Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2018-00

M.Sc. Thesis

Direction-of-Arrival Estimation using an Unsynchronized Array of Acoustic Vector Sensors

Ing. Bart Coonen

Abstract

Direction-Of-Arrival (DOA) estimation of acoustic signals is of great interest in various applications including battlefield acoustic and noise localization. Acoustic sensors are employed in an array configuration to estimate DOAs based on the time differences of arrival DOAs. However, the acoustic sensors in the network have all their own Data AcQuisition (DAQ) unit with independent clocks than, it might not be possible to perfectly synchronize the network which affects the performance of the time differences of arrival reliably.

In this thesis we consider the issue of clock synchronization errors in a network where Acoustic Vector Sensors (AVSs) are used. AVSs are shown to be advantageous in terms of direction finding compared to conventional Acoustic Pressure Sensors (APSs) due to their directional particle velocity measurement capability. Initiallity the measurement model for AVSs is presented. After that the behavior of the clocks is incorporated in the measurement model of the full array setup. Subsequently, the effects of the clocks on the MVDR DOA estimation method is discussed.

The model with clock errors is used in the development of three new DOA-estimation methods. The first two techniques are eigenstructure methods that are capable of finding the DOAs regardless of the accuracy of the synchronization. However, to find the DOAs with high accuracy in a real-time application these methods are not due to their high computational cost. Alternatively, the third proposed algorithm takes the DOA estimate from previous methods with low accuracy as its input. The algorithm estimates the DOA in an iterative fashion with high accuracy based on these estimates with low accuracy.

Finally, measurements are conducted in a controlled environment in order to show that these methods are usable in practical situations.

Direction-of-Arrival Estimation using an Unsynchronized Array of Acoustic Vector Sensors

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Ing. Bart Coonen
born in Venlo, The Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Copyright © 2018 Circuits and Systems Group
All rights reserved.

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Direction-of-Arrival Estimation using an Unsynchronized Array of Acoustic Vector Sensors"** by **Ing. Bart Coonen** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 22 May 2018

Chairman: _____
prof.dr.ir. Geert Leus

Advisor: _____
ir. Krishnaprasad Nambur Ramamohan

Committee Members: _____
dr. Faruk Uysal

_____
dr. Daniel Fernandez Comesana

Abstract

Direction-Of-Arrival (DOA) estimation of acoustic signals is of great interest in various applications including battlefield acoustic and noise localization. Acoustic sensors are employed in an array configuration to estimate DOAs based on the time differences of arrival DOAs. However, the acoustic sensors in the network have all their own Data AcQuisition (DAQ) unit with independent clocks than, it might not be possible to perfectly synchronize the network which affects the performance of the time differences of arrival reliably.

In this thesis we consider the issue of clock synchronization errors in a network where Acoustic Vector Sensors (AVSs) are used. AVSs are shown to be advantageous in terms of direction finding compared to conventional Acoustic Pressure Sensors (APSs) due to their directional particle velocity measurement capability. Initiallity the measurement model for AVSs is presented. After that the behavior of the clocks is incorporated in the measurement model of the full array setup. Subsequently, the effects of the clocks on the MVDR DOA estimation method is discussed.

The model with clock errors is used in the development of three new DOA-estimation methods. The first two techniques are eigenstructure methods that are capable of finding the DOAs regardless of the accuracy of the synchronization. However, to find the DOAs with high accuracy in a real-time application these methods are not due to their high computational cost. Alternatively, the third proposed algorithm takes the DOA estimate from previous methods with low accuracy as its input. The algorithm estimates the DOA in an iterative fashion with high accuracy based on these estimates with low accuracy.

Finally, measurements are conducted in a controlled environment in order to show that these methods are usable in practical situations.

Acknowledgments

The reasearch work in this thesis is conducted in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering at the Delft University of Technology. The research is being supported by Microflown Technologies, Arnhem.

I wish to express my gratitude to Krishnaprasad for his daily supervision and excellent guidance. He teached me to explore all the possibilities. His valuable comments helped me improve my work significantly. His pleasant attitude has made it enjoyable to collaborate with him.

I am also extremely grateful to my professor, Geert Leus, for giving me the opportunity to with him. I am thankful for his useful feedback and guidance throughout the MSc. thesis.

I would also like to thank all other people within Microflown and the Circuits and System group who supported me during the MSc. thesis.

Ing. Bart Coonen
Delft, The Netherlands
22 May 2018

Contents

1 Introduction

In array signal processing multiple sensors are placed in an array in order to obtain space-time information of the measured quantity. This information can be used in order to estimate or detect certain parameters in this space-time domain. Examples of these parameters are Direction Of Arrival (DOA) estimation and estimation of the source spectrum. Array signal processing has applications in fields such as acoustics, radar, wireless communication and many others.

In this work the focus is on DOA estimation of acoustic sources. Traditionally Acoustic Pressure Sensors (APSs) are used to characterize the acoustic field. These APSs measure the variation of pressure induced by the acoustic signals. If the sensors are employed in an array configuration we can find the Time Differences Of Arrival (TDOA) of the signals. The TDOA across all sensors in the array form an interference pattern from which directional information of the acoustic sources can be extracted.

An acoustic field is characterized by two quantities, next to the acoustic pressure there is particle velocity. Particle velocity is a vector-quantity which denotes the flow of particles induced by pressure differences over the field. An APS solely measures the pressure whereas an acoustic vector sensor (AVS) or μ-flown measures both the pressure and particle velocity [1]. It is clear that the particle velocity carries extra information about the DOA due to the directional properties of this quantity. Microflown Technologies is a manufacturer of AVSs [2].

The effects of this extra information in the DOA estimation trough the TDOA are discussed by Krishnaprasad in [3]. For DOA estimation with APSs the array configuration is limited due to the frequency of the signals. AVSs add an extra factor which is independent of frequency. This enables the possibility to use multiple frequencies of signals with a single array.

AVS-array processing for DOA estimation has recently received increased attention in various fields. Such as underwater acoustics [4], service robotics [5] and noise vibration and holography analysis [6, 7].

It is of great importance that all the sensors in the array are perfectly synchronized in order to accurately estimate the TDOAs. In most array signal processing problems the frequencies of the measured signals have a relative small wave length such that the sensors can be placed relatively close to each other. In this way the different sensors can be connected to the same DAQ-unit with only one clock. Therefore it is easy to synchronize the sensors. However, in the field of acoustics the wavelengths may become big and the sensors have to be placed further apart. Consequently it becomes

difficult to connect all sensors to the same DAQ unit. As a result, other methods of synchronization have to be used. Such synchronization can, for example, be performed by means of GPS or through line-of-sight communication.

An application of unsynchronized DOA estimation is found in battlefield acoustics [8, 9]. If the sensors are deployed within a battlefield it might occur that the electromagnetic spectrum is (partially) blocked. Therefore space-time synchronization through GPS will not always be possible. Due to the blocked spectrum it is reasonable to assume that line-of-sight communication is impossible. Hence it is unachievable to accurately synchronize the different clocks of the sensors

This work focuses on the effects of the above described clock errors. Questions answered in this thesis include but are not limited to:

- How can we model an AVS-array that is subjected to clock errors?
- How can we perform DOA estimation on an AVS-array that is subjected to clock errors?

In the following a brief outline of this thesis is given:

Chapter 2 AVS Array - This chapter starts with assumptions on the sensors and the environment these are operating in. Then based on these assumptions a measurement model is build for the sensors. Last some conventional DOA-methods are given so that these can be used as a reference in later chapters. These methods include the Minimum Variance Directional Response (MVDR) beamformer and the MUltiple SIgnal Classification (MUSIC) method.

Chapter 3 Clock Effects on DOA Estimation - This chapter starts with a general discussion about clocks and how these can be modeled. Then these clocks are added to the measurement model. Finally the effects of the clock errors on the MVDR are discussed in the case where one signal is impinging the array.

Chapter 4 Robust DOA Estimation - This chapter focuses on finding an algorithm that includes the clock effects from previous chapter in the DOA estimation. Firstly, an algorithm for an APS array that is independent of clock effects is discussed. Secondly, based on the above algorithm two novel methods are proposed which are independent of clock effects. As these methods are computational expensive a third algorithm is proposed which is not completely independent of clock errors but can be combined with the previous techniques in order to reduce their cost.

Chapter 5 Experimental Results - In this chapter experiments are conducted in a controlled environment, these experiments give a proof of concept of the techniques developed in previous section.

2 Acoustic Vector Sensor Arrays

In this chapter firstly the data model of the AVS Array that is being used throughout this thesis is discussed. The data model is based on the model introduced in [10] and is studied extensively by [3]. This model will be discussed in Section 2.1. Afterwards we will discuss some classical approaches to find the direction of arrival in sensor arrays which are extended to AVS arrays.

2.1 Data Model

In order to build the data model, we will need to make some assumptions on the sensors as well as on its environment [3]:

Co-located sensor components An AVS measures multiple quantities (i.e. pressure and particle velocity). The assumption is that these quantities are measured at the same location in space. Therefore we are able to represent an AVS as a single point. In reality the maximum distance between these measurement points is less then one centimeter. In a setup covering more then 10 meter this centimeter is neglectable.

Free-space environment The acoustic environment is assumed to be a homogeneous isotropic medium. In reality this will not always be the case and the model has to be adjusted to cover also non free space environments when this situation occurs.

Narrow-band signals The source signals are assumed to be narrow-banded such that the signal time delays between sensors is smaller than the inverse bandwith and they can be represented as phase shifts. The measured signals of interest will not be narrow banded but by bandpass filtering the measurement, this broad-band signal can be split up into multiple narrow-band signals.

Far field We assume to be in the far field of the sound source and therefor we will measure a plane wave. This assumption holds whenever the distance between the source and the array is much larger (at least ten times) than the size of the array aperture.

The following assumptions are not necessary in order to build the data model but will be used later throughout the thesis. For convenience these assumptions will be stated here.

Source distribution	The acoustic sources will be assumed to be zero-mean complex Gaussian distributed. Furthermore all sources are assumed to be uncorrelated to each other.
Noise distribution	The noise is assumed to have a zero-mean circular complex Gaussian distribution and is uncorrelated to the acoustic sources.

Based on the first set of assumption the data model will be derived for an acoustic vector sensor.

Under the far field assumption it can be shown [10] that the relation between the acoustic particle velocity $\mathbf{v}(\mathbf{r}, t)$ and acoustic pressure $p(\mathbf{r}, t)$ can be expressed in terms of position $\mathbf{r} = \begin{bmatrix} r_x & r_y \end{bmatrix}$ and time t as:

$$\mathbf{v}(\mathbf{r}, t) = -\frac{\mathbf{u}}{\rho c} p(\mathbf{r}, t), \tag{2.1}$$

where ρ denotes the density of the medium and c the speed of sound in the medium and $\mathbf{u}$ is an unitvector pointing from $\mathbf{r}$ in the direction of the far field source. This equation shows that the pressure and all particle velocity channels will be in phase with each other when working under the far field assumption.

In Figure 2.1 a schematic overview of an AVS array is given. Multiples AVSs are placed freely in 2D-space. For convenience it is useful to place one sensor at the origin of the coordinate system. It is clear from the far field assumption that $\mathbf{u}$ will be the same for all sensors:

$$\mathbf{u} = \begin{bmatrix} u_x & u_y \end{bmatrix}^T = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}^T. \tag{2.2}$$

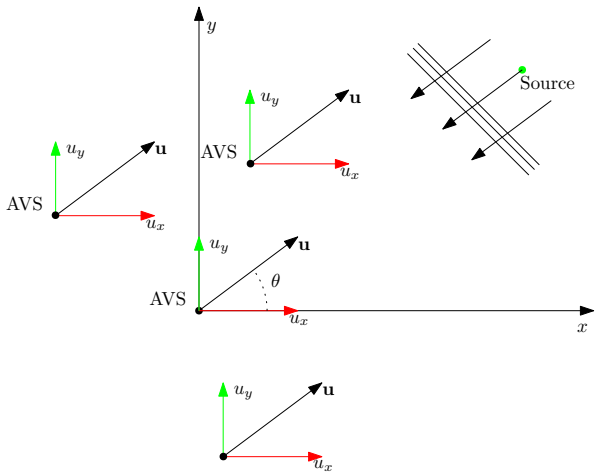


Figure 2.1: A schematic overview of an arbitrary AVS array

If we stack all the sensor outputs at time instant t in a vector $\mathbf{y}(t)$ we can write [3]:

$$\mathbf{y}(t) = \mathbf{a}(\theta) s(t) + \boldsymbol{\eta}(t) \in \mathbb{C}^{3M}, \tag{2.3}$$

where

$$\mathbf{a}(\theta) = \mathbf{a}_{\mathrm{p}}(\theta) \otimes \mathbf{h}(\theta) \in \mathbb{C}^{3M}, \tag{2.4}$$

is the steering vector,

$$\mathbf{a}_{\mathrm{p}}(\theta) = \begin{bmatrix} e^{\mathrm{j}k\mathbf{r}_1^T\mathbf{u}} & e^{\mathrm{j}k\mathbf{r}_2^T\mathbf{u}} & \dots & e^{\mathrm{j}k\mathbf{r}_M^T\mathbf{u}} \end{bmatrix}^T \in \mathbb{C}^M \tag{2.5}$$

the corresponding pressure array steering vector, j the complex number, $\otimes$ denotes the Kronecker product, k the wavenumber, $\mathbf{r}_i = \begin{bmatrix} r_{i,x} & r_{i,y} \end{bmatrix}$ the location vector of sensor i, $\mathbf{h}(\theta) = \begin{bmatrix} 1 & \mathbf{u}^T \end{bmatrix}^T \in \mathbb{R}^4$ the weighting vector for the different channels, $s(t)$ the source signal and $\boldsymbol{\eta}(t)$ the measurement noise.

This model can easily be extended to a multiple source model. Let θ_i denote the angles of arrival of source i and $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_K \end{bmatrix}^T$ be the vector containing these angles where K denotes the amount of sources. Now we could write:

$$\mathbf{y}(t) = \boldsymbol{A}(\boldsymbol{\theta})\mathbf{s}(t) + \boldsymbol{\eta}(t), \tag{2.6}$$

where $\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{a}(\theta_1) & \mathbf{a}(\theta_2) & \dots & \mathbf{a}(\theta_K) \end{bmatrix}$ is the mixing matrix and $\mathbf{s}(t) = \begin{bmatrix} s_1(t) & s_2(t) & \dots & s_K(t) \end{bmatrix}^T$ is the vector containing the source signals where $s_i(t)$ corresponds to the source signal of source i at time instant t.

Expanding the model in time by defining the signal matrix $\boldsymbol{S} = \begin{bmatrix} \mathbf{s}(1) & \mathbf{s}(2) & \dots & \mathbf{s}(N) \end{bmatrix}$ where N equals the amount of samples and defining $\boldsymbol{N} = \begin{bmatrix} \boldsymbol{\eta}(1) & \boldsymbol{\eta}(2) & \dots & \boldsymbol{\eta}(N) \end{bmatrix}$ we can write

$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{S} + \boldsymbol{N}, \tag{2.7}$$

where $\boldsymbol{Y} = \begin{bmatrix} \mathbf{y}(1) & \mathbf{y}(2) & \dots & \mathbf{y}(N) \end{bmatrix}$ contains all received sensor measurements.

2.2 Direction of Arrival Estimation

In Direction of Arrival (DOA) estimation we try to retrieve the angles of arrival $\boldsymbol{\theta}$ from the measurement data $\boldsymbol{Y}$. For simplicity we will discuss in this section only the case where one signal is present and will therefore only focus on finding the angle of arrival θ.

Three well known approaches for finding the DOA are the classical beamformer, the Minimum Variance Distortionless Response (MVDR) and MUltiple SIgnal Clasification (MUSIC). all three will be shortly explained in this section for later usage if one would like to have more information on these one could look into [11]. The classical beamformer and the MVDR create a so-called beamformer vector $\mathbf{w}(\theta)$ which tries to estimate the signal vector $\mathbf{s}_1 = \begin{bmatrix} s(1) & s(2) & \dots & s(N) \end{bmatrix}$ as follows

$$\hat{\mathbf{s}}_1 = \mathbf{w}^T(\theta)\boldsymbol{Y}, \tag{2.8}$$

where $\hat{\mathbf{s}}_1$ denotes the estimate of $\mathbf{s}_1$. These beamformer vectors have a dependency on (θ), so one could perform a grid search over these θ's and take the θ that maximizes the source power as the estimate of the direction of arrival i.e.

$$\hat{\theta} = \arg\max_{\theta'} \hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^H, \tag{2.9}$$

where $\hat{\theta}$ is the estimate of θ.

The classical beamformer is defined by solving (2.8) in a least square sense which leads to:

$$\mathbf{w}^H(\theta) = \mathbf{a}^\dagger(\theta) \tag{2.10}$$

where $[\cdot]^\dagger$ denotes the Moore-Penrose pseudo-inverse. Substituting this in (2.9) leads to:

$$\hat{\theta} = \arg\max_{\theta'} \frac{\mathbf{a}^H(\theta')\boldsymbol{Y}\boldsymbol{Y}^H\mathbf{a}(\theta')}{\mathbf{a}^H(\theta')\mathbf{a}(\theta')\mathbf{a}^H(\theta')\mathbf{a}(\theta')} = \arg\max_{\theta'} \ \mathbf{a}^H(\theta')\hat{\boldsymbol{R}}\mathbf{a}(\theta'), \tag{2.11}$$

where $\hat{\boldsymbol{R}} = N^{-1}\boldsymbol{Y}\boldsymbol{Y}^H$ is the sample covariance matrix and we used the fact that $\mathbf{a}^H(\theta')\mathbf{a}(\theta') = 2M$ is constant over θ. This completes the DOA estimation using the classical beamformer.

The MVDR beamformer minimizes the output power while retaining the output power in the θ direction to be equal to one i.e.:

$$\min_{\mathbf{w}} \ \mathbf{w}^H(\theta)\hat{\boldsymbol{R}}\mathbf{w}(\theta) \qquad\qquad \text{s.t. } \mathbf{w}^H(\theta)\mathbf{a} = 1 \tag{2.12}$$

Solving this using Lagrange multipliers leads to:

$$\mathbf{w}(\theta) = \frac{\hat{\boldsymbol{R}}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\hat{\boldsymbol{R}}^{-1}\mathbf{a}(\theta)}, \tag{2.13}$$

and the estimate for the DOA is found as:

$$\hat{\theta} = \arg\max_{\theta'} \ \frac{1}{\mathbf{a}^H(\theta')\hat{\boldsymbol{R}}^{-1}\mathbf{a}(\theta')} \tag{2.14}$$

Which complements MVDR based DOA estimation.

The MUSIC method uses a subspace based approach on the correlation matrix $\boldsymbol{R}$ to find the DOA. We can find $\boldsymbol{R}$ as follows

$$\boldsymbol{R} = \mathbb{E}\left[\mathbf{y}\mathbf{y}^H\right] = \sigma_s^2\mathbf{a}(\theta)\mathbf{a}^H(\theta) + \sigma_\eta^2\boldsymbol{I}. \tag{2.15}$$

Note that $\hat{\boldsymbol{R}}$ is an estimate of $\boldsymbol{R}$. Now, define the eigenvalue decomposition of $\boldsymbol{R}$

$$\boldsymbol{R} = \sum_{m=1}^{3M} \beta_{\boldsymbol{R},m}\mathbf{u}_{\boldsymbol{R},m}\mathbf{u}_{\boldsymbol{R},m}^H. \tag{2.16}$$

where $\{\beta_{\boldsymbol{R},m}\}_{m=1}^{3M}$ are the eigenvalues of $\boldsymbol{R}$ in descending order with corresponding eigenvectors $\{\mathbf{u}_{\boldsymbol{R},m}\}_{m=1}^{3M}$. From (2.15) it is easy to see that signal subspace of $\boldsymbol{R}$ coincides with the subspace of the rank-1 approximation of $\boldsymbol{R}$. The orthogonal component of this rank-1 approximation will therefore span the noise space of $\boldsymbol{R}$. Note that this noise subspace ($\boldsymbol{U}_{\boldsymbol{R},N}$) can be written as:

$$\boldsymbol{U}_{\boldsymbol{R},N} = \begin{bmatrix}\mathbf{u}_{\boldsymbol{R},2} & \mathbf{u}_{\boldsymbol{R},3} & \dots & \mathbf{u}_{\boldsymbol{R},3M}\end{bmatrix} \tag{2.17}$$

Now, we can find the DOAs by a grid search:

$$\hat{\theta} = \arg\max_{\theta'} \frac{1}{||\mathbf{a}(\theta)\boldsymbol{U}_{\boldsymbol{R},N}||^2} \tag{2.18}$$

As the steering vector $\mathbf{a}(\theta)$ is orthogonal to the noise space the cost function above maximizes for $\theta' = \theta$ and the DOA is found.

Although the above described methods might give good results with the data model described in section 2.1, these estimators may give poor results in practice. The made mistakes are mainly due to that the steering vector $\mathbf{a}(\theta)$ is not exactly known in practice. Placement flaws and clock errors might affect this vector.

The next chapter will focus on clock errors; How can we model them and what influence do they have on the described technique? The MVDR is investigated specifically. Afterwards, Chapter 4 will focus on robust DOA estimation which tries to minimize the miscalculation due to clocks. Inaccuracies due to placement will not be considered in this work. Due to the relative low speed of sound ($c \approx 340$ m/s), these can be considered to be neglectable.

Clock Effects on Direction of Arrival Estimation

3

In the previous chapter we discussed about DOA estimation techniques when the steering vector is exactly known. In this work we will consider arrays where synchronization between sensors is not always possible. In these cases it is common that the time stamping of the data relies on the clocks equipped in the sensors. If these clock are not accurate steering vector mismatches arise. In this chapter, firstly, a model for the different clocks is developed and integrated it in the previously defined model. Secondly, the effects of clock errors on classical DOA estimation with one source will be discussed.

3.1 Clocks

Let A and ν denote the nominal amplitude and frequency of a clock oscillator. Then the clock can be modeled as [12]:

$$a(t) = (A + \delta A\,(t)) \sin\left(2\pi\nu C\,(t)\right), \tag{3.1}$$

where t deotes true time, $\delta A(t)$ the amplitude error as a function of time, $C(t)$ the frequency normalized phase with unit of times and could be written as $C(t) = t+\delta C(t)$. Ideally $C(t) = t$ but in practice time varying errors $\delta C(t)$ exists. Usually $\delta A(t)$ can be eliminated using hard limiting using differential comparison techniques. This leaves the phase error $\delta C(t)$ as the only error.

