spectral density (PSD) of the received signals for frequency ω_k for each microphone in the microphone array at time t. Then the CSM for a single snapshot can be constructed as shown in equation 2.6.

$$\mathbf{C}_{k,t} = \mathbf{X}(\omega_k, t)\mathbf{X}(\omega_k, t)^H \in \mathbb{C}^{M \times M} \tag{2.6}$$

Here, H represents the Hermitian operator (also known as the Hermitian conjugate or complex conjugate transpose).

The signals received by the microphone array are observed over a period of time. In discrete time this means that the signals are represented by a consecutive series of N snapshots, each representing the signal at a single point in time. To construct the CSM for the whole sample period, a CSM must be created for each of these snapshots. Summing and averaging these CSMs will yield a single CSM that describes the cross-power densities over this sample period. The equation for this operation is shown in Equation (2.7).

$$\mathbf{C}_k = \frac{1}{N} \sum_{t=0}^{N} \mathbf{X}(\omega_k, t)\mathbf{X}(\omega_k, t)^H \in \mathbb{C}^{M \times M} \tag{2.7}$$

Conventional Beamforming

Conventional Beamforming (CB) involves utilizing the CSM and steering vectors to create an acoustic image by focusing the microphone array towards each of the L scan directions. Beamforming essentially filters the incoming signals for a specific DOA candidate by attenuating the signals that arrive from other directions than the DOA of interest. The resulting acoustic image holds the power estimates for each of the scan directions. Examples of acoustic images are provided in Figure 2.3. Equation (2.8) shows how a single power estimate is computed.

$$A_k(l) = \mathbf{w}_k(l)^H C_k \mathbf{w}_k(l) \tag{2.8}$$

Where, $\mathbf{w}_k(l)$ is the *weight vector* which is derived from the steering vector $\mathbf{g}_k(l)$ by dividing it by its Frobenius norm [1] as is shown in Equation (2.9)

$$\mathbf{w}_k(l) = \frac{\mathbf{g}(l, \omega_k)}{||\mathbf{g}(l, \omega_k)||} \tag{2.9}$$

With the obtained power estimates for each of the L scan directions, an acoustic image is created. The acoustic images provided in Figure 2.3 were created for an environment containing three narrowband signal sources located at various positions relative to the microphone array. They are of equally powered and located at equal distances from the microphone array. However, each signal source emits a signal at a different frequency. These frequencies are 2400, 2500, and 2600 Hz. The Figure shows that each signal source influence is strongest in their respective acoustic image. As the acoustic image for 2304.7Hz does not closely match any of the signal source's frequencies, it is nearly empty. Detecting the peak positions is essential in determining the locations of the signal sources, but can also be used to determine the frequency of the signals. This allows wideband algorithms to detect multiple signal sources even when placed closely together and if the difference in their frequencies is large enough.

Iterative peak removal

The initial acoustic images obtained with CB are referred to as *dirty maps* and the corresponding CSM $\mathbf{C}_k$ as the *dirty* CSM. This refers to the fact that these images represent

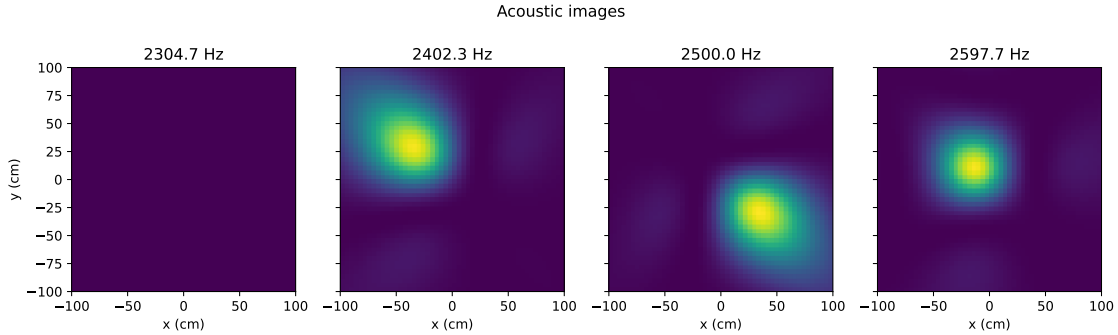


FIGURE 2.3: Acoustic images obtained through CB across multiple frequencies for three equipowered and sources emitting a 2400 Hz, 2500 Hz, and 2600 Hz signal respectively.

both the signal and noise incident onto the array. In the iterative peak removal process, the CLEAN algorithm performs the actual deconvolution by separating the contributions of the source signals from the noise present in the *dirty* CSM. The iterative peak removal process starts by selecting the maximum peak from initial *dirty map* and its corresponding power estimate $A_k(l)$. The array response in the direction of the l-th source is given as the outer product of the steering vector in that direction, with its conjugate transpose and scaled by the *loop gain*, λ as shown in Equation (2.10).

$$\mathbf{R} = \lambda \mathbf{g}(l, \omega_k)\mathbf{g}(l, \omega_k)^H \in \mathbb{C}^{M \times M} \tag{2.10}$$

The *loop gain*, λ, is the source power estimate $A_k(l)$ multiplied with a dampening factor [29] between 0 and 1. This is given in Equation (2.11)

$$\lambda = \zeta A_k(l), \quad 0 < \zeta \leq 1 \tag{2.11}$$

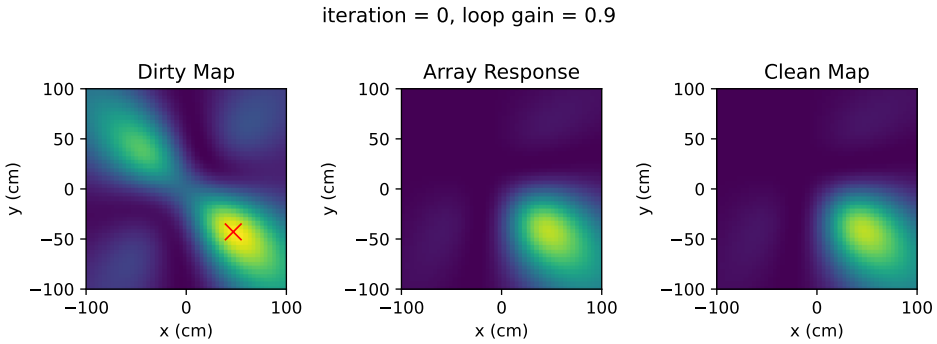
Here, ζ is the dampening factor. The array response is added to the *clean CSM*. This CSM is initially empty, but eventually will contain the sum of all the array responses corresponding with the peaks identified during the iterative peak removal process.

$$C_{\text{clean}} = C_{\text{clean}} + \mathbf{R} \tag{2.12}$$

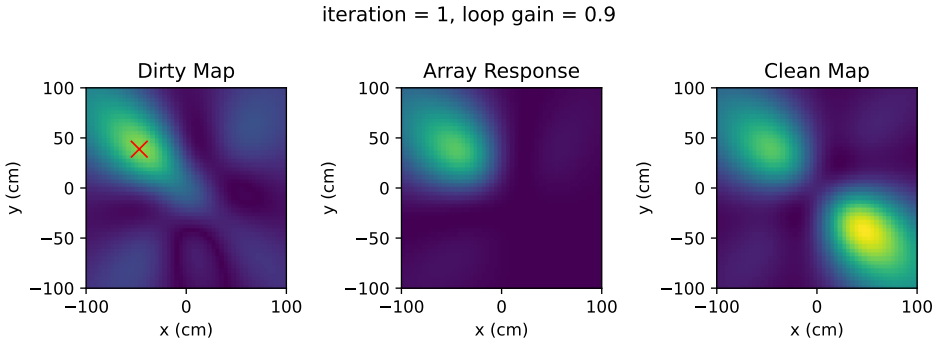
To detect the next peak in the dirty map, the array response is removed from the *dirty CSM* and the resulting CSM is used as input for the next iteration.

$$C_{\text{dirty}} = C_{\text{dirty}} - \mathbf{R} \tag{2.13}$$

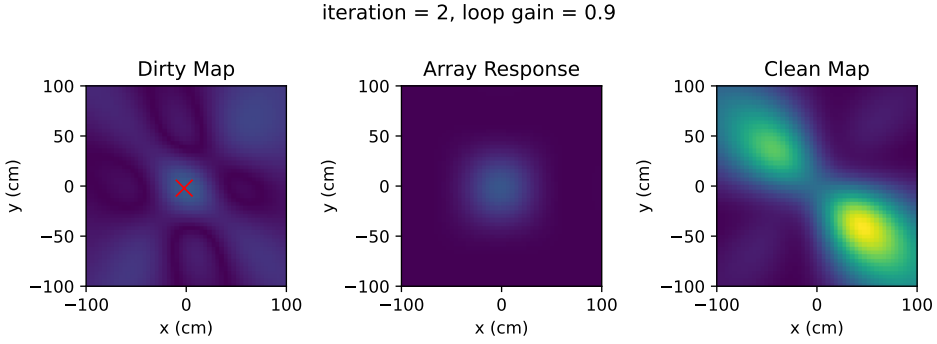
This process repeats until a certain threshold has been reached. For instance, in our implementation used for the simulations presented in Chapter 4 the iterative process was completed if the power estimate was reduced until 15% of the initial power estimate remained or when 10 iterations were reached. Furthermore, a loop gain of 0.9 was employed. The process of iterative peak removal is illustrated in Figure 2.4. From the figure it is apparent how the first peak, indicated by the red cross, is removed from the dirty map using the array response. The array response is stored in the clean map and both the new clean and dirty map are used as input for the next iteration. Eventually, after 4 iterations, all significant peaks are removed, leaving only residuals and the contributions of noise in the final *dirty map*.



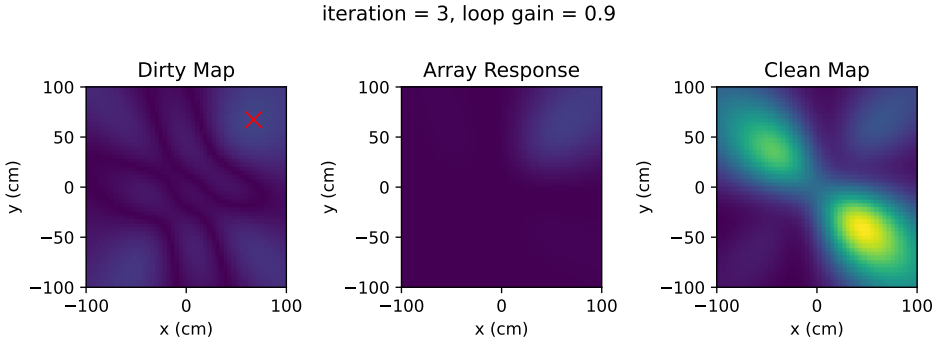
(A)



(B)



(C)



(D)

FIGURE 2.4: Acoustic images illustrating the iterative peak removal process for two signal sources with an signal-to-noise ration of 10dB and no reverberation.

DOA Extraction

Extracting the DOA's of the sources present within the environment is quite simple for CLEAN as each peak is sequentially removed from dirty map. Unlike MUSIC and SRP-PHAT it is not necessary to apply grid-based search methods to find the various peak locations as will be shown in the following sections, where 2D-MUSIC and SRP-PHAT are discussed separately. However, for the wideband implementation of CLEAN, P or less DOA's are obtained per frequency bin. From these KP DOA estimates, P peak locations need to be extracted which correspond with the P source locations. However, this is not as simple as selecting the P peak locations that correspond with the P largest power estimates, as these power estimates might only correspond with one of the two sources as one source can emit a stronger acoustic signal than the other source(s). Furthermore, when the sources are incoherent (exhibit different characteristics) their strongest power estimate can occur in different frequency bins (see Figure 2.3), while coherent sources will have their strongest power estimates associated within the same frequency bin (see Figure 2.4). These factors make it difficult to select the DOA's corresponding with the various signal sources. When dealing with a single source, this problem does not occur, and one can simply select the DOA associated with the strongest power estimate. However, to solve the issue involving multiple sources it has been decided to make use of K-means clustering to create P clusters from the KP obtained DOA's. Then from each cluster the DOA associated with the strongest power estimate is selected, yielding P DOA's associated with P various sources.

2.6 SRP-PHAT

Steered response power (SRP) is another type of beamforming algorithm. Beamforming can be defined as by a *filter-and-sum* process where some temporal filters are applied to the microphone signals before they are summed to form a single focused signal [14]. Temporal filters are filters that attenuate the frequencies of signals such that the signals that are of interest are accentuated while other signals, such as noise, are suppressed. In SRP-PHAT the used filter is the phase transform (PHAT) weighting function. The phase transform is sub-optimal under reverberation-free conditions but is known to perform better in realistic environments [14]. The SRP is equivalent to the sum of generalized cross-correlations (GCCs [37]) of all microphone pairings. The SRP-PHAT algorithm combines both the PHAT and SRP processing by applying the PHAT first and then summing the GCCs of the resulting signals.

Phase Transform

The Phase Transform is a simple operation which involves an element-wise division of the captured STFT signal data of each of the M microphones by its absolute values. Let us represent this data by the matrix $\mathbf{X}$, then the Phase Transform can be given by Equation (2.14).

$$\hat{\mathbf{X}}(\omega_k) = \frac{\mathbf{X}(\omega_k)}{|\mathbf{X}(\omega_k)|} \in \mathbb{C}^{M \times N}, \quad 0 \leq k \leq K \tag{2.14}$$

GCC

After the PHAT has been applied, the GCC is computed for each of the L scan locations by steering the beamformer to each scan direction using the steering vectors.

$$\mathbf{c}(l, \omega_k) = \left| \mathbf{g}(l, \omega_k)^H \hat{\mathbf{X}}(\omega_k) \right|^2 \in \mathbb{R}^{1 \times N} \quad 0 \leq l < L \tag{2.15}$$

SRP spectrum

The SRP for a single scan direction is obtained by summing the GCC for that scan direction.

$$A_k(l) = \sum_{i=0}^{N-1} \mathbf{c}(l, \omega_k)_i \tag{2.16}$$

By computing the SRP for each of the L scan directions, a spectrum is created similar to the acoustic image for CLEAN.

DOA extraction

Similar to CLEAN, in the narrowband scenario, P sources corresponding with the largest response power in the SRP spectrum are selected as the estimated DOA's of the signal sources. These DOA's are extracted from the spectrum containing the contributions of noise and other sources. Therefore, simply selecting the maximum value within the spectrum can result in the P selected DOA's to be associated with only a single source location. As stated in [15], the SRP spectrum can have many local extrema, making it difficult to select those corresponding to the actual source DOA's. To identify the local maxima, a maximum filter with a 3-by-3 structuring element is applied. This involves sliding the structuring element over the acoustic image. At each position, the maximum pixel value within the overlapped region between the structuring element and the image is selected and replaces the value of the corresponding pixel in the output image. In Figure 2.5 the effect of applying the maximum filter is illustrated. The figure shows which pixels have been identified as local maxima. The positions of these local maxima can be mapped back to their corresponding values in the original acoustic image allowing the P largest local maxima to be selected as the DOA's of the signal sources. For the wideband implementation of SRP-PHAT, the narrowband SRP-PHAT algorithm is applied to each of the frequency bins, as is done for every algorithm considered in this paper. P DOA estimates are obtained per frequency bin by applying the aforementioned DOA extraction process. Hence, a total of KP DOA estimates are obtained. The same process as CLEAN is employed by utilizing K-means clustering and selecting the DOA's associated with the largest SRP within each of the P clusters as the final P DOA estimates.

2.7 2D-MUSIC

The MUSIC algorithm is a popular spectral estimation method which has been shown to be particularly effective in localizing narrowband signal sources, even in adverse conditions [22, 36]. Similar to CLEAN and ESPRIT, it utilizes the CSM, or covariance matrix, constructed from the received sound signals after applying the Fourier transform. Though this algorithm also constructs a spectrum (referred to as the pseudospectrum) in which the peaks correspond with the DOAs of the signal sources. However, unlike CLEAN and

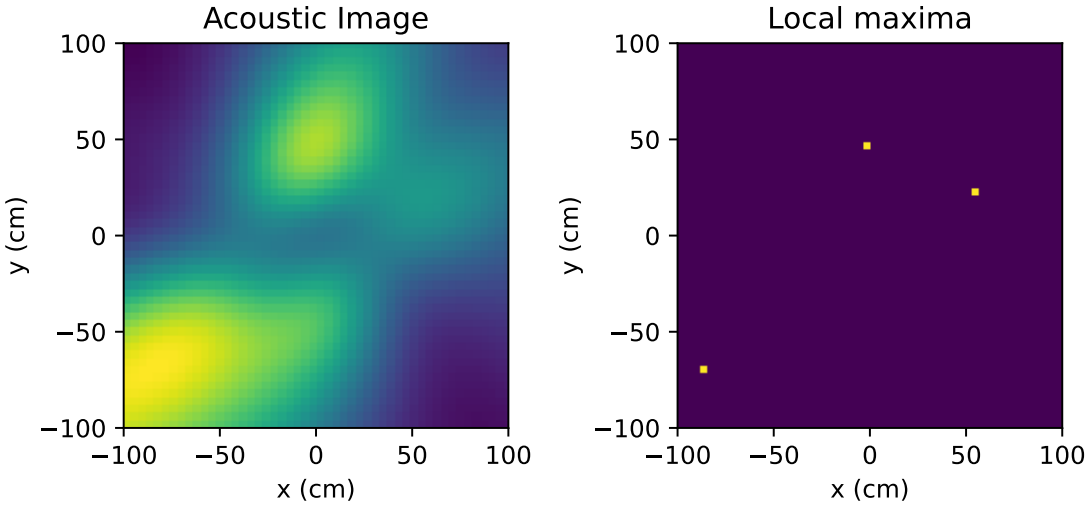


FIGURE 2.5: Acoustic image before and after applying the maximum filter.

SRP-PHAT, it does not utilize CB or GCC to construct this spectrum. Instead, the algorithm uses *eigendecomposition* of the CSM to obtain the noise subspace eigenvectors. The subspace of the signal is orthogonal to that of the noise, as these are uncorrelated, therefore it is possible to construct a spectrum where the peaks correspond with the DOAs of the signal sources, which will be described in more detail in the following sections. Similar to the other scan-grid based algorithms, the resolution of the scan grid needs to be carefully chosen as it affects both the computational load as well as the minimum obtainable error for the DOAs of the signal sources.

Cross-spectral matrix

The construction of the cross spectral matrix $\mathbf{C}_k$ for the MUSIC algorithm is equivalent to that of CLEAN as described in Section 2.5. Therefore, it will not be discussed here.

Eigenvalue decomposition

The next step in the 2D-MUSIC algorithm requires the eigenvalue decomposition of the obtained CSM, $\mathbf{C}_k$. This results in M eigenvalues and eigenvectors. The eigenvalues are not utilized, but the $M - P$ smallest eigenvectors are. These eigenvectors represent the the noise subspace and are used to construct the (pseudo)spectrum from which the DOA's associated with the P source signals can be derived. This is also where a limitation of the MUSIC algorithm becomes apparent. Given the size of the covariance matrix, a maximum of M eigenvalues and eigenvectors can be obtained. Hence, the total number of sources that can theoretically be located by the algorithm is equivalent to the number of microphones in the array.

Pseudospectrum

The noise subspace for the k-th frequency bin, $\mathbf{E}_n(\omega_k) \in \mathbb{C}^{P \times M}$, is constructed from the eigenvectors associated with the smallest eigenvalues after eigendecomposition of the covariance matrix. The noise subspace is orthogonal to the directional vectors of signals. To

construct the spectrum or acoustic image for the k-th frequency bin, the spatial spectrum function of 2D-MUSIC algorithm in Equation (2.17) is utilized.

$$A_k(\theta, \phi) = \frac{1}{\mathbf{g}^H(\theta, \phi, \omega_k)\mathbf{E}_n(\omega_k)\mathbf{E}_n(\omega_k)^H\mathbf{g}(\theta, \phi, \omega_k)} \tag{2.17}$$

Where $\mathbf{g}$ are the steering vectors as described in Section 2.2. After computing the spatial spectrum function for each of the L scan directions, an acoustic image similar to that of SRP-PHAT and CLEAN can be constructed.

DOA extraction

The extraction of the DOA estimates related to the sources is done in an equivalent manner to that of CLEAN and SRP-PHAT. By constructing an acoustic image, a maximum filter can be applied that identifies the local maxima. The largest P local maxima are selected as the DOA estimates for the source signals for a single frequency bin. For the same reasons described in Section 2.5, the selection of P DOA estimates corresponding with P different signal sources from the KP available DOA estimates is not straightforward. Hence, the same methodology as with CLEAN and SRP-PHAT is applied to derive these DOA estimates. This involves forming P data clusters via K-means clustering. Subsequently, the DOA estimates linked with the highest power estimate within each cluster are selected, resulting in the final P DOA estimates.

2.8 2D Unitary ESPRIT

The Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) is a high-resolution signal parameter estimation algorithm that was introduced by Richard Roy and Thomas Kailath in 1989 [65]. It seeks to reduce the computation and storage costs in comparison to MUSIC. ESPRIT also has been shown to be more robust to array imperfections than other techniques including MUSIC [43]. A major difference of ESPRIT compared to the other algorithms is that it uses *subarrays*. The microphone array is split up into subarrays, each of which can be used to obtain spatial information and is combined to obtain the DOA of the signal source more accurately. This does require the microphone array to be centro-symmetric. A possible choice for the selection of subarrays for an URA is given in Figure 2.6. These subarrays both capture the displacement along the x- and y-axis, which is essential to capture both the azimuth and elevation of the DOA. The original algorithm proposed by Roy and Kailath was only applicable to DOA estimation considering either the azimuth or elevation angle. Hence, extensions have been proposed to allow for DOA estimation for both an azimuth and elevation angle. These algorithms work by applying ESPRIT twice. Once for each displacement direction. The obvious downside of this approach is the added computational complexity it introduces. However, 2D Unitary ESPRIT seeks to reduce the computations required for 2D DOA estimation by coupling the azimuth and elevation estimation. Furthermore, the eigenvalue decomposition is performed on real-valued matrices instead of complex-valued ones unlike 2D-MUSIC. However, this is not true for the final eigendecomposition, which is performed on a vector with its dimension equal to the number of microphones in the array [26]. In the remainder of this section the components for the 2D Unitary ESPRIT algorithm will be explained.

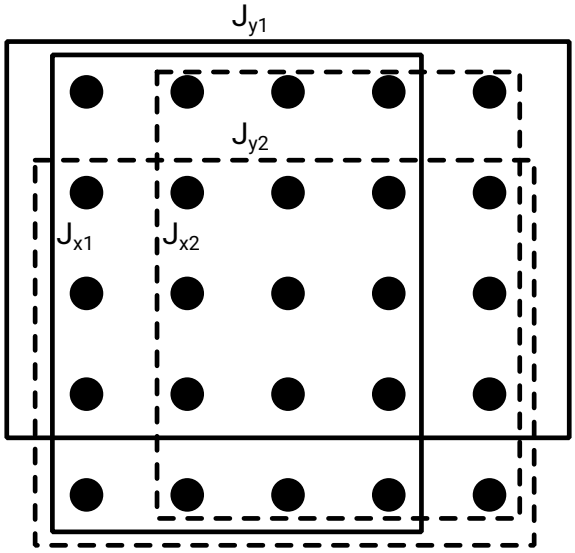


FIGURE 2.6: Four subarrays for the URA with displacement in the x- and y-axis

Selection matrices

As mentioned in the previous section, the ESPRIT algorithm works by dividing the microphone into several subarrays. To achieve this, the algorithm makes use of *selection matrices*. These matrices provide a convenient way of selecting these subarrays through matrix multiplication. Assuming the microphone is a URA containing 25 elements as depicted in Figure 2.6, then define the matrix $\mathbf{J}$ and the *exchange matrix* $\mathbf{\Pi}$ as shown in equations (2.18) and (2.19) respectively.

$$\mathbf{J} = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{vmatrix} \in \{0,1\}^{5\times 5} \tag{2.18}$$

$$\mathbf{\Pi}_p = \begin{vmatrix} & & & & 1 \\ & & & 1 & \\ & & \cdot & & \\ 1 & & & & \end{vmatrix} \in \{0,1\}^{p\times p} \tag{2.19}$$

Then the selection matrix $\mathbf{J}_{x1}$ is constructed as the Kronecker product of the identity matrix $\mathbf{I}_5$ and the matrix $\mathbf{J}$ and the exchange matrix $\mathbf{\Pi}$.

$$\mathbf{J}_{x1} = \mathbf{I}_5 \otimes \mathbf{J} \tag{2.20}$$

The selection matrix $\mathbf{J}_{x2}$ is obtained through matrix multiplication with the exchange matrices $\mathbf{\Pi}_{M_{sub}}$ and $\mathbf{\Pi}_M$ as depicted in Equation (2.21). Here M_{sub} represents the number of elements in a single subarray.

$$\mathbf{J}_{x2} = \mathbf{\Pi}_{M_{sub}} \mathbf{J}_{x1} \mathbf{\Pi}_M \tag{2.21}$$

Similarly, the selection matrices for $\mathbf{J}_{y1}$ and $\mathbf{J}_{y1}$ can be obtained as shown in equations (2.22) and (2.23) respectively.

$$\mathbf{J}_{y1} = \mathbf{J} \otimes \mathbf{I}_5 \tag{2.22}$$

$$\mathbf{J}_{y2} = \mathbf{\Pi}_{M_{sub}} \mathbf{J}_{y1} \mathbf{\Pi}_M \tag{2.23}$$

Extended data matrix

Unitary ESPRIT for the 1D-case makes use of a complex-valued extended data matrix which is constructed from the data matrix $\mathbf{X} \in \mathbb{C}^{M \times N}$ and exchange matrices $\mathbf{\Pi}_M$ and $\mathbf{\Pi}_N$ as defined in equations (2.19) [25]. This allows for the construction of the complex-valued extended matrix as defined in Equation (2.24), which corresponds to a square-root version of the forward-backward averaging scheme.

$$[\mathbf{X} \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N] \in \mathbb{C}^{M \times 2N} \tag{2.24}$$

Matrix transformation

To convert the complex-valued extended data matrix to the real-valued extended data matrix let us first define the *unitary matrix* $\mathbf{Q}_M$ where M still represents the number of microphones.

$$\mathbf{Q}_M = \begin{cases} M \text{is even} & \frac{1}{\sqrt{2}} \begin{vmatrix} \mathbf{I}_n & \mathrm{j}\mathbf{I}_n \\ \mathbf{\Pi}_n & -\mathrm{j}\mathbf{\Pi}_n \end{vmatrix} \\ M \text{is odd} & \frac{1}{\sqrt{2}} \begin{vmatrix} \mathbf{I}_n & \mathbf{0} & \mathrm{j}\mathbf{I}_n \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ \mathbf{\Pi}_n & \mathbf{0} & -\mathrm{j}\mathbf{\Pi}_n \end{vmatrix} \end{cases} \tag{2.25}$$

This matrix is left $\mathbf{\Pi}$-real [25, 40]. *Unitary ESPRIT* is based on the following theorem from [40].

Theorem 1 *Let $\mathbf{Q}_p$ and $\mathbf{Q}_q$ denote unitary left $\mathbf{\Pi}$-real matrices of size $p \times q$ and $q \times q$, respectively. Then the bijective mapping*

$$\varphi : \mathbf{X} \Rightarrow \mathbf{Q}_p^H \mathbf{X} \mathbf{Q}_q$$

maps the set of all $p \times q$ centro-Hermitian matrices onto $\mathbb{R}^{p \times q}$ the set of all real matrices of the same size.

Given the fact that the complex-valued extended data matrix defined in Equation (2.24) is centro-Hermitian [25], it can be transformed into a real-valued matrix of the same size according to theorem 1. Let us define this transformation as $\mathcal{T}(X)$. Then, the transformation can be described as shown in Equation (2.26).

$$\mathcal{T}(\mathbf{X}) \triangleq \mathbf{Q}_M^H \left| \mathbf{X} \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N \right| \mathbf{Q}_{2N} \in \mathbb{R}^{M \times 2N} \tag{2.26}$$

Covariance matrix

The P dominant eigenvectors $\mathbf{E}_s \in \mathbb{R}^{M \times d}$ can be obtained through singular value decomposition (SVD) of $\mathcal{T}(\mathbf{X})$ (direct-data or square root approach). Alternatively, they can be computed through a real-valued eigendecomposition of the covariance matrix as shown in Equation (2.27).

$$\mathcal{T}(\mathbf{X}) \mathcal{T}(\mathbf{X})^H \in \mathbb{R}^{M \times M} \tag{2.27}$$

In this paper only the covariance approach is considered.

Real-valued eigendecomposition

Eigendecomposition of the covariance matrix in Equation (2.27) results in M eigenvalues and eigenvectors. Only the P largest eigenvectors are chosen, which correspond with the signal subspace. These eigenvectors are used to perform the spatial frequency estimation described in the next section.

Spatial frequency estimation

In [25] it was shown that an estimate of the spatial frequencies in x-direction can be more efficiently be obtained from the solutions of the overdetermined real-valued set of equations

$$\mathbf{K}_{x1}\mathbf{E_s}\mathbf{Y}_x \approx \mathbf{K}_{x2}\mathbf{E_s} \tag{2.28}$$

where the selection matrices $\mathbf{K}_{x1}$ and $\mathbf{K}_{x2}$ are derived from $\mathbf{J}_{x1}$ and $\mathbf{J}_{x2}$ according to equations (2.29) and (2.30).

$$\mathbf{K}_{x1} = \mathbf{Q}_{m_x}^H(\mathbf{J}_{x1} + \mathbf{J}_{x2})\mathbf{Q}_M \in \mathbb{R} \tag{2.29}$$

$$\mathbf{K}_{x2} = \mathbf{Q}_{m_x}^H j(\mathbf{J}_{x1} - \mathbf{J}_{x2})\mathbf{Q}_M \in \mathbb{R} \tag{2.30}$$

The estimate of the spatial frequencies in y-direction can be derived similarly from the set of equations

$$\mathbf{K}_{y1}\mathbf{E_s}\mathbf{Y}_y \approx \mathbf{K}_{y2}\mathbf{E_s} \tag{2.31}$$

where the selection matrices $\mathbf{K}_{y1}$ and $\mathbf{K}_{y2}$ are derived from $\mathbf{J}_{y1}$ and $\mathbf{J}_{y2}$ according to equations (2.32) and (2.33).

$$\mathbf{K}_{y1} = \mathbf{Q}_{m_y}^H(\mathbf{J}_{y1} + \mathbf{J}_{y2})\mathbf{Q}_M \in \mathbb{R} \tag{2.32}$$

$$\mathbf{K}_{y2} = \mathbf{Q}_{m_y}^H j(\mathbf{J}_{y1} - \mathbf{J}_{y2})\mathbf{Q}_M \in \mathbb{R} \tag{2.33}$$

The solutions for $\mathbf{Y}_x$ and $\mathbf{Y}_y$ can both be obtained using a Least-Squares (LS) approach. In this paper the utilization of the Total Least-Squares (TLS) is assumed. Finally, the spatial frequency estimates are obtained through complex-valued eigendecomposition of the "complexified" matrix

$$\mathbf{Y} = \mathbf{Y}_x + j\mathbf{Y}_y$$

which automatically pairs the spatial frequency estimation in both the x and y-direction.

Complex-valued Eigendecomposition

The final step involves the complex-valued eigendecomposition of the matrix $\mathbf{Y}$ which results in P complex-valued eigenvalues λ_p. The resulting eigenvalues are converted into azimuth and elevation angles that make up the P DOAs of the sources.

DOA extraction

To convert the obtained complex eigenvalues λ_p to azimuth and elevation angles, first the direction cosines u_p and v_p are computed, which were shown in Figure 2.2.

$$u_p = \frac{2\arctan(\Re(\lambda_p))}{\frac{2\pi\omega_k}{c}d_x}$$

$$v_p = \frac{2\arctan(\Im(\lambda_p))}{\frac{2\pi\omega_k}{c}d_y}$$

To convert the direction cosines into the azimuth and elevation angles, several trigonometric operations need to be performed as described below.

$$\tau = \sqrt{u_p^2 + v_p^2}$$

$$\xi_p = u_p + jv_p$$

$$\theta_p = \angle\xi_p$$

$$\phi_p = \arcsin\left(\frac{u_p}{\cos(v_p/\tau)}\right)$$

Similarly to the other algorithms, a single DOA is obtained for each source for each frequency bin. Hence, the wideband implementation of 2D Unitary ESPRIT also yields KP DOAs. For the same reasons described for the DOA extraction process for CLEAN, and consequently for the SRP-PHAT and 2D-MUSIC algorithms, it is not known which DOA corresponds with which of the P signal sources. Hence, once again, K-means clustering is employed to create P clusters from the KP DOA's based on their estimated azimuth and elevation angles. For each cluster, the DOA associated with the largest eigenvalue for that cluster is selected as the final DOA. Resulting in P DOA's associated with P different sources.

Chapter 3

Complexity analysis and Data Dependencies

This chapter's primary objective is to dissect the SSL algorithms discussed in the previous chapter and compute the complexity of each constituent part of each SSL algorithm to provide the reader with a thorough understanding on how each component influences the algorithm's computational complexity. Moreover, DDGs are constructed that show the inter-element data dependencies and can be used to create parallelized implementation of each algorithm that can accelerate the execution of the various components through parallel processing techniques. Lastly, a comparative analysis is performed on the obtained complexities to determine which algorithm is best-applicable to real-time implementations. Additionally, a comparison of the applicability of each algorithm for parallel processing is provided.

In this chapter each algorithm's components is expressed by its computational complexity using the Big-O notation. These complexities are derived from the total number of *operations* required for each component. Here, *operations* refer to arithmetic operations such as addition and multiplication as well as more advanced operations such as square root and exponentiation. However, the underlying complexity for these operations does vary. For instance, the complexity of an addition is significantly lower than that of a division. Nevertheless, for the sake of simplicity they are all considered equivalent in this paper. Furthermore, it is important to note that no distinction will be made between operations performed on complex numbers versus real numbers. Consequently, the computational complexities obtained in this chapter should be considered to be rough estimates. Common symbols and their meaning are expressed used to formulate the computational complexities are described in Table 3.1.

Symbol	Description
M	Number of microphones
L	Number of scan points
P	Number of signal sources
K	Number of frequency bins

TABLE 3.1: Symbols used to express computational complexities

3.1 Steering matrix

The construction of the steering matrix is given in Equation (2.5). To analyze the complexity, let us rewrite the equation.

$$\mathbf{G}(\omega_k) = \exp(\mathbf{a}_x\mathbf{b}_x + \mathbf{a}_y\mathbf{b}_y) \in \mathbb{C}^{L\times M} \tag{3.1}$$

Where

$$\mathbf{a}_x = \frac{-j2\pi d_x \mathbf{m}_x}{c}\omega_k \in \mathbb{C}^M$$

$$\mathbf{a}_y = \frac{-j2\pi d_y \mathbf{m}_y}{c}\omega_k \in \mathbb{C}^M$$

$$\mathbf{b}_x = \sin(\mathbf{\Theta})\cos(\mathbf{\Phi}) \in \mathbb{R}^L$$

$$\mathbf{b}_y = \sin(\mathbf{\Theta})\sin(\mathbf{\Phi}) \in \mathbb{R}^L$$

Using this notation, the computation required for a single element in the steering matrix can be described as shown in Equation (3.2).

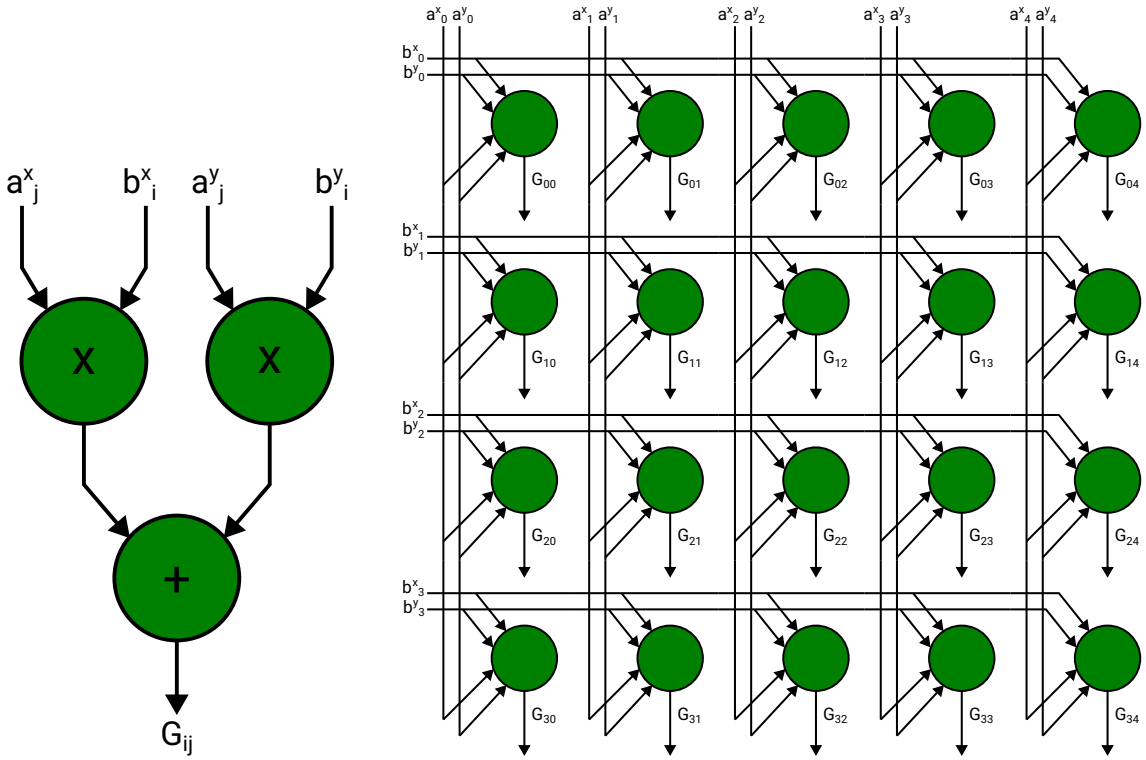
$$\mathbf{G}_{ij} = \mathbf{a}_j^x\mathbf{b}_i^x + \mathbf{a}_j^y\mathbf{b}_i^y, \quad 0 < i < L-1, \quad 0 < j < M-1 \tag{3.2}$$

The equation shows that each element in the steering matrix is computed through two complex multiplications and a complex addition. Hence, the computational complexity of the steering matrix is equal to $O(KLM)$. Clearly, $\mathbf{b}_x$ and $\mathbf{b}_y$ can be computed independently of the frequency ω_k. Furthermore, the vectors $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are directly derived from the scan grid and can therefore be considered as constants. The vectors $\mathbf{a}_x$ and $\mathbf{a}_y$ are dependent on the ω_k, hence they need to be evaluated for each of the frequency bins obtained through STFT. It is possible to compute the steering matrices in advance and access them through a LUT as they are needed. This can contribute to reducing the complexity of the scan-grid based algorithms. However, it is important to note that this may require a large amount of storage. For example, assume a complex number is stored as two 64-bit floating points. Then to store the steering matrices for each of the K frequency bins will require $128KLM$ bits of storage. Therefore, when dealing with many frequency bins, a high-resolution scan grid, or a large microphone array, this option might not be feasible.

Whether or not the steering matrix is pre-computed, it is possible to reduce the computational complexity of the operation through parallel processing techniques. For this purpose, the processing element (PE) and DDGs are defined in Figures 3.1a and 3.1b respectively. The DDG is an example, the complete design is much larger. To be specific, the number of required processing elements is equal to ML. If these resources are available, each element in the steering matrix can be computed in parallel and the computation complexity of the operation is reduced to $O(1)$. Alternatively, a block-based processing approach can be employed. Let us denote the number of available processing elements as B. Then B elements of the steering matrix are computed in parallel. Therefore, the computational complexity for this approach can be given as $O\left(\frac{LM}{B}\right)$.

3.2 CLEAN

To compute the complexity of the CLEAN algorithm, the complexity of each operation will be calculated individually, without considering the frequency bin (narrowband). Afterwards, the obtained complexities will be expressed for the wideband case.



(A) (B)

FIGURE 3.1: PE (A) and DDG (B) for computing the steering matrix.

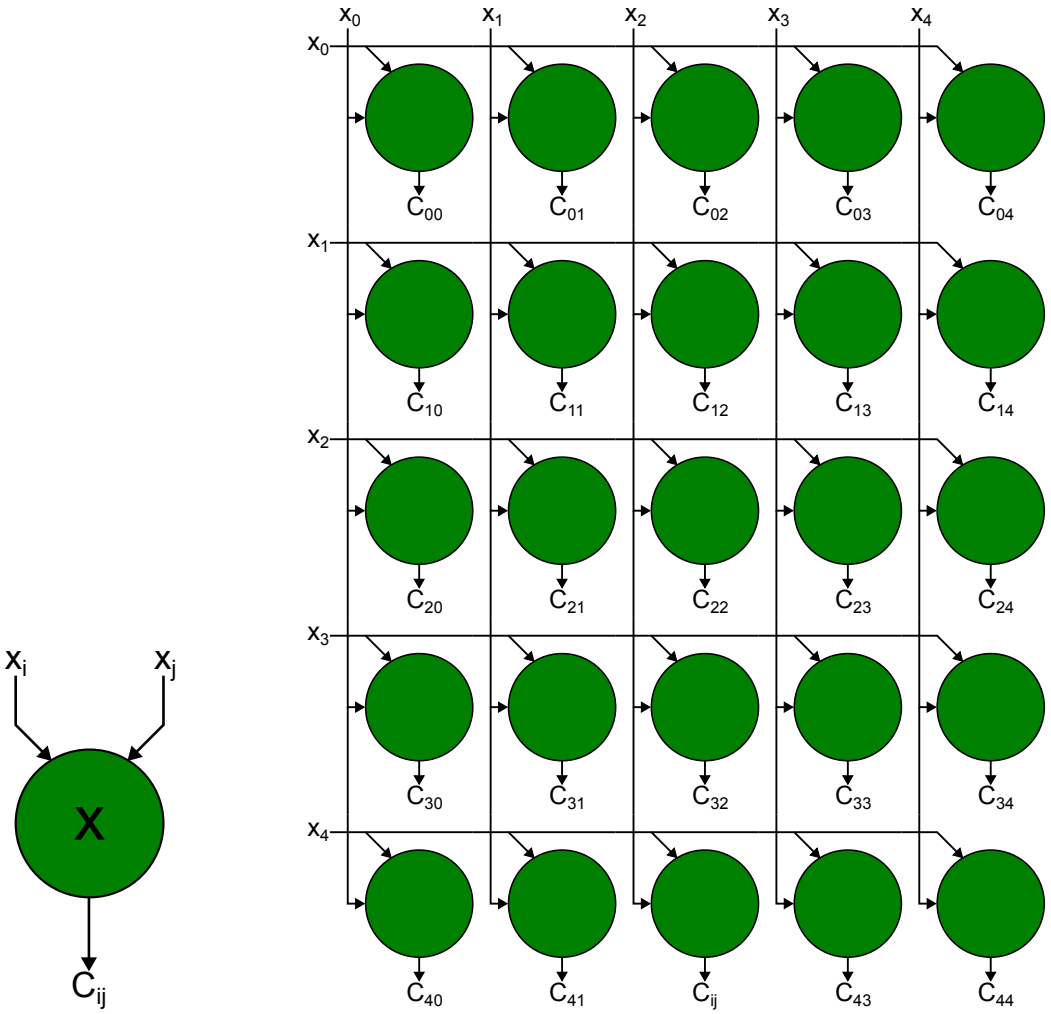
Cross-Spectral Matrix (CSM)

The mathematical equation for constructing the CSM was given in Equation (2.6), which shows that the CSM is the result of the outer product of the received microphone signals at a single point in time $\mathbf{x}$ and its conjugate transpose $\mathbf{x}^H$. Therefore, the operation for computing each element in the CSM can be expressed as shown in Equation 3.3.

$$\mathbf{C}_{ij}(n) = \mathbf{x}_i(n)\mathbf{x}_j^H(n) \tag{3.3}$$

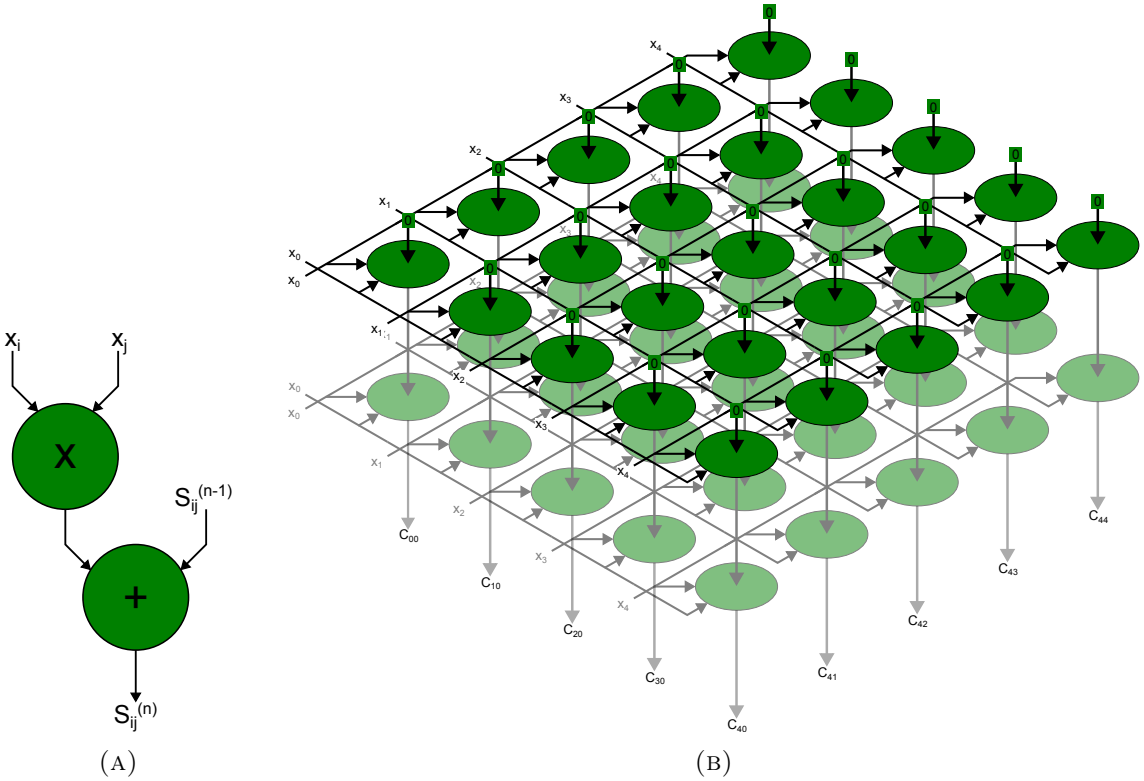
Which means that the computational complexity of computing a CSM is equal to $O(M^2)$, as a total of M^2 multiplications are required. To parallelize the operation, the PE and DDG are given in Figure 3.2. The DDG clearly shows that each element of the CSM can be computed independently, hence the operation can be fully parallelized. As a result, the computational complexity of the parallelized implementation for the CSM is equal to $O(1)$. The computational complexity for a single CSM can be used to derive the computational complexity of the summed and averaged CSM that was shown in Equation (2.7). Let us denote the equations for computing a single element in the summed and averaged CSM, $\hat{\mathbf{C}}$, as shown in Equation (3.4).

$$\hat{\mathbf{C}}_{ij} = \frac{1}{N}\sum_{n=0}^{N}\mathbf{C}_{ij}(n) \tag{3.4}$$



(A) (B)

FIGURE 3.2: PE (A) and DDG (B) for computing the CSM (M = 5).



(A) (B)

FIGURE 3.3: PE (A) and DDG (B) for computing the summed and averaged CSM (M = 5, N = 2).

Then the computational complexity of the operation is given by $O(NM^2)$. To parallelize the operation, the partial sum $S_{ij}^{(n)}$ is defined in Equation (3.5).

$$\begin{aligned} S_{ij}^{(-1)} &= 0 \\ S_{ij}^{(k)} &= S_{ij}^{(k-1)} + \mathbf{x}_i(n)\mathbf{x}_j^H(n) \end{aligned} \tag{3.5}$$

With this definition each element in the summed and averaged CSM can be expressed as $\hat{\mathbf{C}}_{ij} = \frac{1}{N} S_{ij}^{(N-1)}$. The corresponding PE and DDG are presented in Figure 3.3. The longest path in the DDG to compute a single element in the resulting summed and averaged CSM will require N processing elements. Hence, the parallelized computational complexity of computing this CSM is equal to $O(N)$.

Conventional Beamforming

As was described in Section 2.5, the dirty map is constructed through (CB), which involves computing the power estimate for each of the L scan directions. The equation for the power estimate is given in 2.8. This operation consists of a matrix-vector multiplication and a vector-vector multiplication. First, the equation is rewritten as shown in Equation (3.6).

$$A(l) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} g_i(l) C_{ij} g_j^H(l) \tag{3.6}$$

Here the g_i and g_j^H represent the i-th and j-th elements of the steering vectors $\mathbf{g}(l)$ and $\mathbf{g}^H(l)$ respectively. C_{ij} represents the element located at the i-th row and j-th column of the CSM.

From this notation it is determined that a total of $2M^2$ multiplications and M^2 additions must be performed to compute the power estimate for a single candidate DOA. Considering that to construct the complete beamforming map a power estimate needs to be computed for each of the L scan directions, the computational complexity of CB is equal to $O(LM^2)$. To parallelize the operation, the partial sums S_i and S are defined in equations (3.7) and (3.8) respectively.

$$\begin{aligned} S_i^{(-1)} &= 0 \\ S_i^{(n)} &= S_i^{(n-1)} + g_i C_{in} g_n^H \end{aligned} \tag{3.7}$$

$$\begin{aligned} S^{(-1)} &= 0 \\ S^{(n)} &= S^{(n-1)} + S_n^{(M-1)} \end{aligned} \tag{3.8}$$

From these partial sums the PE and DDG in Figure 3.4 are constructed. The DDG shows that each of the partial sums $S_i^{(n)}$ can be computed in parallel. The longest path in this DDG has a length of $2M$. Given the associative properties of the operations involved in CB, other DDG designs are possible. Some of which are more efficient. Take the alternative design in Figure 3.5 for example which has a longest path of length $M+2$. For both DDGs the obtained computational complexity is equal to $O(LM)$. This can actually be reduced to only $O(M)$ as each of the power estimates for each of the L DOAs can be computed in parallel by stacking the DDG L times. However, this will take a significant number of resources. Nevertheless, it shows that CB can significantly benefit from a parallelized implementation.

Extracting the DOA estimates

After CB an acoustic image is formed from which the DOA estimate of a single source is extracted by selecting the associated azimuth and elevation angle with the position of the maximum value within the acoustic image. This will require L comparisons and therefore has a complexity of $O(L)$. A parallelized approach would involve splitting up the acoustic image into several blocks of data containing B elements and determine the maximum of each individual block in parallel. After each maximum is found a final comparison of the $\frac{L}{B}$ obtained maxima needs to be compared to find the final maximum value. The complexity of finding the maximum of each data block is $O(B)$. Therefore, the computational complexity for finding the maximum for all blocks is $\frac{L}{B} \times O(B) = O(L)$. Once the $\frac{L}{B}$ maxima are obtained, selecting the maximum of these maxima has a complexity of $O(\frac{L}{B})$. Therefore, the computational complexity of the complete process is equal to $O(B + \frac{L}{B})$. To minimize the total complexity, the terms L and $\frac{L}{B}$ need to be balanced to find the optimal ratio. Since, L is the dominant term and independent of B, the goal is to ensure that $\frac{L}{B}$ is minimized while keeping the computational efficiency in consideration. A good heuristic is to choose B such that $B \approx \sqrt{L}$. This makes sure that B is neither too small nor too large. Substituting $\sqrt{L}$ for B yields the computational complexity $O(\sqrt{L} + \frac{L}{\sqrt{L}}) = O(2\sqrt{L}) = O(\sqrt{L})$. For the sake of brevity however, B will not be considered further in this paper and

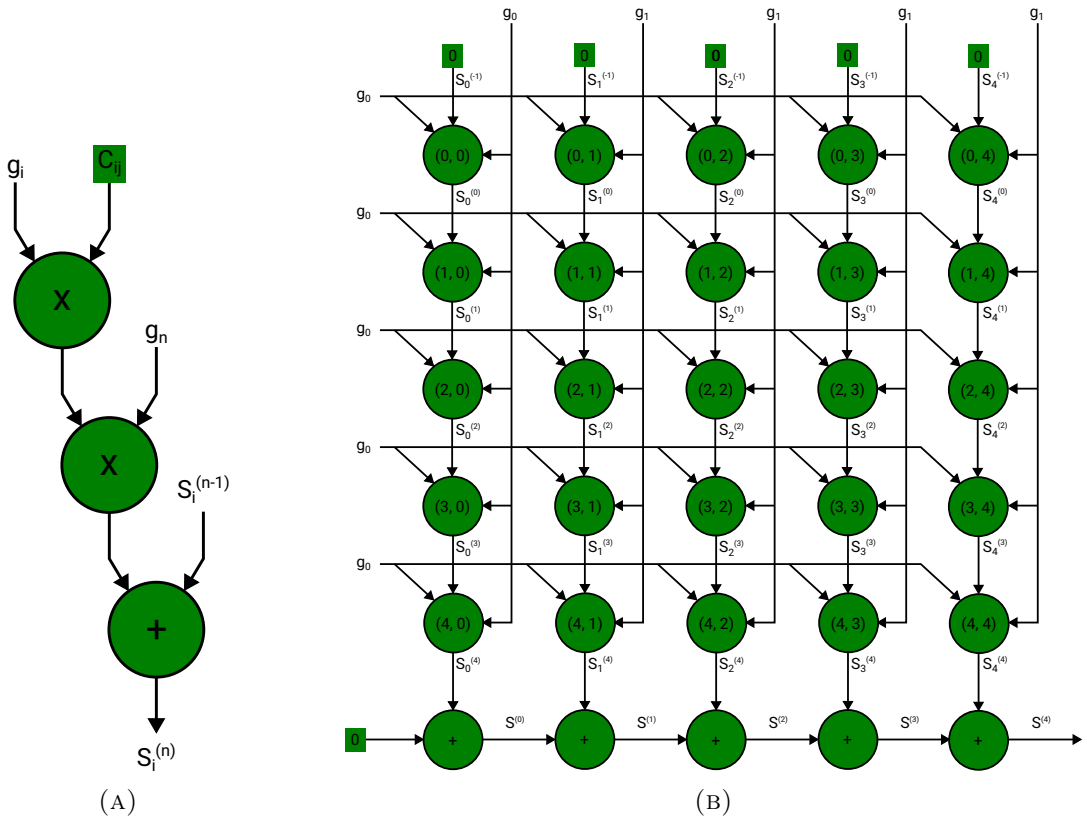


FIGURE 3.4: PE (A) and DDG (B) for CB.

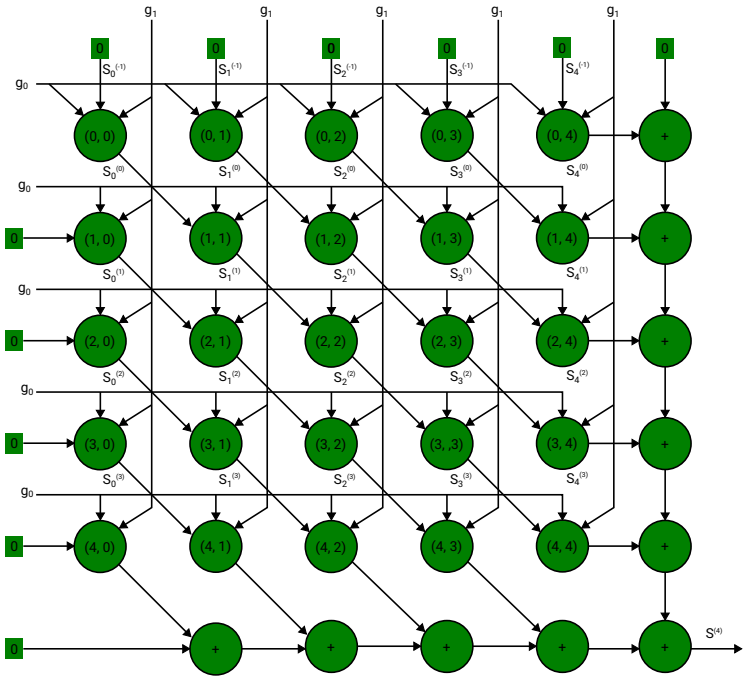


FIGURE 3.5: Alternative DDG for CB

it is assumed that the DOA estimates are obtained by comparing each of the L elements, which has a computational complexity of $O(L)$.

Array response

When a DOA estimate has been identified, the array response in the identified direction can be constructed as shown in Equation (2.10). This operation consists of computing the outer product of the steering vector $\mathbf{g}$ with its conjugate transpose for each of the L scan points. Additionally, each value in the outer product is divided by the norm of the steering vector and multiplied by the loop gain λ.

$$\mathbf{R}_{ij} = \lambda \frac{g_i g_j^H}{\| \mathbf{g} \|^4}, \quad 0 \leq i < M. \quad 0 \leq j < M \tag{3.9}$$

Where $\| \mathbf{g} \|$ represents the Frobenius Norm of the steering vector which is described in Equation (3.10).

$$\| \mathbf{g} \| = \sqrt{\sum_{m=0}^{M-1} |g_m|^2} \tag{3.10}$$

Therefore, Equation (3.9) can be rewritten as:

$$\mathbf{R}_{ij} = \frac{\lambda}{\left(\sum_{m=0}^{M-1} |g_m|^2\right)^2} g_i g_j^H \tag{3.11}$$

To simplify the equation, the term $\frac{\lambda}{\left(\sum_{m=0}^{M-1} |g_m|^2\right)^2}$ is substituted by β.

$$\mathbf{R}_{ij} = \beta g_i g_j^H \tag{3.12}$$

Computing β requires dividing the loop gain, λ, by the squared sum of squares of the vector $\mathbf{g}$. This requires a total of $2M + 2$ operations. The operation $g_i g_j^H$ is repeated for each of the L scan points and is therefore equal in its computational complexity to that of the CSM. Hence, the computational complexity of constructing the array response is equal to $O(LM^2)$. The operation can be parallelized using the PE in Figure 3.6 which can be arranged similarly to the DDG for the CSM in Figure 3.2. Therefore, the parallelized computation complexity of the array response is equal to $O(1)$.

Peak reduction

When the array response is computed, it needs to be subtracted from the dirty CSM to construct the new dirty map for the next iteration of the peak removal process. Let us define the element-wise subtraction of the array response from the dirty CSM as shown in Equation 3.13.

$$\hat{\mathbf{C}}_{ij} = \mathbf{C}_{ij} - \mathbf{R}_{ij} \tag{3.13}$$

Therefore, the sequential computational complexity of the peak reduction is equal to $O(M^2)$. Each subtraction is performed on independent elements. Therefore, the parallelized computational complexity is equal to $O(1)$. To achieve this the PE and DDG in Figure 3.7 are presented. By changing the subtraction in the PE to an addition, the same DDG and PE can also be used to add the array response to the clean CSM, meaning that the parallelized computational complexity of this process is also equal to $O(1)$.