The error $\delta C(t)$ can be understood by expanding it as a polynomial by Taylor expansion of time and plugging it in $C(t)$. This leads to [13]:

$$\begin{aligned} C(t) &= t + \delta C(t) && (3.2)\\ &= \phi + \left(1 + \dot{\phi}\right) t + 0.5\ddot{\phi}t^2 + ... + \eta(t) && (3.3)\\ &= \phi + \omega t + 0.5\ddot{\phi}t^2 + ... + \eta(t), && (3.4) \end{aligned}$$

where ϕ is the phase offset at $t = 0$ and is real valued, $\omega = 1 + \dot{\phi}$ is the clock skew or the clock frequency at $t = 0$, $\eta(t)$ contains stochastic noise as well as other noises like effects from humidity and temperature. An ideal clock has no phase error and therefore can be modeled as $C = t$ and thus $[\phi, \dot{\phi}, \ddot{\phi}] = [0, 0, 0]$.

In this work we will model the clock as an affine system and assume that the coefficients of this system are time-invariant over a later defined period of time:

$$C(t) \approx \omega t + \phi. \tag{3.5}$$

In order to check the validity of this approximation we want to estimate the difference between the last expression and the real clock (3.2). There are various ways

to evaluate the errors on this affine system, such as Allan variance, Overlapping Allan, Hadamard and Theol [14]. We will focus on Allan variance and overlapping Allan variance of which the latter will be denoted as AVAR. Allan variance is an easy to understand characterization, this will help us to explain AVAR, which is the IEEE standard. In order to illustrate these variances we will derive the instantaneous frequency $\tilde{\nu}(t)$ by differentiating the phase in (3.1) and dividing it by 2π:

$$\tilde{\nu}(t) = \frac{1}{2\pi}\frac{d}{dt}\pi\nu C(t) = \nu + \nu\left(\frac{d}{dt}\delta C(t)\right). \tag{3.6}$$

Now we can define the normalized fractional frequency as the relative error on the nominal frequency ν:

$$\zeta(t) \triangleq \frac{\tilde{\nu}(t) - \nu}{\nu} = \frac{d}{dt}\delta C(t). \tag{3.7}$$

Averaging this fractional frequency over a time period τ:

$$\overline{\zeta}(t,\tau) = \frac{1}{\tau}\int_t^{t+\tau} \zeta(t)dt. \tag{3.8}$$

Finally the Allan variance is defined as the expected value of one half of the time-averaged squares of the differences between adjacent fractional frequency deviations $\overline{\zeta}(t,\tau)$ [13]:

$$\sigma^2_{\zeta_{Allan}}(\tau) = 0.5\mathbb{E}\left[(\overline{\zeta}(t+\tau,\tau) - \overline{\zeta}(t,\tau))^2\right], \tag{3.9}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator and $\sigma^2_{\zeta_{Allan}}(\tau)$ the Allan variance for time duration τ. Note that in discrete form this could be estimated as [14]:

$$\sigma^2_{\zeta_{Allan}}(\tau) = \frac{1}{2(\overline{N}-1)}\sum_{k=1}^{\overline{N}-1}(\overline{\zeta}_{k+1,\tau} - \overline{\zeta}_{k,\tau})^2, \tag{3.10}$$

where $\overline{N}$ denotes the amount of samples and $\overline{\zeta}_{k,\tau}$ the averaged fractional frequency corresponding to the k^{th} sample over time duration τ. Now, let $\{C(T_k)\}_{k=1}^{\overline{N}}$ denote a discrete set of time measurements on the clock at true time instants T_k with a nominal spacing of $\tau = T_{k+1} - T_k$, we can write the average fractional frequency of the k^{th} measurement by substituting (3.7) in (3.8):

$$\overline{\zeta}_{k,\tau} = \frac{\delta C(T_k+\tau) - \delta C(T_k)}{\tau}. \tag{3.11}$$

Substituting this in (3.10) leads to:

$$\sigma^2_{\zeta_{Allan}}(\tau) = \frac{1}{2\tau^2(\overline{N}-2)}\sum_{k=1}^{\overline{N}-2}(\delta C(T_k+2\tau) - 2\delta C(T_k+\tau) + \delta C(T_k))^2. \tag{3.12}$$

If we substitute (3.4) in this equation we get:

$$\begin{aligned}\sigma^2_{\zeta_{Allan}}(\tau) = \frac{1}{2\tau^2(\overline{N}-2)}\sum_{k=1}^{\overline{N}-2}\Big(&\phi+\dot{\phi}(T_k+2\tau)+0.5\ddot{\phi}(T_k+2\tau)^2+...+\eta_{k+2}\\ &-2\big(\phi+\dot{\phi}(T_k+\tau)+0.5\ddot{\phi}(T_k+\tau)^2+...+\eta_{k+1}\big)\\ &+\phi+\dot{\phi}(T_k)+0.5\ddot{\phi}(T_k)^2+...+\eta_k\Big)^2\\ = \frac{1}{2(\overline{N}-2)}\sum_{k=1}^{\overline{N}-2}(\ddot{\phi}\tau+\cdots+\ddot{\eta}_k)^2,&\end{aligned} \tag{3.13}$$

where $\ddot{\eta} = \tau^{-1}(\dot{\eta}_{k+1}-\dot{\eta}_k)$ and $\dot{\eta}_k = \tau^{-1}(\eta_{k+1}-\eta_k)$. Note that the frequency drift ($\ddot{\phi}$), the averaged random noise and the higher order terms are in the equation for the Allan variance while the phase and frequency offset are eliminated. Therefore the Allan variance is a measure of noise contributed by higher order terms of the clock during an integration time τ. **Thus our affine model proposed in (3.5) is valid for a coherence time τ given that the Allan variance for the used clock with integration time τ is sufficiently low.** What sufficiently low exactly means will be discussed after the explanation of AVAR in this section.

Recall that the Allan Variance subtracts two adjacent fractional frequency averages ($\overline{\zeta}_{k+1,\tau_0}-\overline{\zeta}_{k,\tau_0}$) with averaging time τ_0. In contrast, AVAR averages over a time $\overline{n}\tau_0$, where $\overline{n}$ is the averaging factor. Later we find that the expected value of both the Alan Variance and AVAR to be equalt. However, [14] shows that the latter has a lower variance. AVAR is formally defined as [14]:

$$\sigma^2_\zeta(\tau=\overline{n}\tau_0) = \frac{1}{2\overline{n}^2(\overline{N}-2\overline{n})}\sum_{l=1}^{\overline{N}-2\overline{n}}\left(\sum_{k=l}^{l+\overline{n}-1}(\zeta_{k+\overline{n},\tau}-\zeta_{k,\tau})\right)^2. \tag{3.14}$$

The inner sum in this equation is in fact an averaging operation and it can therefore easily be rewritten to:

$$\sigma^2_\zeta(\tau) = \frac{1}{2(\overline{N}-\overline{n})}\sum_{k=1}^{\overline{N}-2\overline{n}}\left(\overline{\zeta}_{k+\overline{n},\tau}-\overline{\zeta}_{k,\tau}\right)^2. \tag{3.15}$$

Substituting (3.11) in this equation yields:

$$\sigma^2_\zeta(\tau) = \frac{1}{2\tau^2(\overline{N}-2\overline{n})}\sum_{k=1}^{\overline{N}-2\overline{n}}\left(\delta C(T_k+2\tau)-2\delta C(T_k+\tau)+\delta C(T_k)\right)^2. \tag{3.16}$$

Substituting (3.4) and a comparable derivation as for the normal Allan variance leads to:

$$\sigma^2_\zeta(\tau) = \frac{1}{2(\overline{N}-2\overline{n})}\sum_{k=1}^{\overline{N}-2\overline{n}}\left(\ddot{\phi}\tau+...+\ddot{\eta}_k\right)^2. \tag{3.17}$$

If we compare the last expression with (3.13) we observe that the Allan variance and AVAR are very similar..

The affine clock model given in (3.5) is valid as long as the AVAR ($\sigma_\zeta(\tau)$) is negligible small for a certain coherence time $\tau = \tau_c$. A rough stability requirement by [15] states that the coherence time is the time for which the RMS phase error of the clock remains less than 1 radian:

$$2\pi\nu_o\sigma_\zeta(\tau_c)\tau_c \leq 1, \tag{3.18}$$

where ν_o is the observed frequency. So if we want to have a coherence time of 100 seconds while using a clock with frequency ν_0 equal to the sampling rate of 16kHz, we need an Allan standard deviation of:

$$\sigma_\zeta(\tau = 100) \leq 10^{-8}. \tag{3.19}$$

Typical Allan variances for a temperature compensated quartz oscillator with a coherence time between 1 and 100 seconds are according to [16] between 10^{-8} and 10^{-9}. These oscillator thus have the desired Allan variance. Unfortunately, the clocks currently applied in the AVSs are not temperature compensated which makes it hard to find reliable Allan variances. Although the Allan variance of our clocks might not meet the requirement measurements are conducted in order to show their affine behaviour. These measurements are shown in Appendix A. This appendix shows that the affine model is valid for at least a small period of time.

3.2 Incorporating Clocks into the Model

The affine system describing the clocks from previous section can incorporated in the data model. Note that the clock mismatches will only change the phase of the measured signal. In fact, the steering vector for the pressure channel can be rewritten as follows:

$$\mathbf{a}_{\mathrm{p}_\epsilon}(\theta, t) = \begin{bmatrix} e^{\mathrm{j}kc(t-t_1(t))} \\ e^{\mathrm{j}kc(t-t_2(t))} \\ \vdots \\ e^{\mathrm{j}kc(t-t_M(t))} \end{bmatrix} \odot \mathbf{a}_{\mathrm{p}}, \tag{3.20}$$

where $t_i(t) = \omega_i t + \phi_i$ denotes the time according to sensor i at time t and $\cdot$ denotes the Hadamard product. Note that we could write the offsets ϕ_i in a diagonal matrix $\mathbf{\Phi}$ where the entries are given by $\{\mathbf{\Phi}\}_{ii} = e^{\mathrm{j}kc\phi_i}$. Similarly write the slopes in a diagonal matrix $\mathbf{\Omega}(t)$ with entries $\{\mathbf{\Omega}(t)\}_{ii} = e^{\mathrm{j}kc(1-\omega_i)t}$. The steering vector can be written as:

$$\mathbf{a}_{\mathrm{p}_\epsilon}(\theta, t) = \mathbf{\Phi}\mathbf{\Omega}(t)\mathbf{a}_{\mathrm{p}}(\theta). \tag{3.21}$$

where $\mathbf{a}_{\mathrm{p}}(\theta)$ is given by (2.5). For performing DOA estimation only a few samples in time are taken. If we start to perform the DOA estimation at time T_1 and end at T_2, then the start and end time of the DOA estimation of sensor i could be written as

$t_i(T_1)$ and $t_i(T_2)$ respectively.

Usually, in battlefield acoustics the source signals are present for a short period of time. Therefore it is not possible to use a lot of sample points. Due that only a short period of time is taken for the DOA estimation and $t_i(t)$ is shown to change relatively slow. We conclude that $t_i(T_1) \approx t_i(T_2)$. So we could approximate the clock error for the DOA estimation algorithm as static. For that reason the clock effects are assumed to be constant throughout the algorithm. However the clocks will be simulated as an affine system in order to show that the given approximation does not have a big effect on the estimation. The choice is made to use the average clock error within T_1 and T_2:

$$\begin{aligned} \mathbf{\Gamma} &= \mathbf{\Phi} \left(\mathbf{\Omega} \left(T_1 \right) \mathbf{\Omega} \left(T2 \right) \right)^{1/2}, \\ \mathbf{a}_{\mathrm{p}_\epsilon}(\theta) &\approx \mathbf{\Gamma} \mathbf{a}_{\mathrm{p}}, \end{aligned} \tag{3.22}$$

where $\mathbf{\Gamma}$ is a diagonal matrix containing the clock errors. Note that an index of $\mathbf{\Gamma}$ could be written as:

$$\{\mathbf{\Gamma}\}_{ii} = e^{jkc\gamma_i} = e^{jkc\left(\frac{1}{2}(1-\omega_i)(T_1+T_2)+\phi_i\right)}. \tag{3.23}$$

Note that the approximation in (3.22) will generate time-dependent phase mismatches between our model and reality. In Appendix B, simulations are shown that support the given similarity.

3.3 Effects of Clock Errors on DOA Estimation

In this section, we focus on effects of the clocks on the DOA estimation. This will give a general insight into the problem and shows that correcting for clock errors is essential.

The previously established model will be analyzed for clock errors that follow a Gaussian distribution with given standard deviation. Furthermore, we assume a 2D acoustical space with one source and multiple APSs. These APSs are placed in a Uniformly spaced Linear Array (ULA). Firstly, the MVDR response for a non-errorneous signal is derived. Secondly the MVDR response for an APS with clock errors is derived and analyzed. Lastly, the discussion is extended to the case of an AVS array.

3.3.1 Acoustic Pressure Sensor Array

The model introduced in Chapter 2 is a general model for AVS arrays, in this section however, we will use an APS array. The derived model can be reduced to an APS array by neglecting the $\mathbf{h}(\theta)$-vectors. Firstly, this model is discussed without clock errors. Secondly, a closed form solution will be derived when (small) clock errors are present. Note that if we reduce (2.7) to one source and multiple APSs we will get:

$$\mathbf{Y}_{\mathrm{p}} = \mathbf{a}_{\mathrm{p}}(\theta)\mathbf{s}_1^T + \boldsymbol{N} \in \mathbb{C}^{N \times M}. \tag{3.24}$$

Furthermore, we could express $\mathbf{a}_\mathrm{p}$ for a ULA, where the most left sensor is placed at the origin of the coordinate system, as:

$$\mathbf{a}_\mathrm{p}(\theta) = \begin{bmatrix} e^{\mathrm{j}k(\mathbf{r}_1^T\mathbf{u})} & e^{\mathrm{j}k(\mathbf{r}_2^T\mathbf{u})} & \dots & e^{\mathrm{j}k(\mathbf{r}_M^T\mathbf{u})} \end{bmatrix}^T = \begin{bmatrix} 1 & e^{\mathrm{j}kd\cos\theta} & e^{2\mathrm{j}kd\cos\theta} & \dots & e^{(M-1)\mathrm{j}kd\cos\theta} \end{bmatrix}^T, \tag{3.25}$$

where d is the distance between two adjacent sensors. The correlation matrix from (3.24) can be expressed as:

$$\begin{aligned} \boldsymbol{R}_\mathrm{p} &\triangleq \lim_{N\to\infty} \frac{1}{N}\boldsymbol{Y}_\mathrm{p}\boldsymbol{Y}_\mathrm{p}^H \\ &= \sigma_s^2 \mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta) + \sigma_\eta^2 \boldsymbol{I}_M, \end{aligned} \tag{3.26}$$

where we used $\mathbf{a}_\mathrm{p}^H(\theta)\mathbf{a}_\mathrm{p}(\theta) = M$, defined σ_s^2 as the signal power and σ_η^2 as the noise power. Furthermore, we assumed the signal and the noise to be uncorrelated. We can find the inverse with the Sherman-Morrison matrix inversion lemma:

$$\boldsymbol{R}_\mathrm{p}^{-1} = \sigma_\eta^{-2} I_M - \frac{\sigma_\eta^{-4}\sigma_\mathrm{s}^2 \mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta)}{1 + \sigma_\eta^{-2}\sigma_\mathrm{s}^2 \mathbf{a}_\mathrm{p}^H(\theta)\mathbf{a}_\mathrm{p}(\theta)} = \frac{\left(\sigma_\eta^2 + \sigma_\mathrm{s}^2 M\right)\boldsymbol{I}_M - \sigma_\mathrm{s}^2 \mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta)}{\sigma_\eta^2\left(\sigma_\eta^2 + \sigma_\mathrm{s}^2 M\right)}. \tag{3.27}$$

Substituting this in the equation of the MVDR (2.14) leads to:

$$P_{MVDR} = \frac{1}{\mathbf{a}\mathrm{p}^H(\theta')\boldsymbol{R}_\mathrm{p}^{-1}\mathbf{a}_\mathrm{p}(\theta')} \tag{3.28}$$

$$= \frac{\sigma_\eta^2 M(\sigma_\mathrm{s}^2 + \sigma_\eta^2)}{\mathbf{a}_p^H(\theta')\left(\left(\sigma_\eta^2 + \sigma_\mathrm{s}^2 M\right) I_M - \sigma_\mathrm{s}^2 \mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta)\right)\mathbf{a}_\mathrm{p}(\theta')} \tag{3.29}$$

$$= \frac{\sigma_\eta^2 M(\sigma_\mathrm{s}^2 + \sigma_\eta^2)}{M\left(\sigma_\eta^2 + \sigma_\mathrm{s}^2 M\right) - \sigma_\mathrm{s}^2 \mathbf{a}_\mathrm{p}^H(\theta')\mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta)\mathbf{a}_\mathrm{p}(\theta')}. \tag{3.30}$$

Recall that θ denotes the real DOA, while θ' is the variable over which the grid search is performed. Note the following lemma:

Lemma 3.1.

$$\mathbf{a}_\mathrm{p}^H(\theta')\mathbf{a}_\mathrm{p}(\theta)\mathbf{a}_\mathrm{p}^H(\theta)\mathbf{a}_\mathrm{p}(\theta') = M + 2\sum_{i=1}^{M-1}(M-i)\cos\left[iq(\theta,\theta')\right], \tag{3.31}$$

where we defined $q(\theta,\theta') = kd\left(\cos(\theta) - \cos(\theta')\right)$.

Proof. See Appendix C. ■

Now, we find the MVDR output to be:

$$P_{MVDR} = \frac{\sigma_\eta^2 M(\sigma_\mathrm{s}^2 M + \sigma_\eta^2)}{M\left(\sigma_\eta^2 + \sigma_\mathrm{s}^2 M\right) - \sigma_\mathrm{s}^2\left(M + 2\sum_{i=1}^{M-1}(M-i)\cos\left[iq(\theta,\theta')\right]\right)}. \tag{3.32}$$

This spectrum is valid when a single source with DOA θ impinges a perfectly synchronized array. In the following we will derive the MVDR output in a similar fashion for

a single source impinging an array containing clock errors. By using the model given in (3.22). We can write the received data as:

$$\boldsymbol{Y}_{\mathrm{p}} = \boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}\mathbf{s}^{H} + \boldsymbol{N}. \tag{3.33}$$

Recall that $\boldsymbol{\Gamma}$ is a diagonal matrix containing the clock errors. Computing the new autocorrelation in a similar fashion as before leads to:

$$\boldsymbol{R}_{\mathrm{p}} = \sigma_{\mathrm{s}}^2\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)\mathbf{a}_{\mathrm{p}}^{H}(\theta)\boldsymbol{\Gamma}^{H} + \sigma_{\eta}^2\boldsymbol{I}_M. \tag{3.34}$$

Using the Sherman-Morrison matrix inversion lemma again, while noting that $\boldsymbol{\Gamma}^{H}\boldsymbol{\Gamma} = \boldsymbol{I}_M$, we find the inverse of the correlation matrix to be:

$$\boldsymbol{R}_{\mathrm{p}}^{-1} = \sigma_{\eta}^{-2} - \frac{\sigma_{\eta}^{-4}\sigma_{\mathrm{s}}^2\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)\mathbf{a}_{\mathrm{p}}^{H}(\theta)\boldsymbol{\Gamma}^{H}}{1+\sigma_{\eta}^{-2}\sigma_{\mathrm{s}}^2\mathbf{a}_{\mathrm{p}}^{H}(\theta)\boldsymbol{\Gamma}^{H}\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)} = \frac{\left(\sigma_{\eta}^2+\sigma_{\mathrm{s}}^2M\right)\boldsymbol{I}_M - \sigma_{\mathrm{s}}^2\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)\mathbf{a}_{\mathrm{p}}^{H}(\theta)\boldsymbol{\Gamma}^{H}}{\sigma_{\eta}^2\left(\sigma_{\eta}^2+\sigma_{\mathrm{s}}^2M\right)}. \tag{3.35}$$

Substituting this in the MVDR equation (2.14):

$$\begin{aligned} P_{MVDR} &= \frac{1}{\mathbf{a}_{\mathrm{p}}^{H}(\theta')\boldsymbol{R}_{\boldsymbol{Y}}^{-1}\mathbf{a}_{\mathrm{p}}(\theta')} \\ &= \frac{\sigma_{\eta}^2M(\sigma_{\mathrm{s}}^2+\sigma_{\eta}^2)}{M\left(\sigma_{\eta}^2+\sigma_{\mathrm{s}}^2M\right) - \sigma_{\mathrm{s}}^2\left|\mathbf{a}_{\mathrm{p}}^{H}(\theta')\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)\right|^2}. \end{aligned} \tag{3.36}$$

The last term in the denumerator of previous expression can be rewritten as:

$$\sigma_{\mathrm{s}}^2\left|\mathbf{a}_{\mathrm{p}}^{H}(\theta')\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)\right|^2 = \sigma_{\mathrm{s}}^2\left|\sum_{i=0}^{M-1}e^{\mathrm{j}iq(\theta,\theta')+\mathrm{j}kc\gamma_{i+1}}\right|^2 \leq M^2, \tag{3.37}$$

Substituting (3.37) in (3.36) leads to:

$$P_{MVDR} = \frac{\sigma_{\eta}^2M(\sigma_{\mathrm{s}}^2+\sigma_{\eta}^2)}{M\left(\sigma_{\eta}^2+\sigma_{\mathrm{s}}^2M\right) - \sigma_{\mathrm{s}}^2\left|\sum_{i=0}^{M-1}e^{\mathrm{j}iq(\theta,\theta')+\mathrm{j}kc\gamma_{i+1}}\right|^2}. \tag{3.38}$$

It is easy to observe that the only difference between (3.32) and (3.38) is the bias term in the exponential in the latter equation. It is clear that the θ' that maximizes the MVDR is the one that maximizes the absolute sum of exponentials in (3.38). The θ' that maximizes the absolute sum can be approximated, for small γs, by the θ' that minimizes the difference between the different arguments of the complex exponentials i.e. the θ' for which the arguments are closest to a consensus C in an absolute sense:

$$\hat{\theta} = \arg\max_{\theta'} P_{MVDR} \approx \arg\min_{\theta',C}\sum_{i=0}^{M-1}|iq(\theta,\theta')+kc\gamma_{i+1}-C| = \arg\min_{\theta',C}||\mathbf{f}(\theta',C)||_2^2, \tag{3.39}$$

where $\hat{\theta}$ is the estimated DOA, moreover we defined $\mathbf{f}(\theta',C)\in\mathbb{R}^M$, whose entries are given by

$$\mathbf{f}_i(\theta',C) = (i-1)q(\theta,\theta') + kc\gamma_i - C. \tag{3.40}$$

Note that the approximation given in (3.39) is only valid when the clock errors are relatively small, some simulations supporting this claim are given in Appendix D. If we derive the derivative of the cost function in (3.39) with respect to C we find:

$$\nabla_C \left(\mathbf{f}^T(\theta', C)\mathbf{f}(\theta', C) \right) = -\frac{M(M-1)}{2} q(\theta, \theta') + MC - kc \sum_{i=1}^{M} \gamma_i. \tag{3.41}$$

Equating this to zero and solving for C leads to

$$C = \frac{M-1}{2} q(\theta, \theta') + \frac{kc}{M} \sum_{i=1}^{M} \gamma_i. \tag{3.42}$$

Substituting this in (3.40) removes the dependency on C:

$$\mathbf{f}_i(\theta') = \left(i - \frac{M+1}{2} \right) q(\theta, \theta') + kc\gamma_i - \frac{kc}{M} \sum_{j=1;}^{M} \gamma_j. \tag{3.43}$$

Now, computing the derivative of the cost function in (3.39) with respect to θ', equating to zero and recalling $q(\theta, \theta') = kd(\cos\theta - \cos\theta')$ will result in:

$$\nabla_{\theta'} \left(\mathbf{f}^T(\theta')\mathbf{f}(\theta') \right) = 2kd \sin\theta' \sum_{i=1}^{M} \left[\left(i - \frac{M+1}{2} \right) \left(\left(i - \frac{M+1}{2} \right) q(\theta, \theta') + kc\gamma_i - \frac{kc}{M} \sum_{j=1}^{M} \gamma_j \right) \right] = 0. \tag{3.44}$$

Note that:

$$\sum_{i=1}^{M} \left[\left(i - \frac{M+1}{2} \right) \sum_{j=1}^{M} \gamma_j \right] = 0. \tag{3.45}$$

Substituting above expression into (3.44), while discarding the $\sin\theta'$ factor results in:

$$\frac{M\left(M^2-1\right)}{12} q(\theta, \theta') + kc \sum_{i=1}^{M} \left(i - \frac{M+1}{2} \right) \gamma_i = 0. \tag{3.46}$$

Note that $\theta' = 0°$ and $\theta' = 180°$ are solutions of (3.44) which we discarded in (3.46) both of these solutions will result in a maximum in the cost function and therefore the final result is not influenced by discarding them. Solving the latter for $\cos\theta'$ will lead to:

$$\cos\hat{\theta}' = \cos\theta + \frac{12c}{dM(M^2-1)} \sum_{i=1}^{M} \left(i - \frac{M+1}{2} \right) \gamma_i = \cos\theta + b_{cos}(\boldsymbol{\gamma}), \tag{3.47}$$

where we defined

$$b_{cos}(\boldsymbol{\gamma}) = \frac{12c}{dM(M^2-1)} \sum_{i=1}^{M} \left(i - \frac{M+1}{2} \right) \gamma_i \tag{3.48}$$

to denote the bias on the cosine of the estimated DOA caused by the clock errors $\boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 & \gamma_2 & \dots & \gamma_M \end{bmatrix}^T$. The result in (3.47) has a few interesting properties:

- The bias solely depends on c/d i.e. the inverse distance between the sensors with respect to the speed of sound. Thus the further we place sensors from each other the smaller our error will be. However, due that d is a design variable which should be chosen based on the frequency, it follows that the bias is indirectly dependent on the frequency. Also note that the approximation in (3.39) is not valid for high frequencies.