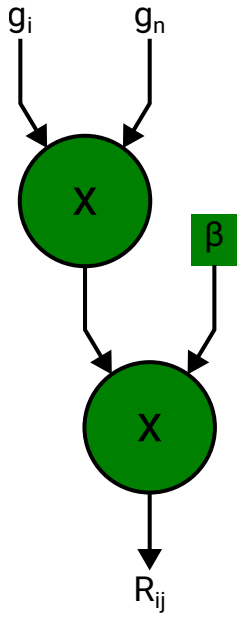
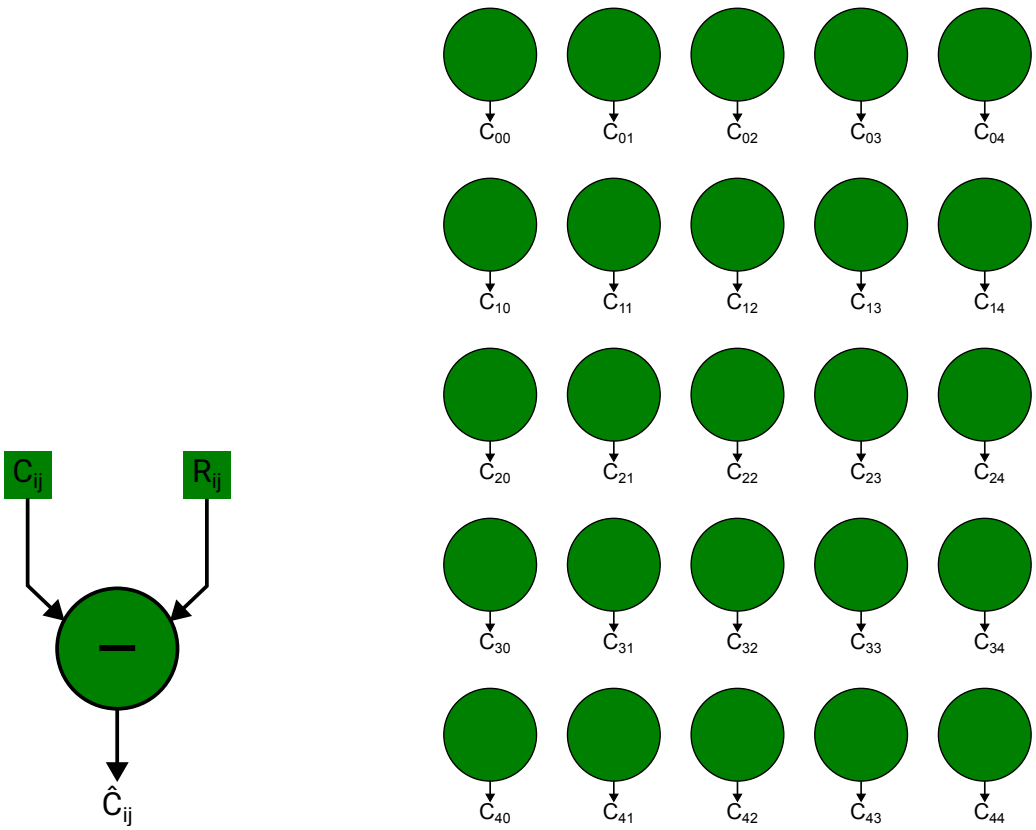


FIGURE 3.6: PE for computing the array response



(A) (B)

FIGURE 3.7: PE (A) and DDG (B) for subtracting the array response from the dirty CSM.

Iterative peak reduction

Removing the influence of each signal source from an acoustic image is an iterative process which involves computing the array response for the DOA of the signal source, removing the array response from the dirty CSM, and adding the array response to the clean CSM as has been described earlier in this section. The number of iterations required to remove the influence of the signal source is determined by the loop gain and the chosen threshold. As mentioned in Section 2.5, the threshold that was chosen for the purpose of this research paper required that the influence of the signal source be reduced to at least 15% of its initial peak value or 10 iterations had been reached. Hence, it took anywhere from 1 to 10 iterations for the peaks to be removed from the dirty acoustic image. Due to this uncertainty on how many iterations are performed, the worst-case scenario is assumed for determining the computational complexity of the iterative peak reduction process. The process starts with an acoustic map from which the DOA is extracted according to the DOA extraction process described in Section 3.2. From this DOA the array response is constructed as previously described and removed from the *dirty CSM* associated with the acoustic map through peak reduction and added to the *clean CSM* in a similar manner to the peak reduction process. This results in a new clean and dirty CSM which is utilized for the next iteration. Let us define the maximum number of iterations required for the peak reduction process as I_{max}. As each iteration is dependent on its previous iteration, it is not possible perform each iteration in parallel. Hence, the computational complexities are derived from the previous sections by summing them together and multiplying them by I_{max}. Therefore, the sequential computational complexity of the iterative peak reduction process is equal to $O(I_{max}LM^2)$. The parallelized computational complexity is equal to $O(I_{max}M)$.

Operation	**Sequential**	**Parallelized**
CSM	$O(NM^2)$	$O(N)$
CB	$O(LM^2)$	$O(M)$
DOA Extraction	$O(L)$	$O(L)$
Array Response	$O(M^2)$	$O(1)$
It. Peak reduction	$O(I_{max}LM^2)$	$O(I_{max}M)$
Total narrowband	$O(NM^2 + LM^2 + L + I_{max}LM^2)$	$O(N + M + L + I_{max}M)$
Total wideband	$O(KNM^2 + KLM^2 + KL + KI_{max}LM^2)$	$O(KN + KM + KL + KI_{max}M)$

TABLE 3.2: Complexities of CLEAN components

Total complexity

Now that all components have been investigated, the results can be combined to obtain the sequential and parallelized computational complexities of the narrowband implementation of the CLEAN algorithm. The sequential computational complexity is equal to $O(NM^2 + LM^2 + L + PI_{max}LM^2)$. The parallelized computational complexity is equal to $O(N + M + L + I_{max}M)$. The wideband computational complexity is easily obtained as it requires the narrowband implementation of CLEAN to be repeated for each of the K frequency bins. Hence the corresponding sequential and parallelized computational complexities are equal to $O(KNM^2 + KLM^2 + KL + KPI_{max}LM^2)$ and $O(KN + KM + KL + KI_{max}M)$ respectively. It is important however, that the wideband implementation of CLEAN will yield KP DOA estimates from which only P need to be selected. For the purpose of

this research, the DOA estimates were divided into P clusters of data. From each cluster, the DOA associated with the largest peak value was chosen, yielding two distinct DOA's. This is one approach to this problem and other choices can be made depending on what the requirements are. Furthermore, it should theoretically be possible to perform the narrowband implementation for each of the K frequency bins in parallel. However, the number of resources required for such an implementation may prove to be problematic. Therefore, it is not considered in this paper.

In Figure 3.8 the total sequential and parallelized computational complexities are plotted against the number of microphones, frequency bins, and scan points. The figure shows that the parallelized approach leads to lower computational complexities when considering each of these parameters. The parallelized approach enhances the algorithm's efficiency, especially when considering the number of microphones or scan points, as the increase in parallelized computational complexity compared to the sequential computational complexity is less significant with these parameters. When considering the number of frequency bins this rate of increase appears to be the same between both the sequential and parallelized approach. In conclusion, the parallelized approach to the CLEAN algorithm leads to an implementation that leads to an overall reduction of computational complexity. In particular, the parallelized complexity scales well against the number of microphones present in the array.

3.3 SRP-PHAT

Phase transform

The phase transform as shown in Equation (2.14) consists of an element-wise division of the data matrix $\mathbf{X} \in \mathbb{C}^{M \times N}$ with its absolute values. Hence, the data matrix after phase transformation $\hat{\mathbf{X}}$ can be described as:

$$\hat{\mathbf{X}}_{ij} = \frac{\mathbf{X}_{ij}}{|\mathbf{X}_{ij}|} \tag{3.14}$$

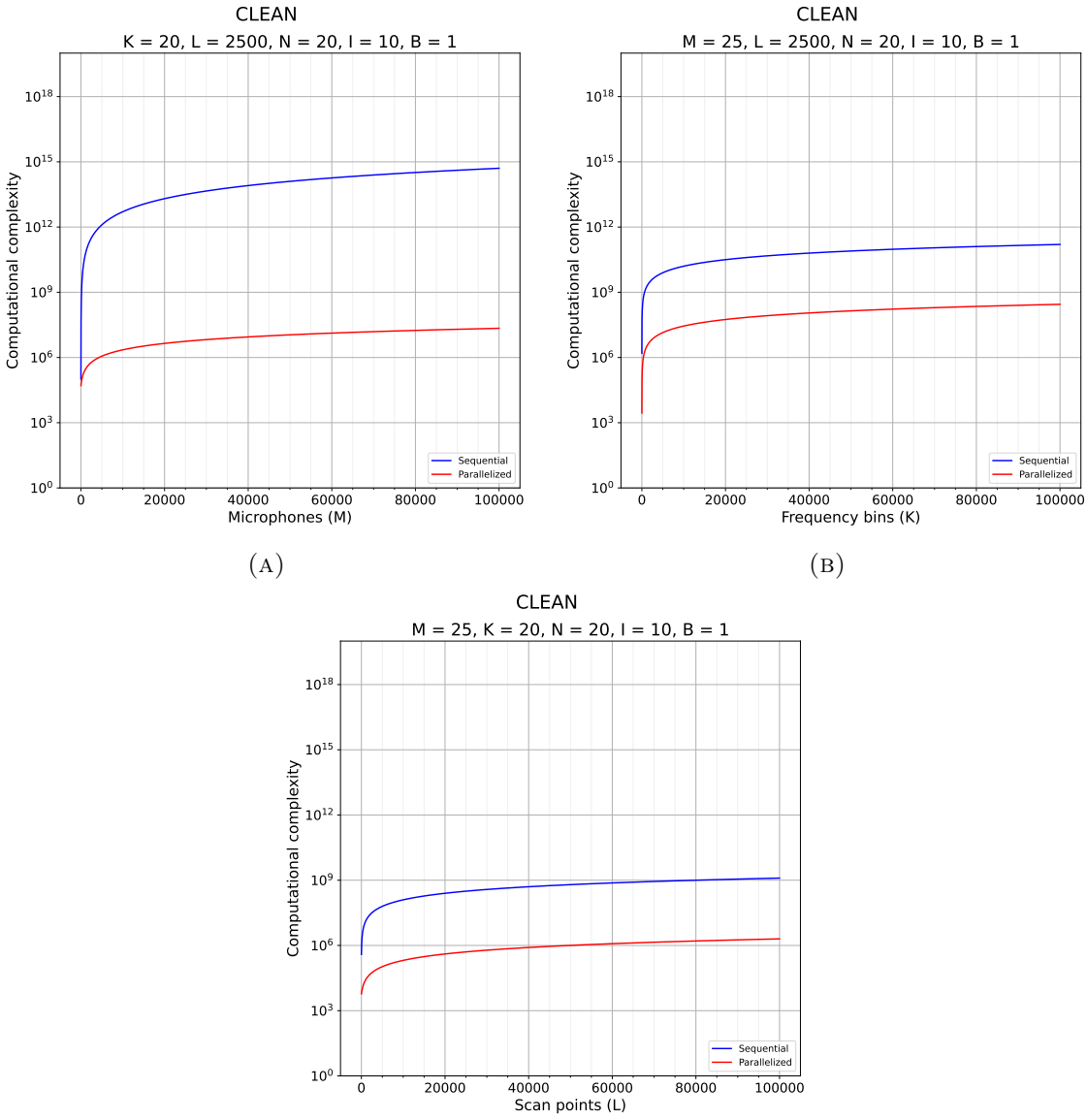
Clearly, this will require MN divisions and has a sequential computational complexity of $O(MN)$. To parallelize the operation, a simple PE is defined and arranged in the DDG presented in Figure 3.9. Clearly, each element of $\mathbf{X}_{ij}$ can be computed independently. Therefore, the operation can be fully parallelized and has a parallelized computational complexity of $O(1)$.

Steered Response Power

For each of the L candidate directions, the SRP has to be computed to construct the complete SRP spectrum. Equation (2.16) describes how the SRP can be computed for a single scan location. For each scan location a vector-matrix multiplication needs to be performed. Let us consider only one frequency bin, then this operation can be described as shown in Equation (3.15).

$$C(l) = \sum_{i=0}^{N-1} \left| \sum_{j=0}^{M-1} \mathbf{g}_j^* \hat{\mathbf{X}}_{ji} \right|^2 \tag{3.15}$$

Which means that, to compute the SRP for a single candidate DOA, a total of NM multiplications and additions are performed. Additionally, N values have to be converted



(A) (B)

(C)

FIGURE 3.8: Sequential and parallelized computational complexity for the CLEAN algorithm plotted against the number of microphones (A), frequency bins (B), and scan points (C).

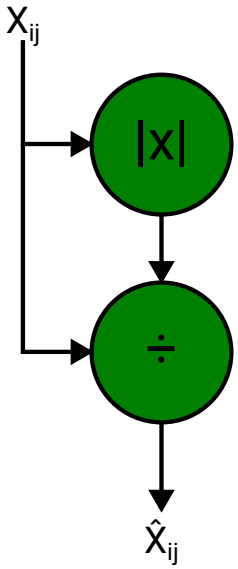


FIGURE 3.9: PE for computing the PHAT.

into their absolute value and squared. Together, this brings the sequential computational complexity of constructing the SRP spectrum to be equal to $O(LNM)$. To assist in the process of constructing a parallelized implementation for this operation, the partial sums S_i and S are defined in equations (3.16) and (3.17) respectively.

$$\begin{aligned} S_i^{(-1)} &= 0 \\ S_i^{(n)} &= S_j^{(n-1)} + \mathbf{g}_n^* \hat{\mathbf{X}}_{ni} \end{aligned} \tag{3.16}$$

$$\begin{aligned} S^{(-1)} &= 0 \\ S^{(n)} &= S^{(n-1)} + \left| S_n^{(M-1)} \right|^2 \end{aligned} \tag{3.17}$$

From the definitions of the partial sums, the PEs and DDG are easily derived. These are presented in Figure 3.10 and show how the operation of computing the SRP for a single candidate DOA can be parallelized. The parallelized computational complexity of constructing the SRP spectrum for a single candidate DOA is equal to $O(MN)$. To construct the complete spectrum, the DDG design can be stacked, allowing each of the SRPs for each of the L candidate DOAs to be computed in parallel, giving a parallelized computational complexity of $O(MN)$ for the complete SRP spectrum. However, one must take into account that the number of resources required to achieve this complexity can become quite substantial. To compute the SRP for a single DOA candidate, $N(M+1)$ PEs are required. Hence, to compute the complete SRP spectrum in parallel a total of $LN(M+1)$ PEs are required.

Extracting DOA estimates

From the SRP spectrum the P largest peaks are associated with the DOAs of the P signal sources in the room. For a single signal source, one only needs to find the maximum value

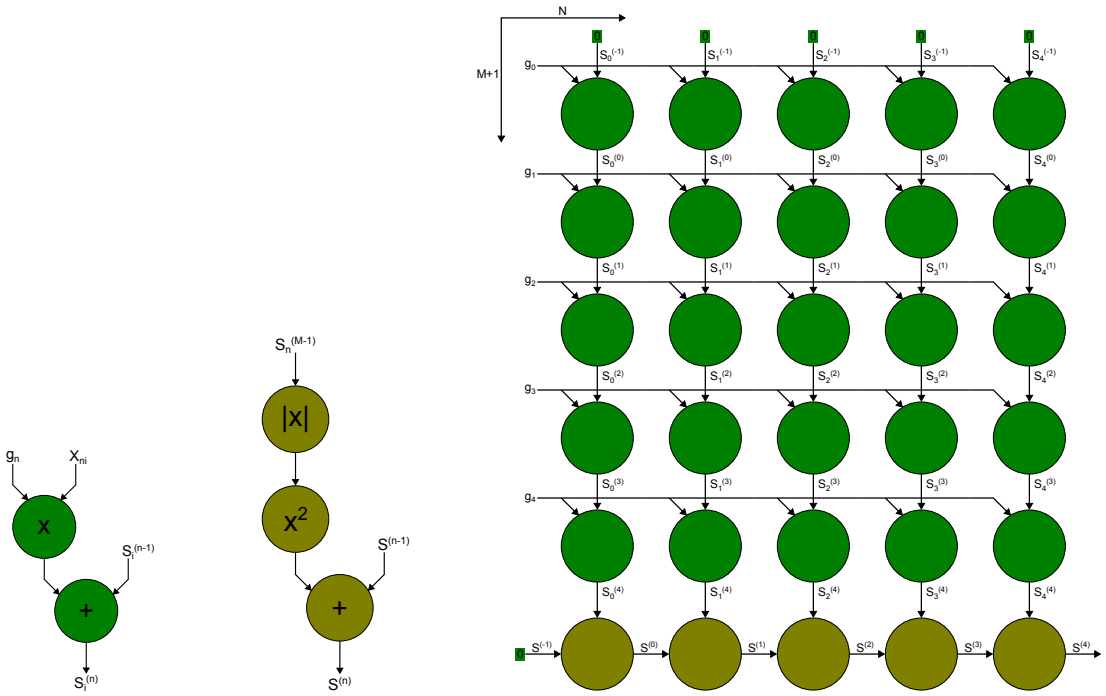


FIGURE 3.10: PEs (A, B) and DDG (B) for SRP.

in the spectrum, whose position in the spectrum determines its DOA. As an SRP value is associated with each of the L scan points, one can sort these values using any preferred sorting algorithm and select the P largest values. However, the issue is that when the P largest values are directly chosen from the SRP spectrum, these values are not necessarily associated with different signal sources as one signal source might be stronger than the other. Therefore, the SRP values surrounding the maximum SRP value can be larger than the value of the second peak. To solve this issue, a maximum filter is applied onto the SRP spectrum to identify the local maxima. From these local maxima, the P largest values can be chosen to obtain the P signal sources, each of which is more likely associated with a different signal source. To apply a maximum filter, a kernel must be defined. The size of the kernel was chosen to be 3-by-3 pixels large. Hence, for each pixel in the SRP spectrum, its neighboring 8 pixels are utilized to determine the filtered value for that pixel. This means that for a spectrum containing L pixels a total of $8L$ comparisons need to be made. Therefore, the computational complexity of applying the maximum filter to the SRP spectrum is equal to $O(L)$. The SRP values associated with the positions of the local maxima are then sorted and the P largest local maxima are used to determine the DOAs of the P signal sources. Given that the number of local maxima positions will be significantly less than the total number of scan positions, the computational complexity of extracting the DOA estimates is roughly equal to $O(L)$. To parallelize the operation, the Herk/Gil-Werman algorithm can be used. This algorithm reduces the number of comparisons that need to be performed for each pixel in the SRP spectrum. Using this algorithm less than 3 comparisons per pixel need to be performed [23], but as was shown in [24] this can actually be further reduced. Nevertheless, each pixel still needs to be processed and the computational complexity in the Big-O notation remains equal to $O(L)$.

Operation	Sequential	Parallelized
Phase Transform	$O(MN)$	$O(1)$
SRP	$O(LMN)$	$O(MN)$
DOA Extraction	$O(L)$	$O(L)$
Total narrowband	$O(LMN + MN + L)$	$O(MN + L)$
Total wideband	$O(KLMN + KMN + KL)$	$O(KMN + KL)$

TABLE 3.3: Narrowband computational complexities of SRP-PHAT components

Total complexity

The computational complexities for each individual component of the SRP-PHAT algorithm have been summarized in Table 3.3. From this table the total computational complexity of the sequential narrowband implementation of SRP-PHAT algorithm is derived and is equal to $O(LMN + MN + L)$. Through parallelization, the computational complexity can be reduced to $O(MN + L)$.

The wideband complexity is straightforward as each of components needs to be executed once for each of the K frequency bins. Hence, the sequential wideband computational complexity of SRP-PHAT is equal to $O(KLMN+KMN+KL)$. . Through parallelization, the computational complexity can be reduced to $O(KMN + KL)$.

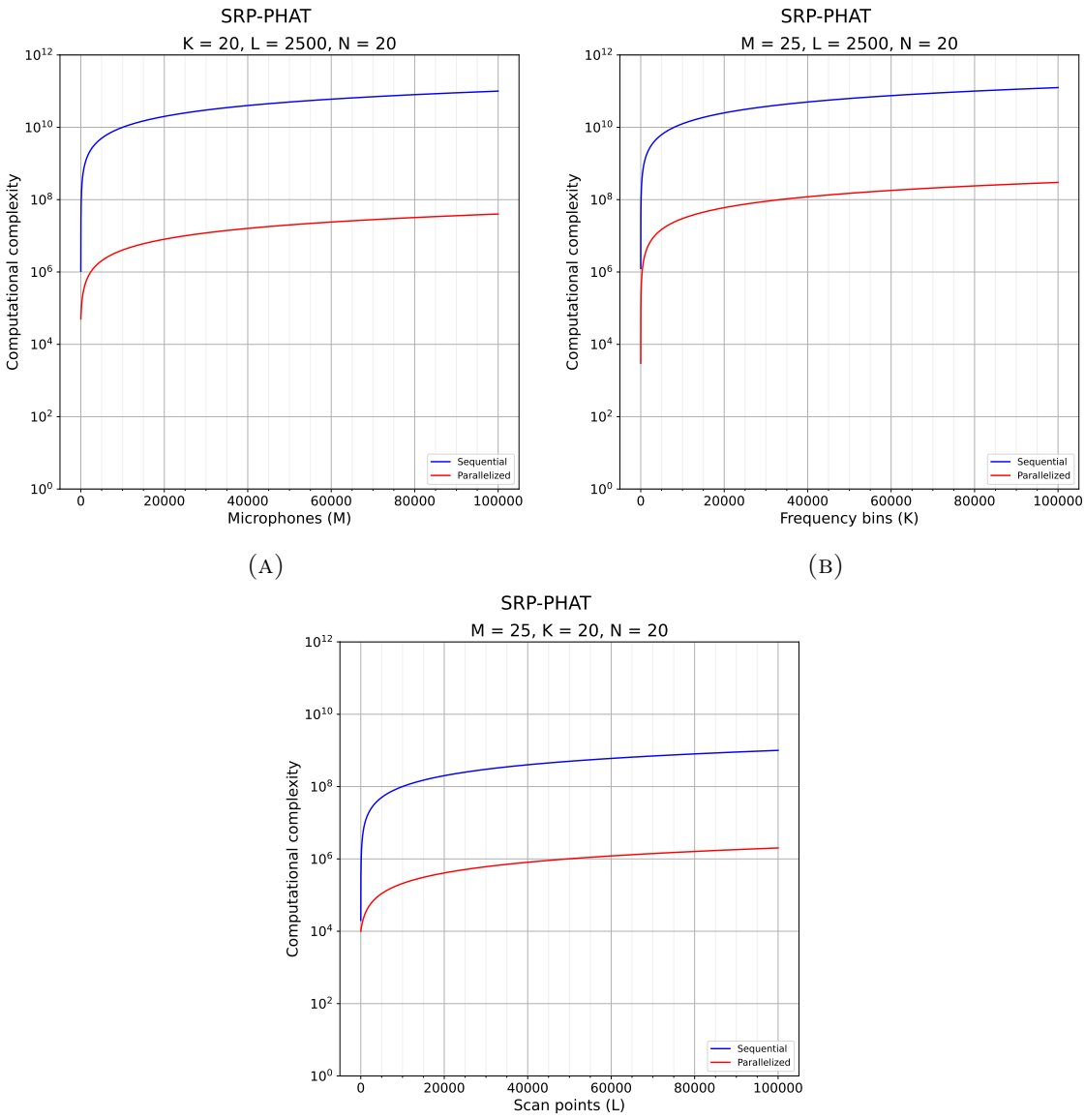
In Figure 3.11 the total sequential and parallelized computational complexities are plotted against the number of microphones, frequency bins, and scan points. The figure shows that the parallelized approach leads to lower computational complexities when considering each of these parameters. In particular, the parallelized approach seems to affect the efficiency of the algorithm when considering the number of microphones or the number of scan points as the parallelized computational complexity can be seen to increase less significantly with these parameters. When considering the number of frequency bins this increase appears to be the same between both the sequential and parallelized approach. In conclusion, the parallelized approach to the SRP-PHAT algorithm leads to an implementation that leads to an overall reduction of the computational complexity. In particular, the parallelized complexity scales well against the number of microphones and number of scan points in the scan grid.

3.4 2D-MUSIC

Similar to the other algorithms discussed thus far, the computational complexity of the 2D-MUSIC algorithm will be assessed through the computation of both sequential and parallelized complexities associated with each constituent component comprising the algorithm. Subsequently, the determined computational complexities of these individual components will be integrated to formulate the computational complexities for both the narrowband and wideband implementation of the algorithm.

Cross-spectral matrix

The formulation of the CSM for the MUSIC algorithm mirrors that of the CLEAN algorithm and has undergone comprehensive examination in Section 3.2. Consequently, to maintain conciseness in this report, detailed discussion on this matter will be omitted in the current section.



(A) (B)

(C)

FIGURE 3.11: Sequential and parallelized computational complexity for the SRP-PHAT algorithm plotted against the number of microphones (A), frequency bins (B), and scan points (C).

Eigendecomposition

Eigendecomposition of a matrix is a complex operation which converts the matrix into an equivalent representation in terms of its eigenvalues and eigenvectors. It is a well-studied subject with multiple research papers analyzing its computational complexity and methods to reduce the computational load of the operation [11, 63, 3, 67, 81, 19].

Conventional methods, such as the Jacobi method approach the eigendecomposition of matrix using an iterative approach. They compute the eigenvalues and eventually the eigenvectors with a complexity of $O(n^3)$ for n-by-n matrices. Due to the high complexity of this approach, many other algorithms have been developed with the intention of reducing the computational load of eigendecomposition. Direct methods are less commonly used as they are limited to small matrices ($n < 4$). Semi-direct methods such as Lanczos and the Householder algorithm construct an intermediate matrix and continue with an iterative method [39, 30]. Strassen's algorithm, developed by Volker Strassen in 1969, also known as fast matrix multiplication, was the first discovered algorithm to reduce the problem to a theoretical complexity of $O(n^{2.373})$ [75]. More recently in 2023, Williams, Xu and Zhou, showed a computational complexity of $O(n^{2.371552})$ [80]. However, both of these algorithms cannot be used in practice as the constant coefficient that they utilize is so large that they are only worthwhile for matrices that cannot be handled on present-day computers. So far, only approaches to general matrices have been considered. However, eigendecomposition in the 2D-MUSIC algorithm is performed on a Hermitian matrix whose properties can be exploited to achieve faster computation of the eigendecomposition. Specialized algorithms exist that achieve lower computational loads for such matrices [48]. In this paper, the computational complexity of eigendecomposition of the covariance matrix is based on the Jacobi method and no distinction is made between the sequential and parallel computational complexity. Hence, the computational complexity of eigendecomposition of the CSM is equal to $O(M^3)$ for both cases.

Pseudospectrum

The pseudospectrum is constructed by computing the power estimates for each of the L candidate DOAs. The equation to compute the power estimate for a single DOA is given in Equation (2.17). Let us simplify the notation and consider only a single frequency bin. Then the corresponding equation can be written as shown in Equation (3.18).

$$\frac{1}{A} = \sum_{p=0}^{M-1} \sum_{q=0}^{M-P-1} \sum_{r=0}^{M-1} \mathbf{g}_r^H \mathbf{E}_{rq} \mathbf{E}_{pq}^H \mathbf{g}_p \tag{3.18}$$

This operation is repeated for each of the L DOA candidates in the scan grid. Hence, the sequential computational complexity is equal to $O(M^3 - PM^2)$ for a single DOA candidate. The sequential computational complexity for the complete pseudospectrum is equal to $O(LM^3 - LPM^2)$. To parallelize the operation, first define the partial sum S_{pq}.

$$S_{pq}^{(-1)} = 0, \quad 0 \leq p < M, \quad 0 \leq q < (M - P)$$

$$S_{pq}^{(n)} = S_{pq}^{(n-1)} + \mathbf{g}_n^H \mathbf{E}_{nq} \mathbf{E}_{pq}^H \mathbf{g}_p, \quad 0 \leq n < M$$

Next, define the second partial sum S_q.

$$S_q^{(-1)} = 0$$

$$S_q^{(n)} = S_q^{(n-1)} + S_{nq}^{(M-1)}$$

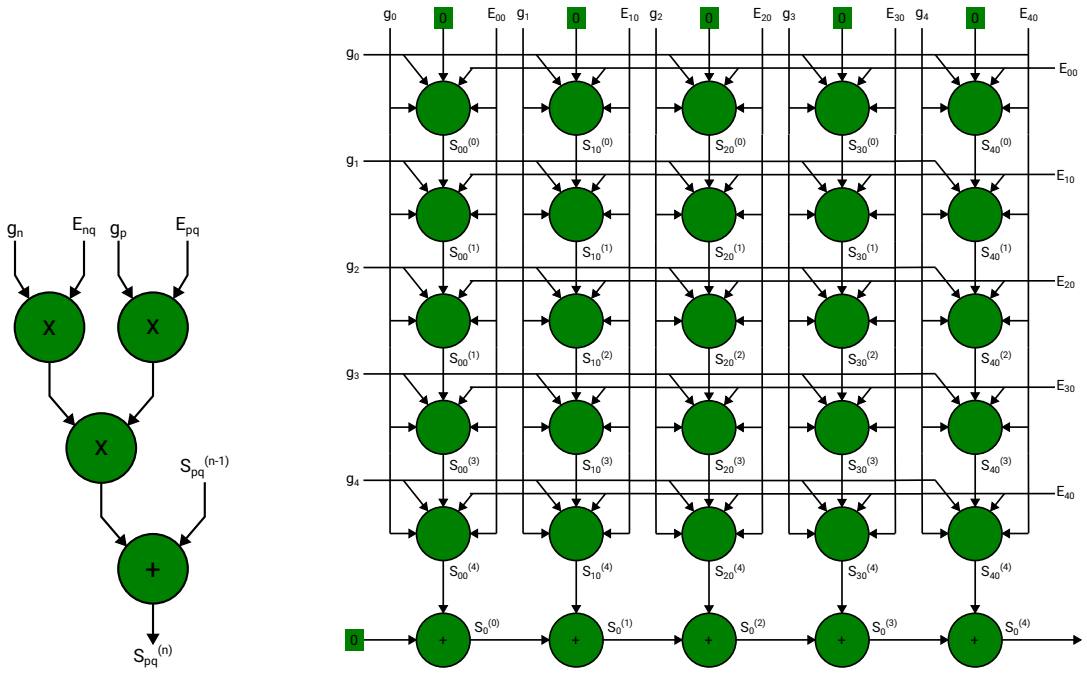


FIGURE 3.12: PE (A) and DDG (B) for computing a single candidate DOA in the pseudospectrum.