- The bias works in the cosine space of the estimate $\cos(\hat{\theta})$, thus the resulting absolute error $|\hat{\theta} - \theta|$ will be bigger around $\theta = 0°$ or $\theta = 180°$ and small around $\theta = 90°$.

- The bias induced by a sensor depends not only on the clock but also the location of the sensor. For example the sensor in the center of an odd numbered ULA does not introduce any error on the estimate regardless of the clock. On the other hand sensors placed on the ends of the ULA will introduce the biggest errors.

- Although the bias decreases when the amount of sensors increases asymptoticly with a rate of $1/M$, this is mainly due to the bigger size of the array. If we decrease the sensor distance d while increasing the total amount of sensors M, such that the total array size dM stays constant. We will observe an asymptoticly decay of $M/(M+1)$ thus the gain for adding more sensors while not increasing the total size of the array is neglectable for big M. However, increasing the total size of the array (with or without adding new sensors) does decrease the bias. In fact the bias decreases with $1/(dM)$.

- The bias of the estimate does not depend on the signal or noise power in the single source scenario.

The nature of the bias can be explained with the schematic overview in Figure 3.1. In this figure three APSs are deployed in a ULA, the two most left sensors have perfect clock while the most right sensor is lagging behind. A farfield source is emitting a plane wave towards the sensors. The two most left sensors will "see" the waves correct while the right sensor will have a delay. The complete array will see the wave as the dotted line in the figure. It is imaginable that the eventual estimate of θ will be biased anti-clockwise.

In Figure 3.2 various Mean Square Errors (MSE: $1/K \sum_{k=1}^{K} (\theta'_k - \theta)^2$, with K the number of realizations) are plotted against the real DOA (recall that we took $N \to \infty$ snapshots). The amount of sensors used are varying ($M = 2, 3, 5, 7$) and the clock errors are created using a Gaussian process with different standard deviations ($\sigma_\gamma = 1, 10, 100, 1000\mu$s), $K = 1000$ instants are computed and the MSE is given in the plots in Figure 3.2. It is observable that the DOA estimation for $\sigma_\gamma = 1$ and 10μs are acceptable (neglecting the endfire) regardless of the amount of sensors. Furthermore. the maximum error for $\sigma_\gamma = 100\mu$s stays low for the seven sensor setup (MSE $< 1°$ within the $40° < \theta < 140°$ interval. Note that our approximation in (3.39) is not valid at the endfire of the array. Therefore the values outside said interval might not be valid. Increasing σ_γ to 1ms results into bad performance for all setups but it is clear

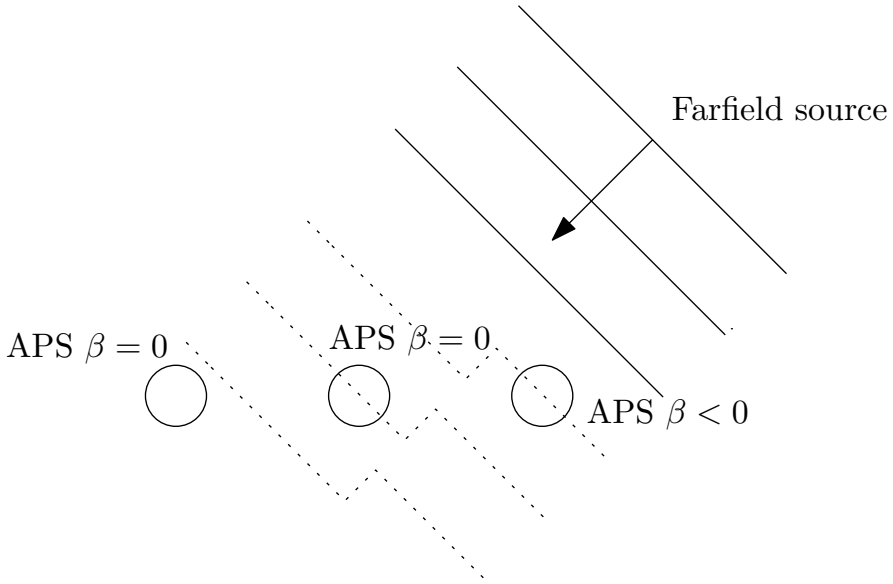


Figure 3.1: Schematic overview on the effects of clock errors

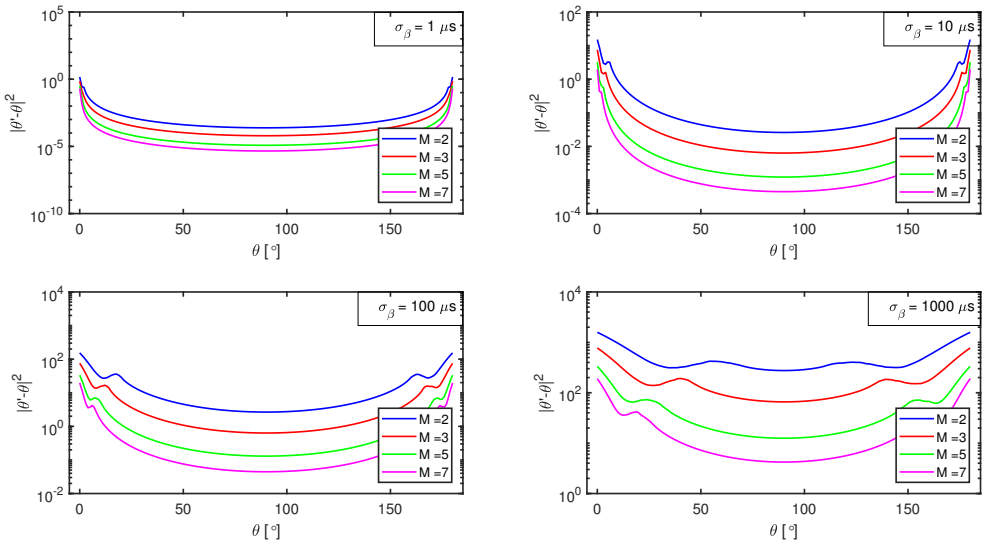


Figure 3.2: A few examples of DOA estimation errors caused by static clock errors for an ULA, with sensor distance $d = 0.5$m.

that the amount of sensors (i.e. the aperture of the array) has a big influence on the DOA estimation.

Figure 3.3 shows the MSE due to static clock errors with $\sigma_\gamma = 100\mu s$, instead of keeping the same sensor distance, while adding sensors. The total size of the array-aperture is kept constant at 5 meter (i.e. $dM = 5$m). Observe that adding a third sensor to a two sensor array will improve the DOA estimation significantly, while for

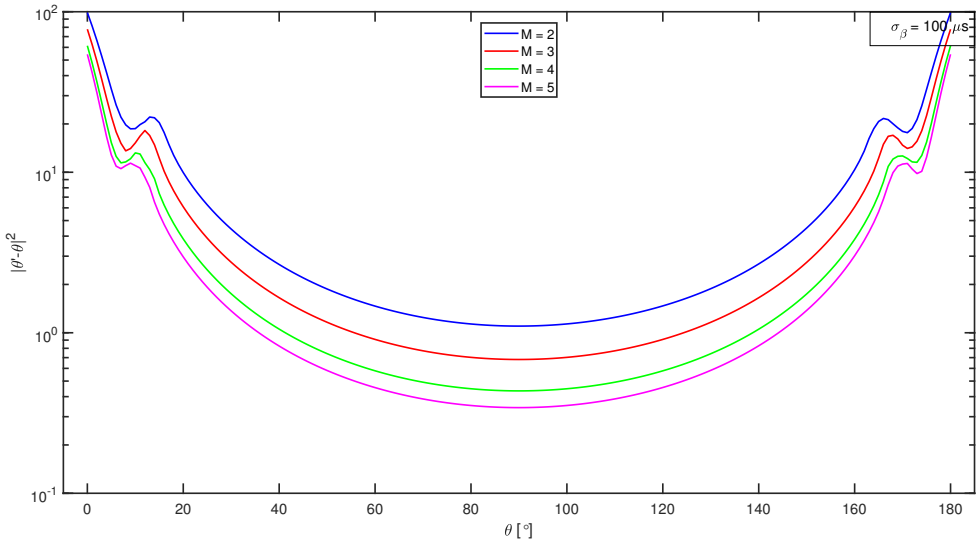


Figure 3.3: A few examples of DOA estimation errors caused by clock errors for an ULA with a different amount of sensors with the same total array length ($dM = 5$m.

each extra sensor added after that the improvement gets less.

It should be noted that for the case with one source, the DOA approximation and their errors due to clock-synchronization faults are equal for the classical beamformer, MVDR and MUSIC. As discussed in Section 2.2, when multiple sources are impinging the array their performance may vary. It is argued that the classical beamformer is more robust against small clock errors when more signals are present. This is due to that this method directly uses a projection on the steeringvector $\mathbf{a}(\theta)$, while the other two methods take the inverse of a projection over the steeringvector, which amplifies the errors in this vector.

3.3.2 Acoustic Vector Sensor Array

Up to this point, we analyzed the behaviour of the MVDR beamformer when a single signal is impinging an APS-array that is exposed to clock errors. In the remainder of this chapter, the behaviour of the MVDR while using an AVS-array is discussed. We can add static clock errors in the 1-source AVS model by replacing the steering vector $\mathbf{a}(\theta)$ by:

$$\mathbf{a}(\theta, \mathbf{\Gamma}) = (\mathbf{\Gamma}\mathbf{a}_{\mathrm{p}}(\theta)) \otimes \mathbf{h}(\theta). \tag{3.49}$$

Note that the MVDR power received by an AVS ULA can be derived in almost exactly the same fashion as for the APS ULA. Doing so leads to:

$$P_{MVDR} = \frac{\sigma_\eta^2 M(2\sigma_{\mathrm{s}}^2 + \sigma_\eta^2)}{M\left(\sigma_\eta^2 + 2\sigma_{\mathrm{s}}^2 M\right) - \sigma_{\mathrm{s}}^2 \left|\mathbf{h}^T(\theta')\mathbf{h}(\theta) \sum_{i=0}^{M-1} e^{\mathrm{j}iq(\theta,\theta') + \mathrm{j}kc\gamma_{i+1}}\right|^2}. \tag{3.50}$$

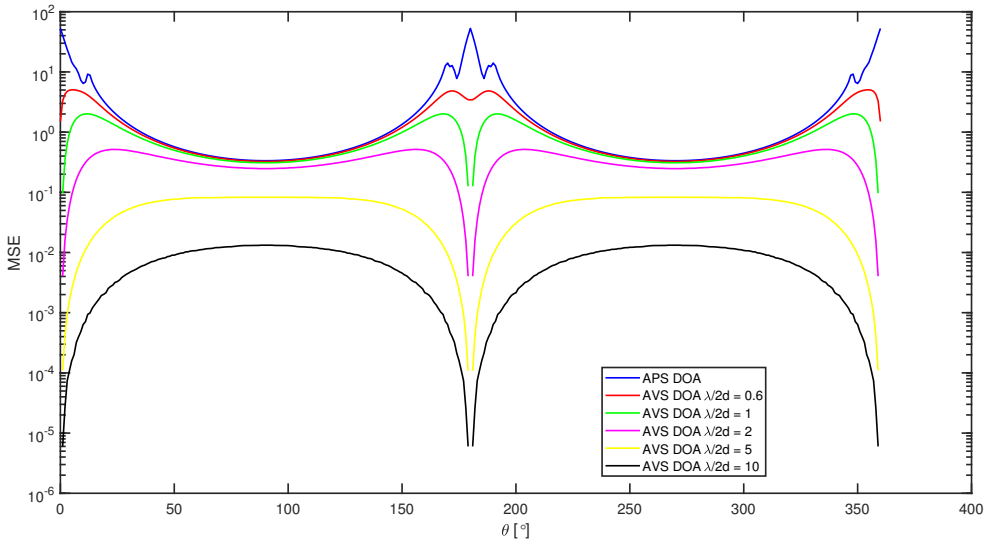


Figure 3.4: MSE for APS and AVS with M = 4, $\sigma_\gamma = 100\mu$s, $\Delta\theta' = 0.01°$, $d = 1,5$m and $K = 100$.

Computing the estimated θ' is in this case not as straightforward as for the APSs and will therefore not be evaluated in this work. It is clear that the AVS has a gain in comparison with the APS, this gain is due to the $\mathbf{h}^T(\theta')\mathbf{h}(\theta)$ factor, also denoted as Velocity Gain Modulation (VGM) in [3]. Note that we can find the θ' for which the MVDR maximizes as follow:

$$\hat{\theta}' = \arg\max_{\theta'} P_{MVDR} = \arg\max_{\theta'} \left| \mathbf{h}^T(\theta')\mathbf{h}(\theta) \sum_{i=0}^{M-1} e^{\mathrm{j}iq(\theta,\theta')+\mathrm{j}kc\gamma_{i+1}} \right|^2 . \tag{3.51}$$

We can visualize this by providing a grid search over θ'. Figure 3.4 shows MSE errors for a setting with $\sigma_\gamma = 1000\mu s$ and $M = 4$ for an APS array and multiple AVS arrays with changing wavelengths. Note that a wavelength of $\frac{\lambda}{2d} = 1$ corresponds with Nyquist spatial sampling rate, longer wavelengths with an oversampled sampling rate and shorter wavelengths with undersampled sampling rates.

The figure shows that if the sensors spatially sample at nyquist rate or higher our MSE disappears at the endfire which means that the MVDR is an asymptotic unbiased estimator for $\theta \in [0°, 180°]$. This can be explained by placing the source in Figure 3.1 in the endfire like in Figure 3.5, where the normal lines are the real waves and the dotted the waves as seen by the array. Although the dotted waves are not completely the same as the real waves it is obvious that the DOA estimation is not influenced.

Furthermore, we observe in Figure 3.4 that if the wavelength decreases that eventually a MSE at the endfire appears.

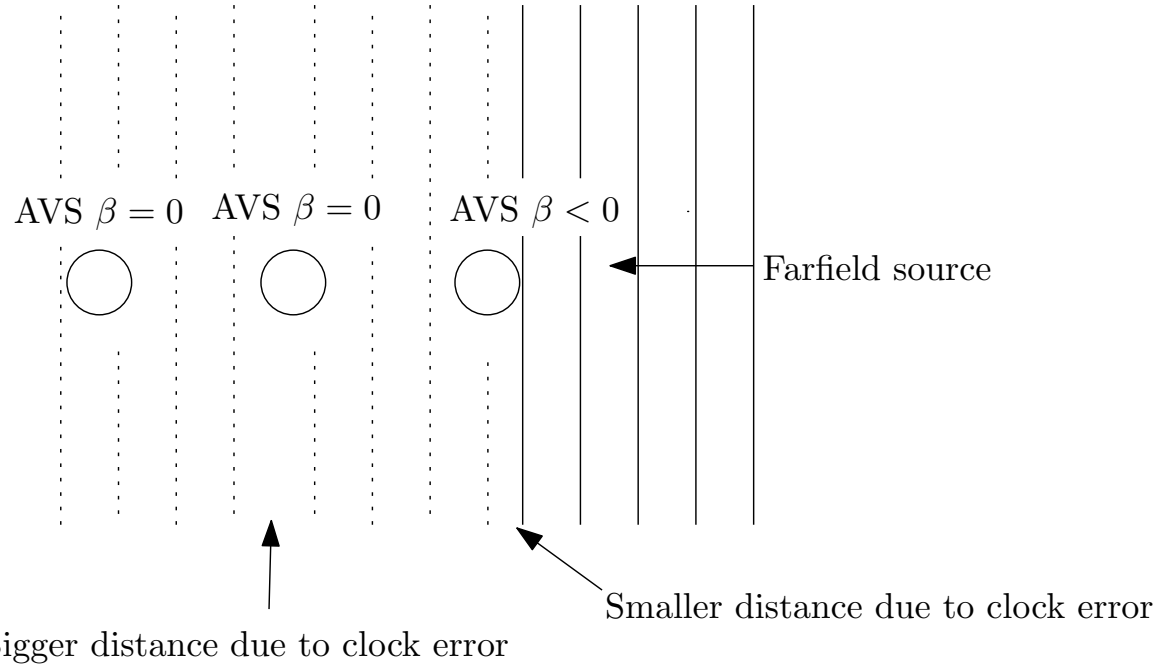


Figure 3.5: MSE for APS and AVS with M = 4, $\sigma_\gamma = 100\mu s$, $\Delta\theta' = 0.01°$, $d = 1,5m$, speed of sound $c = 340m/s$ and $K = 100$

Robust Direction of Arrival Estimation

4

In this chapter we will derive methods for robust DOA estimation in the case the AVS sensor data is subject to synchronization issues. In the first section, robust beamforming approaches are discussed. After that, it is made clear why these approaches are not appropriate for this problem. In Section 4.2 and Section 4.3 two clock independent eigenstructure methods are introduced. In Section 4.4 an iterative approach to solve the synchronization problem is proposed.

4.1 Robust beamforming approaches

It is well known that beamformers are designed to reconstruct a signal send over a wireless channel. These beamformers can easily be changed into a DOA estimation algorithm. By performing a search over different angles, changing the modeled array aperture. Taking the angle with the maximum output, results in an estimate of the direction from which the most power is approaching the array. With this in mind, it is clear that a robust beamformer is not necessary to be robust in terms of DOA estimation.

There are several beamformers developed for the case where the steering vector is not exactly known (e.g. if the measurements are affected by clock errors). One of these robust beamformers is in literature known as Vorobyov's beamformer, first introduced in [17] and developed by Vorobyov, Gershman and Luo. This technique is shortly explained in this section. From this example, it becomes obvious why it is not appropriate to use robust beamforming approaches to perform robust DOA estimation.

Vorobyov et al. assumes that the total error on the steering vector $\mathbf{a}_\epsilon(\theta) - \mathbf{a}(\theta)$ (e.g. phase and gain errors) can be bounded by a constant $\epsilon > 0$:

$$||(\mathbf{a}_\epsilon(\theta) - \mathbf{a}(\theta)|| \leq \epsilon. \tag{4.1}$$

Which directly implies that the actual steering vector $\mathbf{a}_\epsilon(\theta)$ lies within the following set:

$$\mathbf{a}_\epsilon(\theta) \in \mathcal{A}(\theta, \epsilon) = \{\mathbf{c} | \mathbf{c} = \mathbf{a}(\theta) + \mathbf{e}, ||\mathbf{e}|| \leq \epsilon\}. \tag{4.2}$$

Now, the new beamformer is defined similar to the MVDR beamformer. Recall the definition of the MVDR beamformer (2.12):

$$\mathbf{w}(\theta) = \min_{\mathbf{w}'(\theta)} \mathbf{w}'^H(\theta)\hat{\boldsymbol{R}}\mathbf{w}'(\theta) \qquad \text{s.t. } \mathbf{w}'^H(\theta)\mathbf{a} = 1. \tag{4.3}$$

Where the MVDR keeps the power within the θ-direction equal to one, the new, robust beamformer keeps the power for all vectors that lie within $\mathcal{A}(\theta, \epsilon)$ to have an absolute value of at least one:

$$\mathbf{w}(\theta) = \arg\min_{\mathbf{w}'(\theta)} \mathbf{w}'^H(\theta)\hat{\boldsymbol{R}}\mathbf{w}'(\theta) \qquad \text{s.t. } |\mathbf{w}'^H(\theta)\mathbf{c}| \geq 1 \ \forall \mathbf{c} \in \mathcal{A}(\theta, \epsilon). \tag{4.4}$$

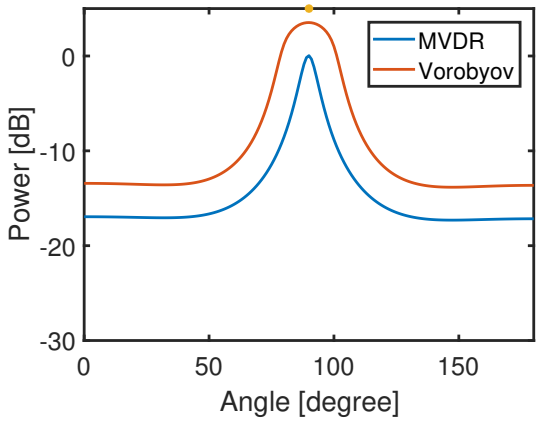


Figure 4.1: Spatial spectra for the MVDR and Vorobyov beamformer.

Although this problems seems to be non-convex due to its constraint, it is proven by [17] that (4.4) can be written as a convex second order cone program and therefore can be solved relatively efficient.

In Figure 4.1 spatial spectra of the MVDR and of Vorobyov's method are shown. There is one signal present which has a DOA of 90° and no clock errors are present. As can be seen in the figure Vorobyovs method has an offset and widens the mainlobe of the MVDR. Due to this widening of the mainlobe the reconstruction of the signal becomes less sensitive to minor errors on the steering vector. Furthermore, when we look at the definition of Vorobyovs method (4.4) that the the power of Vorobyovs method at θ is a summation of the MVDR at θ and the region around it bounded by ϵ. Thus, when the MVDR method gives a symmetric response as in this example the estimate for the DOA will be the same for MVDR and Vorobyovs method. When the MVDR is not symmetric there is no guarantee that the DOA estimate obtained through Vorobyovs method will move in the correct direction.

4.2 Clock Independent Eigenstructure Method

In [18], Liu et al. propose a DOA estimation for an APS-array which is independent of phase errors. This method is based on the Hadamard product between the measurement vector and its complex conjugate. It uses the resulting vector as the new measurement vector on which a MUSIC-like algorithm is performed. The new measurement vector can be written as:

$$\tilde{\mathbf{y}}_{\mathrm{p}}(t) = \mathbf{y}_{\mathrm{p}}(t) \odot \mathbf{y}_{\mathrm{p}}^{*}(t) \in \mathbb{R}^{M}, \tag{4.5}$$

where $\odot$ denotes the Hadamard product.

In this section, firstly, the method developed by Liu et al. is reproduced briefly. Secondly, this method will be expand for usage with AVSs, so that it becomes clear what the differences and advantageous are between the two methods.

4.2.1 Acoustic Pressure Sensor Array

The method developed by Liu defines a covariance matrix based on the new measurement vector $\tilde{\mathbf{y}}_p(t)$:

$$\tilde{\boldsymbol{R}}_{\mathrm{p}} = \mathbb{E}\left[\tilde{\mathbf{y}}_{\mathrm{p}}(t)\tilde{\mathbf{y}}_{\mathrm{p}}^{H}(t)\right]. \tag{4.6}$$

Now, we will investigate the new covariance matrix in the case of two sources.

Theorem 4.1. *When there are two sources impinging the array we could expand $\tilde{\boldsymbol{R}}_p$ as follows:*

$$\tilde{\boldsymbol{R}}_{\mathrm{p}} = \boldsymbol{C}_{\mathrm{p}}\boldsymbol{R}_{\tilde{s}_{\mathrm{p}}}\boldsymbol{C}_{\mathrm{p}}^{T} + \left(2\left(\sigma_{\mathrm{s}1}^2 + \sigma_{\mathrm{s}2}^2\right)\sigma_{\eta}^2 + \sigma_{\eta}^4\right)\boldsymbol{I}, \tag{4.7}$$

where

$$\boldsymbol{C}_{\mathrm{p}} = \begin{bmatrix}\mathbf{1} & \Re\left\{\mathbf{a}_{\mathrm{p}}(\theta_1)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta_2)\right\} & \Im\left\{\mathbf{a}_{\mathrm{p}}(\theta_1)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta_2)\right\} & \mathbf{1}\end{bmatrix}, \tag{4.8}$$

$$\mathbf{1} = \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^{T}, \tag{4.9}$$

$$\boldsymbol{R}_{\tilde{s}_{\mathrm{p}}} = \mathbb{E}\begin{bmatrix}\tilde{\mathbf{s}}_{\mathrm{p}}\tilde{\mathbf{s}}_{\mathrm{p}}^{H} & 0\\ 0 & 2\sigma_{\eta}^2\left(\sigma_{\mathrm{s}1}^2+\sigma_{\mathrm{s}2}^2\right)+\sigma_{\eta}^4\end{bmatrix}, \tag{4.10}$$

$$\tilde{\mathbf{s}}_{\mathrm{p}} = \begin{bmatrix}s_1 s_1^{*} + s_2 s_2^{*} & 2\Re\left\{s_1 s_2^{*}\right\} & -2\Im\left\{s_1 s_2^{*}\right\}\end{bmatrix}^{T}. \tag{4.11}$$

Proof. See [18]. ■

If we compare this with the equation for the covariance of the normal measurement vector, we observe certain similarities. The matrix $\boldsymbol{R}_{\tilde{\mathbf{s}}_{\mathrm{p}}}$ could be seen as the signal correlation matrix ($\boldsymbol{R}_s$) and $\boldsymbol{C}_{\mathrm{p}}$ as the matrix containing the steering vectors. With this in mind it is easy to develop a MUSIC-like algorithm. Define the eigenvalue decomposition of $\tilde{\boldsymbol{R}}_{\mathrm{p}}$ as follows:

$$\tilde{\boldsymbol{R}}_{\mathrm{p}} = \sum_{m=1}^{M}\beta_{\tilde{\boldsymbol{R}}_{\mathrm{p},m}}\mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},m}}\mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},m}}^{H} \tag{4.12}$$

where $\left\{\beta_{\tilde{\boldsymbol{R}}_{\mathrm{p},m}}\right\}_{m=1}^{M}$ are the eigenvalues of $\tilde{\boldsymbol{R}}_{\mathrm{p}}$ with corresponding eigenvectors $\left\{\mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},m}}\right\}_{m=1}^{M}$. By analyzing $\boldsymbol{C}_{\mathrm{p}}$, we find that the signal subspace is spanned by three vectors, such that:

$$\operatorname{span}\left\{\mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},1}}, \mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},2}}, \mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},3}}\right\} = \operatorname{span}\left\{\mathbf{1}, \Re\left\{\mathbf{a}_{\mathrm{p}}(\theta_1)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta_2)\right\}, \Im\left\{\mathbf{a}_{\mathrm{p}}(\theta_1)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta_2)\right\}\right\}. \tag{4.13}$$

Now, a two dimensional spatial spectrum is defined based on the signal subspace above:

$$P_{\mathrm{p}}(\theta,\theta') = \left(||\tilde{\boldsymbol{U}}_{\tilde{\boldsymbol{R}}_{\mathrm{p}}}^{H}\Re\left\{\mathbf{a}_{\mathrm{p}}(\theta)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta')\right\}||_2^2 + ||\tilde{\boldsymbol{U}}_{\tilde{\boldsymbol{R}}_{\mathrm{p}}}^{H}\Im\left\{\mathbf{a}_{\mathrm{p}}(\theta)\odot\mathbf{a}_{\mathrm{p}}^{*}(\theta')\right\}||_2^2\right)^{-1}, \tag{4.14}$$

where $\tilde{\boldsymbol{U}}_{\tilde{\boldsymbol{R}}_{\mathrm{p}}}\begin{bmatrix}\mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},4}} & \mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},5}} & \dots & \mathbf{u}_{\tilde{\boldsymbol{R}}_{\mathrm{p},5}}\end{bmatrix}$. The DOAs can be estimated by searching for the peaks in the spectrum defined by $P_{\mathrm{p}}(\theta,\theta')$:

$$\left\{\hat{\theta}_1, \hat{\theta}_2\right\} = \arg\max_{\theta,\theta'} P_{\mathrm{p}}(\theta,\theta'), \tag{4.15}$$

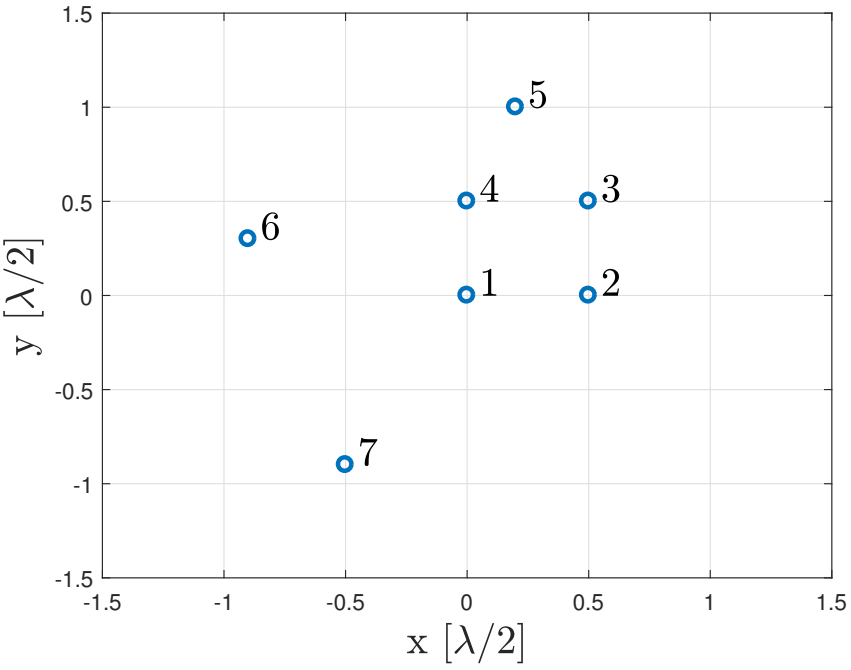


Figure 4.2: The used array configuration.

where $\hat{\theta}_i$ are the estimates for the DOAs.

From [18] it follows that the DOA estimates acquired by this method are unambiguous if the following statements hold:

- The array consists out of at least four sensors and are placed such that four sensors are located on the four vertices of a square with sides of no more than $\frac{\lambda}{4}$.

- The DOAs are located within $[0°, 180°]$.

Liu et al. propose an array-configuration that satisfies above conditions. We will use the same configuration so that it is easy to compare the different algorithms. The array-configuration is given in Figure 4.2.

The above is simulated such that the two dimensional spectrum $P_{\mathrm{p}}(\theta, \theta')$ is obtained. For this simulation the discussed array is used. Furthermore two signals with DOAs of $\theta_1 = 30°$ and $\theta_2 = 130°$ are impinging the setup, the signals have equal source power with a SNR of 12dB and a frequency of 1 kHz. The sensors use a sampling frequency of 16 kHz and 500 sample points are used. Moreover the clocks of the seven sensors are

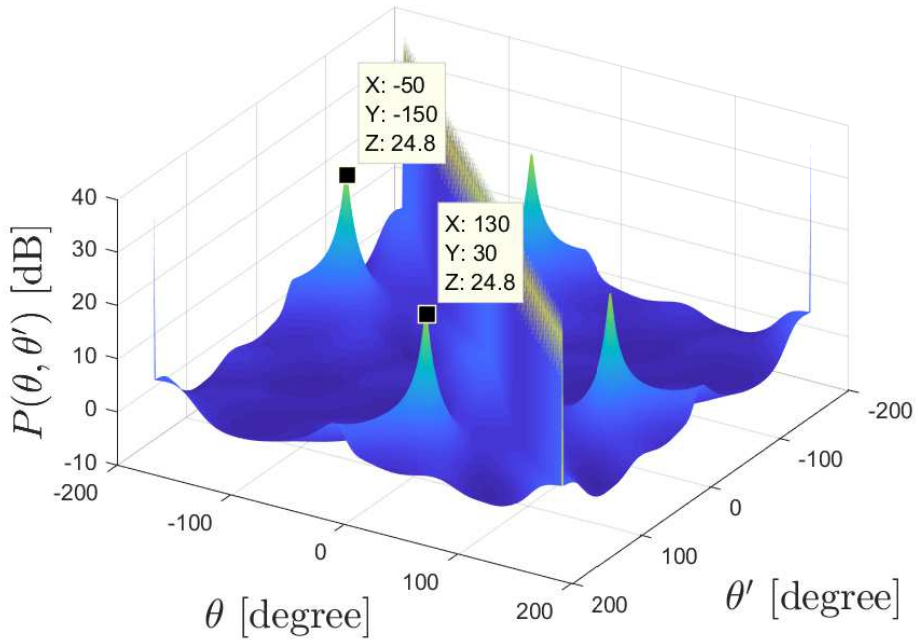


Figure 4.3: Results using the method proposed by Liu et al. in [18] the peaks are located at (-50,-150), (-150,-50), (30,130) and (130,30)

simulated with the following properties:

$$\boldsymbol{\omega} = \left(10^6 + \begin{bmatrix} 34.3 \\ 357.8 \\ 276.9 \\ -135.0 \\ 303.5 \\ 72.5 \\ -6.3 \end{bmatrix} \right) [\mu s/s], \qquad \boldsymbol{\phi} = \begin{bmatrix} 107.5 \\ 366.8 \\ -451.8 \\ 172.4 \\ 63.8 \\ -261.5 \\ 86.7 \end{bmatrix} [\mu s].$$

From which the time-vector $\mathbf{t} = \boldsymbol{\omega} t + \boldsymbol{\phi}$ is extracted. Here t is the true time and $\mathbf{t}$ contains the times according to the sensors.

From Figure 4.3 it is clear that we are able to estimate the DOAs if we assume that these are within the interval $[0°, 180°]$. However, it is not possible to retrieve the DOAs within the full interval $[-180°, 180°]$ due to the ambiguities at $-50°$ and $-150°$. The ambiguities follow from the following equality:

$$\mathbf{a}_{\mathrm{p}}(\theta) \odot \mathbf{a}_{\mathrm{p}}^*(\theta') = \mathbf{a}_{\mathrm{p}}(\theta' - 180°) \odot \mathbf{a}_{\mathrm{p}}^*(\theta - 180°) \tag{4.16}$$

Note that this equality is a direct result from the fact that the new steering vectors take the differences between two DOAs (i.e. $\cos(\theta) - \cos(\theta')$ and $\sin(\theta - \theta')$) instead of the direct cosines and sines. The difference of two DOAs have the following properties:

$$\cos\theta - \cos\theta' = \cos(\theta' - 180°) - \cos(\theta - 180°) \tag{4.17}$$

$$\sin\theta - \sin\theta' = \sin(\theta' - 180°) - \sin(\theta - 180°) \tag{4.18}$$

So one of the disadvantages of this method is the ambiguity in the interval $[-180°, 180°]$. Below the biggest disadvantages of this method are indicated:

1. The ambiguity within the interval $[-180°, 180°]$ described above.
2. As we have seen, the array configuration is fairly limited and it is depended on the frequency of the signals.
3. The $\mathbf{1}$-vector is always in the steering and vector the line $\theta = \theta'$ within $P_{\mathrm{p}}(\theta, \theta')$ will always be in the signal-subspace. This makes it impossible to find the DOA when only one signal is present. Which of course makes sense when APSs are used while the time references of the different sensors are unknown, there is simply not enough information available to find the DOA of one single source.
4. The new correlation matrix is defined as a fourth-order statistic (therefore the name correlation matrix might be incorrect), this may cause issues when the signals and the noise are not completely Gaussian distributed.
5. This method uses a two dimensional grid search, therefore the technique becomes computational expensive when compared with other algorithms.
6. The relative high amount of sensors that are needed. Where the common MUSIC method needs 1 more sensor than sources this methods needs for K sources $K(K-1)+1$ sensors [18] to estimate the DOAs.

Although these disadvantages might seem to make this method unsatisfying there is still one big advantage of this method. This one is namely completely independent of the errors in the phases of the signals and will thus work regardless of the accuracy of synchronization.

In the remaining parts of this chapter, we will try to find solutions for the disadvantages of Liu's method. In the next section, we will see that when we use an AVS-array and make a relative simple change to the algorithm we are able to tackle the first three disadvantages. Then in Section 4.3 a method proposed by Cao et al. in [19], which deals with the fourth disadvantage, is merged with our method. While the method developed by Cao et al. does not deal with the first three disadvantages, our proposed method deals with the first four disadvantages. Liu et al. propose in [18] a method to partly solve the fifth issue themselves, in Section 4.4 we simply alter this technique in order to use it in our proposed algorithm. This partly solves the fourth issue. How to solve the fifth and sixth issue of the method is not treated in this work, and therefore stays an open question.

4.2.2 Acoustic Vector Sensor Array

In this subsection, Liu's method described above will be changed in order to be suitable for an AVS array. Simply copying the mentioned method to an AVS-array (i.e. $\tilde{\mathbf{y}}(t) = \mathbf{y}(t) \odot \mathbf{y}^*(t)$) is not sufficient. As the multiplication will square the entries of the, by AVSs obtained, $\mathbf{h}(\theta)$-vector. One of the main advantages of an AVS-array over an APS-array is that the $\mathbf{h}(\theta)$-vector added by the AVS-system has an unique solution

within the interval $[-180°, 180°]$. By squaring the entries of $\mathbf{h}(\theta)$ we will be left with an ambiguity in this factor, namely:

$$\mathbf{h}^{\odot 2}(\theta) = \mathbf{h}^{\odot 2}(\theta - 180), \tag{4.19}$$

where $[\cdot]^{\odot i}$ denotes the entrywise i^{th} power of $[\cdot]$. By recalling the ambiguity for Liu's method we find that the two ambiguities to be equal. It is clear that this behavior is unwanted. In this section we will redefine the method proposed by Liu et al. for an AVS-array that keeps this uniqueness. As the clock errors are assumed to be the same for all channels on a sensor (i.e. the phase error of the pressure channel is equal to that of the velocity channels of that same sensor). It is easy to see that multiplying a velocity channel with the complex conjugate of its corresponding pressure channel we end up with a result that is independent of clock errors. Doing so leads to:

$$\tilde{\mathbf{y}}(t) = \mathbf{y}(t) \odot \left(\mathbf{y}_{\mathrm{p}}^{*}(t) \otimes \mathbf{1}_3\right) \in \mathbb{R}^{3M}. \tag{4.20}$$

Firstly, $\tilde{\mathbf{y}}(t)$ is investigated in the case of two sources and a correlation matrix ($\tilde{\boldsymbol{R}}$) based on this vector will be defined. Secondly, the obtained correlation matrix will be extended to the more general case with an arbitrary number of sources. By investigating this correlation matrix a DOA estimation method is proposed. It is likely to be not robust when certain assumptions that are made might not be completely true. This is discussed in the end of this section which leads to an intuitive interpretation of the method that will be proposed in the next section.

Finding the new autocorrelation matrix

Note that the actual steering vector $\mathbf{a}_{\epsilon}(\theta)$ could be written as:

$$\mathbf{a}_{\epsilon}(\theta) = \boldsymbol{\Pi}\mathbf{a}(\theta), \tag{4.21}$$

where $\boldsymbol{\Pi} = \boldsymbol{\Gamma} \otimes \boldsymbol{I}_3$. Now we can write (4.20) in the case of two sources as follows:

$$\tilde{\mathbf{y}}(t) = \left(\boldsymbol{\Pi}\mathbf{a}(\theta_1)s_1(t) + \boldsymbol{\Pi}\mathbf{a}(\theta_2)s_2(t) + \boldsymbol{\eta}\right) \odot \left[\left(\boldsymbol{\Gamma}^{*}\mathbf{a}_{\mathrm{p}}(\theta_1)^{*}s_1(t)^{*} + \boldsymbol{\Gamma}^{*}\mathbf{a}_{\mathrm{p}}^{*}(\theta_2)s_2(t)^{*} + \boldsymbol{\eta}_{\mathrm{p}}^{*}\right) \otimes \mathbf{1}_3\right], \tag{4.22}$$

where the entries of $\boldsymbol{\eta}_{\mathrm{p}}$ denote the noise on the pressure channels. By defining the following identities, we can write the current and following equations in a more compact form:

$$\mathbf{a}_i = \mathbf{a}(\theta_i), \tag{4.23}$$
$$\mathbf{a}_{\mathrm{p},i} = \mathbf{a}_{\mathrm{p}}(\theta_i), \tag{4.24}$$
$$\mathbf{h}_i = \mathbf{h}(\theta_i), \tag{4.25}$$
$$\boldsymbol{\xi}_i = \mathbf{1}_M \otimes \mathbf{h}_i. \tag{4.26}$$

Now, by denoting that

$$\boldsymbol{\Pi}\mathbf{a}_i s_i \odot \left(\boldsymbol{\Gamma}^{*}\mathbf{a}_{\mathrm{p},i}^{*}s_i^{*} \otimes \mathbf{1}_3\right) = \boldsymbol{\xi}_i s_i s_i^{*}. \tag{4.27}$$

We can expand (4.22) while dropping the explicit time dependency (t):

$$\begin{aligned}\tilde{\mathbf{y}} =& \boldsymbol{\xi}_1 s_1 s_1^* \\ &+ \boldsymbol{\xi}_2 s_2 s_2^* \\ &+ s_1 s_2^* \left(\mathbf{a}_{\mathrm{p},1} \odot \mathbf{a}_{\mathrm{p},2}^*\right) \otimes \mathbf{h}_1 \\ &+ s_1^* s_2 \left(\mathbf{a}_{\mathrm{p},1}^* \odot \mathbf{a}_{\mathrm{p},2}\right) \otimes \mathbf{h}_2 \\ &+ \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},1} \otimes \mathbf{h}_1\right) \odot \left(\boldsymbol{\eta}_{\mathrm{p}}^* \otimes \mathbf{1}_3\right) s_1 \\ &+ \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},2} \otimes \mathbf{h}_2\right) \odot \left(\boldsymbol{\eta}_{\mathrm{p}}^* \otimes \mathbf{1}_3\right) s_2 \\ &+ \boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^* \mathbf{a}_{\mathrm{p},1}^* \otimes \mathbf{1}_3\right) s_1^* \\ &+ \boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^* \mathbf{a}_{\mathrm{p},2}^* \otimes \mathbf{1}_3\right) s_2^* \\ &+ \boldsymbol{\eta} \odot \left(\boldsymbol{\eta}_{\mathrm{p}}^* \otimes \mathbf{1}_3\right) \end{aligned} \tag{4.28}$$

where we made use of the following property of the Hadamard and Kronecker product: $(\mathbf{a} \otimes \mathbf{b}) \odot (\mathbf{c} \otimes \mathbf{d}) = (\mathbf{a} \odot \mathbf{b}) \otimes (\mathbf{b} \odot \mathbf{d})$. Now we can divide the terms of this equation in different groups: The first four terms only contain the signal without any noise, the 5^{th} and 6^{th} term are a mixture of the AVS-array response and noise. The 7^{th} and 8^{th} term are mixtures of the APS-array response and noise and the last term contains noise only.