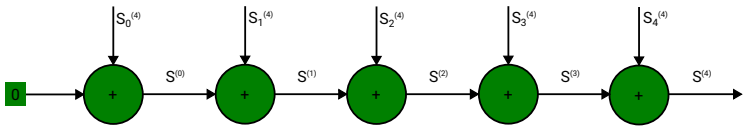


FIGURE 3.13: DDG for summing the outputs of the DDG in Figure 3.12b

which will yield M results. To obtain the power estimate, the final partial sum S is defined.

$$S^{(-1)} = 0$$

$$S^{(n)} = S^{(n-1)} + S_n^{(M-P-1)}$$

The PEs corresponding with the second and third partial sums are straightforward. However, the PE for the first partial sum is more complex and defined in Figure 3.12a. The DDG will be three dimensional and is achieved by stacking the DDG in Figure 3.12b. However, the DDG can be further optimized by diagonally summing the outputs of each PE diagonally as was shown for the DDG for CB in Figure 3.5. When the DDG is stacked as shown in Figure 3.14, a single output is produced per layer which needs to be summed together. This can be achieved with the design presented in Figure 3.13. The complete DDG to compute a single power estimate is therefore given as shown in Figure 3.14. From this implementation, the longest path of PEs is equal to $(M + 2) + (M - P)$. Hence, the complexity of the parallelized approach to computing a single value of the pseudospectrum has a complexity of $O(M)$. The computational complexity of this design for the complete pseudospectrum is obtained by scaling this complexity by the number of scan directions, L. Hence, the parallelized computational complexity for computing the is equal to $O(LM)$. However, due to the data independence between each of the L scan points, it is theoretically possible to compute the complete pseudospectrum with a parallelized computational

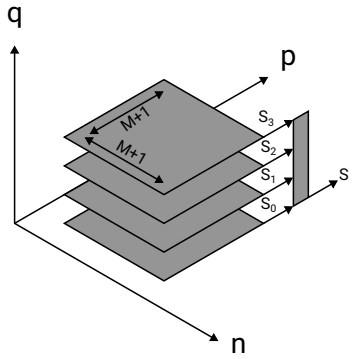


FIGURE 3.14: Stacked DDG for computing a single power estimate ($M - P = 4$)

Operation	**Sequential**	**Parallelized**
Cross-spectral matrix	$O(NM^2)$	$O(N)$
Pseudospectrum	$O(LM^3 - LPM^2)$	$O(M)$
Eigendecomposition	$O(M^3)$	$O(M^3)$
DOA Extraction	$O(L)$	$O(L)$
Total Narrowband	$O(LM^3 + NM^2 - LPM^2)$	$O(M^3 + N + L)$
Total Wideband	$O(KLM^3 + KNM^2 - KLPM^2)$	$O(KM^3 + KN + KL)$

TABLE 3.4: Complexities of MUSIC components

complexity of only $O(M)$ by computing each of the L DOAs in parallel. However, this will require a significant amount of resources which might prove to be infeasible depending on the number of available resources.

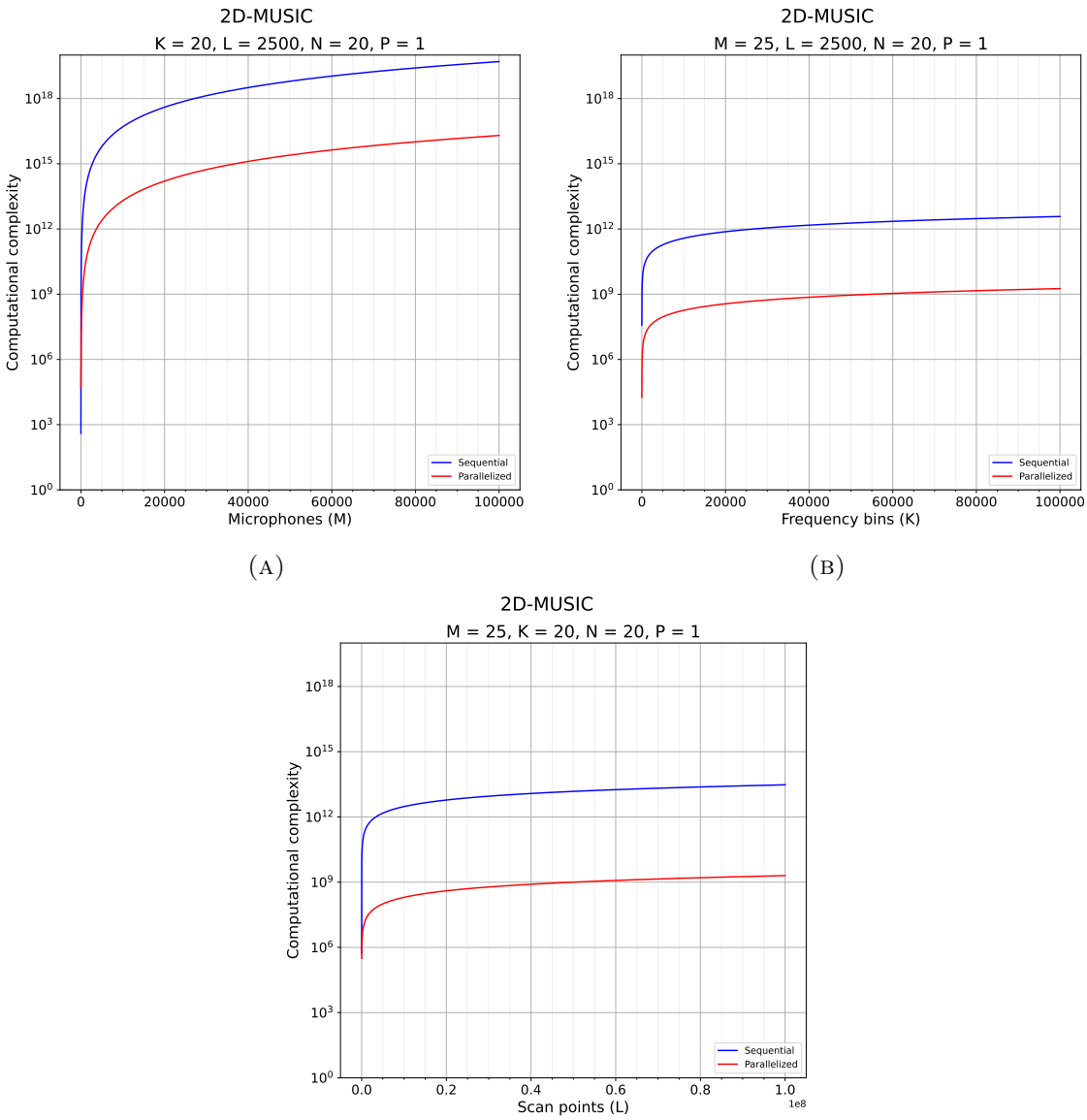
Extracting the DOA-estimates

Extracting the DOA estimates from the pseudospectrum is done similarly to SRP-PHAT described in Section 3.3. However, instead of using a SRP spectrum, the pseudospectrum is utilized instead. Hence, the sequential and parallelized computational complexity of this operation is equal to $O(L)$.

Total complexity

Similar to the other algorithms, the narrowband complexity of the 2D-MUSIC algorithm is easily derived by summing each of the complexities of its individual components. Hence, the sequential complexity of 2D-MUSIC is equal to $O(NM^2 + LM^3 - LPM^2 + M^3 + L)$. By eliminating the least significant terms, the total complexity can be simplified to $O(NM^2 + LM^3 - LPM^2)$. The parallelized narrowband computational complexity is equal to $O(N + M + M^3 + L)$ which can be simplified to $O(N + M^3 + L)$. The wideband complexity is obtained through multiplication with the number of frequency bins that need to be obtained. Hence the sequential computational complexity of the algorithm is equal to $O(KNM^2 + KLM^3 - KLPM^2)$ and the parallelized computational complexity is equal to $O(KN + KM^3 + KL)$.

In Figure 3.15 the total sequential and parallelized computational complexities are plotted against the number of microphones, frequency bins, and scan points. The figure shows that the parallelized approach leads to lower computational complexities when considering each of these parameters. In particular, the parallelized approach seems to



(A) (B)

(C)

FIGURE 3.15: Sequential and parallelized computational complexity for the 2D-MUSIC algorithm plotted against the number of microphones (A), frequency bins (B), and scan points (C).

affect the efficiency of the algorithm when considering the number of scan points as the parallelized computational complexity can be seen to increase less fast for this parameter. When considering the number of frequency bins this increase appears to be the same between both the sequential and parallelized approach. The number of microphones seem to have the most influence on the magnitude of the computational complexity. It can also be seen that for a small number of microphones the difference between the parallelized and sequential approach for the algorithm is minimal. Only when considering a large number of microphones will the difference between these approaches become apparent. However, having a microphone array with that many microphones would be unrealistic for real-world application. In conclusion, the parallelized approach to the 2D-MUSIC algorithms leads to an implementation that leads to an overall reduction of the computational complexity. When considering the number of microphones, the algorithms parallelized approach will not yield noticeable results unless an unrealistic number of microphones is used. However, the parallelized complexity scales well against the number of scan points in the scan grid.

3.5 2D Unitary ESPRIT

In Section 2.8 2D Unitary ESPRIT was introduced and each of its components were explained. In this section each of its components are explored in more detail regarding their computational complexities and data dependencies.

Selection matrices

The selection matrices used to select the microphone signals for each of the subarrays presented in Figure 2.6 remain constant. Hence, they only need to be computed once and can be accessed using a LUT. Therefore, the construction of these selection matrices does not contribute to the complexity of the algorithm.

Extended Data Matrix

The construction of the extended data matrix is shown in Equation (2.24). It involves horizontally stacking the data matrix $\mathbf{X}$ with the data exchange matrix $\mathbf{\Pi}_M\mathbf{X}\mathbf{\Pi}_N$. From the definition of $\mathbf{\Pi}$ matrices in Equation (2.19) it is clear that this matrix has ones on its antidiagonal while the other values are zero. Therefore, it is possible to define the expression $\mathbf{\Pi}_M\mathbf{X}\mathbf{\Pi}_N$ as shown in Equation (3.19).

$$(\mathbf{\Pi}_M\mathbf{X}\mathbf{\Pi}_N)_{ij} = X_{(M-i)(N-j)}, \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \tag{3.19}$$

Hence, the operation involves flipping the data matrix along both its axes. No arithmetic operations are required for this operation. Thus, the construction of the extended data matrix does not contribute to the algorithm's complexity.

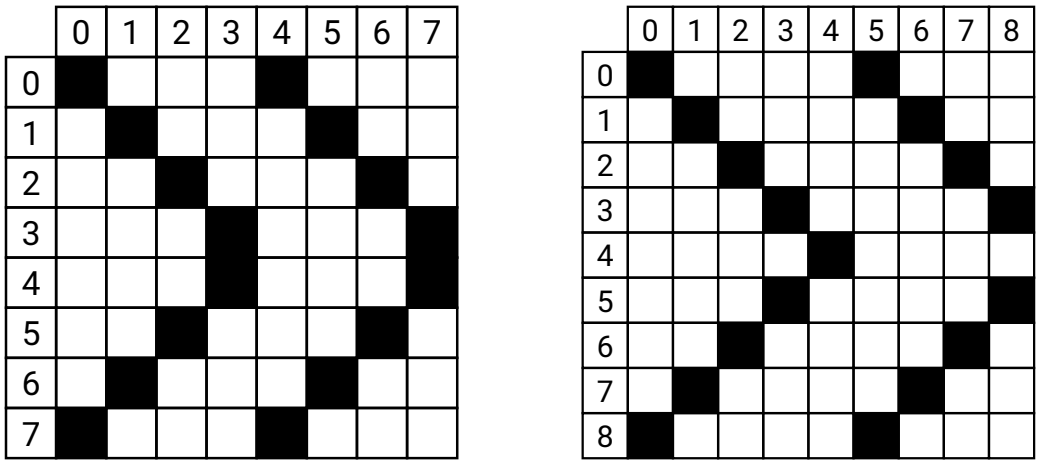
Matrix transformation

The conversion of the complex-valued extended data matrix into the real-valued data matrix $\mathcal{T}(\mathbf{X})$ is given in Equation (2.26). Let us simplify the notation to

$$\mathcal{T}(\mathbf{X}) \triangleq \mathbf{Q}_M^H\hat{\mathbf{X}}\mathbf{Q}_{2N} \in \mathbb{R}^{M\times 2N} \tag{3.20}$$

where $\hat{\mathbf{X}}$ represents the extended data matrix $\begin{vmatrix}\mathbf{X} & \mathbf{\Pi}_M\mathbf{X}^*\mathbf{\Pi}_N\end{vmatrix}$ From the definition of the sparse unitary matrix, it can be observed that $\mathbb{Q}_M$ has $2M$ non-zero elements if M is even

and $4\lfloor M/2 \rfloor + 1$ non-zero elements when M is odd. These elements are distributed such that each row of the sparse unitary matrix contains two non-zero elements. When M is odd the center row will only contain one element. This is illustrated in Figure 3.16. Due to



(A) Sparse unitary matrix for even arguments ($2n = 8$)

(B) Sparse unitary matrix for odd arguments ($2n + 1 = 9$)

FIGURE 3.16: Structure of the sparse unitary matrix Q for odd and even arguments. The black boxes represent the non-zero values.

the structure of the unitary matrices, the operation $\mathbf{R} = \mathbf{Q}_M^H \hat{X}$ can be described as shown in Equation (3.21).

$$\mathbf{R}_{ij} = (\mathbf{Q}_M^H)_{ip}\hat{\mathbf{X}}_{pj} + (\mathbf{Q}_M^H)_{iq}\hat{\mathbf{X}}_{qj}, \quad 0 \leq i \leq (M-1) \quad 0 \leq j \leq (2N-1) \tag{3.21}$$

Where $p = i \mod (M/2)$ and $q = (M-1) - p$. Then, the operation $\bar{\mathbf{X}} = \mathbf{R}(\mathbf{Q}_{2N})$ can be described as shown in Equation (3.22).

$$\bar{\mathbf{X}}_{ij} = \mathbf{R}_{ir}(\mathbf{Q}_{2N})_{rj} + \mathbf{R}_{is}(\mathbf{Q}_{2N})_{sj}, \quad 0 \leq i \leq (2N-1) \quad 0 \leq j \leq (M-1) \tag{3.22}$$

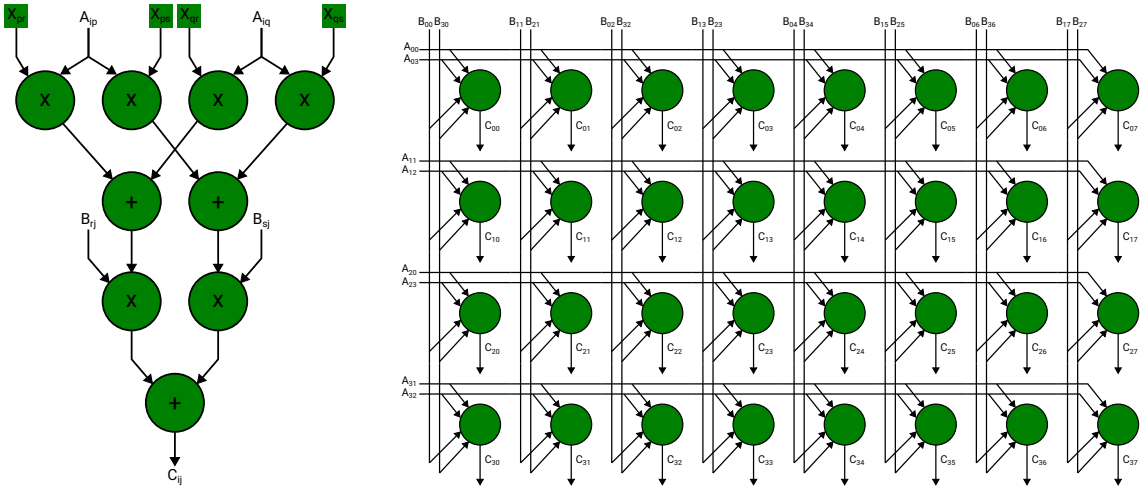
Where $r = j \mod (M/2)$ and $s = (M-1) - r$. Substituting Equation (3.21) in Equation (3.22) yields the equation in (3.23) which describes how each element in the transformed data matrix can be computed.

$$\mathcal{T}(\mathbf{X})_{ij} = (\mathbf{Q}_{2N})_{rj}((\mathbf{Q}_M^H)_{ip}\hat{\mathbf{X}}_{pr} + (\mathbf{Q}_M^H)_{iq}\hat{\mathbf{X}}_{qr}) + (\mathbf{Q}_{2N})_{sj}((\mathbf{Q}_M^H)_{ip}\hat{\mathbf{X}}_{ps} + (\mathbf{Q}_M^H)_{iq}\hat{\mathbf{X}}_{qs}) \tag{3.23}$$

These equations hold when M is even, but as shown in Figure 3.16, when M is odd, the center row of $\mathbf{A}$ will only contain one non-zero element. Hence, for $i = \lfloor M/2 \rfloor$ the equation can be expressed as shown in Equation (3.24).

$$\mathcal{T}(\mathbf{X})_{ij} = (\mathbf{Q}_M^H)_{ii}\hat{\mathbf{X}}_{ij}, \quad i = \lfloor M/2 \rfloor, \quad 0 \leq j \leq (2N-1) \tag{3.24}$$

Therefore, computing a single element of $\mathcal{T}(\mathbf{X})$ requires 9 operations. Hence, the number of operations required to transform the data matrix is equal to $18NM$ when M is even and $18NM - 2N$ when M is odd. Consequently, its complexity is equal to $O(NM)$ for both cases. The operation can be parallelized with the PE and DDG presented in Figure 3.17. It is important to note that this DDG is only applicable when M is even. However, the DDG for when M is odd will not yield a different computational complexity as each PE remains independent. Therefore, the computational complexity of the fully parallelized operation for computing $\mathcal{T}(\mathbf{X})$ is equal to $O(1)$.



(A) (B)

FIGURE 3.17: PE (A) and DDG (B) for transforming the data matrix.

Covariance matrix

Constructing the covariance matrix consists of a matrix multiplication of the transformed data matrix $\mathcal{T}(\mathbf{X}) \in \mathbb{C}^{M \times 2N}$ and its conjugate transpose. Hence, the operation can be described as shown in Equation (3.25).

$$\mathbf{C}_{ij} = \sum_{k=0}^{2N-1} \mathcal{T}(\mathbf{X})_{ik} \mathcal{T}(\mathbf{X})_{kj}^{H} \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (M-1) \tag{3.25}$$

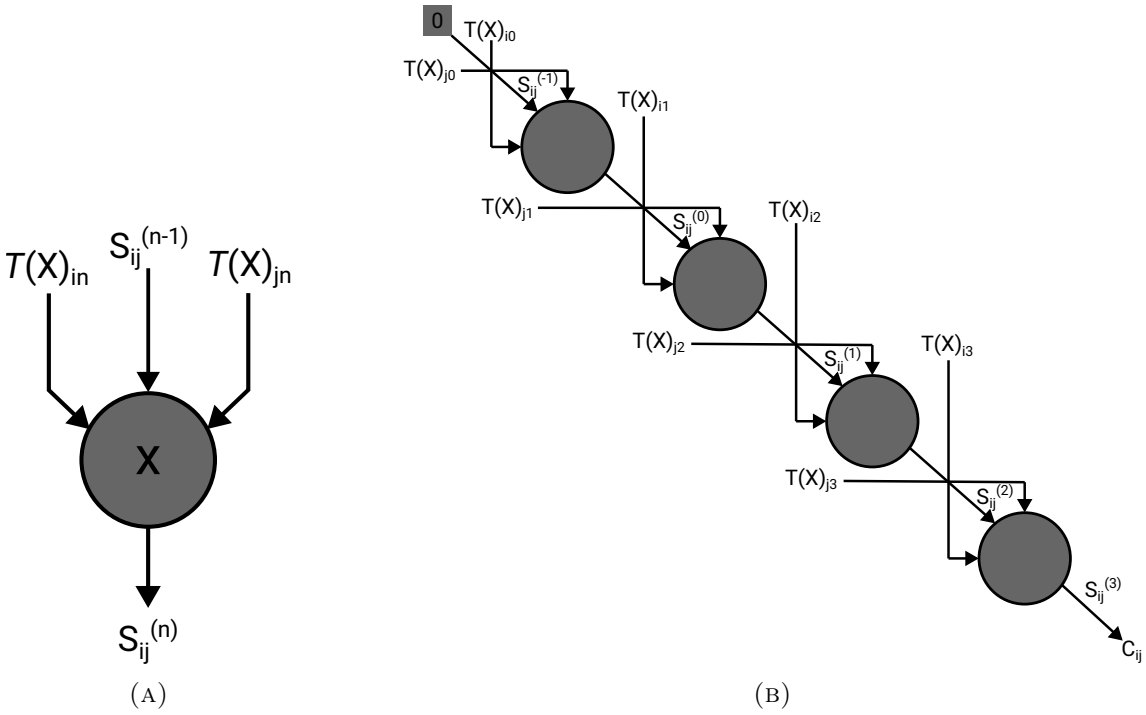
However, the transpose in Equation (3.25) can be omitted by rearranging the indexing parameters as shown in Equation (3.26).

$$\mathbf{C}_{ij} = \sum_{k=0}^{2N-1} \mathcal{T}(\mathbf{X})_{ik} \mathcal{T}(\mathbf{X})_{jk}^{*} \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (M-1) \tag{3.26}$$

The number of operations required to compute a single element of the covariance matrix requires $4N-1$ operations. The covariance matrix has dimensions $M \times M$. Consequently, $4NM^2 - M^2$ operations will need to be performed. Which yields a sequential computational complexity of $O(NM^2)$. The number of operations can be reduced when considering that the covariance matrix is Hermitian. Therefore, only half of its elements would need to be computed to be able to construct the covariance matrix. This does not affect the sequential computational complexity in the Big-O notation. When analyzing the data dependencies of this operations, it became clear that constructing a single DDG for computing each element of the covariance matrix proved difficult. However, the operation can certainly be parallelized. Let us define the partial sum in Equation (3.27).

$$\begin{aligned} S_{ij}^{(-1)} &= 0 \\ S_{ij}^{(n)} &= S_{ij}^{(n-1)} + \mathcal{T}(X)_{in} \mathcal{T}(X)_{jn} \quad 0 \leq n \leq 2N-1 \end{aligned} \tag{3.27}$$

The corresponding PE and DDG for computing a single element of the covariance matrix are defined in Figure 3.18. It is possible to stack the DDG such that each of the covariance matrix's elements can be computed in parallel. This means that the parallelized computational complexity of this operation is equal to $O(N)$.



(A) (B)

FIGURE 3.18: PE (A) and DDG (B) for computing a single element of the covariance matrix.

Real-valued Eigendecomposition

The computational complexity of eigendecomposition of a matrix has been addressed in Section 3.4. After eigenvalue decomposition of the covariance matrix, one obtains M eigenvectors. The largest P eigenvalues are selected and used as input for the next step in the algorithm. An important difference between 2D-MUSIC and 2D Unitary ESPRIT however, is the fact that instead of a complex-valued matrix, the decomposition is performed on a real-valued matrix. Hence, the total number of operations required for decomposition will be less than that of 2D-MUSIC. Nevertheless, the computational complexity of the operation is equal to $O(M^3)$.

Least-squares

The least-squares operation is performed on the obtained eigenvectors after eigenvalue decomposition. Furthermore, four subarrays are utilized, containing 20 microphones each. After selecting the values from the eigenvectors corresponding with the microphones of each subarray, four $\mathbb{R}^{20\times P}$ eigenvectors are obtained. To find both the azimuth an elevation of the incident source signals, two least-squares operations are performed to find the solution for $Y_u \in \mathbb{R}^{P\times P}$ and $Y_v \in \mathbb{R}^{P\times P}$ according to the equations $Y_u K_{U_1} = K_{U_2}$ and $Y_v K_{V_1} = K_{V_2}$ respectively. Similarly to eigenvalue decomposition, the exact computational complexity of the operation is not straightforward, and several papers have been published that explain the exact computational complexity in more detail. The main takeaway from the papers is that a least-squares problem of m degree polynomial regression with n measured values (Hence, $M_{sub} \geq P + 1$), has a complexity of $O(M_{sub}^2 P)$ using traditional methods. However, L.Li [44] shows that the arithmetic complexity of this problem does not exceed $O(M_{sub} \log^2 2P)$. Parallelization of the least-square problems is a complex problem due to

Operation	Sequential	Parallelized
Matrix Transformation	$O(MN)$	$O(1)$
Covariance Matrix	$O(NM^2)$	$O(N)$
Real-valued Eigendecomposition	$O(M^3)$	$O(M^3)$
Least Squares	$O(PM_{sub}^2)$	$O(PM_{sub}^2)$
Complex-valued Eigendecomposition	$O(P^3)$	$O(P^3)$
DOA Conversion	$O(P)$	$O(1)$
Total narrowband	$O(M^3 + P^3 + NM^2 + PM_{sub}^2)$	$O(M^3 + P^3 + PM_{sub}^2 + N)$
Total wideband	$O(KM^3 + KP^3 + KNM^2 + KPM_{sub}^2)$	$O(KM^3 + KP^3 + KPM_{sub}^2 + KN)$

TABLE 3.5: Complexities of ESPRIT components

the presence of data contra-flow and feedback loops in the underlying signal flow graph [60, 33]. Fortunately, these papers also provide solutions to the systolic array designs for the least-squares. However, as the operation is performed on relatively small matrices, the effort might not outweigh the gained reduction in complexity of the operation. Hence, the benefit of a parallelized approach might only become noticeable when dealing with a large number of microphones and a desire to find a large number of signal sources.

Complex-valued eigenvalue decomposition

After the least-squares operation for the four subarrays that are utilized, a $P \times P$ matrix is obtained. 2D Unitary ESPRIT performs a final complex-valued eigendecomposition on this matrix to obtain P eigenvalues which will be used to compute the final azimuth and elevation angles for each of the sources. It is known the complexity of this operation will therefore be equal to $O(P^3)$.

Extracting the DOA estimates

After the final two complex eigenvalues are obtained for 2D Unitary ESPRIT, all that remains is converting the real and imaginary part of these eigenvalues to azimuth and elevation angles using several trigonometric functions. Unlike the other algorithms, this does not require a grid-based search, but instead the P source directions can be directly derived from the eigenvalues obtained after the complex-valued eigenvalue decomposition mentioned in the previous section. This is a big benefit of ESPRIT, as this operation has a low computational complexity and this complexity is independent of the chosen scan range. To be more specific the computational complexity of converting the complex eigenvalues to DOA estimates can be expressed as $O(P)$. As each of the P eigenvalues can be processed independently, the parallelized computational complexity is equal to $O(1)$.

Total complexity

Similar to the other algorithms, the narrowband complexity of an algorithm is obtained by summing all of the obtained number of operations and computational complexities for each individual component. The narrowband sequential computational complexity is equal to $O(M^3 + P^3 + NM^2 + PM_{sub}^2)$. The parallelized computational complexity is equal to $O(M^3 + P^3 + PM_{sub}^2 + N)$. The wideband sequential and parallelized computational

complexity is equal to $O(KM^3 + KP^3 + KNM^2 + KPM_{sub}^2)$ and $O(KM^3 + KP^3 + KPM_{sub}^2 + KN)$.

Chapter 4

Simulation results

To evaluate the performance of each algorithm in determining the location of signal sources a set of simulations was performed. For this purpose, each algorithm was implemented using the python programming language. The input for these algorithms was provided through use of the *pyroomacoustics* library [70]. This library was chosen as it provides a convenient interface to construct simulation scenarios involving microphone arrays, multiple sound sources and three-dimensional rooms. For the purpose of this research a single room was defined. The dimensions of which are $12 \times 12 \times 12$ metres as shown in Figure 4.1. The microphone array was defined according to the URA in Figure 2.2 and placed on the floor of the room. Additionally, the library provides functionality to specify signal-to-noise (SNR) ratios and absorption coefficients among other variables. The SNR describes the ratio of the signal power against the power of the noise and is expressed in decibels (dB). A negative SNR indicates that the noise is stronger than the signal, whereas a positive SNR signifies the opposite scenario. The absorption coefficient determines how efficient the walls of the room are in absorbing sound waves. This coefficient has no unit. A value of 1.0 means that the signal source is completely absorbed when it hits a wall. A value of 0.0 means that it is completely reflected. By varying both the SNR and absorption coefficient, the algorithms can be evaluated for multiple sets of parameters and provide a comprehensive dataset to determine their performance. In this chapter, the simulation results for two distinct scenarios are presented. The first scenario involves a single sound source placed at different positions in the room. For the second scenario two coherent sound sources are placed at opposite sides of the room and are increasingly placed closer together. In both scenarios the generated noise is random, therefore Monte Carlo simulations were performed. The obtained data is represented in graphs and analyzed to determine each algorithm's performance and limitations in determining not only a single source, but two coherent sources as well.

4.1 Distance errors

The input of each algorithm are the signals obtained by each microphone in the array. The output of these algorithms is one DOA for each signal source. These estimated DOA's are in spherical coordinates. Hence, they can be converted to Cartesian coordinates using the equation in (4.1).

$$\begin{aligned} \dot{x} &= r\sin(\phi)\cos(\theta) \\ \dot{y} &= r\sin(\phi)\sin(\theta) \\ \dot{z} &= r\sin(\phi) \end{aligned} \tag{4.1}$$

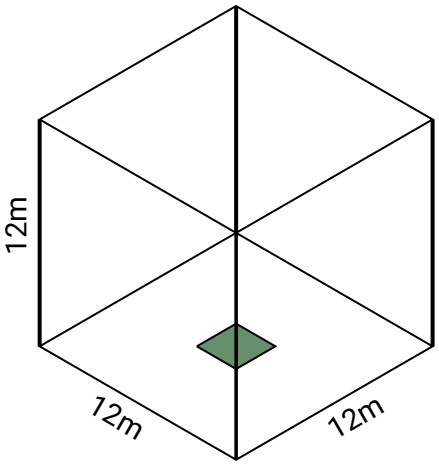


FIGURE 4.1: Room and array location used for simulations in *pyroomacoustic* (not to scale)

Then these coordinates can be projected on the scan grid, which is a plane located at one meter from, and parallel to the microphone array's origin. Equation (4.2) describes how these coordinates are projected onto the scan grid which is located at 1 meter from the array's origin.

$$\begin{aligned} x &= \dot{x}\frac{1}{\dot{z}} \\ y &= \dot{y}\frac{1}{\dot{z}} \\ z &= 1 \end{aligned} \tag{4.2}$$

Using these positions, the Euclidean distance between the projected estimated position $[x_0, y_0, 1]$ and the projected actual position $[x_1, y_1, 1]$ can be computed according to Equation (4.3).