Now by focusing on the first four signal term we can define:

$$\boldsymbol{\alpha}_{i,j} = \left(\mathbf{a}_{\mathrm{p},i} \odot \mathbf{a}_{\mathrm{p},j}^*\right) \otimes \mathbf{h}_i, \tag{4.29}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \boldsymbol{\alpha}_{1,2} & \boldsymbol{\alpha}_{2,1} \end{bmatrix} \in \mathbb{C}^{3M \times 4}, \tag{4.30}$$

$$\tilde{\mathbf{s}} = \begin{bmatrix} s_1 s_1^* & s_2 s_2^* & s_1 s_2^* & s_1^* s_2 \end{bmatrix}^T \in \mathbb{C}^4. \tag{4.31}$$

Which enables us to write the first four terms as $\boldsymbol{C}\tilde{\mathbf{s}}$. Now we define:

$$\boldsymbol{\nu}_{\mathrm{v},1} = \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},1} \odot \boldsymbol{\eta}_{\mathrm{p}}^*\right) \otimes \mathbf{h}_1, \tag{4.32}$$

$$\boldsymbol{\nu}_{\mathrm{v},2} = \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},2} \odot \boldsymbol{\eta}_{\mathrm{p}}^*\right) \otimes \mathbf{h}_2, \tag{4.33}$$

$$\boldsymbol{\nu}_{\mathrm{v}} = \boldsymbol{\nu}_{\mathrm{v},1} s_1 + \boldsymbol{\nu}_{\mathrm{v},2} s_2. \tag{4.34}$$

It is clear that $\boldsymbol{\nu}_{\mathrm{v}}$ contains the mixture of the combined channels and the noise. Doing the same for the mixture of pressure channel and noise:

$$\boldsymbol{\nu}_{\mathrm{p},1} = \boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^* \mathbf{a}_{\mathrm{p},1}^* \otimes \mathbf{1}_3\right), \tag{4.35}$$

$$\boldsymbol{\nu}_{\mathrm{p},2} = \boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^* \mathbf{a}_{\mathrm{p},2}^* \otimes \mathbf{1}_3\right), \tag{4.36}$$

$$\boldsymbol{\nu}_{\mathrm{p}} = \boldsymbol{\nu}_{\mathrm{p},1} s_1^* + \boldsymbol{\nu}_{\mathrm{p},2} s_2^*. \tag{4.37}$$

The term that contains noise only can be written as:

$$\boldsymbol{\nu}_{\mathrm{n}} = \boldsymbol{\eta} \odot \left(\boldsymbol{\eta}_{\mathrm{p}}^* \otimes \mathbf{1}_3\right). \tag{4.38}$$

Now, (4.28) can be simplified as:

$$\tilde{\mathbf{y}} = \boldsymbol{C}\tilde{\mathbf{s}} + \boldsymbol{\nu}_{\mathrm{p}} + \boldsymbol{\nu}_{\mathrm{v}} + \boldsymbol{\nu}_{\mathrm{n}}. \tag{4.39}$$

Like the correlation matrix $\tilde{\boldsymbol{R}}_{\mathrm{p}}$ defined by Liu, we similarly define a correlation matrix $\tilde{\boldsymbol{R}}$ for the adjusted measurement data $\tilde{\mathbf{y}}$:

$$\tilde{\boldsymbol{R}} = \mathbb{E}\left[\tilde{\mathbf{y}}\tilde{\mathbf{y}}^H\right] \in \mathbb{R}^{3M\times 3M} \tag{4.40}$$

Theorem 4.2. *When there are two sources impinging the array we could expand $\tilde{\boldsymbol{R}}$ as follows:*

$$\begin{aligned}\tilde{\boldsymbol{R}} =& \boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^H + \sigma_\eta^2\left(\sigma_{\mathrm{s},1}^2\boldsymbol{\xi}_1 + \sigma_{\mathrm{s},2}^2\boldsymbol{\xi}_2\right)\mathbf{p}^T + \sigma_\eta^2\mathbf{p}\left(\sigma_{\mathrm{s},1}^2\boldsymbol{\xi}_1 + \sigma_{\mathrm{s},2}^2\boldsymbol{\xi}_2\right)^T \\ &+ \sigma_\eta^2\left(\sigma_{\mathrm{s},1}^2\left(\boldsymbol{H}_1 + \boldsymbol{I}\right) + \sigma_{\mathrm{s},2}^2\left(\boldsymbol{H}_2 + \boldsymbol{I}\right)\right) + \sigma_\eta^4\left(\boldsymbol{I} + \mathbf{p}\mathbf{p}^T\right)\end{aligned} \tag{4.41}$$

where:

$$\boldsymbol{R}_{\tilde{\mathbf{s}}} = \mathbb{E}\left[\tilde{\mathbf{s}}\tilde{\mathbf{s}}^H\right] = \begin{bmatrix} 2\sigma_{\mathrm{s},1}^4 & \sigma_{\mathrm{s},1}^2\sigma_{\mathrm{s},2}^2 & 0 & 0 \\ \sigma_{\mathrm{s},1}^2\sigma_{\mathrm{s},2}^2 & 2\sigma_{\mathrm{s},2}^4 & 0 & 0 \\ 0 & 0 & \sigma_{\mathrm{s},1}^2\sigma_{\mathrm{s},2}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\mathrm{s},1}^2\sigma_{\mathrm{s},2}^2 \end{bmatrix}, \tag{4.42}$$

$$\mathbf{p} = \mathbf{1}_M \otimes \mathbf{e}_1 \in \mathbb{R}^{3M}, \tag{4.43}$$

$$\boldsymbol{H}_i = \boldsymbol{I} \otimes \mathbf{h}_i\mathbf{h}_i^T, \tag{4.44}$$

$\sigma_{\mathrm{s},i}^2$ the power of the i^{th} source and $\mathbf{e}_i$ is the i^{th} column of the identity matrix of appropriate size.

Proof. See Appendix E ■

By substituting (4.42) into the first term of (4.41) we obtain:

$$\boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^H = 2\sigma_{\mathrm{s},1}^4\boldsymbol{\xi}_1\boldsymbol{\xi}_1 + 2\sigma_{\mathrm{s},2}^4\boldsymbol{\xi}_2\boldsymbol{\xi}_2 + \sigma_{\mathrm{s},1}^2\sigma_{s2}^2\left(\boldsymbol{\xi}_1\boldsymbol{\xi}_2^H + \boldsymbol{\xi}_2\boldsymbol{\xi}_1^H + \boldsymbol{\alpha}_{1,2}\boldsymbol{\alpha}_{1,2}^H + \boldsymbol{\alpha}_{2,1}\boldsymbol{\alpha}_{2,1}^H\right). \tag{4.45}$$

Note that we can write this in a more general form by denoting K sources:

$$\boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^H = 2\sum_{i=1}^{K}\sigma_{\mathrm{s},i}^4\boldsymbol{\xi}_i\boldsymbol{\xi}_i^H + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sigma_{\mathrm{s},i}^2\sigma_{sj}^2\boldsymbol{\xi}_i\boldsymbol{\xi}_j^H + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},j}^2\boldsymbol{\alpha}_{i,j}\boldsymbol{\alpha}_{i,j}^H. \tag{4.46}$$

From this point it is easy to see that we can expand (4.41) for multiple sources as follows :

$$\begin{aligned}\tilde{\boldsymbol{R}} =& 2\sum_{i=1}^{K}\sigma_{\mathrm{s},i}^4\boldsymbol{\xi}_i\boldsymbol{\xi}_i^H + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},j}^2\boldsymbol{\xi}_i\boldsymbol{\xi}_j^H + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},,j}^2\boldsymbol{\alpha}_{i,j}\boldsymbol{\alpha}_{i,j}^H \\ &+ \sigma_\eta^2\sum_{i=1}^{K}\sigma_{\mathrm{s},i}^2\left(\boldsymbol{\xi}_i\mathbf{p}^T + \mathbf{p}\boldsymbol{\xi}_i^T\right) + \sigma_\eta^2\sum_{i=1}^{K}\sigma_{\mathrm{s},i}^2\boldsymbol{H}_i + \sigma_\eta^4\mathbf{p}\mathbf{p}^T + \sigma_\eta^2\left(\sigma_\eta^2 + \sum_{i=1}^{K}\sigma_{\mathrm{s},i}^2\right)\boldsymbol{I}_{3M}\end{aligned} \tag{4.47}$$

$$= \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},j}^2\boldsymbol{\alpha}_{i,j}\boldsymbol{\alpha}_{i,j}^H + \sum_{i=1}^{K}\sigma_{\mathrm{s},i}^4\boldsymbol{\xi}_i\boldsymbol{\xi}_i^H + \boldsymbol{R}_{\mathrm{e}} + \boldsymbol{R}_{\mathrm{H}} + \sigma_\eta^2\sum_{i=1}^{K}\sigma_{\mathrm{s},i}^2\boldsymbol{I}_{3M}, \tag{4.48}$$

where we defined

$$\boldsymbol{R}_{\mathrm{e}} = \sum_{i=1}^{K}\sum_{j=1}^{K} \sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},j}^2 \boldsymbol{\xi}_i\boldsymbol{\xi}_j^T + \sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \left(\boldsymbol{\xi}_i\mathbf{p}^T + \mathbf{p}\boldsymbol{\xi}_i^T\right) + \sigma_\eta^4 \mathbf{p}\mathbf{p}^T \tag{4.49}$$

and

$$\boldsymbol{R}_{\mathrm{H}} = \sigma_\eta^2 \left(\sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \boldsymbol{H}_i + \sigma_\eta^2 \boldsymbol{I}_{3M} \right). \tag{4.50}$$

Defining a DOA estimator
Now $\tilde{\boldsymbol{R}}$ is analyzed, we are going to define a DOA estimator based on this matrix. We have seen that for an APS-arrays $\tilde{\boldsymbol{R}}_{\mathrm{p}}$ is sufficient to find the DOAs. In the APS case $\boldsymbol{R}_{\mathrm{e}}$ could be written as $\boldsymbol{R}_e \underset{\mathrm{APS}}{\longrightarrow} \alpha_{\boldsymbol{R}_{\mathrm{e}}} \mathbf{1}\mathbf{1}^T$ for a specific $\alpha_{\boldsymbol{R}_{\mathrm{e}}}$ and $\boldsymbol{R}_{\mathrm{H}}$ as $\boldsymbol{R}_{\mathrm{H}} \underset{\mathrm{APS}}{\longrightarrow} \alpha_{\boldsymbol{R}_{\mathrm{H}}} \boldsymbol{I}$ for a specific $\alpha_{\boldsymbol{R}_{\mathrm{H}}}$. This leads to an easy to find eigenstructure of the matrix $\boldsymbol{R}_d$. However due to the AVS setup cross terms between the different channels appear in these two matrices which makes it hard to find an eigenstructure. In the following we focus on estimating $\boldsymbol{R}_{\mathrm{e}}$ and $\boldsymbol{R}_{\mathrm{H}}$ so that these can be eliminated.

Lemma 4.1.

$$\boldsymbol{R}_{\mathrm{e}} = \mathbb{E}\left[\tilde{\mathbf{y}}\right] \mathbb{E}\left[\tilde{\mathbf{y}}^H\right]. \tag{4.51}$$

Proof. By evaluating $\mathbb{E}\left[\tilde{\mathbf{y}}\right]$ we find:

$$\mathbb{E}\left[\tilde{\mathbf{y}}\right] = \sum_{i=1}^{K} \boldsymbol{\xi}_i \sigma_{\mathrm{s},i}^2 + \sigma_\eta^2 \mathbf{p}. \tag{4.52}$$

So:

$$\begin{aligned}
\mathbb{E}\left[\tilde{\mathbf{y}}\right] \mathbb{E}\left[\tilde{\mathbf{y}}^H\right] &= \left(\sum_{i=1}^{K} \boldsymbol{\xi}_i \sigma_{\mathrm{s},i}^2 + \sigma_\eta^2 \mathbf{p} \right) \left(\sum_{i=1}^{K} \boldsymbol{\xi}_i^T \sigma_{\mathrm{s},i}^2 + \sigma_\eta^2 \mathbf{p}^T \right) && (4.53)\\
&= \sum_{i=1}^{K}\sum_{j=1}^{K} \sigma_{\mathrm{s},i}^2\sigma_{\mathrm{s},j}^2 \boldsymbol{\xi}_i\boldsymbol{\xi}_j^T + \sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \left(\boldsymbol{\xi}_i\mathbf{p}^T + \mathbf{p}\boldsymbol{\xi}_i^T\right) + \sigma_\eta^4 \mathbf{p}\mathbf{p}^T && (4.54)\\
&= \boldsymbol{R}_{\mathrm{e}}. && (4.55)
\end{aligned}$$

Which completes the proof. ■

Lemma 4.2.

$$\begin{aligned}
\boldsymbol{R}_{\mathrm{H}} &= \sigma_\eta^2 blkdiag\left\{\boldsymbol{R}_{\mathrm{H}_1}, \boldsymbol{R}_{\mathrm{H}_2}, \ldots, \boldsymbol{R}_{\mathrm{H}_M}\right\}, && (4.56)\\
\boldsymbol{R}_{\mathrm{H}_i} &= \mathbb{E}\left[\mathbf{y}_i\mathbf{y}_i^H\right] && (4.57)
\end{aligned}$$

where $\mathbf{y}_i$ is the output vector of sensor i.

Proof. Define $a_i(\theta_l)$ as the pressure transfer function from the lth source to the ith sensor, γ_i the clock error of the ith sensor and $\mathbf{n}_i \in \mathbb{C}^3$ the noise vector on the ith sensor. Now, we can expand (4.57) to:

$$\boldsymbol{R}_{\mathrm{H}_i} = \mathbb{E}\left[\left(e^{j2\pi f\gamma_i}\sum_{l=1}^{K} a_i(\theta_l)s_l\mathbf{h}_l + \mathbf{n}_i\right)\left(e^{-j2\pi f\gamma_i}\sum_{l=1}^{K} a_i^*(\theta_l)s_l^*\mathbf{h}_l^T + \mathbf{n}_i^H\right)\right] \tag{4.58}$$

$$= \sum_{l=1}^{K}\sigma_{\mathrm{s},l}^2\mathbf{h}_l\mathbf{h}_l^T + \sigma_\eta^2\boldsymbol{I}_3. \tag{4.59}$$

where we made use of the assumption that the signals are uncorrelated to each other and uncorrelated to the noise. By substituting (4.59) into $\sigma_\eta^2\mathrm{blkdiag}\left\{\boldsymbol{R}_{\mathrm{H}_1}, \boldsymbol{R}_{\mathrm{H}_2}, \ldots, \boldsymbol{R}_{\mathrm{H}_M}\right\}$ we find:

$$\sigma_\eta^2\mathrm{blkdiag}\left\{\boldsymbol{R}_{\mathrm{H}_1}, \boldsymbol{R}_{\mathrm{H}_2}, \ldots, \boldsymbol{R}_{\mathrm{H}_M}\right\} = \sigma_\eta^2\boldsymbol{I}_M \otimes \left(\sum_{l=1}^{K}\sigma_{\mathrm{s},l}^2\mathbf{h}_l\mathbf{h}_l^T + \sigma_\eta^2\boldsymbol{I}_3\right) \tag{4.60}$$

$$= \sigma_\eta^2\left(\sum_{l=1}^{K}\sigma_{\mathrm{s},l}^2\boldsymbol{H}_l + \sigma_\eta^2\boldsymbol{I}_{3M}\right) \tag{4.61}$$

$$= \boldsymbol{R}_{\mathrm{H}}. \tag{4.62}$$

Which completes the proof. ■

In order to get an estimate for the noise power, recall the eigendecomposition of our original covariance matrix (2.16):

$$\boldsymbol{R} = \sum_{m=1}^{3M}\beta_{\boldsymbol{R},m}\mathbf{u}_{\boldsymbol{R},m}\mathbf{u}_{\boldsymbol{R},m}^H. \tag{4.63}$$

Now, an estimate for the noise power is given by [20]:

$$\hat{\sigma}_\eta^2 = \frac{1}{3M-K}\sum_{m=K+1}^{3M}\beta_{\boldsymbol{R},m}. \tag{4.64}$$

As $\hat{\sigma}_\eta^2$ is the maximum likelihood estimate of σ_η^2 [20] we have that $\hat{\sigma}_\eta^2 \simeq \sigma_\eta^2$. Note that we could estimate $\boldsymbol{R}_{\mathrm{e}}$ from the data using N data points as follows:

$$\hat{\boldsymbol{R}}_{\mathrm{e}} = \frac{1}{N^2}\left(\sum_{n=1}^{N}\tilde{\mathbf{y}}(n)\right)\left(\sum_{n=1}^{N}\tilde{\mathbf{y}}(n)\right)^H, \tag{4.65}$$

where $\hat{\boldsymbol{R}}_{\mathrm{e}}$ is the estimate for $\boldsymbol{R}_{\mathrm{e}}$. Similar we could estimate $\boldsymbol{R}_{\mathrm{H}}$ by $\hat{\boldsymbol{R}}_{\mathrm{H}}$ as:

$$\hat{\boldsymbol{R}}_{\mathrm{H}} = \hat{\sigma}_\eta^2\mathrm{blkdiag}\left\{\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_1(n)\mathbf{y}_1^H(n), \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_2(n)\mathbf{y}_2^H(n), \ldots, \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_M(n)\mathbf{y}_M^H(n)\right\}. \tag{4.66}$$

Now define:

$$\boldsymbol{R}_{\mathrm{x}} = \tilde{\boldsymbol{R}} - \hat{\boldsymbol{R}}_{\mathrm{e}} - \hat{\boldsymbol{R}}_{\mathrm{H}} \tag{4.67}$$

$$\simeq \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \sigma_{si}^2 \sigma_{sj}^2 \boldsymbol{\alpha}_{i,j} \boldsymbol{\alpha}_{i,j}^H + \sum_{i=1}^{K} \sigma_{si}^4 \boldsymbol{\xi}_i \boldsymbol{\xi}_i^H + \sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \boldsymbol{I}_{3M}. \tag{4.68}$$

It is our goal to find the subspace spanned by the first term in (4.68). We could use a subspace based filtering method by for example defining a generalized eigenvalue decomposition on the matrix tuple $\left\{\tilde{\boldsymbol{R}}, \tilde{\boldsymbol{R}}_{\mathrm{e}} + \tilde{\boldsymbol{R}}_{\mathrm{H}}\right\}$. However due to that the term $\sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \boldsymbol{I}_{3M}$. would appear in the signal subspace. Of course it is possible to find an estimate for this term. However, For now, we will simply use the subtraction defined in (4.67). Although this subtraction makes our system less robust against certain assumptions (e.g. the uncorrelated sources assumption) this will give us a good insight in the behavior of $\boldsymbol{R}_{\mathrm{x}}$. In the next section we will propose a method that directly estimates $\boldsymbol{R}_{\mathrm{x}} - \sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \boldsymbol{I}_{3M}$.

Now, define the eigenvalue decomposition of $\boldsymbol{R}_{\mathrm{x}}$ as follows:

$$\boldsymbol{R}_{\mathrm{x}} = \sum_{m=1}^{3M} \beta_{\boldsymbol{R}_{\mathrm{x}},m} \mathbf{u}_{\boldsymbol{R}_{\mathrm{x}},m} \mathbf{u}_{\boldsymbol{R}_{\mathrm{x}},m}^H, \tag{4.69}$$

where $\{\beta_{\boldsymbol{R}_{\mathrm{x}},m}\}_{m=1}^{3M}$ are the eigenvalues of $\boldsymbol{R}_{\mathrm{x}}$ in descending order corresponding to the eigenvectors $\{\mathbf{u}_{\boldsymbol{R}_{\mathrm{x}},m}\}_{m=1}^{3M}$.

By analyzing the expected eigenvalues of $\boldsymbol{R}_{\mathrm{x}}$ we find that the noise floor will be at $\sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2$. From (4.68) we find for $K < 3M$ sources for which $\theta_i \neq \theta_j; \forall i \neq j \in [1, 2, \ldots, K]$ that, depending on the array-aperture, the vectors in the first sum of (4.68) will all be orthogonal to each other. Furthermore we have that the second sum is orthogonal to the first sum and that the maximum rank of the resulting matrix of the sum equals three. It is easy to see that

$$\operatorname{rank}\left\{\sum_{i=1}^{K} \sigma_{si}^4 \boldsymbol{\xi}_i \boldsymbol{\xi}_i^H\right\} = \min\{K, 3\}. \tag{4.70}$$

From which we conclude

$$r_{\boldsymbol{R}_{\mathrm{x}}} = K(K-1) + \min\{K, 3\}, \tag{4.71}$$

where $r_{\boldsymbol{R}_{\mathrm{x}}}$ denotes the rank of the signal subspace of $\boldsymbol{R}_{\mathrm{x}}$. With this information we can use a MUSIC like method to recover the DOA's. For this we define a spatial spectrum as follows:

$$P_{\mathrm{x}}(\theta, \theta') = \left(||\tilde{\boldsymbol{U}}_{\boldsymbol{R}_{\mathrm{x}}}^H \boldsymbol{\alpha}(\theta, \theta')||_2^2 + ||\tilde{\boldsymbol{U}}_{\boldsymbol{R}_{\mathrm{x}}}^H \boldsymbol{\alpha}(\theta', \theta)||_2^2\right)^{-1}, \tag{4.72}$$

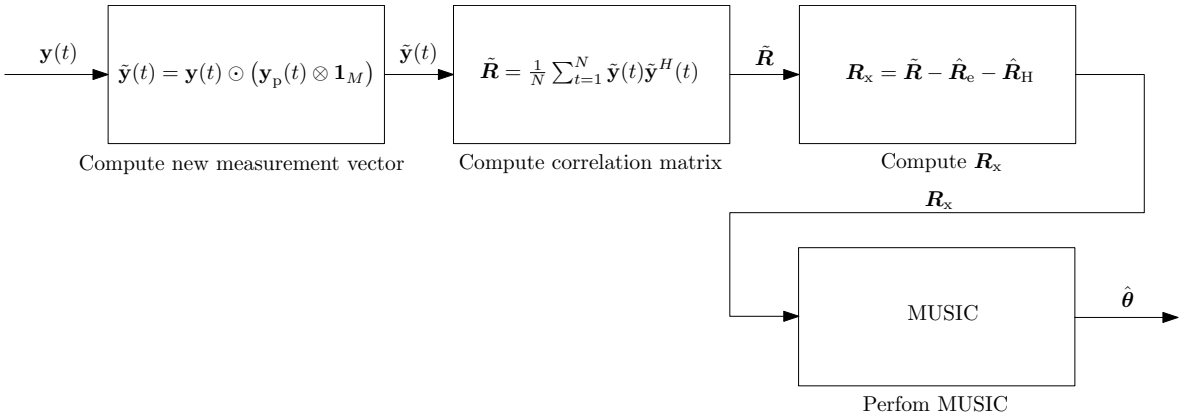


Figure 4.4: Brief overview of the method described in this section.

where we defined $\tilde{\boldsymbol{U}}_{\boldsymbol{R}_x} = \begin{bmatrix} \mathbf{u}_{\boldsymbol{R}_x, r_{\boldsymbol{R}_x+1}} & \mathbf{u}_{\boldsymbol{R}_x, r_{\boldsymbol{R}_x+2}} & \dots & \mathbf{u}_{\boldsymbol{R}_x, r_{4M}} \end{bmatrix}$. Now we can find the DOA's by searching for the peaks in this spectrum:

$$\left\{ \hat{\theta}_1, \hat{\theta}_2 \right\} = \arg \max_{\theta, \theta'} P_x(\theta, \theta'). \tag{4.73}$$

A brief overview of this method is given in Figure 4.4.

Analyzing the estimator

In the following, the estimates of θ_i for a varying number of sources K will be analyzed. The analization will be based on theory and simulations. Firstly the simple situation where only one source is present (i.e. $K = 1$) is discussed. Secondly, the situation of two sources. Lastly, their follows a generalization for the case where three or more sources are present.

Note that if $\theta = \theta'$ than $\boldsymbol{\alpha}(\theta, \theta') = \boldsymbol{\xi}(\theta)$. So if there is only one source ($K = 1$) with direction of arrival θ_1 and we search over the intervals $-180° < \theta < 180°$ and $-180° < \theta' < 180°$ we will find a peak at $\theta = \theta' = \theta_1$.

The above is simulated by the 7-sensor layout proposed by Liu et al. in [18] given in Figure 4.2. Furthermore, a signal with DOA of $\theta_1 = 30°$, and all other simulation variables are the same as used as before. The resulting spectrum is given in Figure 4.5. From this figure it is clear that the estimated DOA will be $\theta = \theta' = \theta_1$ which is as expected. Furthermore it is clear that if we have only one source that we could estimate its DOA with the 1D-search over $\theta = \theta'$. Note that for $\theta = \theta'$ the results will be equal to a simple combination between the sensors without using the time properties of the received data.

By expanding to the situation for two sources ($K = 2$) with the first DOA as before

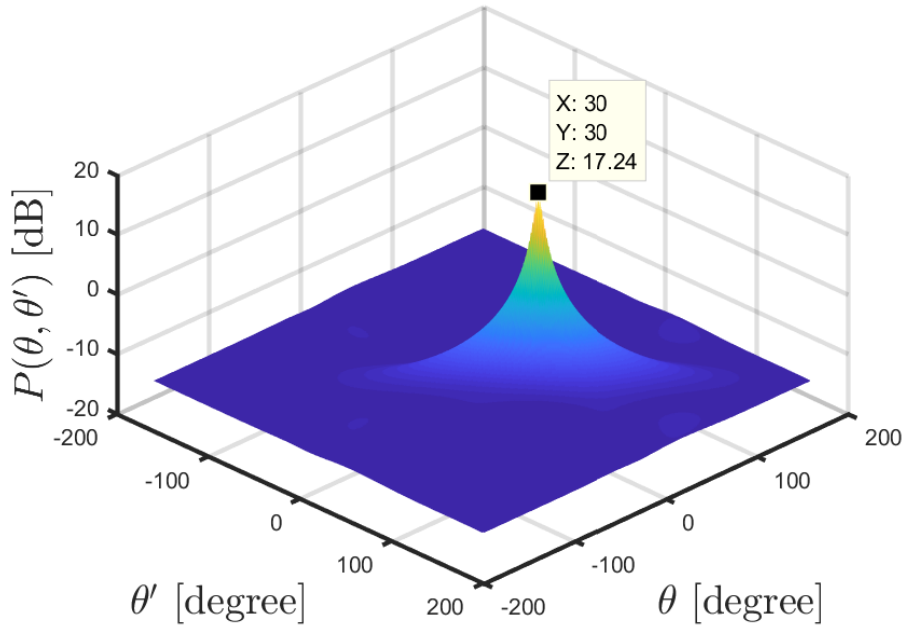


Figure 4.5: Simulation of the described method with $K = 1$ source.

and the second at $\theta_2 \neq \theta_1$ we expect to find 4 peaks, namely at:

$$\begin{aligned}[\theta, \theta'] &= [\theta_1, \theta_1], \\ [\theta, \theta'] &= [\theta_1, \theta_2], \\ [\theta, \theta'] &= [\theta_2, \theta_1], \\ [\theta, \theta'] &= [\theta_2, \theta_2].\end{aligned}$$

Note that for the peaks that lie at the line $\theta = \theta'$ (i.e. $[\theta_i, \theta_i]$) we only make use of the information in the vector channels, while the information of the relative time difference of arrival is omitted. This is due to the fact that $\boldsymbol{\alpha}(\theta, \theta) = \boldsymbol{\xi}(\theta)$. So it is possible to find the DOAs by a 1D-grid search, however more accurate results may be obtained when a 2D-grid search is used. As for $\boldsymbol{\alpha}(\theta, \theta')$ $\forall \theta \neq \theta'$ both the information in the vector channels and the relative time difference of arrival is used.

This is simulated as before but with a second uncorrelated source with equal source power and a DOA $\theta_2 = 130°$. The results of this simulation are shown in Figure 4.6. It is clear that the peaks are located at the places where we expect them. The peaks at the points $(30; 130)$ and $(130; 30)$ are sharper than the other two. This is due to that at the line $\theta = \theta'$ we only make use of the channel information and not the information that is in the timing of the data. In the two sharper points a combination is used between the channel and the (relative) timing information.

Expanding the case furthermore to three or more sources ($K \geq 3$), we have that the last term in (4.68) reaches its maximum rank. From which we conclude that the

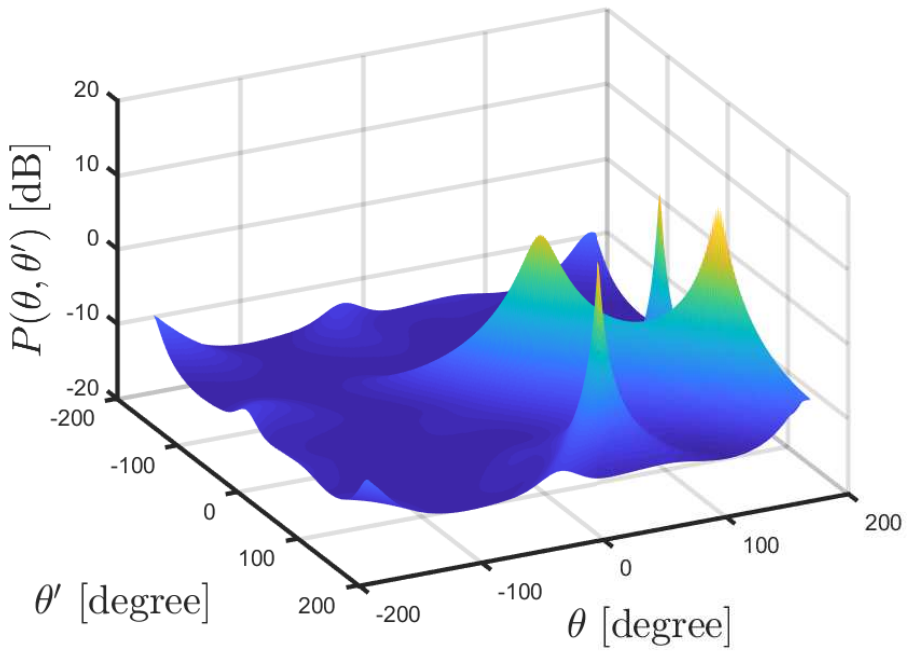


Figure 4.6: Simulation of the described method with $K = 2$ sources. The peaks are located at (30;30), (30;130), (130;30) and (130;130).

line $\theta = \theta'$ fully lies in the signal space of $\boldsymbol{R}_x$. So the values of $p(\theta, \theta')$ will be large if θ is close to θ'. Moreover it is easy to see that all other maximums in $P(\theta, \theta')$ are at:

$$[\theta, \theta'] = [\theta_i, \theta_j] \qquad \forall i \neq j \in [1, 2, \ldots, K].$$

This is simulated as before with a third source at $\theta_3 = -30°$, the results are shown in Figure 4.7. From this figure it is clear that we can not estimate the DOAs anymore by searching at the line $\theta = \theta'$. However the DOAs can be estimated by finding the other peaks.

When we compare this new method with the method described in the previous section we find it to have similar results in the case of 2 or more sensors. However, our method eliminates the first three disadvantages of the earlier technique.

1. As can be seen in the figures, the ambiguity with Lius method within the interval $[-180°, 180°]$ is gone within the new method.

2. The new method does not explicitly give any limitations on the array, although the array configuration has effect on the performance on the algorithm the algorithm will not break down when another aperture or frequency will be used. This will be simulated in the next section.

3. The $\mathbf{1}$-vector in the steering vector in previous section has been replaced by the $\boldsymbol{\xi}_i$-vectors which enables us to find the DOA using this method when only one signal is present.

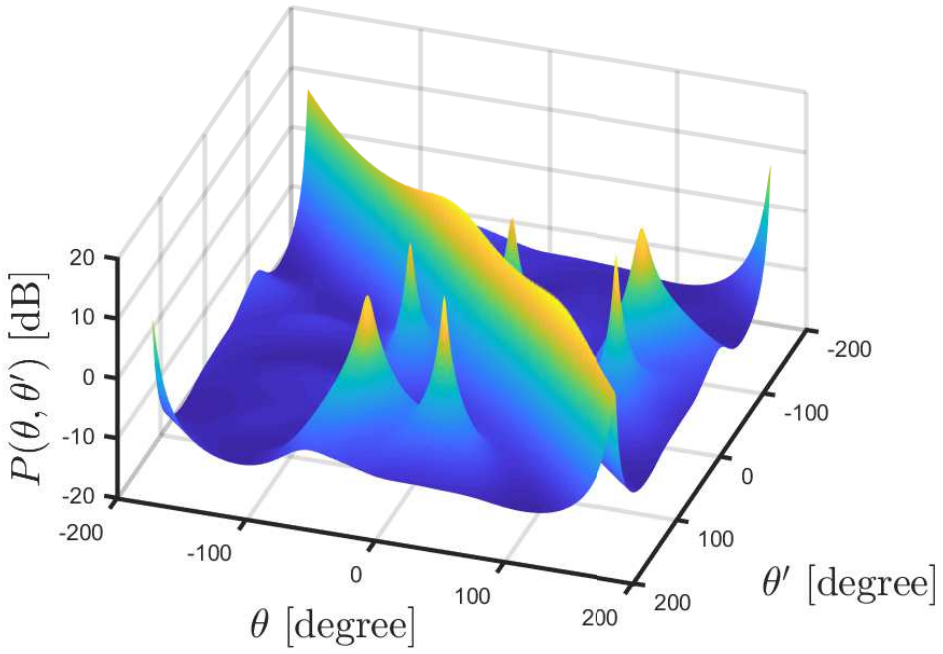


Figure 4.7: Simulation of the described method with $K = 3$ sources. The peaks are located at (-30;30), (-30;130) (30,-30), (30;130) (130,-30) and (130;30).

There are also disadvantages of this method compared with previous:

1. The amount of channels necessary to make the algorithm work has gone up from $K(K-1)+1$ to $K(K-1)+\min\{K,3\}+1$. Although it could be argued that an AVS contains three channels while an APS only contains one, that less sensors are needed in total.

2. The subtraction performed by (4.67) may cause issues when the signals and noise are not completely Gaussian distributed.

Another issue of the previous method that is unaltered by this method, is the dependency on the fourth-order statistics. In the remainder of this subsection, we will give an intuitive approach on how to solve this issue, while at the same time taking care of the second disadvantage of this method. In the next section this new method is discussed more in depth.

From fourth order statistics to second order statistics.
By analyzing our method we find that we can describe $\tilde{\boldsymbol{R}}$ also as:

$$\tilde{\boldsymbol{R}} = \mathbb{E}\left[\left(\mathbf{y} \odot \mathbf{y}_{\mathrm{p}}^{*} \otimes \mathbf{1}_3\right)\left(\mathbf{y} \odot \mathbf{y}_{\mathrm{p}}^{*} \otimes \mathbf{1}_3\right)^{H}\right] \tag{4.74}$$

and $\boldsymbol{R}_e$ as:

$$\boldsymbol{R}_e = \mathbb{E}\left[\mathbf{y} \odot \left(\mathbf{y}_{\mathrm{p}}^{*} \otimes \mathbf{1}_3\right)\right]\mathbb{E}\left[\mathbf{y} \odot \left(\mathbf{y}_{\mathrm{p}}^{*} \otimes \mathbf{1}_3\right)\right]^{H}. \tag{4.75}$$

From [21], we have that the product of four complex Gaussian variables $\{x_i\}_{i=1}^4$ of which at least one is zero mean has the following relation:

$$\mathbb{E}\left[x_1x_2x_3x_4\right] = \mathbb{E}\left[x_1x_2\right]\mathbb{E}\left[x_3x_4\right] + \mathbb{E}\left[x_1x_3\right]\mathbb{E}\left[x_2x_4\right] + \mathbb{E}\left[x_1x_4\right]\mathbb{E}\left[x_2x_3\right]. \tag{4.76}$$

Using the above equation we can expand (4.74) as follows:

$$\begin{aligned}\tilde{\boldsymbol{R}} = \mathbb{E}\left[\mathbf{y}\odot\left(\mathbf{y}_{\mathrm{p}}^*\otimes\mathbf{1}_3\right)\right]\mathbb{E}\left[\mathbf{y}\odot\left(\mathbf{y}_{\mathrm{p}}^*\otimes\mathbf{1}_3\right)\right]^H + \mathbb{E}\left[\mathbf{y}\mathbf{y}^H\right]\odot\mathbb{E}\left[\left(\mathbf{y}_{\mathrm{p}}^*\otimes\mathbf{1}_3\right)\left(\mathbf{y}_{\mathrm{p}}\otimes\mathbf{1}_3\right)^T\right] \\ +\mathbb{E}\left[\mathbf{y}\left(\mathbf{y}_{\mathrm{p}}^T\otimes\mathbf{1}_3^T\right)\right]\odot\mathbb{E}\left[\left(\mathbf{y}_{\mathrm{p}}^*\otimes\mathbf{1}_3\right)\mathbf{y}^H\right].\end{aligned} \tag{4.77}$$

Now it is easy to that the first term is equal to $\boldsymbol{R}_e$ and that under the assumption of circular complex Gaussian sources and noise that the last term equals zero. So by the subtraction of $\boldsymbol{R}_e$ from $\tilde{\boldsymbol{R}}$ we actually get:

$$\begin{aligned}\tilde{\boldsymbol{R}} - \boldsymbol{R}_e =& \mathbb{E}\left[\mathbf{y}\mathbf{y}^H\right]\odot\mathbb{E}\left[\left(\mathbf{y}_{\mathrm{p}}^*\otimes\mathbf{1}_3\right)\left(\mathbf{y}_{\mathrm{p}}\otimes\mathbf{1}_3\right)^T\right] && (4.78)\\ =& \boldsymbol{R}\odot\left(\boldsymbol{R}_{\mathrm{p}}^*\otimes\mathbf{1}_3\mathbf{1}_3^T\right), && (4.79)\end{aligned}$$

where we made us of $(\mathbf{a}\otimes\mathbf{b})(\mathbf{c}\otimes\mathbf{d}) = (\mathbf{ac})\otimes(\mathbf{bd})$ and defined $\boldsymbol{R}_{\mathrm{p}} = \mathbb{E}\left[\mathbf{y}_{\mathrm{p}}\mathbf{y}_{\mathrm{p}}^H\right]$. So by first taking the correlation matrices and then the Hadamard product between the two matrices will lead to a similar result. A method developed by Cao et al. in [19] for APS-arrays uses the product $(\boldsymbol{R}_{\mathrm{p}}\odot\boldsymbol{R}_{\mathrm{p}}^*)$ and will further be referred to as Caos method. Based on Caos method a new method for AVS-arrays is developed in the next section.

4.3 Cao-Based Eigenstructure Method

In [19], Cao et al. propose an eigenstructure method which is also based on the Hadamard product. Instead of applying the Hadamard product directly on the data the product is performed on a manipulated form of the correlation matrix $\boldsymbol{R}$ as can be seen later. In this section we will derive a method based on [19] applicable for an AVS-array. First we will derive a new autocorrelation matrix aftwer which we will anlyze the new estimator that follows from this.

Finding the new correlation matrix
Recall the data-model:

$$\mathbf{y}(t) = \boldsymbol{\Pi}\boldsymbol{A}\mathbf{s}(t) + \boldsymbol{\eta}(t), \tag{4.80}$$

where $\mathbf{y}(t)\in\mathbb{C}^{3M}$ is the sensor output vector at time instant t, M the number of sensors, $\boldsymbol{\Pi} = \boldsymbol{\Gamma}\otimes\boldsymbol{I}_3$, $\boldsymbol{\Gamma}\in\mathbb{C}^{M\times M}$ the diagonal matrix containing the clock errors, $\boldsymbol{A}\in\mathbb{C}^{3M\times K}$ the mixing matrix from source to sensor, K the number of sources, $\mathbf{s}(t)\in\mathbb{C}^K$ the source signal at time instant t and $\boldsymbol{\eta}(t)\in\mathbb{C}^{3M}$ contains the sensor noise. Computing the data correlation matrix $\boldsymbol{R}_{\mathrm{y}}$ based on this model while omitting the time dependency results in:

$$\boldsymbol{R}_{\mathrm{y}} = \mathbb{E}\left[\mathbf{y}\mathbf{y}^H\right] = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{R}_{\mathrm{s}}\boldsymbol{A}^H\boldsymbol{\Pi}^H + \boldsymbol{R}_{\boldsymbol{\eta}}, \tag{4.81}$$

where $\boldsymbol{R}_{\mathrm{s}} = \mathbb{E}\left[\mathbf{s}\mathbf{s}^H\right]$ is the source correlation matrix and $\boldsymbol{R}_{\boldsymbol{\eta}} = \mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^H\right]$ the noise correlation matrix. Observe that this correlation matrix is equal to the correlation matrix $\boldsymbol{R}$ defined before, the y subscript is added to improve the readability later in this section. Note that we can write the data vector based on (4.80) for the pressure channels as follows:

$$\mathbf{y}_{\mathrm{p}} = \boldsymbol{\Gamma}\boldsymbol{A}_{\mathrm{p}}\mathbf{s} + \boldsymbol{\eta}_{\mathrm{p}}, \tag{4.82}$$

where $\mathbf{y}_{\mathrm{p}} \in \mathbb{C}^M$ is the pressure channel output, $\boldsymbol{A}_{\mathrm{p}} \in \mathbb{C}^{M\times K}$ the mixing matrix from the sources to the pressure channels and $\boldsymbol{\eta}_{\mathrm{p}}$ the pressure channel noise vector. Now, define a pressure channel correlation matrix similar to the data correlation matrix:

$$\boldsymbol{R}_{\mathrm{p}} = \mathbb{E}\left[\mathbf{y}_{\mathrm{p}}\mathbf{y}_{\mathrm{p}}^H\right] = \boldsymbol{\Gamma}\boldsymbol{A}_{\mathrm{p}}\boldsymbol{R}_{\mathrm{s}}\boldsymbol{A}_{\mathrm{p}}^H\boldsymbol{\Gamma}^H + \boldsymbol{R}_{\boldsymbol{\eta}_{\mathrm{p}}}, \tag{4.83}$$

where $\boldsymbol{R}_{\boldsymbol{\eta}_{\mathrm{p}}} = \mathbb{E}\left[\boldsymbol{\eta}_{\mathrm{p}}\boldsymbol{\eta}_{\mathrm{p}}^H\right]$ is the pressure channel noise correlation matrix. If we assume the noise to be independently white circular Gaussian distributed we find:

$$\boldsymbol{R}_{\boldsymbol{\eta}} = \mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^H\right] = \sigma_\eta^2\boldsymbol{I}_{3M} \tag{4.84}$$

and

$$\boldsymbol{R}_{\boldsymbol{\eta}_{\mathrm{p}}} = \mathbb{E}\left[\boldsymbol{\eta}_{\mathrm{p}}\boldsymbol{\eta}_{\mathrm{p}}^H\right] = \sigma_\eta^2\boldsymbol{I}_M, \tag{4.85}$$

where σ_η^2 is the noise power. Note that we have an estimate $\hat{\sigma}_\eta^2$ for the noise power given through the eigenvalue decomposition of $\boldsymbol{R}_{\mathrm{y}}$ given by (4.64). Subtracting this estimate for the noise correlation matrices from the data correlation matrices results in

$$\tilde{\boldsymbol{R}}_{\mathrm{y}} = \boldsymbol{R}_{\mathrm{y}} - \hat{\sigma}_\eta^2\boldsymbol{I}_{3M} \simeq \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{R}_{\mathrm{s}}\boldsymbol{A}^H\boldsymbol{\Pi}^H \tag{4.86}$$

and

$$\tilde{\boldsymbol{R}}_{\mathrm{p}} = \boldsymbol{R}_{\mathrm{p}} - \hat{\sigma}_\eta^2\boldsymbol{I}_M \simeq \boldsymbol{\Gamma}\boldsymbol{A}_{\mathrm{p}}\boldsymbol{R}_{\mathrm{s}}\boldsymbol{A}_{\mathrm{p}}^H\boldsymbol{\Gamma}^H. \tag{4.87}$$

At this point the clock errors are still in the correlation matrices. However by taking the Hadamard product between either of these matrices and its complex conjugate will remove these errors. In order to keep the final solutions unique and independent of frequency we define this Hadamard product as follows:

$$\tilde{\boldsymbol{R}}_{\mathrm{f}} = \tilde{\boldsymbol{R}}_{\mathrm{y}} \odot \left(\tilde{\boldsymbol{R}}_{\mathrm{p}}^* \otimes \mathbf{1}_3\mathbf{1}_3^T\right). \tag{4.88}$$

Note that we can write one index of $\tilde{\boldsymbol{R}}_{\mathrm{p}}$ as follows:

$$\left[\tilde{\boldsymbol{R}}_{\mathrm{p}}\right]_{p,q} = e^{\mathrm{j}2\pi f(\gamma_p-\gamma_q)}\sum_{s=1}^{K}\sigma_s^2 e^{jk\mathbf{u}_s^T(\mathbf{r}_p-\mathbf{r}_q)}, \tag{4.89}$$

where $\mathbf{u}_s = \mathbf{u}(\theta_s)$. By splitting the matrix $\tilde{\boldsymbol{R}}_{\mathrm{y}}$ in blocks of 3×3 we can write:

$$\tilde{\boldsymbol{R}}_{\mathrm{y}} = \begin{bmatrix} \tilde{\mathcal{R}}_{\mathrm{y},11} & \tilde{\mathcal{R}}_{\mathrm{y},12} & \dots & \tilde{\mathcal{R}}_{\mathrm{y},1M} \\ \tilde{\mathcal{R}}_{\mathrm{y},21} & \tilde{\mathcal{R}}_{\mathrm{y},22} & \dots & \tilde{\mathcal{R}}_{\mathrm{y},2M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{R}}_{\mathrm{y},M1} & \tilde{\mathcal{R}}_{\mathrm{y},M2} & \dots & \tilde{\mathcal{R}}_{\mathrm{y},MM} \end{bmatrix}, \tag{4.90}$$

where every block can be written as:

$$\tilde{\mathcal{R}}_{\mathrm{y},pq} = e^{j2\pi f(\phi_p - \phi_q)} \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \mathbf{h}_i \mathbf{h}_i^T e^{jk\mathbf{u}_i^T(\mathbf{r}_p - \mathbf{r}_q)}. \tag{4.91}$$