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \tag{4.3}$$

4.2 Single Source Localization

The first simulations involved a single sound source emitting a 2500 Hz signal placed at one of the 80 pre-determined locations in the room, each of which is located 3 meters from the array's origin. Their projected locations on the 2-by-2 meter scan grid located at 1 meter from the array's origin is given in Figure 4.2. The frequency of 2500 Hz was chosen as it avoids both time and spatial aliasing, the concepts of which were introduced in section 2.1. For each algorithm, a single simulation results in a single DOA estimate for the signal source. For each combination of signal source position, SNR, and absorption coefficient, 1000 simulations were performed. To measure the performance of each algorithm in estimating each of the signal source position shown in Figure 4.2, the estimated distance errors are categorized by the SNR and absorption coefficient applied to the set of 1000 simulations. The Euclidian distance between the estimated position and the i-th signal source position obtained from the n-th simulation is defined as $d_i^{(n)}(SNR, a)$. Using this notation, the average distance error for all source positions for a given SNR and absorption

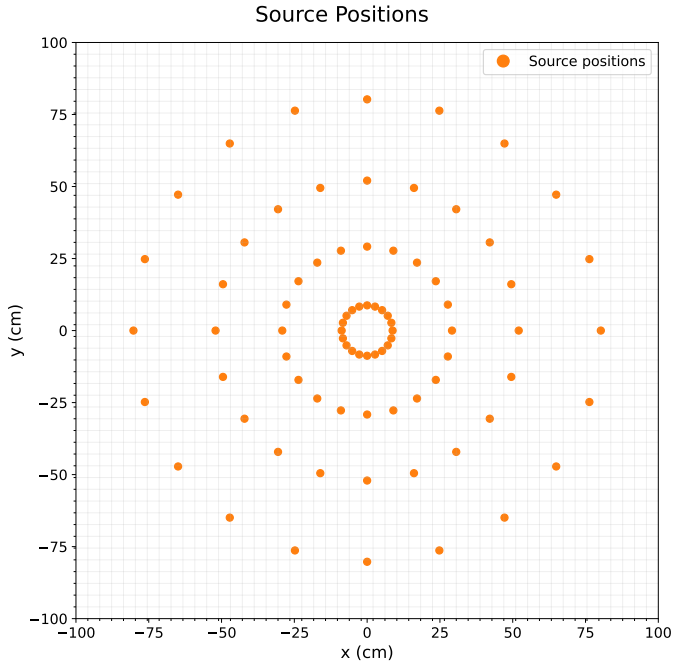


FIGURE 4.2: Signal source positions projected onto the scan grid that are utilized for the single-source simulations

coefficient can be defined as shown in Equation (4.4).

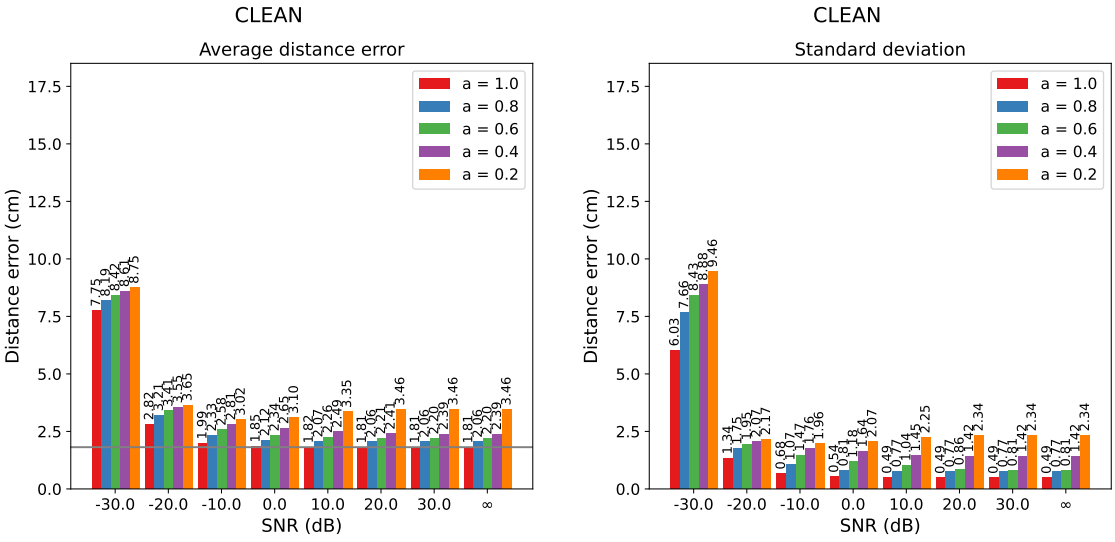
$$d_{avg}(SNR, a) = \frac{\sum_{i=0}^{80} \sum_{n=0}^{1000} d_i^{(n)}(SNR, a)}{80 \cdot 1000} \tag{4.4}$$

To evaluate the precision of each algorithm, we also compute the standard deviation across all estimated distance errors.

$$\sigma(SNR, a) = \sqrt{\frac{\sum_{i=0}^{80} \sum_{n=0}^{1000} (d_i^{(n)}(SNR, a) - d_{\text{avg}}(SNR, a))^2}{80 \cdot 1000}} \tag{4.5}$$

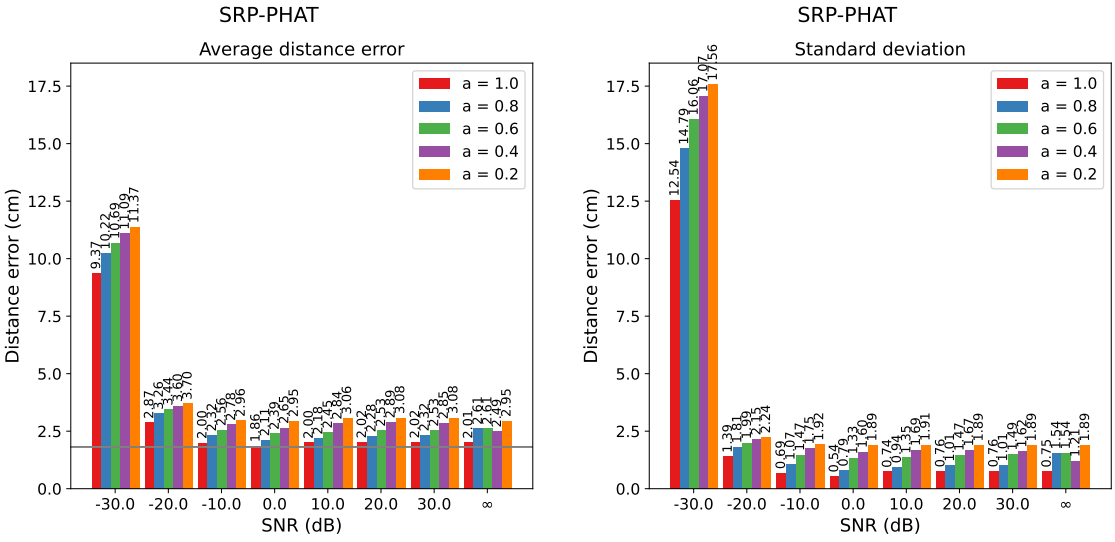
For each algorithm, bar charts showing the average distance error and the corresponding standard deviation are presented in Figures 4.3, 4.4, 4.5, and 4.6 for the CLEAN, SRP-PHAT, 2D-MUSIC and 2D Unitary ESPRIT algorithms respectively. As the projected actual source positions are not chosen to directly coincide with the scan grid positions there is an inherent systematic error present for the algorithms that utilize a scan grid. These algorithms cannot achieve a lower average distance error than this systematic error. The systematic error is equal to the average distance for each of the source positions to its closest neighboring scan grid point, which has been determined to be equal to approximately 1.81 cm. In the graphs this is represented by the grey horizontal line along with the simulation results in these bar charts.

In the next sections, the simulation results will be discussed separately for each SSL algorithm. However, one observation that applies to each algorithm can already be made. First, at an SNR of $-30dB$, each of the algorithms experiences a significant drop in their accuracy and precision. At this point, the noise is too strong for each algorithm to accurately and reliably identify the location of the signal source. Next each of the algorithms' results will be discussed separately, starting with the CLEAN algorithm.



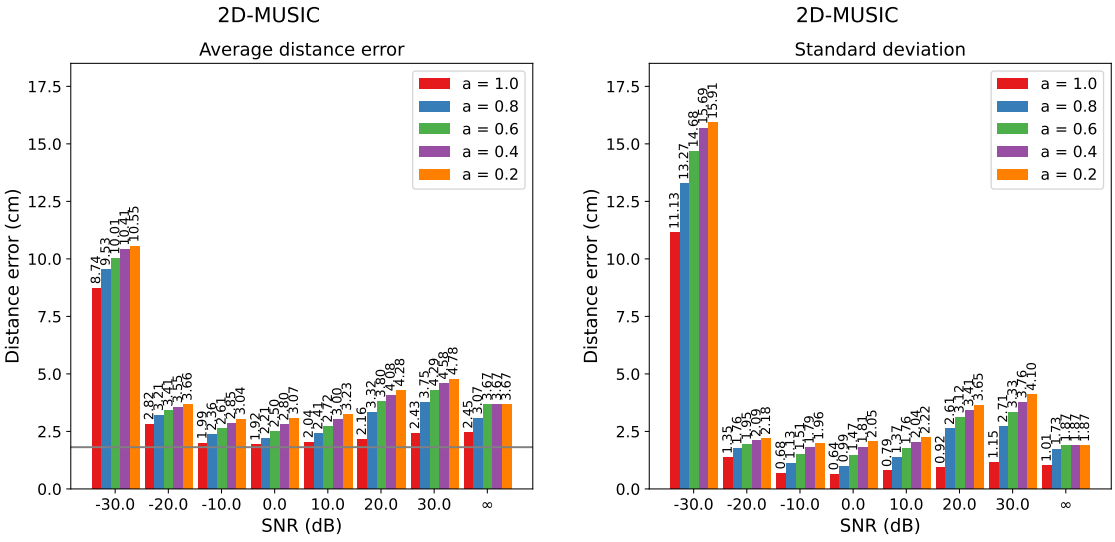
(A) Average distance error (B) Standard deviation

FIGURE 4.3: Single source simulation results for the CLEAN algorithm



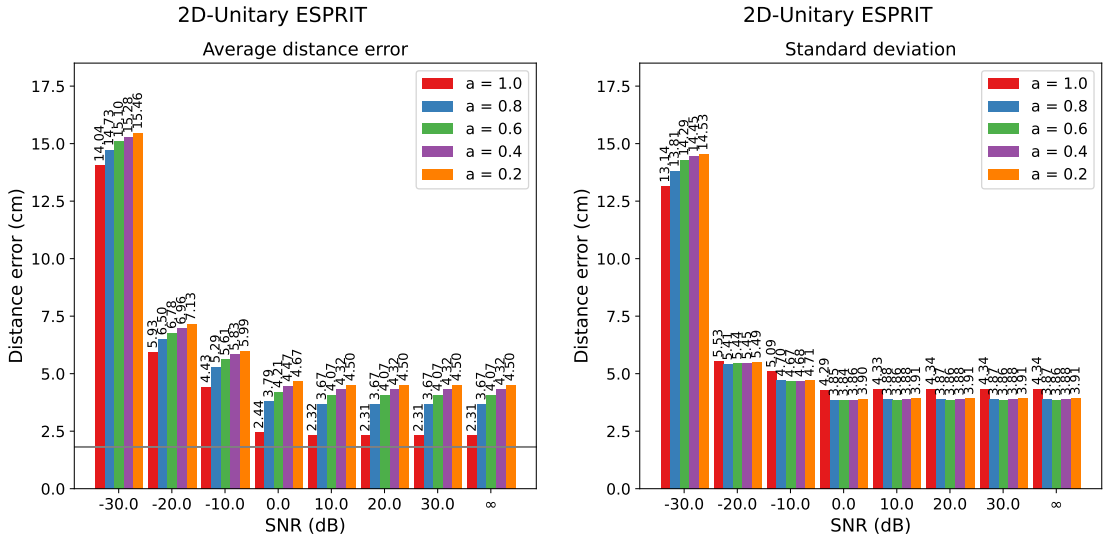
(A) Average distance error (B) Standard deviation

FIGURE 4.4: Single source simulation results for the SRP-PHAT algorithm



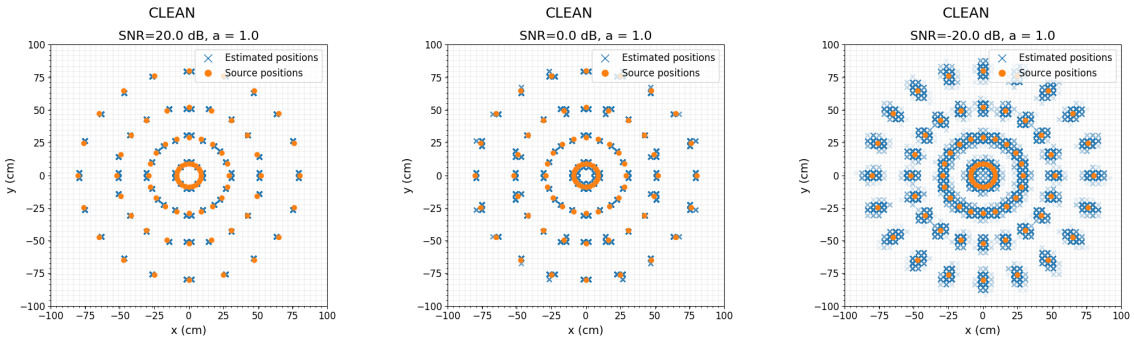
(A) Average distance error (B) Standard deviation

FIGURE 4.5: Single source simulation results for the MUSIC algorithm



(A) Average distance error (B) Standard deviation

FIGURE 4.6: Single source simulation results for the 2D Unitary ESPRIT algorithm



(A) (B) (C)

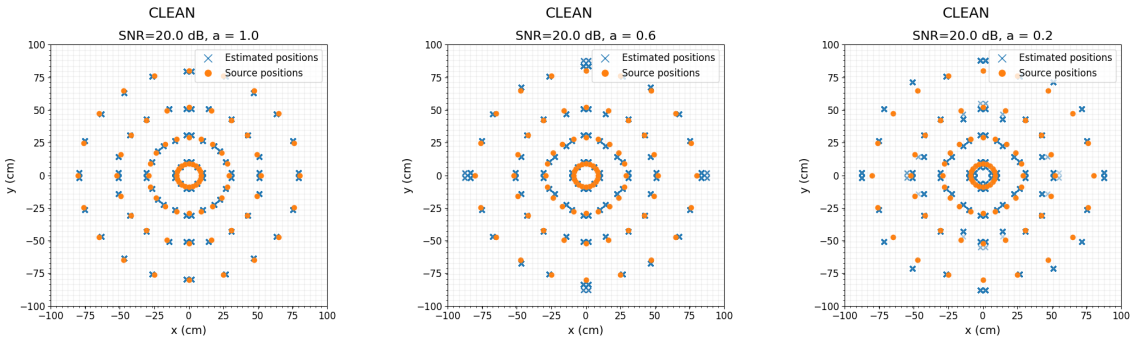
FIGURE 4.7: 80000 estimated and actual source positions illustrating the effect of noise on the CLEAN algorithm.

CLEAN

For ideal circumstances where there is no noise and no reverberation the CLEAN algorithm performs nearly optimal and achieves the lowest attainable average distance error of 1.81cm and a standard deviation of 0.49cm. This behavior is consistent for an absorption rate of 1.0 until the noise becomes stronger or equal to strength of the source signal. To better illustrate this, the estimated and actual signal source positions are plotted using scatterplots in Figure 4.7 for an absorption rate of 1.0 and three different SNRs. In each of the scatterplots the results of 1000 simulations are shown for each of the 80 source signal positions. Therefore, a total of 80000 estimated positions are plotted. When there is a little noise, the CLEAN algorithm is able to consistently and accurately determine the closest location to the actual source position. As noise increases, so does the variance in the estimated positions and consequently the average distance error and the standard deviation of the algorithm increases as well. Besides noise, the algorithm is also negatively affected by reverberation. Though this decline in performance is gradual, there is a significant drop in both accuracy and precision when the lowest absorption coefficient of 0.2 is utilized. When the noise becomes stronger than the source signal this drop is less exaggerated due to noise having a more profound effect on the average distance error than the low absorption coefficient. To better illustrate the effect of reverberation on the estimated positions three scatterplots are provided in Figure 4.8. Once again, the estimated positions are plotted in each scatterplot. However, now the absorption coefficient is different in each plot while the noise level remains equal at 20dB for each plot. The figure shows that reverberation causes the estimated positions to be located further from the actual source positions and introduces some variance in the estimated positions. Interestingly, the variance in the estimations is predominant for sources that are closely located to the imaginary x and y axis originating in the center of the scan grid. The exact reason for this could not be determined with the data that was obtained during this research. It might be caused by the shape of the environment, the positioning of the signal sources or perhaps due to the shape of the microphone array.

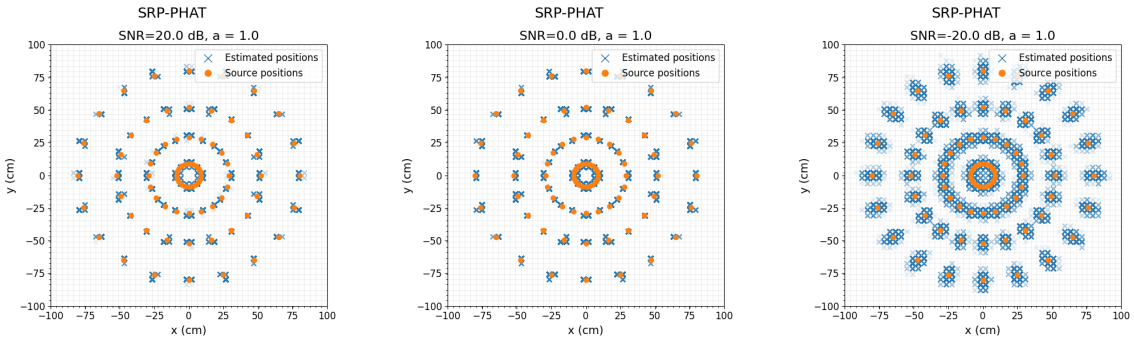
SRP-PHAT

Next is the SRP-PHAT algorithm. This algorithm achieves its best performance when the strength of the noise is equal to the strength of the signal sources. Although it is not able to achieve the minimum achievable average distance error of 1.81cm, it can come



(A) (B) (C)

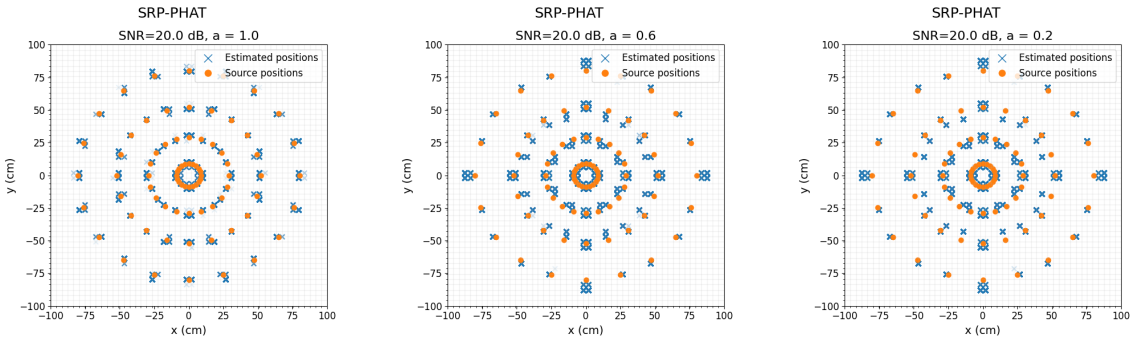
FIGURE 4.8: 80000 estimated and actual source positions illustrating the effect of reverberation on the CLEAN algorithm.



(A) (B) (C)

FIGURE 4.9: 80000 estimated and actual source positions illustrating the effect of noise on the SRP-PHAT algorithm.

very close with an average distance error of 1.86cm and a standard deviation of 0.54cm for an SNR of 0dB and no reverberation. To illustrate the effect of noise on the SRP-PHAT algorithm, three scatterplots are presented in Figure 4.9. Each scatterplot shows the positions of the 80000 estimated positions obtained from the simulations where there was no reverberation, but for different SNRs. For an SNR of 20dB and 0dB, there is some variation in the locations of the estimated positions. For the latter SNR there is slightly less variation than the former SNR although it is difficult to see. For an SNR of -20dB there is significantly more variation in the estimated positions. The SRP-PHAT algorithm is negatively affected by reverberation. Its performance decreases gradually as the absorption coefficient is decreased. The effect of reverberation is illustrated in the scatterplots in Figure 4.10. The scatterplots show the estimated positions for 1000 simulations for each signal source position and an applied SNR of 20dB. The scatterplots only differ in the applied absorption coefficient. For an absorption coefficient of 1.0 there is some variation in the estimated positions which is partially due to the positioning of the source positions. Source positions that are located in between scan grid points will result in slight variation in the estimated positions for those source positions. Source positions that are located very close to a single scan grid position will result in estimated positions that exhibit a more deterministic pattern with little to no variation. The estimations change as more reverberation is introduced to the simulations. Depending on the location of the source position, reverberation causes the estimated position to be observed either closer or



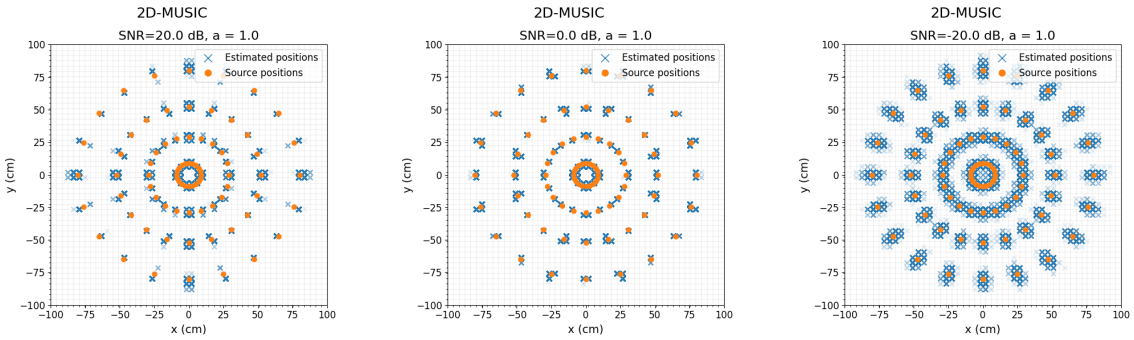
(A) (B) (C)

FIGURE 4.10: 80000 estimated and actual source positions illustrating the effect of reverberation on the SRP-PHAT algorithm.

further away from the scan grid's origin. Consequently, this observed position might cause either more or less variation to the estimated positions. Although the estimated positions change, the effect is less significant compared to the effect that noise can have. It can even be observed that for some signal source positions reverberation has no effect. Nevertheless, the overall average distance between estimated and actual source positions does increase.

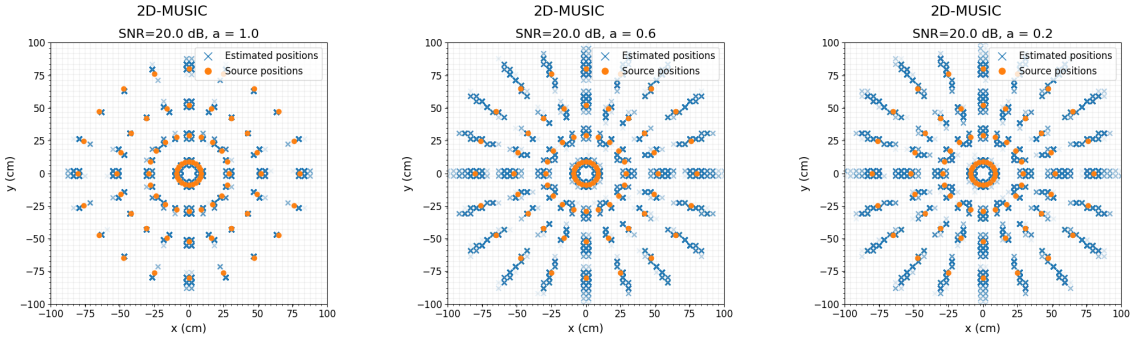
2D-MUSIC

The simulation results obtained using the 2D-MUSIC algorithm show that the algorithm achieves the best performance when the strength of the noise is equal to the strength of the source signal. The lowest obtained average distance error and standard deviation obtained by the algorithm are equal to 1.92cm and 0.64cm, respectively. To show the effect of noise on the algorithm, another three scatterplots are presented in Figure 4.11. For each plot, the estimated positions obtained from the 80000 simulations are plotted for a SNR of -20, 0, and 20 decibels while no reverberation was applied. In these plots it can be seen that at an SNR of 20dB, the algorithm has some variation in its estimations and occasionally estimated the source position at scan grid point that are not located nearest to the actual location of the source position. For an SNR of 0dB this is also true, however the estimations are more concentrated around the actual source position. For these SNRs it can also be observed that the variations predominantly occur for signal source positions that are located near the center between of two or more neighboring scan grid points. In contrast, source signal positions that are located very close to scan grid points experience less variance in their respective estimations. At an SNR of -20dB it can be observed that the strength of noise causes variation in the obtained estimated positions. Besides noise or lack thereof, the algorithms are also affected by the presence of reverberation. The higher the reverberation the lower the performance. In particular, when there is little noise present, the difference between the results obtained in absence of reverberation or presence of reverberation is exacerbated as indicated by the prominent increase in the average distance error and standard deviation. To better illustrate the effect reverberation has on the estimated positions three additional scatterplots are given in Figure 4.12. The plots show that the presence of reverberation causes more variance in the estimated positions. The maximum distances between the estimated and actual source positions do seem to be largest for sources that are placed furthest away from the scan grid's origin. Additionally, the estimated positions seem to vary along an imaginary lines that originate from the scan grid's origin and intersects with the source signal positions.



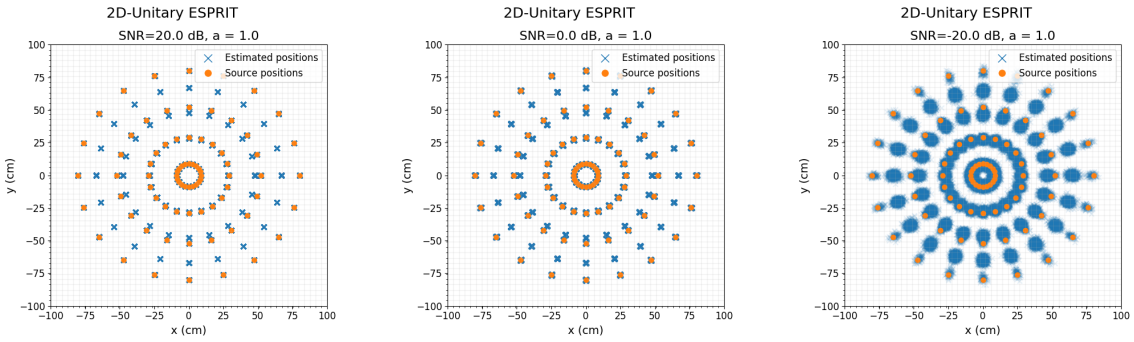
(A) (B) (C)

FIGURE 4.11: 80000 estimated and actual source positions illustrating the effect of noise on the 2D-MUSIC algorithm.



(A) (B) (C)

FIGURE 4.12: 80000 estimated and actual source positions illustrating the effect of reverberation on the 2D-MUSIC algorithm.

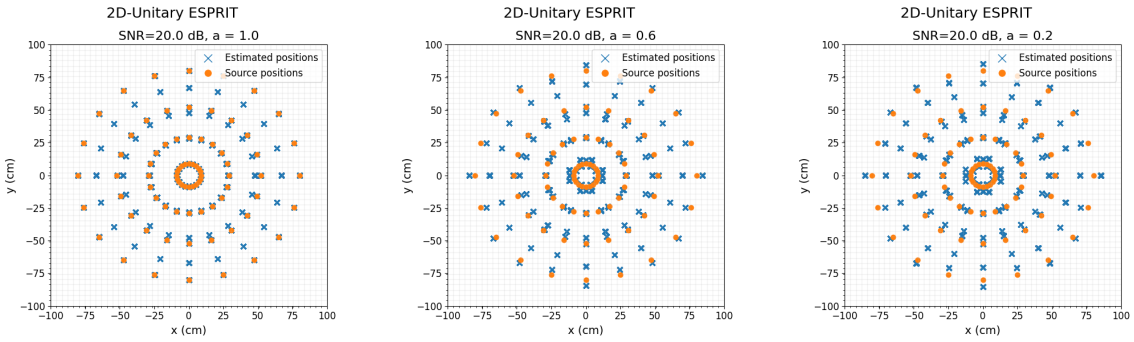


(A) (B) (C)

FIGURE 4.13: 80000 estimated and actual source positions illustrating the effect of noise on the 2D Unitary ESPRIT algorithm.

2D Unitary ESPRIT

The simulation results obtained for the 2D Unitary ESPRIT algorithm show that the lowest obtained for the average distance error is equal to 2.31cm while the lowest obtained standard deviation is equal to 3.84cm. Although these minima do not occur for the same combination of SNR and absorption coefficient. The lowest average distance error occurs for SNRs larger than 0dB and an absorption coefficient of 1.0, while the lowest obtained standard deviation occurs for an SNR of 0dB and an absorption coefficient of 0.6. It can also be seen that the algorithm's average distance error increases along with noise and reverberation. The algorithm exhibits a constant average distance error and standard deviation until the SNR drops below 10dB after which both the error and standard deviation start to increase. To better illustrate the effect of noise on the algorithm, the estimated positions obtained from the simulations are plotted in Figure 4.13. The results show that the estimated positions form two groups for each source location. One of the groups almost perfectly coincides with the source location. The second group of estimated locations is located towards the scan grid's origin. The further away from the center of the scan grid the source location is located, the larger the distance between these two groups. When the noise is increased, the area that these groups of estimated positions cover will grow. It also reveals that the number of estimated locations is higher in the group that is closest to the scan grid's origin. To better illustrate the effect of reverberation on the estimated positions, three additional scatterplots are given in Figure 4.14. The results show that in the absence of reverberation one of the two groups of estimated positions for each source location almost perfectly coincides with the that source location while the other group has an offset towards the scan grid's origin. When the absorption coefficient is set to 0.6, estimated positions for the center ring of source locations form a cross in the center of the scan grid. In the second-most-inner ring, the displacements occur to the left or right of the source position. The only exceptions are for the source locations that are directly on the x- or y-axes originating from the scan grids origin. These estimated positions only show slight displacement towards the scan grid's origin. This last observation also applies to the outer two rings of source locations. However, only for some source locations the estimated positions are displaced towards the scan grid's origin, while for other source locations the estimated positions move away from the array's origin. When the absorption coefficient is decreased further to 0.2, this displacement observed for all estimated positions increases by a small amount. However, the distance between each set of two groups of estimated positions remains nearly equal. This explains why the standard deviation remains constant



(A) (B) (C)

FIGURE 4.14: 80000 estimated and actual source positions illustrating the effect of reverberation on the 2D Unitary ESPRIT algorithm.

for scenarios where there is reverberation, while the average distance error does increase as seen in the bar charts in Figure 4.6.