By splitting the matrix $\tilde{\boldsymbol{R}}_{\mathrm{f}}$ in blocks of 3×3 we can write:

$$\tilde{\boldsymbol{R}}_f = \begin{bmatrix} \tilde{\mathcal{R}}_{\mathrm{f},11} & \tilde{\mathcal{R}}_{\mathrm{f},12} & \dots & \tilde{\mathcal{R}}_{\mathrm{f},1M} \\ \tilde{\mathcal{R}}_{\mathrm{f},21} & \tilde{\mathcal{R}}_{\mathrm{f},22} & \dots & \tilde{\mathcal{R}}_{\mathrm{f},2M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{R}}_{\mathrm{f},M1} & \tilde{\mathcal{R}}_{\mathrm{f},M2} & \dots & \tilde{\mathcal{R}}_{\mathrm{f},MM} \end{bmatrix}. \tag{4.92}$$

Note that we can find the blocks of $\tilde{\boldsymbol{R}}_{\mathrm{f}}$ as follows:

$$\tilde{\mathcal{R}}_{\mathrm{f},pq} = \tilde{\mathcal{R}}_{\mathrm{y},pq} \left[\tilde{\boldsymbol{R}}_{\mathrm{p}}^*\right]_{p,q} \tag{4.93}$$

$$= \left(\sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \mathbf{h}_i \mathbf{h}_i^T e^{jk\mathbf{u}_s^T(\mathbf{r}_p - \mathbf{r}_q)} \right) \left(\sum_{s=1}^{K} \sigma_{\mathrm{s},i}^2 e^{jk\mathbf{u}_i^T(\mathbf{r}_q - \mathbf{r}_p)} \right) \tag{4.94}$$

$$= \sum_{i=1}^{K} \sum_{l=1}^{K} \sigma_{\mathrm{s},i}^2 \sigma_{\mathrm{s},l}^2 \mathbf{h}_i \mathbf{h}_i^T e^{jk(\mathbf{u}_i - \mathbf{u}_l)(\mathbf{r}_p - \mathbf{r}_q)}. \tag{4.95}$$

Recall the definition of $\boldsymbol{\alpha}_{i,l}$ in (4.29):

$$\boldsymbol{\alpha}_{i,l} = \left(\mathbf{a}_{\mathrm{p}}(\theta_i) \odot \mathbf{a}_{\mathrm{p}}^*(\theta_l)\right) \otimes \mathbf{h}_i. \tag{4.96}$$

Splitting $\boldsymbol{\alpha}_{i,l}$ in vectors of length three leads to:

$$\boldsymbol{\alpha}_{i,l} = \begin{bmatrix} \boldsymbol{\alpha}_1^T(i,l) & \boldsymbol{\alpha}_2^T(i,l) & \dots & \boldsymbol{\alpha}_M^T(i,l) \end{bmatrix}^T. \tag{4.97}$$

We have that:

$$\boldsymbol{\alpha}_m(i,l) = \mathbf{h}_i e^{jk\mathbf{r}_m(\mathbf{u}_i - \mathbf{u}_l)}. \tag{4.98}$$

Now, by substituting (4.98) into (4.95) we find:

$$\tilde{\mathcal{R}}_{\mathrm{f},pq} = \sum_{i=1}^{K} \sum_{l=1}^{K} \sigma_{\mathrm{s},i}^2 \sigma_l^2 \boldsymbol{\alpha}_p(i,l) \boldsymbol{\alpha}_q^H(i,l). \tag{4.99}$$

Which indicates that

$$\tilde{\boldsymbol{R}}_{\mathrm{f}} = \sum_{i=1}^{K} \sum_{l=1}^{K} \sigma_{\mathrm{s},i}^2 \sigma_l^2 \boldsymbol{\alpha}_{i,l} \boldsymbol{\alpha}_{i,l}^H. \tag{4.100}$$

Note that for $l = i$ we have that

$$\boldsymbol{\alpha}_{i,i} = \boldsymbol{\xi}_i. \tag{4.101}$$

Now, substitute (4.101) into (4.100)

$$\tilde{\boldsymbol{R}}_{\mathrm{f}} = \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^4 \boldsymbol{\xi}_{\mathrm{s},i} \boldsymbol{\xi}_{\mathrm{s},i}^H + \sum_{i=1}^{K} \sum_{\substack{l=1 \\ l \neq i}}^{K} \sigma_{\mathrm{s},i}^2 \sigma_{\mathrm{s},l}^2 \boldsymbol{\alpha}_{i,l} \boldsymbol{\alpha}_{i,l}^H \tag{4.102}$$

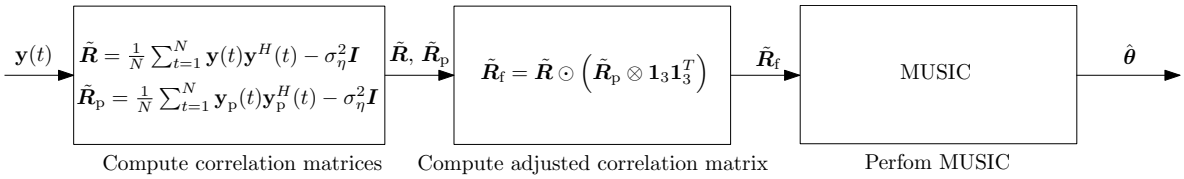


Figure 4.8: Brief overview of the method described in this section.

from which we conclude that $\tilde{\boldsymbol{R}}_\mathrm{f} = \boldsymbol{R}_x - \sigma_\eta^2 \sum_{i=1}^{K} \sigma_{\mathrm{s},i}^2 \boldsymbol{I}$, and that the DOAs can be found in the same fashion as discussed in the previous section. However this method solely relies on the second order statistics of the data where the method described in the previous section also relies on the fourth order statistics. This makes the latter less reliable in practice where the signals might not be perfectly Gaussian. Furthermore the subtraction performed in this section by (4.86) and (4.87) only makes an assumption on the noise. While the subtraction performed in the previous section in (4.67) makes use of the second order statistics of the data and will therefore reduce the performance if the sources are not completely uncorrelated.

A brief overview of the method described in this section is given in Figure 4.8.

Analyzing the new DOA estimator

Recall from previous section that, when two sources are impinging the array we expect peaks at:

$$\begin{aligned}[\theta, \theta'] =& [\theta_1, \theta_1],\\ [\theta, \theta'] =& [\theta_1, \theta_2],\\ [\theta, \theta'] =& [\theta_2, \theta_1],\\ [\theta, \theta'] =& [\theta_2, \theta_2].\end{aligned}$$

Imagine two sources that are close to each other (so θ_1 is close to θ_2) impinging the array, we expect the above given peaks to be close to each other. Which may, when the sources are to close to each other, merge into one single big peak from which the separate peaks might not be extracted anymore.

In previous section we have seen that if 3 sources impinge the array we would find peaks at:

$$\begin{aligned}[\theta, \theta'] =& [\theta_1, \theta_2],\\ [\theta, \theta'] =& [\theta_1, \theta_3],\\ [\theta, \theta'] =& [\theta_2, \theta_1],\\ [\theta, \theta'] =& [\theta_2, \theta_3],\\ [\theta, \theta'] =& [\theta_3, \theta_1],\\ [\theta, \theta'] =& [\theta_3, \theta_2].\end{aligned}$$

Furthermore due to that the $\boldsymbol{\xi}_i$-vectors span full signal space at the line $\theta = \theta'$ we will observe high values at this line. When two sources are located close to each other these peaks might merge with this line such that the peaks are not observable anymore.

In the method proposed by Cao APSs are used, as for an APS the $\mathbf{h}$-vector does not exist. The $\boldsymbol{\xi}$ vector reduces in this case:

$$\boldsymbol{\xi}_i = \mathbf{1}_M \otimes \mathbf{h}_i \underset{\text{APS}}{\longrightarrow} \mathbf{1}_M. \tag{4.103}$$

Note that this implies:

$$\boldsymbol{\alpha}_{i,i} = \boldsymbol{\xi}_i \underset{\text{APS}}{\longrightarrow} \mathbf{1}_M. \tag{4.104}$$

Observe that this implies that for Caos method the line $\theta = \theta'$ will always have high values regardless of the amount of sensor. However this also implies that the first term in (4.102) can be written as:

$$\sum_{i=1}^{K} \sigma_{\text{s},i}^4 \boldsymbol{\xi}_{\text{s},i} \boldsymbol{\xi}_{\text{s},i}^H \underset{\text{APS}}{\longrightarrow} b \mathbf{1}_M \mathbf{1}_M^T \tag{4.105}$$

where we defined b as a scalar. In [19] Cao et al. present a non-convex optimization problem that estimates b. This problem is solved by performing a grid search.

Note that for our method we could write the above as:

$$\sum_{i=1}^{K} \sigma_{\text{s},i}^4 \boldsymbol{\xi}_{\text{s},i} \boldsymbol{\xi}_{\text{s},i}^H = \sum_{i=1}^{K} \sigma_{\text{s},i}^4 \left(\mathbf{1}_M \mathbf{1}_M^T \otimes \mathbf{h}_i \mathbf{h}_i^T \right) = \mathbf{1}_M \mathbf{1}_M^T \otimes \boldsymbol{B} \tag{4.106}$$

where we defined matrix $\boldsymbol{B} \in \mathbb{R}^{3\times 3}$ which is hermitian. If we want to use a comparable method as used in [19] we need to estimate the entries of $\boldsymbol{B}$. Due that $\boldsymbol{B}$ is hermitian we would need to estimate entries. From (4.3) it follows that:

$$[\boldsymbol{B}]_{1,1} = [\boldsymbol{B}]_{2,2}^2 + [\boldsymbol{B}]_{3,3}^2 \tag{4.107}$$

where $[\boldsymbol{B}]_{i,j}$ denotes the (i,j) entry of $\boldsymbol{B}$. From which we conclude that we need to estimate five entries of the matrix to find the full matrix. As the proposed algorithm to solve the discussed optimization problem involves a grid search, it follows that if we want to suppress the high values at the line $\theta = \theta'$ we would need a five-dimensional grid search. As a five-dimensional grid search is computational expensive, this is not an appropriate method.

This results in the fact that in some circumstances the method proposed by Cao for an APS-array behaves better than our method that makes use of an AVS-array. An example of this behavior is given in the simulation below, for this simulation we use the same setup as before (i.e. using Figure 4.2 as sensor layout and the same SNR and frequencies as before). In this simulation two equal powered sources impinge the sensors: one at $\theta_1 = -5°$ and the other at $\theta_2 = 5°$. The result when using the technique described in this section is shown in Figure 4.9. From this figure it is clear that we can not estimates the DOAs in this situation. In Figure 4.10 the result of using the technique described in [19] when using an APS-array. In this plot we

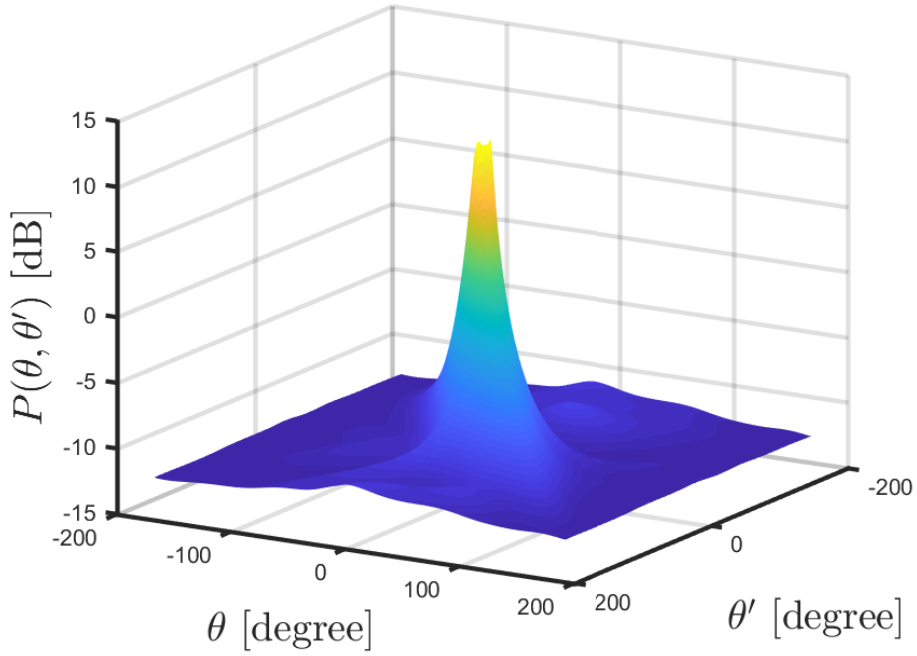


Figure 4.9: AVS result for closely separated sources using the technique described in the section. The DOA $\theta_1 = -5°$ and $\theta_2 = 5°$ can not be determined from this plot.

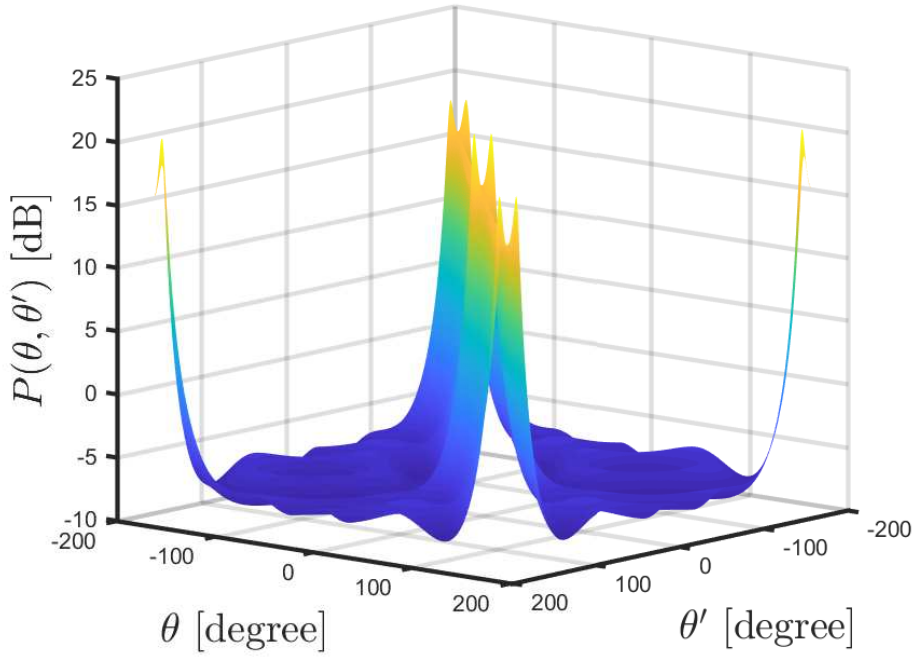


Figure 4.10: APS result for closely separated sources using the technique described in [19]. The DOA $\theta_1 = -5°$ and $\theta_2 = 5°$ can be determined from this plot.

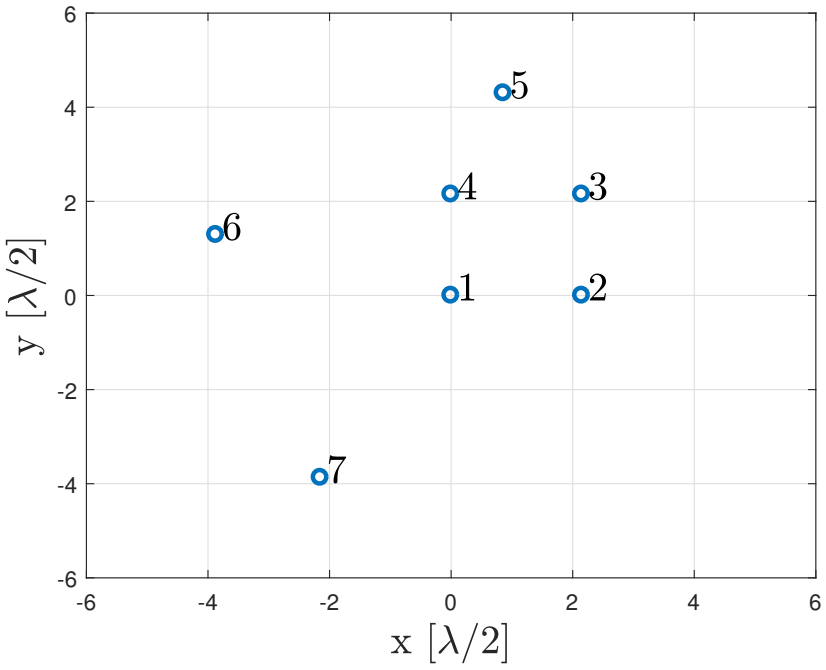


Figure 4.11: The array-aperture with increased inter-sensor spacing.

can clearly see peaks at the points $(-5°, 5°)$ and $(5°, -5°)$ as expected. However also peaks appear at $(-175°, 175°)$ and $(175°, -175°)$. This is due to that Caos method has only unique solutions for the angles (θ) that lie within $-90° \leq \theta < 90°$. Furthermore Caos method has a limited choice on sensor placement, in order to receive the above mentioned uniqueness of solutions. The array should have a sensor at all 4 sides of a square where the side of the square is equal to $\lambda/4$. λ here is the wavelength of the signal. Opposite to this the uniqueness of solution for the method described in this section is valid for $-180° \leq \theta < 180$ regardless of the setup of the array.

A nice feature of this method is that the DOAs can be found independent of the inter-sensor spacing of the array. This will be demonstrated below. By changing the spacing in the current sensor layout which also proves that the DOA can be found regardless of the frequency of the signals.

By increasing the distance between the sensors with a factor of 4.3 in Figure 4.2 we get the layout shown in Figure 4.11. This is simulated with the described array aperture, DOAs at $\theta_1 = 30°$ and $\theta_2 = 130°$ and the other settings as before. The results of this simulation are shown in Figure 4.12. By observing this figure, it is clear that although some minor sidelobes appear it is still possible to estimate the DOAs from this figure. In comparison, simulation with Coas method using APSs gives the result shown in Figure 4.13 it is clear that from this plot it is impossible to estimate the DOAs.

Decreasing the distance between the sensors with a factor of 4.3 in Figure 4.2 leads to the layout shown in Figure 4.14. Keeping the rest of the simulation as before. The

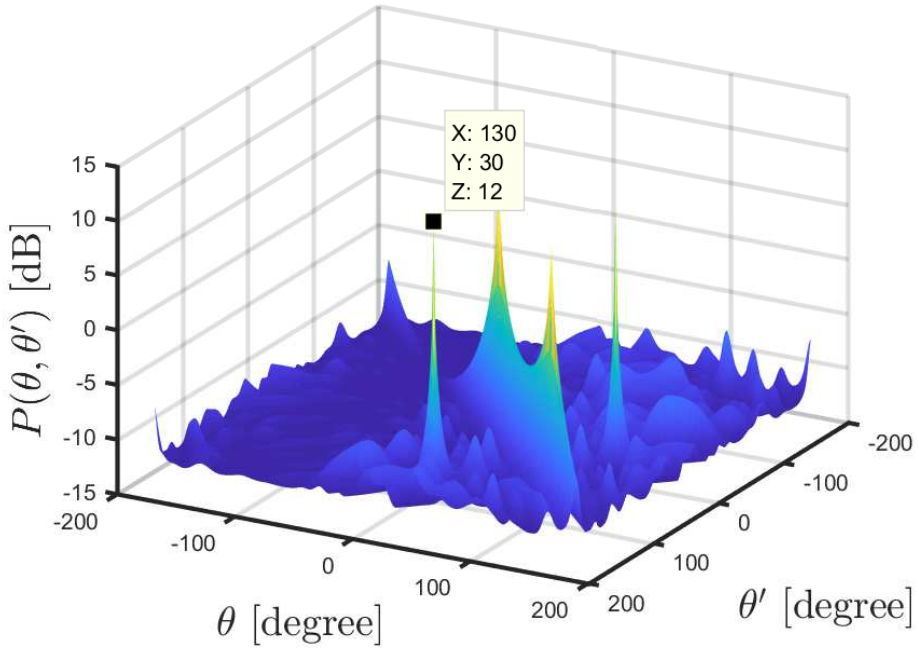


Figure 4.12: The AVS response with the big array-aperture. The peaks are located at (30,30), (30,130), (130,30) and (130,130).

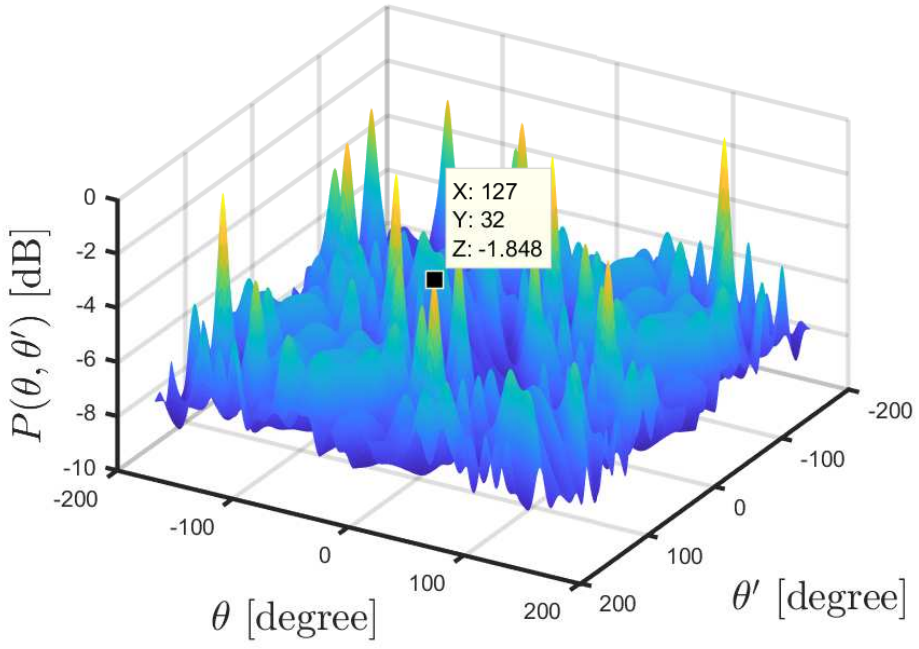


Figure 4.13: The APS response with the big array-aperture. The peaks are located at more or less random positions.