4.3 Multiple Coherent Source localization

Next, simulations are performed using the same SNR and absorption coefficients used in the previous section. However, instead of a single source, two signal sources are placed at opposite sides of the microphone array at an azimuth of 50 and 130 degrees respectively. Each signal source is located at a distance of 3 meters from the microphone array's center. Both sources are coherent and emit a 2500Hz signal with equal amplitudes. Initially, the sources are placed at an elevation of 25 degrees. When 1000 simulations have been performed, the elevation is decreased by 1 degree while maintaining the distance of 3 meters from the array's center, which moves both signal sources closer towards each other. This process repeats until both signal sources reach an elevation angle of 10 degrees for a total of 16000 simulations. This is considered a single simulation run. Multiple simulation runs are performed. For each run, a different SNR and absorption rate is applied. The applied SNRs are -20 dB, -10 dB, 0 dB, 10 dB, and 20 dB. The applied absorption coefficients are equal to 0.2, 0.4, 0.6, 0.8, and 1.0. Hence, to cover each possible combination of SNR and absorption coefficient, a total of 25 simulation runs are performed.

The 2D Unitary ESPRIT algorithms always produce two DOA estimates for each frequency bin. However, the CLEAN, SRP-PHAT and 2D-MUSIC algorithms produce at most two DOA estimates for each frequency bin as it depends on the number of local maxima identified in their respective spectra represented by acoustic images, which can be less than two. Consequently, the number of DOA estimates obtained for each signal source are not necessarily equivalent. From this set of DOA estimates obtained from each frequency bin, only two are selected as the final two DOA estimates to represent the DOA estimate for each signal source. It is unknown which of the DOA estimates are associated with the first signal source and which is related to the second signal source. Therefore, the data is clustered into two data sets based on their azimuth and elevation angle. From each of these data clusters, a single DOA estimate associated with the largest intensity is selected to obtain the final two DOA estimates. Here, the intensity refers to the largest power estimate obtained in the algorithm's spectrum as explained in Chapter 2 or in the case of 2D Unitary ESPRIT, it refers to the value of the eigenvalue associated with the eigenvector corresponding to the DOA estimate.

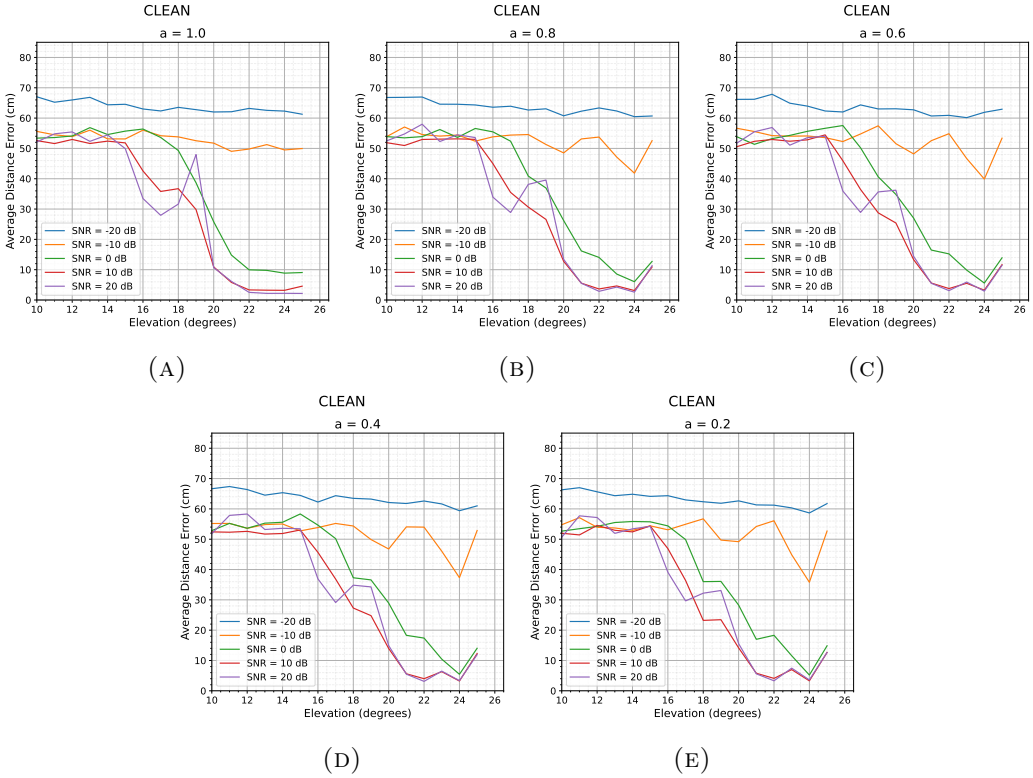
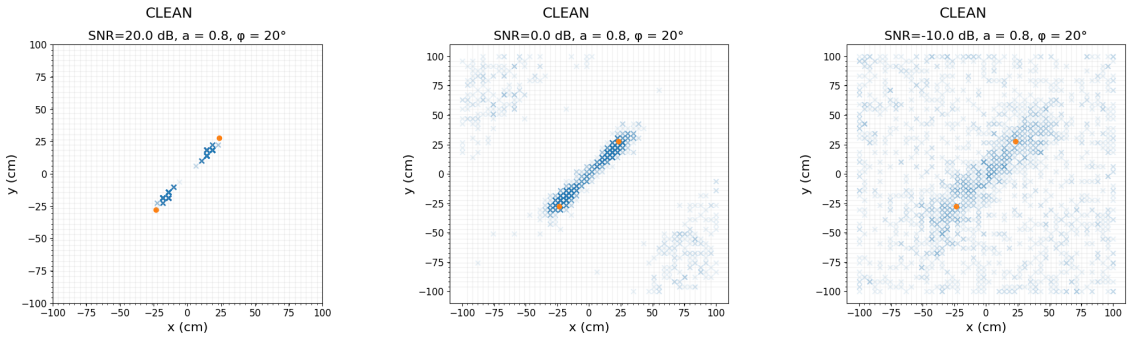


FIGURE 4.15: Average distance error for the CLEAN algorithm for two multiple coherent sources.

CLEAN

The results for the CLEAN algorithm indicate that the algorithm achieves the lowest average distance errors for scenarios where the source signals are stronger than the noise. In the scenarios where the noise is as strong as, or stronger than the signal sources, the average distance error of CLEAN is larger than the previously discussed scenarios. Figure 4.16 illustrates how decreasing the SNR influences the estimated positions for the signal source positions. For each plot in the figure, the estimated positions and the actual signal source positions from 1000 simulations are plotted for an absorption coefficient of 0.8 and elevation 20° while the SNR varies per plot. In Figure 4.16a the utilized SNR is $30dB$. It can be observed that the CLEAN algorithm is able to estimate the position of both signal sources, however, there is some variation in the estimations due to noise. The stronger the noise, the more variation there is in the estimated position. It can also be seen that sometimes the algorithm is unable to identify one of the two signal source, instead choosing a position that is associated with noise. As long as the signals are stronger than the noise, the local maxima in the acoustic image will always correspond with the source signals. However, when noise is as strong as the signal sources, the estimated positions are sometimes associated with noise. When noise is stronger than the signals, the estimated positions are more often associated with noise. In the iterative peak process explained in Section 2.5 it was shown that the CLEAN algorithm will use the largest peaks detected in the acoustic image to construct the clean map, which is then used to find the local maxima. From these local maxima, the P largest local maxima are selected. Hence, if noise is stronger than the signals, the largest local maxima will be associated with noise and consequently the estimated positions will be associated with noise.



(A) (B) (C)

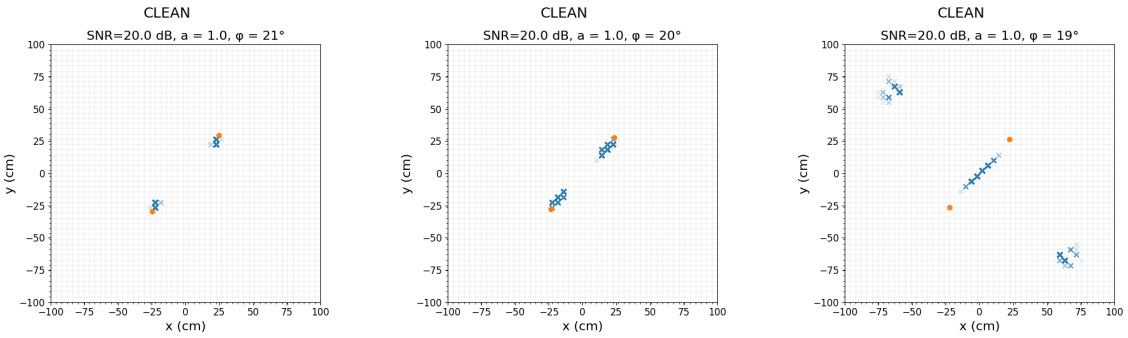
FIGURE 4.16: 2000 estimated and actual source positions of the CLEAN algorithm for two coherent signal sources for varying SNRs.

Furthermore, as the signal sources are coherent, they are more likely to exert influence on each other as their influence on each other will be most prominent when both sources are present in the same acoustic image. However, when the signal sources are placed further apart, their influence on each other will decrease. This is also illustrated in Figure 4.17, where each of the plots contain the estimated positions of 1000 simulations where the utilized absorption coefficient was 1.0 and the SNR is 20 dB. For each plot, a different elevation angle was used for the signal sources, placing them increasingly closer together. Figure 4.17a shows that at an elevation of 21° there is little to no influence of the signal sources on each other. At an elevation of 20° the estimated positions tend towards the center of the scan grid as illustrated by Figure 4.17b. At an elevation of 19°, as shown in Figure 4.17c, the estimated position of the first signal source will be in between both signal sources and will be detected as a single source after applying the maximum filter. Hence, the second estimated signal source position will therefore be associated with noise or a side lobe of the radiation pattern obtained through CB. At a certain elevation, the signal sources are placed so close together that they cannot be properly distinguished. This results in a significant peak in the average distance error. The elevation at which this occurs will be referred to as the algorithm's *breaking point*. For the CLEAN algorithm, the breaking point occurs at an elevation of 19°.

Figure 4.18 shows how reverberation can affect the estimated positions produced by the algorithm. For the three sub-figures the SNR and elevation is kept constant. Decreasing the absorption coefficient, causes more variation in the estimated positions which in turn causes the average distance error to increase. However, the effect that the reverberation has on the estimated positions seems to be largely dependent on the position of the signal sources as illustrated by the plots in Figure 4.15.

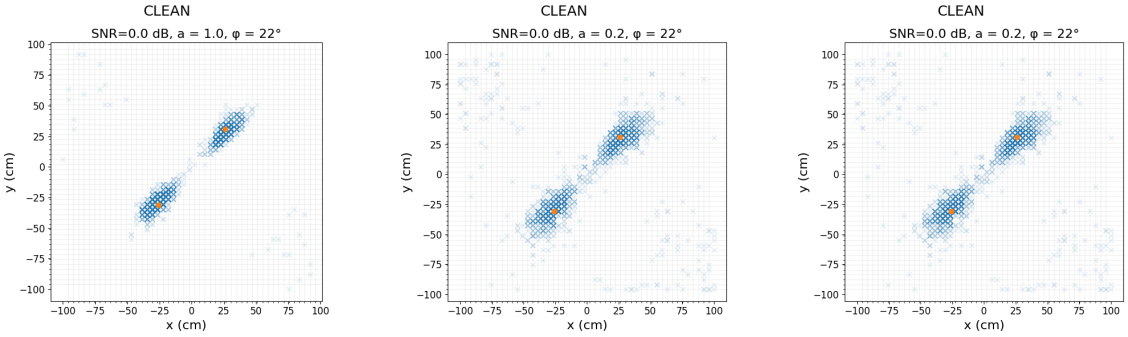
SRP-PHAT

When observing the simulation results for the SRP-PHAT algorithm, it can be determined that the algorithm is robust against noise as the average distance errors for different SNRs do not vary greatly. This is also illustrated by Figure 4.20, where the estimated positions remain very close to the actual source positions. At -20 dB the algorithm starts to occasionally associate its estimated positions with noise. The SRP-PHAT algorithm has a low average distance error when the signal sources are placed further apart than 19° in their elevation and an absorption coefficient of 1.0. After this point, the error significantly increases as the algorithm is not able to distinguish the two signal sources as its estimates



(A) (B) (C)

FIGURE 4.17: 2000 estimated and actual source positions of the CLEAN algorithm for two coherent signal sources for varying elevation angles.



(A) (B) (C)

FIGURE 4.18: 2000 estimated and actual source positions of the CLEAN algorithm for two coherent signal sources for varying absorption coefficients.

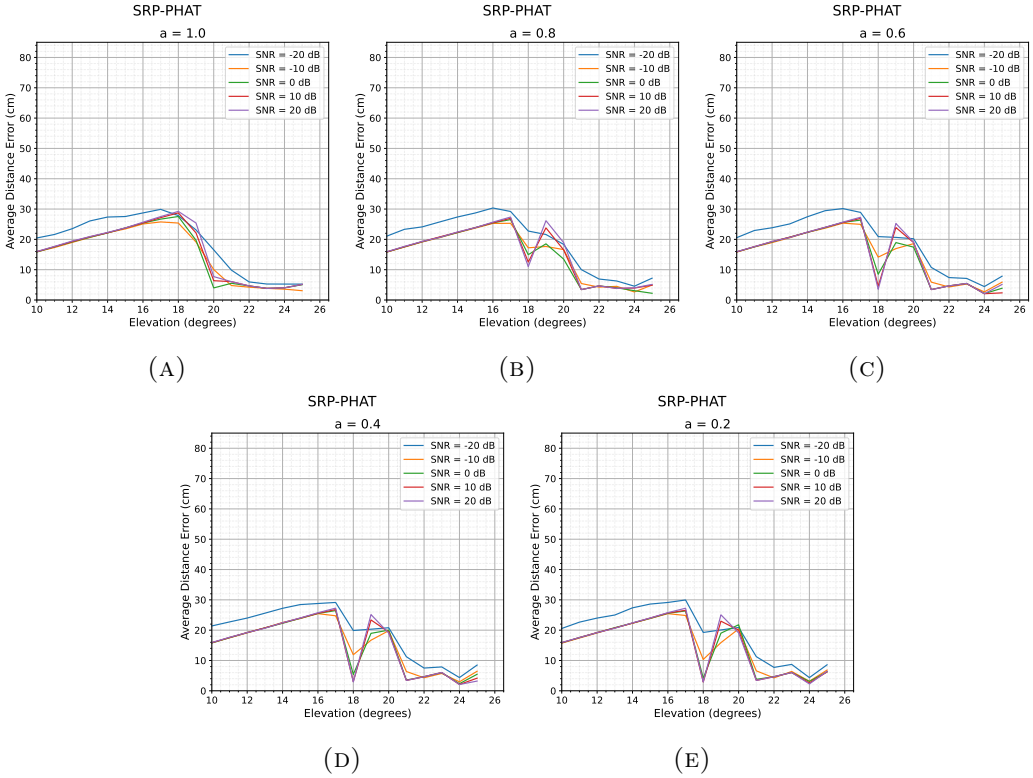
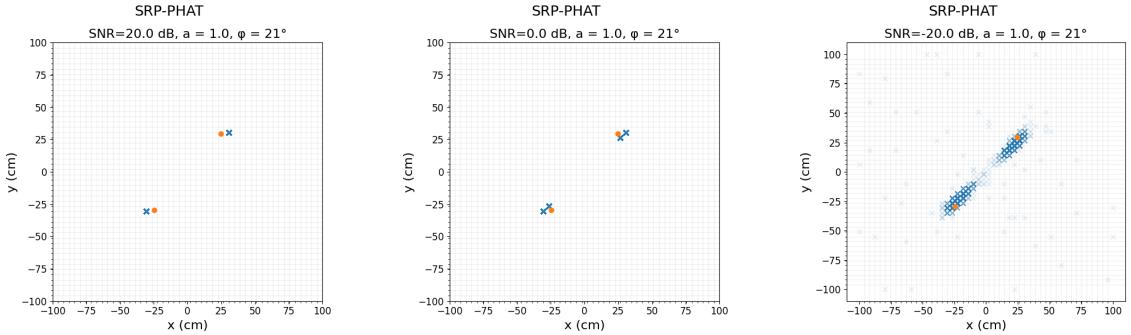


FIGURE 4.19: Average distance error for the SRP-PHAT algorithm for two multiple coherent sources.

tend towards the center point between the signal sources as is illustrated in Figure 4.21. However, instead of one estimated source position being associated with the center between both signal sources and the other with noise, both estimated positions are associated with a position between both signal sources. This also explains why for lower elevations, the average distance error decreases, as the actual signal source positions are placed increasingly closer to the estimated source position in the center of the scan grid. The elevation angle at which the algorithm's ability to distinguish multiple sources significantly decreases, or in other words its *breaking point* is determined to be at an elevation of 19°. Besides being robust against noise, the algorithm is also robust against reverberation. The average distance error only increases slightly as the absorption coefficient is increased. However, for an elevation of 18° the algorithm's average distance error decreases. This particular scenario is illustrated in Figure 4.22. It is assumed that the reverberation causes a shift in the phases of the signals which just happens to perfectly align when received by the microphone array, which allows the algorithm to accurately determine the locations of the signal sources. However, this is only a hypothesis and should be investigated further.

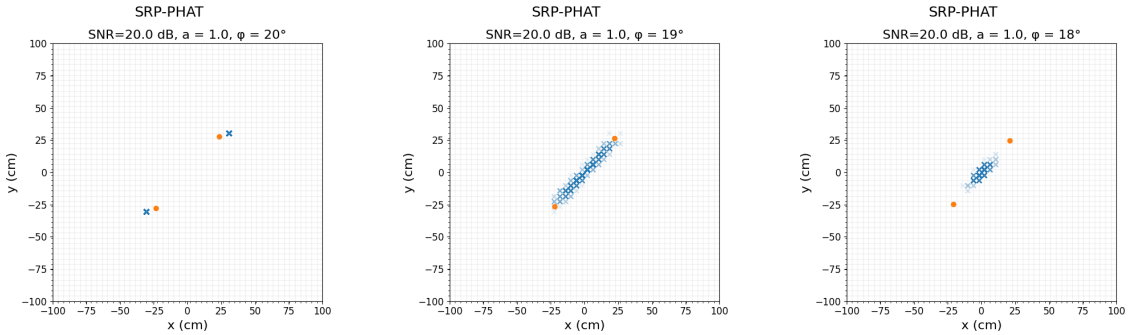
2D-MUSIC

The results for the MUSIC algorithm in Figure 4.23 show that the algorithm is robust against noise as its average distance error across every applied SNR remains constant. In Figure 4.24 the estimated positions are plotted for different SNRs which show how the estimated positions remain close to the actual signal source positions, even for a low SNR of -20 dB. However, at such a low SNR it can be seen that some estimated positions are located further away, which means that these estimated positions are likely associated



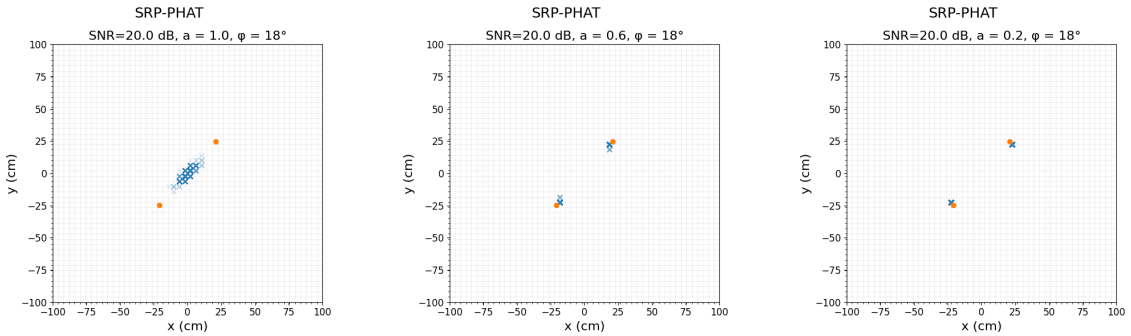
(A) (B) (C)

FIGURE 4.20: 2000 estimated and actual source positions of the SRP-PHAT algorithm for two coherent signal sources for varying SNRs.



(A) (B) (C)

FIGURE 4.21: 2000 estimated and actual source positions of the SRP-PHAT algorithm for two coherent signal sources for varying elevations.



(A) (B) (C)

FIGURE 4.22: 2000 estimated and actual source positions of the SRP-PHAT algorithm for two coherent signal sources for varying absorption coefficients.

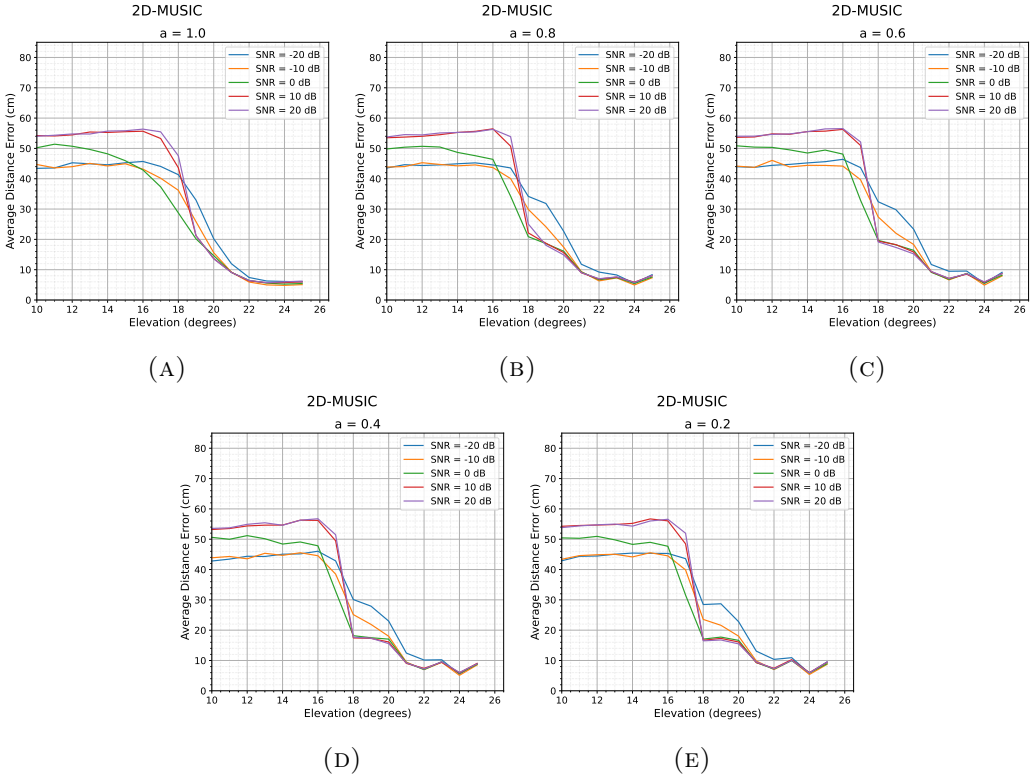
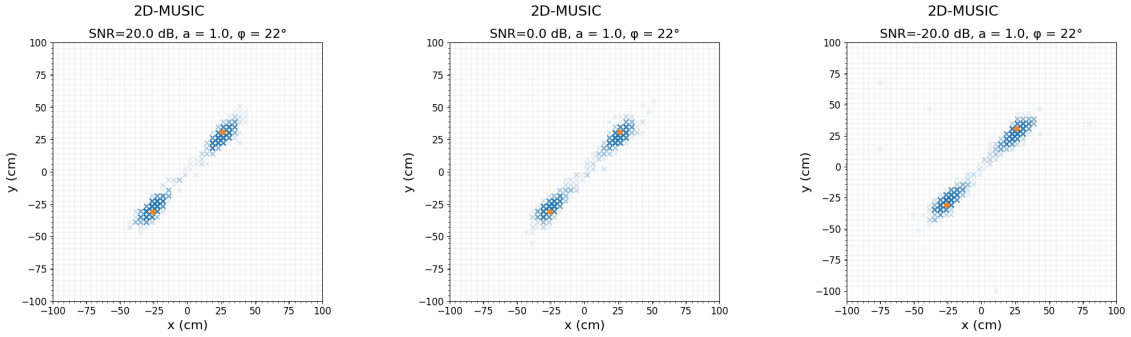


FIGURE 4.23: Average distance error for the MUSIC algorithm for two multiple coherent sources.

with noise. However, for both high and low SNRs, the algorithm always has some variation in its estimated positions which causes the estimated positions to form groups instead of concentrated points. The plots in Figure 4.23 also suggest that the algorithm has a *breaking point*. In Figure 4.25 the estimated positions are plotted for different elevations while no reverberation and an SNR of 20 dB is applied. At an elevation of 21° the estimated positions form two groups, each located around one of the actual source positions. However, some of the estimated position already start to tend towards the center of both source positions. At an elevation of 20° even more estimated position tend towards the center between both signal source positions. Finally, at an elevation of 19°, almost all estimated positions are located between the actual source positions and some start to be associated with random noise. Hence, the *breaking point* of the MUSIC algorithm occurs at an elevation of 19°. When it comes to reverberation, the MUSIC algorithm is also robust. Although decreasing the absorption coefficient causes some increase in the average distance error, this seems to be largely dependent on the position of the signal sources. Nevertheless, the algorithm is not as robust against reverberation as it is to noise.

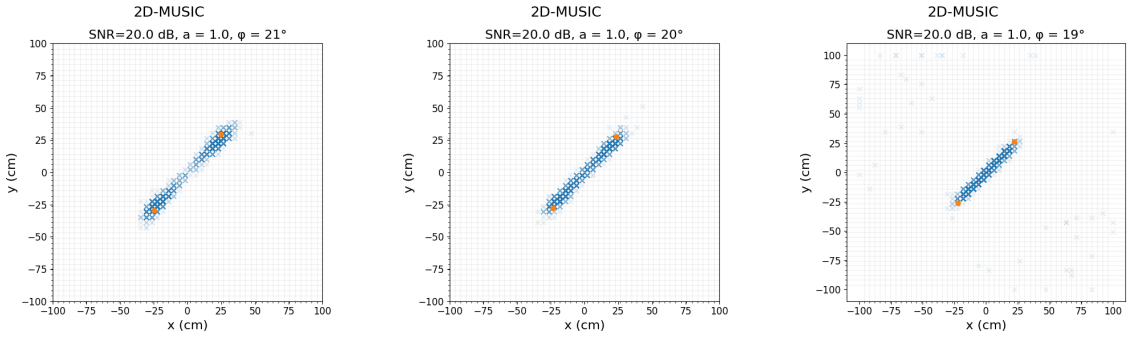
2D Unitary ESPRIT

The average distance errors of the 2D Unitary ESPRIT algorithm are presented in Figure 4.27. The algorithm was observed to always estimate the position of the signal sources to be in the center of the actual signal source positions. No matter the applied noise or absorption coefficient, the results always stay the same. As a result, the average distance error decreases as the signal sources are placed increasingly closer towards the center by decreasing the elevation.



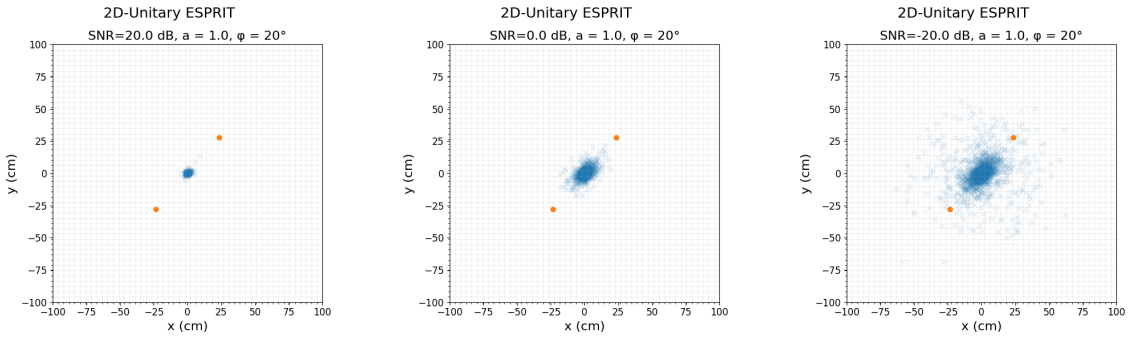
(A) (B) (C)

FIGURE 4.24: 2000 estimated and actual source positions of the 2D-MUSIC algorithm for two coherent signal sources for varying SNRs.



(A) (B) (C)

FIGURE 4.25: 2000 estimated and actual source positions of the 2D-MUSIC algorithm for two coherent signal sources for varying elevations.



(A) (B) (C)

FIGURE 4.26: 2000 estimated and actual source positions of the 2D Unitary ESPRIT algorithm for two coherent signal sources for varying SNRs.

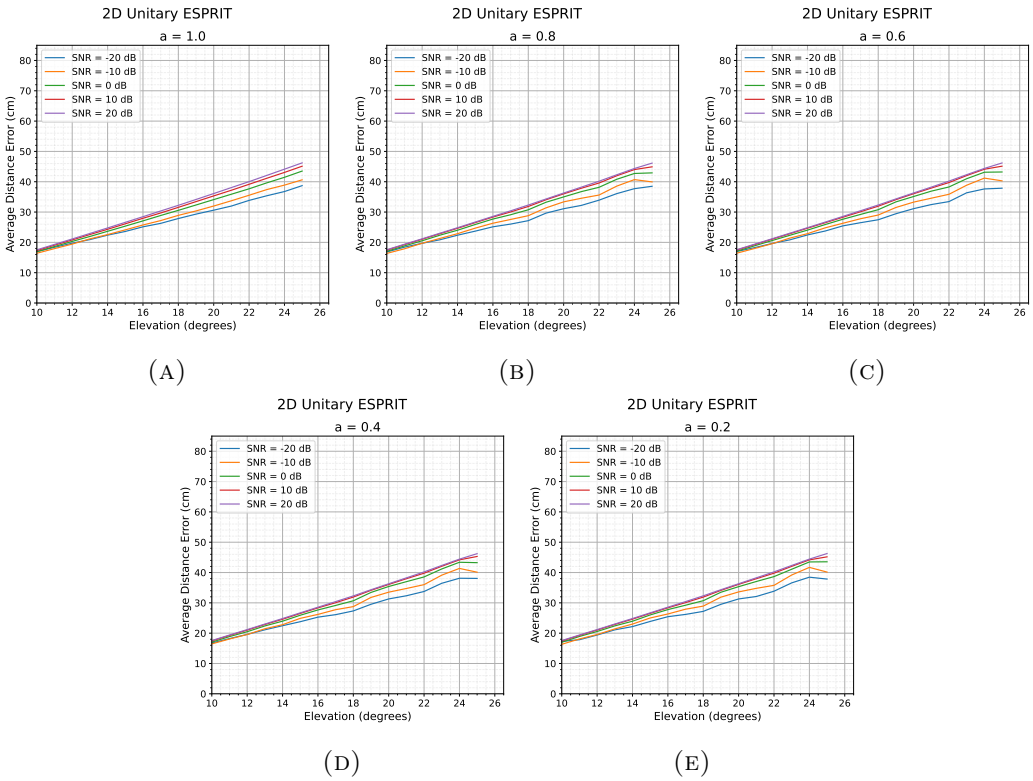


FIGURE 4.27: Average distance error for the ESPRIT algorithm for two multiple coherent sources.

The average distance error also decreases as more noise is introduced. The effect of noise on the algorithm is illustrated in Figure 4.26. As noise increases, so does the variation in the estimated position, causing the estimated position to spread out over a larger area. Consequently, the average distance error increases as more noise is introduced. Overall, the algorithm is incapable of distinguishing the two coherent sources. Therefore, it will not be considered in the next chapter where each SSL algorithm is compared to each other.

Chapter 5

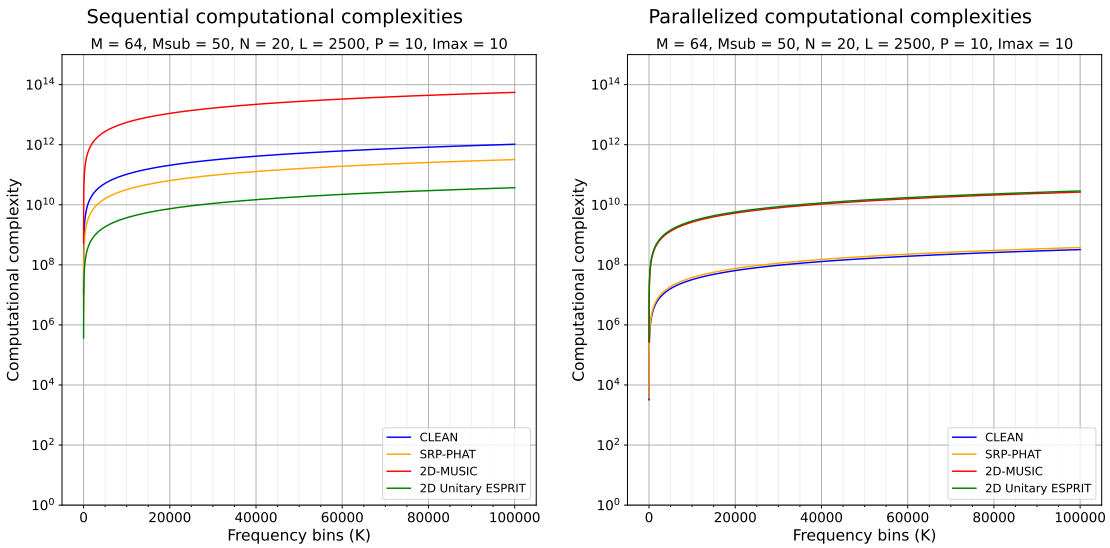
Discussion and Evaluation

In Chapter 3 we explored the complexity of each algorithm. In Chapter 4 we presented the results of various simulations that were run to determine the accuracy of each algorithm in regard to DOA estimation. In this chapter the results obtained in both of these chapters will be combined to provide a comparative analysis of the algorithms, considering both their accuracy and complexities.

5.1 Computational Complexities

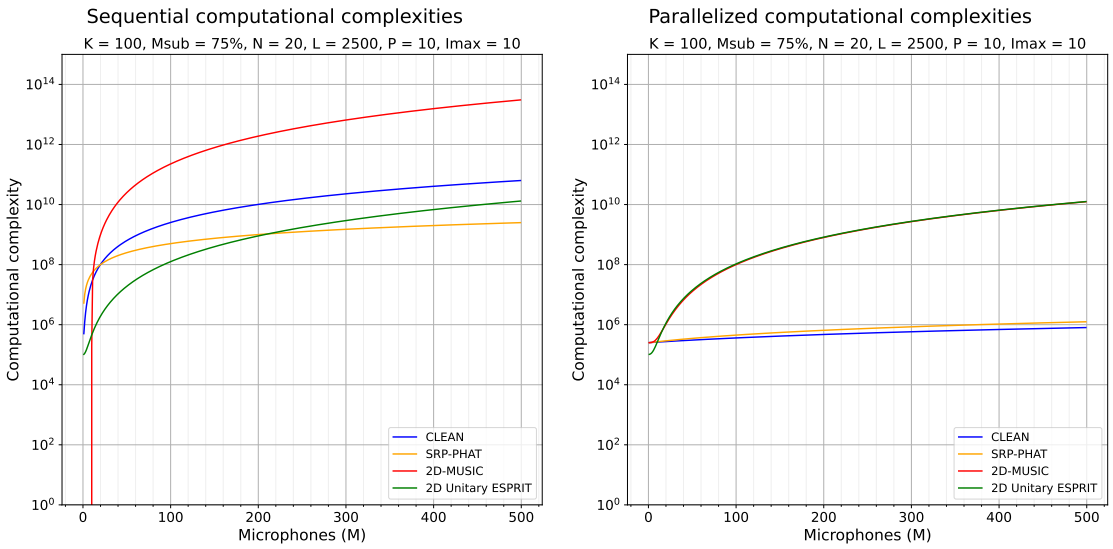
Given the many variables influencing the obtained computational complexities it would be impractical to show how each algorithm's computational complexities scales by modifying each variable. Moreover, some variables such as the number of microphones or snapshots are expected to remain constant when the algorithm is implemented. Therefore, attention is directed towards a single variable that is expected to be adjustable within an application of the SSL algorithms. This variable is the number of frequency bins, denoted as K throughout this paper. This decision is also based on the design of the Fluke ii910 Sonic Industrial Imager. The device features 64 microphones arranged in a sunflower pattern and can detect sound signals within a frequency range of 2kHz to 100kHz. The device also supports real-time selection of any sub-range of frequencies within this spectrum. Logically, a larger scanning sub-range will require the evaluation of more frequency bins compared to a smaller sub-range. In Figure 5.1 the computational complexity of each algorithm versus the number of frequency bins that need to be processed is plotted. In Figure 5.2 the computational complexities are plotted against the number of microphones. To make sure that the the number of microphones in the subarrays used by the 2D Unitary ESPRIT algorithm scale realistically with the total number of microphones available, the subarray will contain 75% of the total microphones available in the microphone array.

The figures confirm that each algorithm benefits from a parallelized approach, resulting in a significantly lower compared to their sequential computational complexities. The only exception being 2D Unitary ESPRIT, which only slightly improves from its sequential complexity. Although 2D Unitary ESPRIT has the lowest sequential computational complexity out of all the algorithms, its parallelized computational complexity is the highest of all the algorithms. It is slightly outperformed by 2D-MUSIC for the parallelized approach. The sequential approach of 2D-MUSIC reflects the high complexity of the algorithm with its computational complexity being the largest of all the algorithms. The large difference between 2D-MUSIC's sequential and parallelized computational complexity is due to the utilization of the CSM and pseudospectrum. These constructs have shown to be particularly suitable for a parallelized approach, greatly reducing the computational



(A) (B)

FIGURE 5.1: Computational complexities versus the number of frequency bins.



(A) (B)

FIGURE 5.2: Computational complexities versus the number of microphones.

complexity and in the case of the pseudospectrum, largely eliminating the dependence of the parallelized computational complexity on the resolution of the scan grid. Consequently, the parallelized complexity is largely determined by the number of microphones in the array due to the complexity of eigendecomposition. This also explains why the 2D-MUSIC and 2D Unitary ESPRIT algorithm have similar parallelized computational complexities. Both depend mostly on the number of microphones for their parallelized complexities due to eigendecomposition and 2D Unitary ESPRIT also does not depend on the resolution of the scan grid. Unlike 2D-MUSIC, however, it also does not depend on the scan grid resolution for its sequential complexity, explaining why its complexity is lower in that case.

The SRP-PHAT algorithm has a higher complexity than the CLEAN algorithm for the given parameters. Both algorithms perform better than 2D-MUSIC for the sequential implementation and better than 2D Unitary ESPRIT. Using the parallelized implementation both CLEAN and SRP-PHAT have a lower computational complexity than 2D-MUSIC and 2D Unitary ESPRIT. The SRP-PHAT algorithm has a lower sequential computational complexity than CLEAN. However, the inverse is true for their parallelized computational complexities. The equations for the computational complexities of CLEAN and SRP-PHAT are presented in Tables 3.2 and 3.3 respectively. From these equations, it is clear that the CLEAN algorithm is dependent on the maximum number of iterations for peak removal. What is not shown however, is that more iterations are required if more peaks are found. This causes CLEAN to require more time when the SNR is low, as more peaks associated with noise will be present in the *dirty map*. However, SRP-PHAT is not dependent on these parameters. Instead, it is largely influenced by the number of snapshots taken and the number of microphones. For the parameters utilized in Figure 5.1 this results in the CLEAN algorithm achieving a lower parallelized complexity than SRP-PHAT. However, for different parameters it is possible for SRP-PHAT to achieve a lower complexity, especially when the number of iterations allowed for iterative peak removal is high. The formulas presented in the tables presented in Chapter 3 can be used to determine the algorithm with the lowest computational complexity for any set of parameters considered in this paper.

5.2 Performance

Single Source Localization

In the previous chapter the simulation results for single sources and two coherent sources were discussed. In this section the results obtained for each algorithm will be presented in such a way that they can easily be compared to one another. First the algorithms will be evaluated with respect to each other for the single source simulation results. In Figure 5.3 the average distance error is plotted for each algorithm. Each plot shows the average distance error and how it evolves with the applied SNR and a single absorption coefficient. The values that are presented directly correspond with the bar charts presented in Figures 4.3 - 4.6. Similar plots for the standard deviation are given in Figure 5.4. From these comparative graphs it can be seen that the 2D Unitary ESPRIT algorithm has the highest average distance error and standard deviation out of all the algorithms. Only in a couple of scenarios where the SNR is high does the 2D-MUSIC algorithm perform worse. In figure 4.13 it can be seen that the poor performance of the ESPRIT algorithm is due to its estimated locations forming two groups for each of the utilized source locations with the larger group of estimations being closer towards the scan grid's origin. The further away the source location is from the scan grid's origin, the larger the distance between

the groups of estimations. Consequently, average distance error and standard deviation is mostly influenced by source locations that are placed further away from the origin. This also suggests that the effective range of the 2D Unitary ESPRIT algorithm seems to be smaller than that of the other algorithms. Whether the existence of two groups of estimations for each source location is due to a behavior of the algorithm, a result of a faulty implementation, or another unknown reason has not been determined and should be investigated in future research.

The 2D-MUSIC algorithm exhibits a larger average distance error and standard deviation than the CLEAN and SRP-PHAT algorithms. This is particularly obvious for scenarios where the strength of the noise is less than the strength of the source signal. As previously mentioned, it sometimes has a lower performance than the 2D Unitary ESPRIT algorithm. To be more precise, the average distance error is larger than that of 2D Unitary ESPRIT at an SNR of 30dB for each absorption coefficient. However, in the absence of noise it does perform better than 2D Unitary ESPRIT with the only exception occurring when there is also an absence of reverberation. It can also be seen that the difference in the performance of the 2D MUSIC algorithm is accentuated by the absorption coefficient. As the absorption coefficient decreases, its performance decreases. The effect on the algorithm's performance is particularly detrimental when reverberation is present, and the signal is stronger than the noise. However, as noise increases, the average distance error and the standard deviation decreases. Eventually, when the noise is stronger than the source signals, the algorithm achieves comparable performance to the CLEAN and SRP-PHAT algorithms. For an absorption coefficient of 0.2, it actually achieves slightly better performance than CLEAN at an SNR of 0dB and 10dB. The difference between the algorithm's performance for different SNRs is likely due to the algorithm's dependence on the noise subspace to construct its pseudospectrum. The noise subspace is determined by the eigenvectors associated with noise obtained after eigendecomposition. Due to this dependence, when there is little noise present, the estimated positions obtained from the pseudospectrum will be less accurate. However, the plots show that the absorption coefficient has a much more detrimental effect on the accuracy and precision of the 2D MUSIC algorithm.

The SRP-PHAT algorithm performs better than the 2D Unitary ESPRIT and MUSIC algorithms. When the noise is equal to or stronger than the noise, the algorithm's performance is equal to CLEAN. However, when noise is stronger the algorithm performs worse than the CLEAN algorithm for these scenarios. However, for the lowest applied absorption coefficient of 0.2 the SRP-PHAT algorithm's performance is better than the CLEAN algorithm as indicated by the lower average distance error and standard deviation.

Lastly, let us discuss the CLEAN algorithm. The CLEAN algorithm obtains the lowest average distance error of all the algorithms considered in this thesis. When there is little noise present, the algorithm actually performs optimally and is able to select the scan grid location that is closest to the source location. However, at a SNR of 10dB and lower the algorithm's performance gradually decreases. Nevertheless, it is only outperformed by the SRP-PHAT algorithm at the minimum applied absorption coefficient of 0.2.

In conclusion, when it comes to locating a single signal source, CLEAN shows the highest accuracy and precision. The only exception occurs at an absorption coefficient of 0.2 where it is outperformed by SRP-PHAT. However, the same cannot be stated when it comes to locating multiple coherent signal sources. Similar to 2D Unitary ESPRIT, the estimated positions were always located at the center, directly between both signal sources. 2D Unitary ESPRIT, however, also performs poorly when subjected to a single signal source with the algorithm obtaining the lowest accuracy and precision. This was

unexpected as the lack of a scan grid would suggest that 2D Unitary ESPRIT would be able to attain a lower average distance error than the other, scan-grid based algorithms.

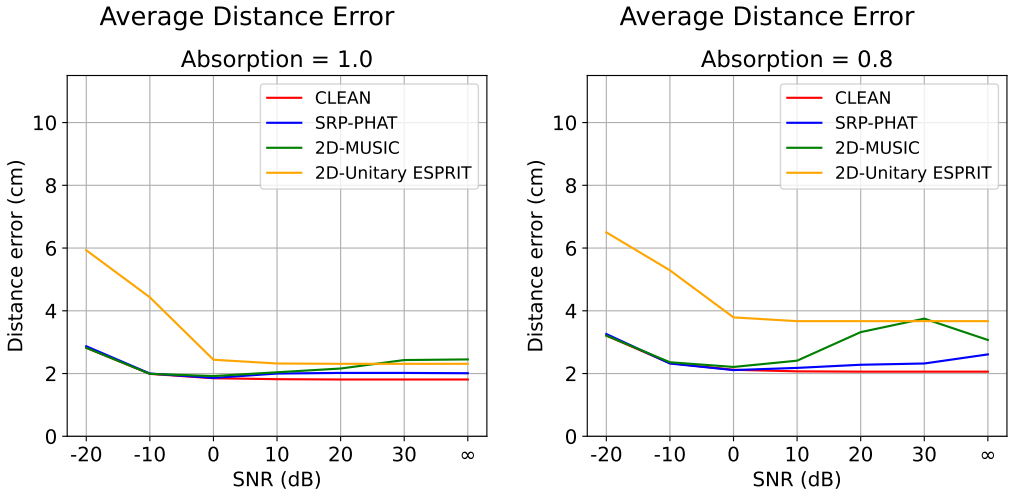
Multiple Coherent Source Localization

Next, the simulation results obtained for each of the SSL algorithms for two coherent signal sources are analyzed to determine how each of the algorithms compare to each other. The results of the 2D Unitary ESPRIT algorithm are not included, because as shown in the previous chapter, this algorithm cannot distinguish between two coherent signal sources within the parameters used in this thesis. In the previous chapter it was determined that the CLEAN, SRP-PHAT, and 2D-MUSIC algorithm were able to distinguish the two coherent sources. Hence, these three algorithms are used to create the comparative graphs in Figure 5.5, which display the average distance errors of each algorithm for multiple SNRs and elevation angles that were applied to the simulations. From the comparative graph in Figure 5.5a it becomes clear that, compared to the other algorithms, CLEAN is severely affected by noise. As noise increases, its average distance error increases quite dramatically compared to the other algorithms. In scenarios where there is little noise, the CLEAN algorithm also has a higher average distance error than the other 2D-MUSIC and SRP-PHAT algorithm when the signal sources are placed at an elevation of 21 degrees or larger. The 2D-MUSIC algorithm achieves a respectable average distance error when the signal sources are sufficiently distanced from each other. However, it is generally outperformed by the SRP-PHAT algorithm. Similar to the SRP-PHAT algorithm, the 2D-MUSIC algorithm's average distance error starts to increase around an elevation angle of 23 degrees. However, for the 2D-MUSIC algorithm this increase is more significant than that for SRP-PHAT and leads to a larger average distance error. The comparative graphs in Figure 5.5 shows that the SRP-PHAT algorithm achieves the lowest average distance error as long as the signal sources are both placed at an elevation of 21 degrees. Between 21 and 13 degrees, the SRP-PHAT algorithm is sometimes outperformed by the CLEAN algorithm in scenarios where the source signals are stronger than the noise. When reverberation is present, SRP-PHAT is sometimes also outperformed by the 2D-MUSIC algorithm for SNRs larger than -20dB. When reverberation is present, each algorithm's performance is afflicted. The CLEAN algorithm exhibits increased variance in its average distance errors between consecutive elevations, especially for low SNRs. This can also be observed for the 2D-MUSIC algorithm, although the variance is less noticeable. The same can be stated for the SRP-PHAT algorithm. However, at an elevation of 18 degrees there is a sudden decrease in the average distance error. The exact reason is not exactly known and should be further investigated.

5.3 Evaluation

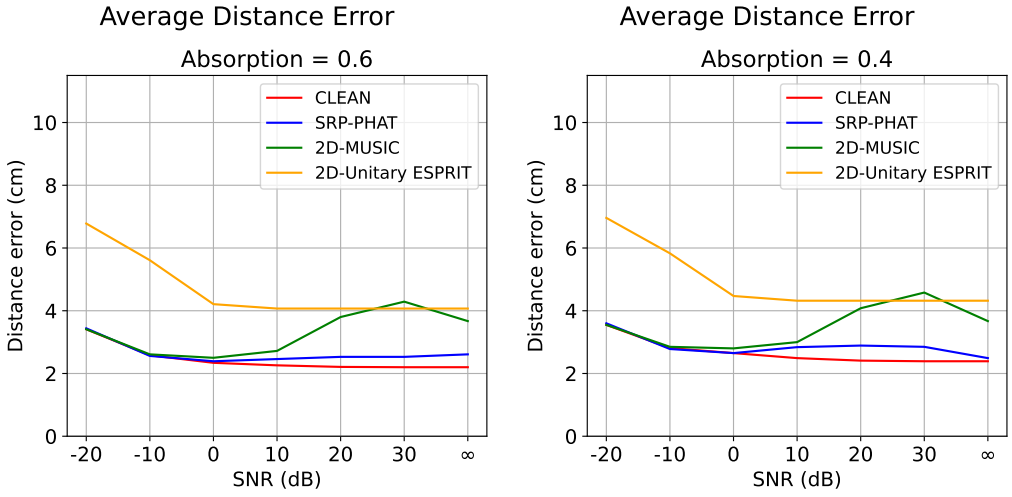
In this section, each algorithms computational complexity and performance regarding single source and multiple coherent source simulations is summarized to provide the final evaluation.

Figure 5.1 shows that the 2D-MUSIC algorithm has a high sequential computational complexity. This is due to its dependence on eigendecomposition and its pseudospectrum, which are both computationally intensive operations. The 2D Unitary ESPRIT algorithm also depends on eigendecomposition but does not require the construction of a spectrum from which to derive its estimates for the signal sources. As a consequence, it has the lowest sequential computational complexity of all SSL algorithms presented in this the-



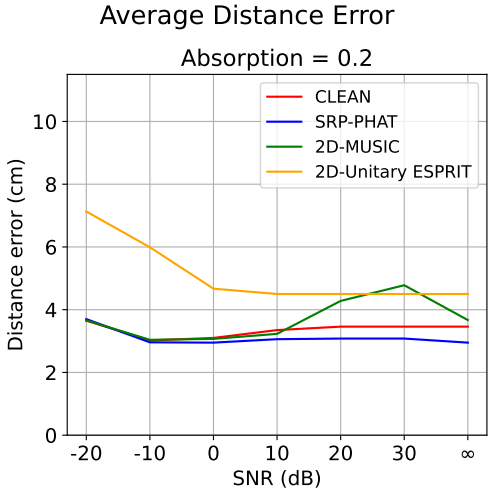
(A)

(B)



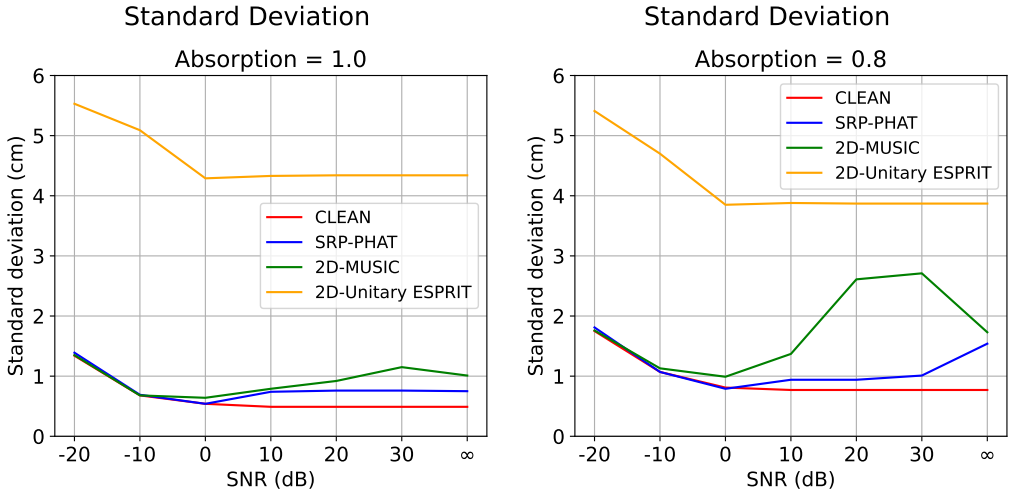
(C)

(D)



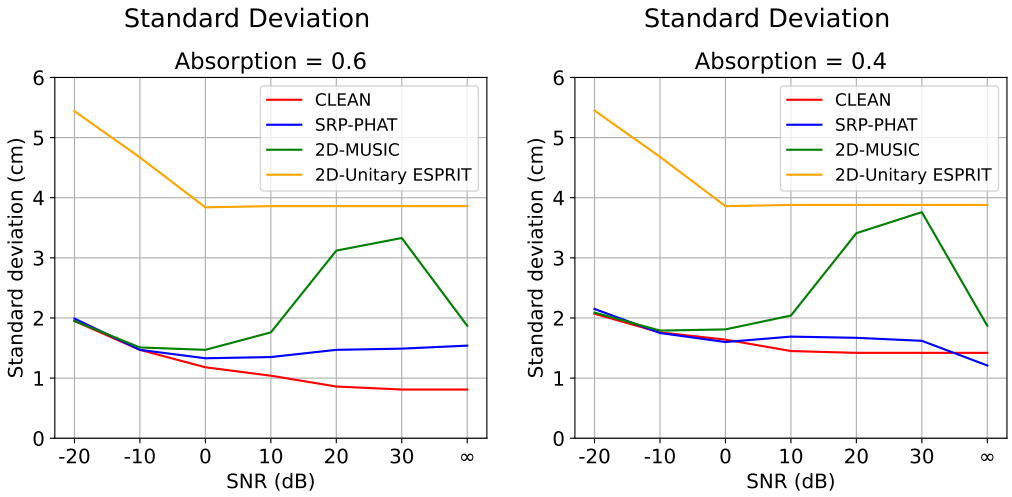
(E)

FIGURE 5.3: Comparison of the average distance error for single-source localization



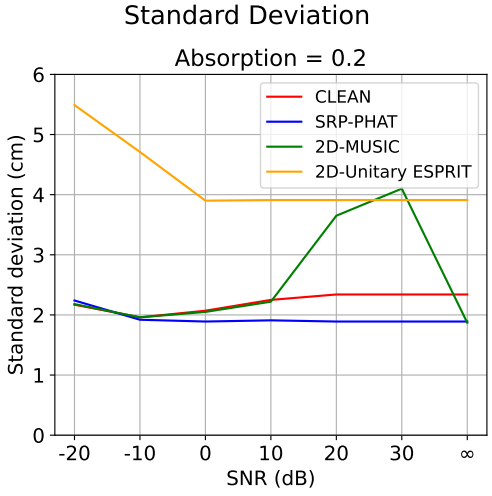
(A)

(B)



(C)

(D)



(E)

FIGURE 5.4: Comparison of the standard deviation for single-source localization

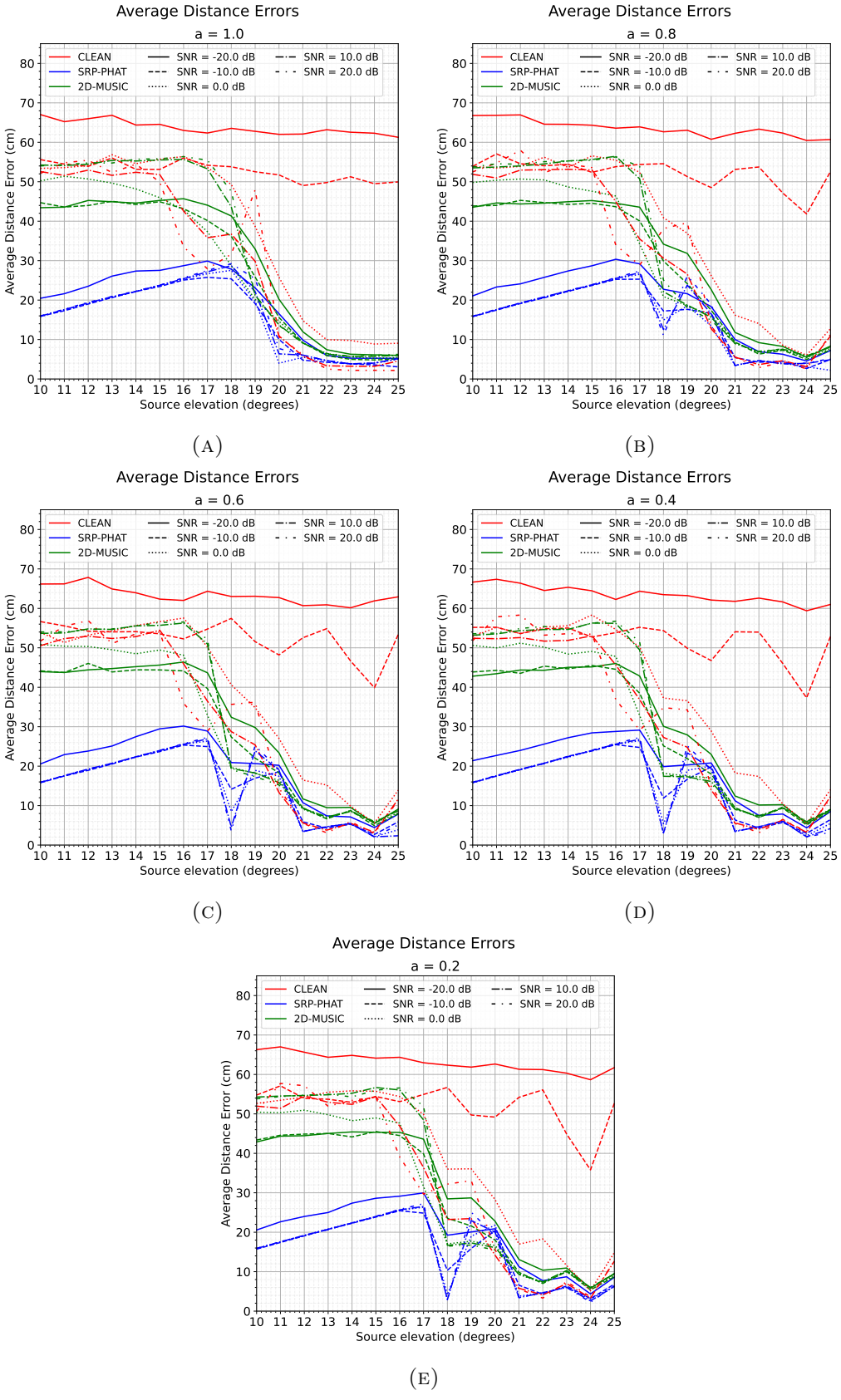


FIGURE 5.5: Comparison of the average distance errors for multiple coherent sources localization

sis. However, the construction of 2D-MUSIC's pseudospectrum can be highly optimized through the parallelization techniques presented in Chapter 3. Eigendecomposition is an operation that is much harder to parallelize and warrants its own research. Eigendecomposition for both these algorithms scales poorly with the number of microphones present in the microphone array. Consequently, using the parameters used in Figure 5.1 causes the 2D Unitary ESPRIT and 2D MUSIC algorithm to have the worst parallelized computational complexity out of all the SSL algorithms presented in this thesis. The CLEAN algorithm has an average sequential computational complexity. However, its operations can be well parallelized, leading to a significantly lower parallelized computational complexity. However, a limiting factor of the CLEAN algorithm is that it can only process each of the signal sources sequentially. Hence, the algorithm's computational performance scales poorly with the number of sources that must be detected. The SRP-PHAT algorithm has a significantly lower sequential and higher parallelized computational complexity than CLEAN for the parameters used in Figures 5.1 and 5.2. When it comes to locating a single signal source, the 2D Unitary ESPRIT had the worst performance out of all algorithms for every scenario that was applied. Only the 2D-MUSIC algorithm performed slightly worse in scenarios where very little noise was present. The 2D-MUSIC algorithm had similar performance to the SRP-PHAT and CLEAN algorithm when noise was stronger than the signal source. However, when the signal source was stronger than the noise, its performance was significantly worse. When reverberation is introduced, this exacerbated its poor performance for low-noise scenarios. CLEAN and SRP-PHAT had very similar performance with CLEAN slightly outperforming the SRP-PHAT algorithm. Only when the absorption coefficient was decreased to its minimum of 0.2 did the SRP-PHAT algorithm outperform the CLEAN algorithm. Lastly, when locating two coherent signal sources at opposite sides in their azimuth location, the 2D Unitary ESPRIT algorithm was unable to identify the locations of the two signal sources. Instead it always selects the location in the middle of the signal sources. The CLEAN, 2D-MUSIC, and SRP-PHAT algorithms were able to distinguish between both signal sources under specific scenarios. As long as the signal sources are stronger that the noise and they are both located at an elevation of at least 20 degrees, the CLEAN algorithm is able to locate both signal sources. The 2D-MUSIC algorithm was also able to distinguish the two signal sources and did it more accurately. Furthermore, it was able to do so even when the noise was stronger than the signal sources. Nevertheless, at an elevation of 22 degrees or lower, the algorithm's performance starts to decrease, eventually only being able to locate its first signal source estimate to be located in between both signal sources, while its other estimate is associated with random noise. The SRP-PHAT algorithm generally achieves the best performance when locating the two coherent signal sources, even in adverse conditions where there is reverberation and noise present. Similar to CLEAN, the algorithm's performance starts to deteriorate around an elevation of 20 degrees or lower. At an elevation of 18 degrees or lower its estimates for the first and second signal sources are located in the middle of both signal sources and the signal sources cannot be effectively distinguished.

Chapter 6

Conclusions & Future Work

In this study four different SSL algorithms have been analyzed to determine their computational complexities, accuracy, and precision. In Chapter 2, each of the algorithms were dissected and each of their components were individually explained.

Chapter 3 explored the computational complexity of each of the algorithm's components identified in Chapter 2. Each component's corresponding equations were rewritten to clearly express its computational complexity using the Big-O notation. Each of these equations were expressed as partial sums such that corresponding PEs could be constructed and arranged in a DDG. These DDGs show how each of the components can be implemented using parallel processing techniques and the resulting reduction in computational complexity was also expressed using the Big-O notation. The computational complexities obtained for each algorithm's components were combined to express the sequential and parallelized narrowband computational complexity for each algorithm. Lastly, these computational complexities were scaled with the variable K, representing the number of frequency-bins, to obtain the sequential and parallelized computational complexities for their wideband implementation.

Chapter 4 presented an in-depth exploration of the simulation results for each algorithm. The first set of simulations considered a single source signal. The second set considered two coherent signal sources placed at opposite azimuth angles and equivalent elevation angles relative to the microphone array's center. These simulations allowed for the determination of each algorithm's accuracy and precision for both a single narrowband source and two coherent narrowband sources. To determine the robustness of each algorithm, different levels of noise and reverberation were applied. Lastly, the two coherent sources were placed increasingly closer to determine the minimum elevation angles at which each algorithm was unable to distinguish the locations of both signal sources. In Chapter 5, these findings were juxtaposed with the previously obtained computational complexities for each algorithm to provide a final evaluation of the results.

Regarding their performance, the single source simulations showed how each algorithm's accuracy and precision were affected by factors such as noise and reverberation. CLEAN and SRP-PHAT showed robust performance across all applied reverberation and noise levels. 2D-MUSIC showed that it depends on noise to maintain satisfactory performance as its performance degrades when little noise is present. Lastly, 2D Unitary ESPRIT had the worst performance, as its estimates tended to be situated in two separate groups. One of these groups coincided with the actual location of the signal source while the other was located closer towards the center of the scan grid. The distance between these groups grew with the distance of the actual source location from the origin of the scan grid, suggesting that the effective range of 2D Unitary ESPRIT is limited compared to the other algorithms.

The results obtained for multiple coherent sources showed that the 2D Unitary ESPRIT algorithm was not able to distinguish the locations of both signal sources regardless of their relative position to each other and the applied noise and reverberation. The algorithm always estimated the position of both signal sources to be in the middle of both signal sources. The other algorithms were able to distinguish both signal sources to different extents. As the 2D Unitary ESPRIT algorithm was not able to distinguish the two coherent sources, it was not included in the comparative graphs presented in Chapter 5. The CLEAN algorithm's performance when subjected to two coherent signal sources has been shown to significantly deteriorate in scenarios where the noise is as strong or stronger than the source signals. When the source signals were stronger than the noise the algorithm was able to distinguish the locations of both signal sources until each of these sources was located at an elevation of 19 degrees, at which point the sources were placed so closely together that the algorithm could not effectively distinguish them. The 2D-MUSIC algorithm was also able to distinguish the two coherent sources, but as the signal sources are placed increasingly closer together, this ability started to rapidly diminish eventually leading to a large average distance error. However, compared to the CLEAN algorithm, its performance is much more consistent than CLEAN across the applied SNRs. Nevertheless, each algorithm was outperformed by the SRP-PHAT algorithm. This algorithm showed the highest accuracy even when subjected to noise and reverberation.

The computational complexities of each algorithm were compared for the same parameters for both their sequential and parallelized computational complexities. Using these parameters, it was shown that 2D Unitary ESPRIT has the lowest sequential computational complexity followed by SRP-PHAT, CLEAN, and finally 2D-MUSIC which had the highest computational complexity. This is largely due to its dependence on both the construction of a spectrum and eigendecomposition. Eigendecomposition is a complex operation with intricate data dependencies which makes it difficult to apply parallel processing techniques to it. In fact, this should be regarded as a separate subject of study and has therefore not been explored in this thesis. The construction of spectrums, however, is an operation that loans itself to be optimized through parallelization. Consequently, the parallelized computational complexity of 2D-MUSIC is almost equal to 2D Unitary ESPRIT, which does not utilize a spectrum for DOA estimation. Other spectrum-based algorithms such as CLEAN, and SRP-PHAT also significantly benefit from parallel processing techniques. In fact, CLEAN's parallelized computational complexity was the lowest out of all algorithms closely followed by SRP-PHAT. 2D-MUSIC and 2D Unitary ESPRIT also had similar parallelized computational complexities, although much greater than CLEAN and SRP-PHAT. Importantly, one must be aware that these results are applicable to the parameters that were used in this thesis, other parameters can yield different results. Nevertheless, it shows that the spectrum-based algorithms all benefit from parallelization.

With these results, the research questions presented in this paper's introduction can be answered as follows.

1. **Which of the selected algorithms achieves the highest accuracy and precision for a single signal source when subjected to noise and reverberation?** The CLEAN and SRP-PHAT algorithm show similar performance with the CLEAN algorithm having the highest accuracy and precision. he only exception occurs when the absorption coefficient is equal to 0.2, at which point SRP-PHAT achieves higher accuracy.

2. **What are the sequential and parallelized computational complexities of each algorithm?** The computational complexities of each algorithm are highly de-

pendent on the values chosen for the many parameters of which the complexities consist of. However, the 2D Unitary ESPRIT has a low sequential complexity due to its independence on a scan grid unlike the other algorithms. However, both 2D Unitary ESPRIT and 2D-MUSIC have higher parallelized computational complexities due to their dependence on eigendecomposition, which is difficult to parallelize. SRP-PHAT and CLEAN have a similar sequential and parallelized computational complexity. However, the CLEAN algorithm is dependent on the number of signal sources that must be found. Due to this algorithm having to process each signal source sequentially, it cannot be parallelized to the same extent that the SRP-PHAT algorithm can.

3. **Which of the algorithms is able to estimate the direction of multiple coherent sources?** The 2D Unitary ESPRIT is unable to distinguish between two coherent signal sources and always selects the position in between both signal sources. The CLEAN algorithm can locate both signal sources, but only when little to no noise is present. The 2D-MUSIC algorithm can also locate both signal sources and do so more accurately than CLEAN, but when the signal sources are placed too close together, the algorithm's performance quickly deteriorates as its first estimate is located between both signal sources, while the second one is associated with noise. The SRP-PHAT algorithm achieved the highest accuracy of all SSL algorithms considered in this thesis. The algorithms starts to break down at the same point as the 2D-MUSIC algorithm, however, not as dramatically. Eventually, both of its estimations are associated with a position in between both signal sources.

6.1 Future Work

In this section suggestions for future research are presented stemming from the observations obtained in this thesis. Although many suggestions can be given, for the sake of conciseness only three have been selected which are considered the most promising by the author of this thesis.

Frequency estimation

One benefit of wideband SSL algorithms that was not explored in this paper is their ability to estimate the frequencies of source signals. This ability was observed during the simulations as the strongest peaks were detected in the frequency bins that are closest to the frequency of the utilized signal sources. However, in this study each of the acoustic signals considered had equivalent frequencies. Evaluating the ability of each algorithm to estimate the frequency of the utilized source signals will require performing additional simulations with signal sources covering a variety of different frequencies. Additionally, one should evaluate possible influences that can tamper with the algorithm's ability to estimate the frequency of the signal. This can include the applied SNR, reverberation, the size and shape of the room, and whether there are multiple coherent or incoherent signal sources present in the environment.

Multiple incoherent sources

Another benefit of wideband SSL algorithms is their ability to distinguish multiple sources. In this paper only two coherent sources were considered. However, when dealing with incoherent sources, these sources might influence each other to a lesser extent as they will

be more prominent in their respective frequency bins as was shown in Figure 2.3. The minimum separation in frequency required for two incoherent sources to be able to be identified as two separate sources can be an interesting subject for future research. It would involve identifying what determines this minimum required distance in frequencies. This can be due to the preprocessing of the signals through STFT, the properties of the microphone array, or the utilized SSL algorithm.

Complex Environments

The simulations conducted within this algorithm focused on two key parameters: the SNR and the absorption coefficient. Additionally, all simulations were performed within the same room. However, *pyroomacoustics* offers many additional options for evaluating SSL algorithms. Users can customize the size and shape of the room, as well as apply different absorption coefficients to each of its surfaces. Furthermore, *pyroomacoustics* supports an experimental hybrid simulator that combines the Image Source Method (ISM) [4] and ray tracing (RT) [78, 71], enabling the modeling of additional parameters such as scattering. While simulations provide only an abstraction of real-life scenarios, these additional options bring us even closer to replicating real-world conditions. Therefore, future research can consider utilizing these features to obtain more realistic results than presented in this paper.

Bibliography

[1] 1 - introductory material. In G. Meurant, editor, *Computer Solution of Large Linear Systems*, volume 28 of *Studies in Mathematics and Its Applications*, pages 1–68. Elsevier, 1999. URL: https://www.sciencedirect.com/science/article/pii/S0168202499800022, doi:https://doi.org/10.1016/S0168-2024(99)80002-2.

[2] Muhammad Afridon and Wan M Faizal. Determining human voice source location using tdoa. In *2018 International Conference on Applied Science and Technology (iCAST)*, pages 57–60, 2018. doi:10.1109/iCAST1.2018.8751513.

[3] A. Ahmedsaid, A. Amira, and A. Bouridane. Efficient systolic array for singular value and eigenvalue decomposition. In *2003 46th Midwest Symposium on Circuits and Systems*, volume 2, pages 835–838 Vol. 2, 2003. doi:10.1109/MWSCAS.2003.1562416.

[4] Jont Allen and David Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65:943–950, 04 1979. doi:10.1121/1.382599.

[5] Thomas F. Brooks and William M. Humphreys. A deconvolution approach for the mapping of acoustic sources (damas) determined from phased microphone arrays. *Journal of Sound and Vibration*, 294(4):856–879, 2006. URL: https://www.sciencedirect.com/science/article/pii/S0022460X06000289, doi:https://doi.org/10.1016/j.jsv.2005.12.046.

[6] Kainan Chen, Jürgen Geiger, Wenyu Jin, Mohammad Taghizadeh, and Walter Kellermann. Robust phase replication method for spatial aliasing problem in multiple sound sources localization. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 195–199, 2017. doi:10.1109/WASPAA.2017.8170022.

[7] Long Chen, Weize Sun, Lei Huang, and Liang Yu. Broadband sound source localisation via non-synchronous measurements for service robots: A tensor completion approach. *IEEE Robotics and Automation Letters*, 7(4):12193–12200, 2022. doi:10.1109/LRA.2022.3212665.

[8] Yang Chong, Zhuang Rui, Zhang Guoguang, Wang Wenyun, Jiang Chunhua, Jiao Yucheng, and Li Guotao. Structural design and finite element analysis of underwater gas leakage intelligent monitoring array. In *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pages 618–624, 2022. doi:10.1109/EEBDA53927.2022.9744985.

[9] Ning Chu, Yue Ning, Liang Yu, Qin Liu, Qian Huang, Dazhuan Wu, and Peng Hou. Acoustic source localization in a reverberant environment based on sound

field morphological component analysis and alternating direction method of multipliers. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021. doi:10.1109/TIM.2021.3077670.

[10] Zhigang Chu and Yang Yang. Comparison of deconvolution methods for the visualization of acoustic sources based on cross-spectral imaging function beamforming. *Mechanical Systems and Signal Processing*, 48(1):404–422, 2014. URL: https://www.sciencedirect.com/science/article/pii/S0888327014000995, doi:https://doi.org/10.1016/j.ymssp.2014.03.012.

[11] Henry Y. H. Chuang, Ling Chen, and Dayuan Qian. A size-independent systolic array for matrix triangularization and eigenvalue computation. *Circuits, Systems and Signal Processing*, 7(2):173–189, Jun 1988. doi:10.1007/BF01602096.

[12] Maximo Cobos, Amparo Marti, and Jose J. Lopez. A modified srp-phat functional for robust real-time sound source localization with scalable spatial sampling. *IEEE Signal Processing Letters*, 18(1):71–74, 2011. doi:10.1109/LSP.2010.2091502.

[13] Cornwell, T. J. Hogbom's clean algorithm. impact on astronomy and beyond - commentary on: Högbom j. a., 1974, a&as, 15, 417. *A&A*, 500(1):65–66, 2009. doi:10.1051/0004-6361/200912148.

[14] Joseph Hector Dibiase. *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. PhD thesis, Brown University, Rhode Island, August 2000.

[15] Joseph Hector DiBiase. *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Brown University, 2000.

[16] Xiaofeng Ding, Chengliang Wang, Heping Liu, Zhihai Zhang, Xianzhang Chen, Yujuan Tan, Duo Liu, and Ao Ren. Frl: Fast and reconfigurable accelerator for distributed sound source localization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):3922–3933, 2022. doi:10.1109/TCAD.2022.3197537.

[17] Jacek Dmochowski, Jacob Benesty, and SofiÈne Affes. On spatial aliasing in microphone arrays. *IEEE Transactions on Signal Processing*, 57(4):1383–1395, 2009. doi:10.1109/TSP.2008.2010596.

[18] Klaus Ehrenfried and Lars Koop. Comparison of iterative deconvolution algorithms for the mapping of acoustic sources. *AIAA Journal*, 45(7):1584–1595, 2007. arXiv:https://doi.org/10.2514/1.26320, doi:10.2514/1.26320.

[19] D.J Evans and K Margaritis. Systolic designs for eigenvalue-eigenvector computations using matrix powers. *Parallel Computing*, 14(1):77–87, 1990. URL: https://www.sciencedirect.com/science/article/pii/016781919090097S, doi:https://doi.org/10.1016/0167-8191(90)90097-S.

[20] Maurice F. Fallon and Simon Godsill. Acoustic source localization and tracking using track before detect. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1228–1242, 2010. doi:10.1109/TASL.2009.2031781.

[21] Mojtaba Farmani, Michael Syskind Pedersen, Zheng-Hua Tan, and Jesper Jensen. Maximum likelihood approach to "informed" sound source localization for hearing aid applications. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20, 2015. doi:10.1109/ICASSP.2015.7177923.

[22] Umut Firat and Tayfun Akgül. Spectral estimation of cavitation related narrowband ship radiated noise based on fractional lower order statistics and multiple signal classification. In *2013 OCEANS - San Diego*, pages 1–6, 2013. doi:10.23919/OCEANS.2013.6741192.

[23] J.Y. Gil and R. Kimmel. Efficient dilation, erosion, opening, and closing algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1606–1617, 2002. doi:10.1109/TPAMI.2002.1114852.

[24] David Guevorkian, J. Astola, and Samvel Atourian. Improving gil-werman algorithm for running min and max filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19:526 – 529, 06 1997. doi:10.1109/34.589214.

[25] M. Haardt and J.A. Nossek. Unitary esprit: how to obtain increased estimation accuracy with a reduced computational burden. *IEEE Transactions on Signal Processing*, 43(5):1232–1242, 1995. doi:10.1109/78.382406.

[26] M. Haardt, M.D. Zoltowski, C.P. Mathews, and J. Nossek. 2d unitary esprit for efficient 2d parameter estimation. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 2096–2099 vol.3, 1995. doi:10.1109/ICASSP.1995.478488.

[27] Seif Eddine Hamdi, Zoubir-Mehdi Sbartaï, Antoine Boniface, Jacqueline Saliba, and Jean-Marie Henault. Pressure-induced damage monitoring in prestressed concrete of nuclear containment wall segments using acoustic emission technique – application to vercors containment building. *Engineering Fracture Mechanics*, 281:109089, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0013794423000474, doi:https://doi.org/10.1016/j.engfracmech.2023.109089.

[28] Pengcheng Han, Qun Yan, Feng Hou, and Xiao Han. Application of microphone array using compressed sensing algorithm for abnormal sound localization in aircraft strength test. In *2023 6th International Conference on Information Communication and Signal Processing (ICICSP)*, pages 1–6, 2023. doi:10.1109/ICICSP59554.2023.10390755.

[29] J. A. Högbom. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. , 15:417, June 1974.

[30] Alston S. Householder. Unitary triangularization of a nonsymmetric matrix. *J. ACM*, 5(4):339–342, oct 1958. doi:10.1145/320941.320947.

[31] Yiteng Huang, J. Benesty, G.W. Elko, and R.M. Mersereati. Real-time passive source localization: a practical linear-correction least-squares approach. *IEEE Transactions on Speech and Audio Processing*, 9(8):943–956, 2001. doi:10.1109/89.966097.

[32] Gal Itzhak and Israel Cohen. Differential and constant-beamwidth beamforming with uniform rectangular arrays. In *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5, 2022. doi:10.1109/IWAENC53105.2022.9914769.

[33] S.Z. Kalson and K. Yao. A class of least-squares filtering and identification algorithms with systolic array architectures. *IEEE Transactions on Information Theory*, 37(1):43–52, 1991. doi:10.1109/18.61101.

[34] Jyoti M. Katagi and Pandurangarao N. Kulkarni. Effect of multi-band frequency compression for improving speech perception in monaural hearing aids on source localization. In *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, pages 1–6, 2022. doi:10.1109/NKCon56289.2022.10127030.

[35] Gavin Kearney and Dermot Furlong. Sound enhancement for classroom based e-learning. In *2006 IET Irish Signals and Systems Conference*, pages 527–532, 2006.

[36] Piotr Kijanka, Bo Qiang, Pengfei Song, Carolina Amador Carrascal, Shigao Chen, and Matthew W. Urban. Robust phase velocity dispersion estimation of viscoelastic materials used for medical applications based on the multiple signal classification method. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 65(3):423–439, 2018. doi:10.1109/TUFFC.2018.2792324.

[37] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976. doi:10.1109/TASSP.1976.1162830.

[38] Tribikram Kundu. Acoustic source localization. *Ultrasonics*, 54(1):25–38, 2014. URL: https://www.sciencedirect.com/science/article/pii/S0041624X13001819, doi:https://doi.org/10.1016/j.ultras.2013.06.009.

[39] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[40] Anna Lee. Centrohermitian and skew-centrohermitian matrices. *Linear Algebra and its Applications*, 29:205–210, 1980. Special Volume Dedicated to Alson S. Householder. URL: https://www.sciencedirect.com/science/article/pii/0024379580902414, doi:https://doi.org/10.1016/0024-3795(80)90241-4.

[41] Sangmoon Lee, Sungmok Hwang, Youngjin Park, and Youn-sik Park. Sound source localization in median plane using artificial ear. In *2008 International Conference on Control, Automation and Systems*, pages 246–250, 2008. doi:10.1109/ICCAS.2008.4694558.

[42] Soo Young Lee, Jiho Chang, and Seungchul Lee. Deep learning-enabled high-resolution and fast sound source localization in spherical microphone array system. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022. doi:10.1109/TIM.2022.3161693.

[43] A.N. Lemma, A.-J. Van der Veen, and E.F. Depreterre. On the multi-resolution esprit algorithm. In *Ninth IEEE Signal Processing Workshop on Statistical Signal and Array Processing (Cat. No.98TH8381)*, pages 248–251, 1998. doi:10.1109/SSAP.1998.739381.

[44] L. Li. A new complexity bound for the least-squares problem. *Computers Mathematics with Applications*, 31(12):15–16, 1996. URL: https://www.sciencedirect.com/science/article/pii/0898122196000727, doi:https://doi.org/10.1016/0898-1221(96)00072-7.

[45] Xiaoqiang Li, Jianfeng Chen, Wentao Qi, and Rongyan Zhou. A distributed sound source surveillance system using autonomous vehicle network. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 42–46, 2018. doi:10.1109/ICIEA.2018.8397686.

[46] Ruiyu Liang, Qingyun Wang, Guichen Tang, and Li Zhao. Multiband sound source localization algorithm for directional enhancement in hearing aids. *IEEJ Transactions on Electrical and Electronic Engineering*, 11(3):331–338, 2016. URL: https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/abs/10.1002/tee.22222, arXiv:https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/pdf/10.1002/tee.22222, doi:https://doi-org.ezproxy2.utwente.nl/10.1002/tee.22222.

[47] Wei Liu and Stephan Weiss. *Wideband Beamforming — Concepts and Techniques*. 03 2010. doi:10.1002/9780470661178.

[48] G. Lucius, F. Le Roy, D. Aulagnier, and S. Azou. An algorithm for extremal eigenvectors computation of hermitian matrices and its fpga implementation. In *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1407–1410, 2013. doi:10.1109/MWSCAS.2013.6674920.

[49] Wei MA and Xun LIU. Improving the efficiency of damas for sound source localization via wavelet compression computational grid. *Journal of Sound and Vibration*, 395:341–353, 2017. URL: https://www.sciencedirect.com/science/article/pii/S0022460X17300998, doi:https://doi.org/10.1016/j.jsv.2017.02.005.

[50] Xuekang Ma and Lei Qin. Application of leakage detection based on acoustic emission technology. In *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, pages 504–508, 2023. doi:10.1109/ICDSCA59871.2023.10392936.

[51] Wageesha Manamperi, Thushara D. Abhayapala, Jihui Zhang, and Prasanga N. Samarasinghe. Drone audition: Sound source localization using on-board microphones. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:508–519, 2022. doi:10.1109/TASLP.2022.3140550.

[52] Ivo Merks, Gerald Enzner, and Tao Zhang. Sound source localization with binaural hearing aids using adaptive blind channel identification. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 438–442, 2013. doi:10.1109/ICASSP.2013.6637685.

[53] Kazuhiro Nakadai, Daisuke Matsuura, Hiroshi G. Okuno, and Hiroshi Tsujino. Improvement of recognition of simultaneous speech signals using av integration and scattering theory for humanoid robots. *Speech Communication*, 44(1):97–112, 2004. Special Issue on Audio Visual speech processing. URL: https://www.sciencedirect.com/science/article/pii/S0167639304001086, doi:https://doi.org/10.1016/j.specom.2004.10.010.

[54] Sarah Nigar and Md. Faruk Hossain. Performance comparison among robust acoustic linear, rectangular, and circular microphone array beamformers. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–4, 2023. doi:10.1109/ICCIT60459.2023.10441203.

[55] Madhab Pal, Rajib Roy, Joyanta Basu, and Milton S. Bepari. Blind source separation: A review and analysis. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–5, Nov 2013. doi:10.1109/ICSDA.2013.6709849.

[56] Sohel J. Patel, Steven L. Grant, Maciej Zawodniok, and Jacob Benesty. On the design of optimal linear microphone array geometries. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 501–505, 2018. doi:10.1109/IWAENC.2018.8521335.

[57] Z. Prime and Con Doolan. A comparison of popular beamforming arrays. pages 151–157, 01 2013. URL: https://hdl.handle.net/2440/84294.

[58] John G. Proakis and Dimitris K Manolakis. *Digital Signal Processing (4th Edition)*. Prentice Hall, 4 edition, 2006.

[59] John G. Proakis, Chrysostomos L. Nikias, Charles M. Rader, Fuyun Ling, Marc Moonen, and Ian K. Proudler. *Algorithms for Statistical Signal Processing*. Prentice Hall PTR, USA, 1st edition, 2001.

[60] I.K. Proudler, J.G. McWhirter, M. Moonen, and G. Hekstra. Formal derivation of a systolic array for recursive least squares estimation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(3):247–254, 1996. doi:10.1109/82.486470.

[61] Xiaoyu Qian, Xunrui Tang, Ziyuan Wang, and Qiao Meng. A narrowband sound source localization method and system based on a non-uniform linear microphone array. In *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*, pages 1760–1764, 2022. doi:10.1109/ICCT56141.2022.10072510.

[62] Rayleigh. Xxxi. investigations in optics, with special reference to the spectroscope. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(49):261–274, 1879. arXiv:https://doi.org/10.1080/14786447908639684, doi:10.1080/14786447908639684.

[63] James P. Reilly and Ming Kin Law. A fast high-performance array processing technique for angle-of-arrival estimation and detection of the number of incident signals. *Canadian Journal of Electrical and Computer Engineering*, 14(2):38–45, 1989. doi:10.1109/CJECE.1989.6592702.

[64] Antonio Rodà and Christian Micheloni. Tracking sound sources by means of hmm. In *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 83–88, 2011. doi:10.1109/AVSS.2011.6027299.

[65] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989. doi:10.1109/29.32276.

[66] R. Roy, A. Paulraj, and T. Kailath. Estimation of signal parameters via rotational invariance techniques - esprit. In *MILCOM 1986 - IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, volume 3, pages 41.6.1–41.6.5, 1986. doi:10.1109/MILCOM.1986.4805850.

[67] Muhammad Sajjad, Mohd Zuki Yusoff, Norashikin Yahya, and Ali Shahbaz Haider. An efficient vlsi architecture for fastica by using the algebraic jacobi method for evd. *IEEE Access*, 9:58287–58305, 2021. doi:10.1109/ACCESS.2021.3072495.

[68] Daniele Salvati and Sergio Canazza. Adaptive time delay estimation using filter length constraints for source localization in reverberant acoustic environments. *IEEE Signal Processing Letters*, 20(5):507–510, 2013. doi:10.1109/LSP.2013.2253319.

[69] Yoko Sasaki, Satoshi Kagami, and Hiroshi Mizoguchi. Multiple sound source mapping for a mobile robot by self-motion triangulation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 380–385, 2006. doi:10.1109/IROS.2006.281797.

[70] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355, April 2018. doi:10.1109/ICASSP.2018.8461310.

[71] Dirk Schröder. *Physically based real-time auralization of interactive virtual environments*, volume 11. Logos Verlag Berlin GmbH, 2011.

[72] Pieter Sijtsma. Clean based on spatial source coherence. *International Journal of Aeroacoustics*, 6(4):357–374, 2007. arXiv:https://doi.org/10.1260/147547207783359459, doi:10.1260/147547207783359459.

[73] Pieter Sijtsma, Roberto Merino-Martinez, Anwar MN Malgoezar, and Mirjam Snellen. High-resolution clean-sc: Theory and experimental validation. *International Journal of Aeroacoustics*, 16(4-5):274–298, 2017. arXiv:https://doi.org/10.1177/1475472X17713034, doi:10.1177/1475472X17713034.

[74] Jacek Stachurski, Lorin Netsch, and Randy Cole. Sound source localization for video surveillance camera. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 93–98, 2013. doi:10.1109/AVSS.2013.6636622.

[75] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, Aug 1969. doi:10.1007/BF02165411.

[76] Takao Suzuki. Damas2 using a point-spread function weakly varying in space. *AIAA Journal*, 48(9):2165–2169, 2010. arXiv:https://doi.org/10.2514/1.J050462, doi:10.2514/1.J050462.

[77] Quoc Nguyen Viet, HoSeok Kang, Sun-Tae Chung, Seongwon Cho, Keeseong Lee, and Taein Seol. Real-time audio surveillance system for ptz camera. In *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, pages 392–397, 2013. doi:10.1109/ATC.2013.6698143.

[78] Michael Vorlaender and Jason Summers. Auralization: Fundamentals of acoustics, modelling, simulation, algorithms, and acoustic virtual reality. *The Journal of the Acoustical Society of America*, 123:4028, 07 2008. doi:10.1121/1.2908264.

[79] Fei Wang, Xiang Tian, Xuesong Liu, Boxuan Gu, Fan Zhou, and Yaowu Chen. Combination complex-valued bayesian compressive sensing method for sparsity constrained deconvolution beamforming. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13, 2022. doi:10.1109/TIM.2022.3169537.

[80] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega, 2023. arXiv:2307.07970.

[81] Shuiping Zhang, Xin Tian, Chengyi Xiong, Jinwen Tian, and Delie Ming. Fast implementation for the singular value and eigenvalue decomposition based on fpga. *Chinese Journal of Electronics*, 26(1):132–136, 2017. URL: https://ietresearch-onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/abs/10.

1049/cje.2016.06.033, arXiv:https://ietresearch-onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/pdf/10.1049/cje.2016.06.033, doi:https://doi-org.ezproxy2.utwente.nl/10.1049/cje.2016.06.033.

[82] Liheng Zhao, Jacob Benesty, and Jingdong Chen. Optimal design of directivity patterns for endfire linear microphone arrays. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 295–299, 2015. doi:10.1109/ICASSP.2015.7177978.

[83] Quanfu Zheng, Lingen Luo, Hui Song, Gehao Sheng, and Xiuchen Jiang. A rssi-aoa-based uhf partial discharge localization method using music algorithm. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021. doi:10.1109/TIM.2021.3070617.

[84] M.F. Zuandi, Mareska Pratiwi Maharani, and W. Lim. Performance comparison between steered response power and generalized cross correlation in microphone arrays for sound source localization. *ARPN Journal of Engineering and Applied Sciences*, 13:3093–3100, 05 2018.

[85] Mert Burkay Çöteli, Orhun Olgun, and Hüseyin Hacıhabiboğlu. Multiple sound source localization with steered response power density and hierarchical grid refinement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2215–2229, 2018. doi:10.1109/TASLP.2018.2858932.