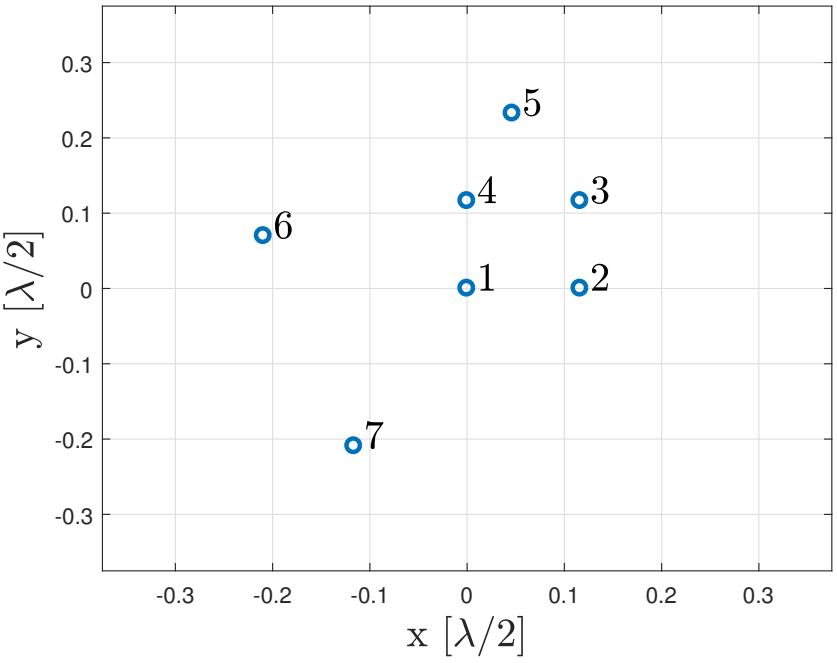


Figure 4.14: The array-aperture with decreased inter-sensor spacing.

results of the AVS method described here are shown in Figure 4.15. Although the peaks are less sharp than before, these still are clear and at the correct spots. So we conclude that for a smaller array it is still possible to retrieve to the DOAs. Comparing this with Caos method of which the results are shown in Figure 4.16. Observing this figure we see that there is no clear peak and that the peaks are not at the correct spots.

Note that changing inter-sensor spacing would have the same effect as changing the frequency of the signals. As the found DOAs appear to be independent of the spacing between the sensors, when two sources are present, we assume the result to be independent of the frequency as well. This can also be observed by analyzing $\boldsymbol{\alpha}_{i,j}$. Recall the definition of $\boldsymbol{\alpha}_{i,j}$:

$$\boldsymbol{\alpha}_{i,j} = \left(\mathbf{a}_{p,i} \odot \mathbf{a}_{p,j}^{*}\right) \otimes \mathbf{h}_i \tag{4.108}$$

The correct DOAs will be found, if all $\boldsymbol{\alpha}_{i,j}$s are orthogonal to each other. So for the purpose of no ambiguity the array-aperture should be designed in such a way that for the desired frequencies all $\boldsymbol{\alpha}_{i,j}$s are orthogonal to each other for all possible combinations of DOAs. This is the case when there are less then $K < 3$ sources. For $K \geq 3$ sources there might appear ambiguities as the $\mathbf{h}_i$-vectors span there maximum space. When these ambiguities appear and how to design the array to keep it unambiguous is an open question for further research. However simulation results suggest that scaled variants of the array-configuration in Figure (4.2) are unambiguous.

The method discussed in this section is independent of clock errors and is more

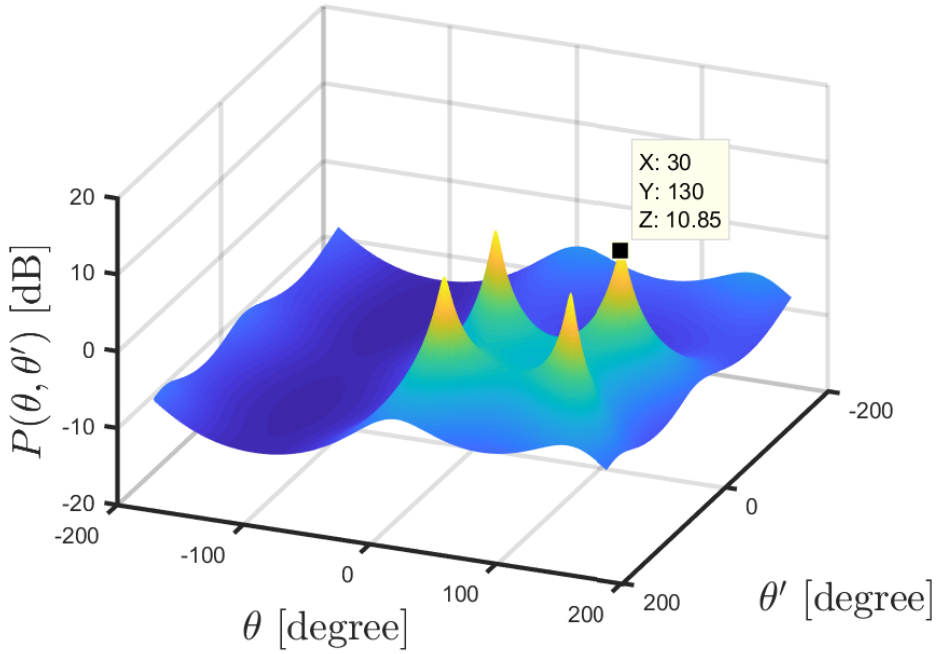


Figure 4.15: The AVS response with the small array-aperture. The peaks are located at (30,30), (30,130), (130,30) and (130,130).

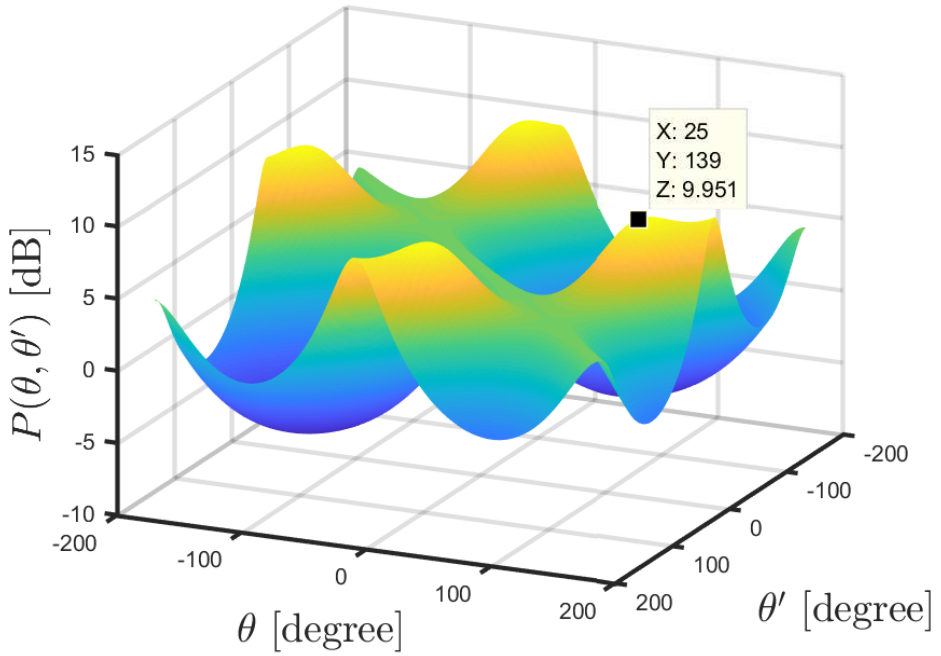


Figure 4.16: The APS response with the small array-aperture.

robust against the assumptions than the method discussed in Section 4.2. Despite this method seems to work well, a 2D-search is necessary in order to retrieve the DOAs. As a 2D search is computable expensive this is not desired, in the next section we propose an iterative method which uses a 1D-search multiple times. This method, however, is not completely independent of clock errors but has a certain maximum error for which it is able to correct. A way to overcome this problem, is by firstly, find an estimate for the DOA using the method described in this section with a sparse grid. Then, secondly, use the estimated DOAs as an input in the method of the next section while using a more dense grid.

As a summary, the properties of the method discussed in this section are listed below:

- There is no ambiguity within the interval $[-180, 180°]$.
- The method holds for every array configuration when $K \leq 2$. It is not clear how the method behaves for $K \geq 3$, this is a question which is still open for research.
- When only one signal is present, the method solely relies on the information of the velocity channels.
- The method relies on the second order statistics of the received data.
- The method uses a two dimensional search which makes the method computational expensive.
- For K sources, the method needs at least $K(K-1) + \min\{K, 3\} + 1$ channels.

4.4 An Iterative Approach

Over the course of time many DOA estimation algorithms are developed that are robust against phase errors (e.g. clock errors). However, most of these methods depend on the phase errors and might fail if they become to big. The method we described in the previous section is independent of phase errors. However it is computational expensive when the DOAs need to be estimated with a high accuracy. Most of the robust DOA estimation algorithms have an iterative approach and alter between estimating phase errors and estimating the DOAs. So if we retrieve a low resolution DOA estimation from the Cao-based method, we could used this as an input of the phase estimation of the iterative method. This is schematically shown in Figure 4.17.

As mentioned before, there are many iterative methods developed over the years. The method that should be used in practice depends on the situation, for example the Reiterative Superresolution algorithm develped by Blunt et al. in [22] could be used. This method behaves well when there are only a few sample points available. The l_p-MUSIC algorithm developed by Zeng et al. in [23] operates well within impulsive noise environments. It should be noted that these algorithms are developed for APS system although the extension towards AVS is usually quite straight forward.

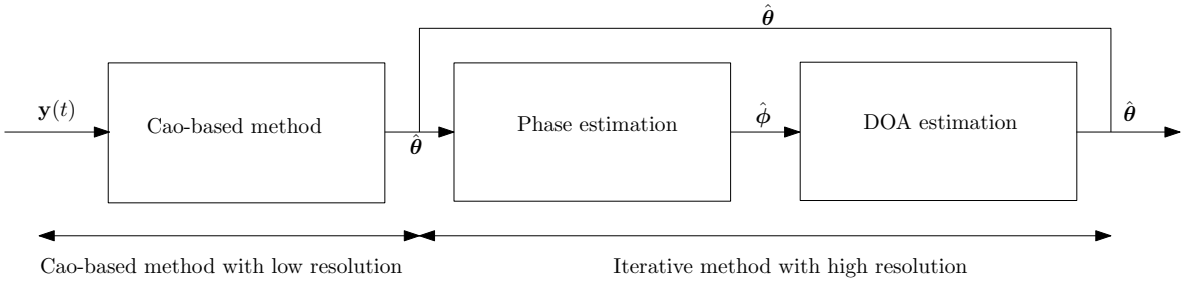


Figure 4.17: Schematic overview of the combined strategy.

In this section we will discuss the more general method developed by Weiss and Friedlander in [24] which is well known as the Weiss and Friedlander (WF) algorithm. The WF-algorithm relies on the same assumptions on the data as we did before and is therefore a logical choice. We will first discuss the phase estimation after which the DOA estimation is discussed. This section will be concluded with simulations of the combined strategy.

It should be noted that not all methods for joined DOA and phase estimation are iterative. If the sensors are placed in an ULA one could for example also used the blind calibration method developed by Krishnaprasad et al. in [25].

4.4.1 Phase Estimation

Liu et al also proposes an algorithm for phase estimation in [18], this algorithm is proven to be independent of phase errors. On further inspection it follows that Liu's estimator is equal to the that of the WF-algorithm. Although the expansion of this method from APS to AVS is quite straightforward we will discuss this fully to proof that the phase independence holds for an AVS-array as well.

Note that we could write $\boldsymbol{R}$ as:

$$\boldsymbol{R} = \boldsymbol{\Pi} \boldsymbol{A} \boldsymbol{R}_s \boldsymbol{A}^H \boldsymbol{\Pi}^H + \sigma_\eta^2 \boldsymbol{I} \tag{4.109}$$

$$\boldsymbol{\Pi}^H \boldsymbol{R} \boldsymbol{\Pi} = \boldsymbol{A} \boldsymbol{R}_\mathrm{s} \boldsymbol{A}^H + \sigma_\eta^2 \boldsymbol{I} \tag{4.110}$$

Observe that $\boldsymbol{\Pi}^H \boldsymbol{R} \boldsymbol{\Pi}$ is independent of phase errors. By writing the eigendecomposition of $\boldsymbol{R}$:

$$\boldsymbol{R} = \sum_{m=1}^{3M} = \beta_m \mathbf{u}_m \mathbf{u}_m^H \tag{4.111}$$

we find the eigendecomposition of $\boldsymbol{\Pi}^H \boldsymbol{R} \boldsymbol{\Pi}$ as:

$$\boldsymbol{\Pi}^H \boldsymbol{R} \boldsymbol{\Pi} = \sum_{m=1}^{3M} \beta_m \boldsymbol{\Pi}^H \mathbf{u}_m \mathbf{u}_m^H \boldsymbol{\Pi} \tag{4.112}$$

By defining the eigenvalues of $\boldsymbol{A} \boldsymbol{R}_\mathrm{s} \boldsymbol{A}^H$ as $\{\xi_m\}_{m=1}^{3M}$. from (4.110) we have that the eigenvalues of $\boldsymbol{\Pi}^H \boldsymbol{R} \boldsymbol{\Pi}$ are equal to $\left\{\xi_m + \sigma_\eta^2\right\}_{m=1}^{3M}$. From which we conclude that the

eigenvalues $\{\beta_m\}_{m=1}^{3M}$ are independent of phase errors. Define:

$$\boldsymbol{V}_0 = \begin{bmatrix} \mathbf{u}_{k+1} & \dots & \mathbf{u}_{3M} \end{bmatrix}, \tag{4.113}$$

$$\boldsymbol{V}_1 = \boldsymbol{\Pi}^H \boldsymbol{V}_0, \tag{4.114}$$

$$\boldsymbol{Q}_1 = \sum_{k=1}^{K} \operatorname{diag}\left\{\mathbf{a}(\theta_k)^*\right\} \boldsymbol{V}_1 \boldsymbol{V}_1^H \operatorname{diag}\left\{\mathbf{a}(\theta_k)\right\}. \tag{4.115}$$

Denote that $\boldsymbol{V}_1$ and $\boldsymbol{Q}_1$ are both independent of phases errors. Now, define $\mathbf{z}$ as:

$$\mathbf{z} = \frac{\boldsymbol{\Pi}\boldsymbol{Q}_1^{-1}\mathbf{e}_m}{\mathbf{e}_m^T\boldsymbol{Q}_1^{-1}\mathbf{e}_m} \tag{4.116}$$

If we define $\mathbf{q}_m$ as the m^{th} column of $\boldsymbol{Q}_1^{-1}$ and q_{mm} the m^{th} element of $\mathbf{q}_m$ we find:

$$\mathbf{z} = \frac{\boldsymbol{\Pi}\mathbf{q}_m}{q_{mm}} \tag{4.117}$$

Taking the complex argument of this will give us our estimate of the phase errors:

$$\hat{\boldsymbol{\beta}}_3 = \arg\{\mathbf{z}\} \tag{4.118}$$

$$= \boldsymbol{\beta} \otimes \mathbf{1}_3 + \arg\left\{\frac{\mathbf{q}_m}{q_{mm}}\right\} \tag{4.119}$$

where $\hat{\boldsymbol{\beta}}_3 = \hat{\boldsymbol{\beta}} \otimes \hat{\mathbf{1}}_3$By the definition of $\boldsymbol{Q}_1$ in (4.115) it follows that $\boldsymbol{Q}_1$ is hermitian and therefor $\boldsymbol{Q}_1^{-1}$ is hermitian and q_{mm} is thus real. So our estimate reduces to:

$$\hat{\boldsymbol{\beta}}_3 = \boldsymbol{\beta} \otimes \mathbf{1}_3 + \arg\{\mathbf{q}_m\} \tag{4.120}$$

As $\boldsymbol{Q}_1$ is independent of phase errors we have that $\mathbf{q}_m$ is independent of phase errors and we can conclude that the error of our estimate is independent of phase errors. Note that we do not have direct access to $\boldsymbol{Q}_1$ nor $\boldsymbol{\Pi}$ but if we observe that $\boldsymbol{\Pi}^H\mathbf{e_m} = \mathbf{e_m}$ we can write $\mathbf{z}$ as follows:

$$\mathbf{z} = \frac{\boldsymbol{\Pi}\boldsymbol{Q}_1^{-1}\boldsymbol{\Pi}^H\mathbf{e}_m}{\mathbf{e}_m^T\boldsymbol{\Pi}\boldsymbol{Q}_1^{-1}\boldsymbol{\Pi}^H\mathbf{e}_m} = \frac{\boldsymbol{Q}_0^{-1}\mathbf{e}_m}{\mathbf{e}_m^T\boldsymbol{Q}_0^{-1}\mathbf{e}_m} \tag{4.121}$$

where we used

$$\boldsymbol{Q}_0 = \boldsymbol{\Pi}\boldsymbol{Q}_1\boldsymbol{\Pi}^H = \sum_{k=1}^{K} \operatorname{diag}\left\{\mathbf{a}(\theta_k)^*\right\} \boldsymbol{V}_0 \boldsymbol{V}_0^H \operatorname{diag}\left\{\mathbf{a}(\theta_k)\right\} \tag{4.122}$$

which we can estimate if we have an estimate for the DOAs. Note that this algorithm estimates $\boldsymbol{\beta}_3$ rather than $\boldsymbol{\beta}$ we can easily find this estimate $\hat{\boldsymbol{\beta}}$ by averaging over the corresponding entries of $\hat{\boldsymbol{\beta}}_3$. A summary of this algorithm is given in algorithm 1.

Algorithm 1 Phase Errors

1: **procedure** FIND PHASE ERRORS
2: Perform eigenvalue decomposition $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^H = \boldsymbol{R}$
3: Find the nullspace of $\boldsymbol{R}$, $\boldsymbol{V}_0$
4: Compute $\boldsymbol{Q}_0$ with (4.122)
5: Compute $\mathbf{z}$ with (4.121)
6: $\boldsymbol{\beta}_3 = \arg\{z\}$
7: Find $\boldsymbol{\beta}$ by averaging over $\boldsymbol{\beta_3}$

4.4.2 DOA Estimation

From previous sections it is clear that we can use the MUSIC methodology where we permute the steering vectors with the phase delays in order to find an estimate of the DOA i.e. minimizing

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{K} \left|\left| \boldsymbol{V}_0^H \hat{\boldsymbol{\Pi}} \mathbf{a}(\theta_k) \right|\right|^2 \tag{4.123}$$

where $\hat{\boldsymbol{\Pi}}$ is an estimate of $\boldsymbol{\Pi}$. From which we can conclude that we can find an estimate of the DOA by performing a search:

$$\hat{\theta} = \arg\max_{\theta'} \frac{1}{\mathbf{a}(\theta')^H \hat{\boldsymbol{\Pi}}^H \boldsymbol{V}_0 \boldsymbol{V}_0^H \hat{\boldsymbol{\Pi}} \mathbf{a}(\theta')}. \tag{4.124}$$

Now, by combining the procedure described in algorithm 1 and equation (4.124) we can iteratively estimate the DOAs and the phase (e.g. clock) errors where we use some starting condition. Based on the starting condition the iterative estimate may or may not converge to the optimal solution or to a local minimum.

4.4.3 Simulation of the combined strategy

In this section we will simulate the combined strategy as given schematically in Figure 4.17. We will use the array with the increased sensor spacing where 3 sources are impinging this array at $\theta_1 = -50.2°$, $\theta_2 = 31.1°$ and $\theta_3 = 132.8°$. All other settings are as before. For the first step, the Cao-based method, described in Section 4.3 where we use a grid-search with 1° spacing between the grid points. After which we will use the result of this method as an input of the iterative method described above where we will use a grid with 0.1° spacing between the grid points. Coa-based algorithm leads to a spectrum shown in Figure 4.18. It is easy to see that we can extract the DOAs $\hat{\boldsymbol{\theta}} = \begin{bmatrix} -50° & 31° & 133° \end{bmatrix}^T$ from this spectrum.

The WF-algorithm is set with a grid of 0.1° spacing between the grid points and the algorithm is set to stop if the DOAs of the previous iteration is equal to the DOA of the current iteration. Running this with the DOAs obtained previously leads to the spectrum given in Figure 4.19. Note that the algorithm stopped after 2 iterations (i.e. the first iteration was already spot on) and we could extract the DOAs as $\hat{\boldsymbol{\theta}} =$

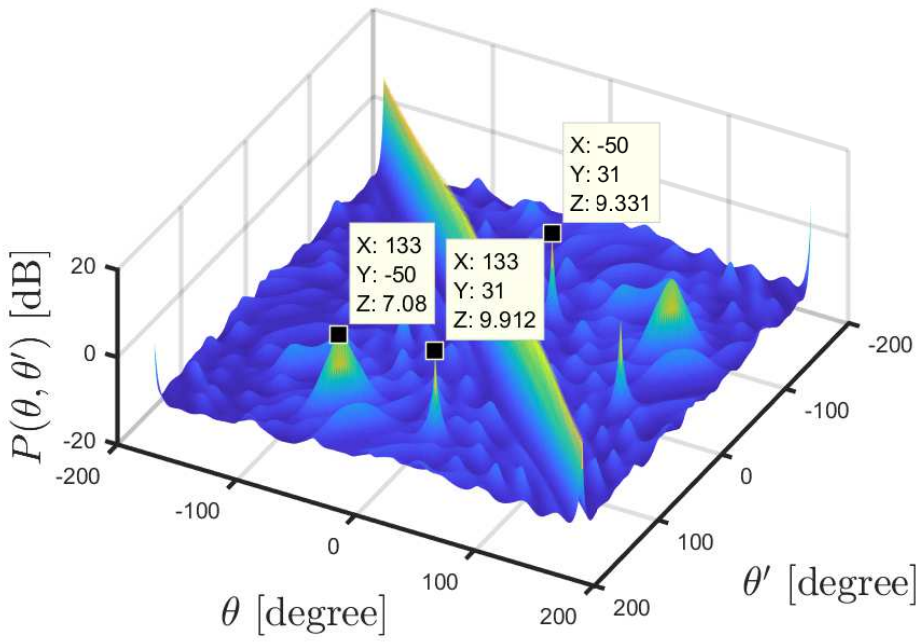


Figure 4.18: The spectrum obtained through the method described in Section 4.3.

$\begin{bmatrix}-50.2^\circ & 31^\circ & 132.8^\circ\end{bmatrix}^T$ which is, although $\theta_2 = 31.1^\circ$ is 0.1° off, as desired.

In order to show that, in this case, the method proposed in Section 4.3 is necessary to obtain a guess that could be used as an input of WF-algorithm we also give the spectrum that is obtained when we take as initial guess that the clocks in the array are perfectly synchronized. This spectrum is given in Figure 4.20. In this figure we observe that the the algorithm takes 13 iterations to converge to a incorrect minimum which proves that the WF-algorithm needs a more appropriate first guess in order to converge to the desired result.

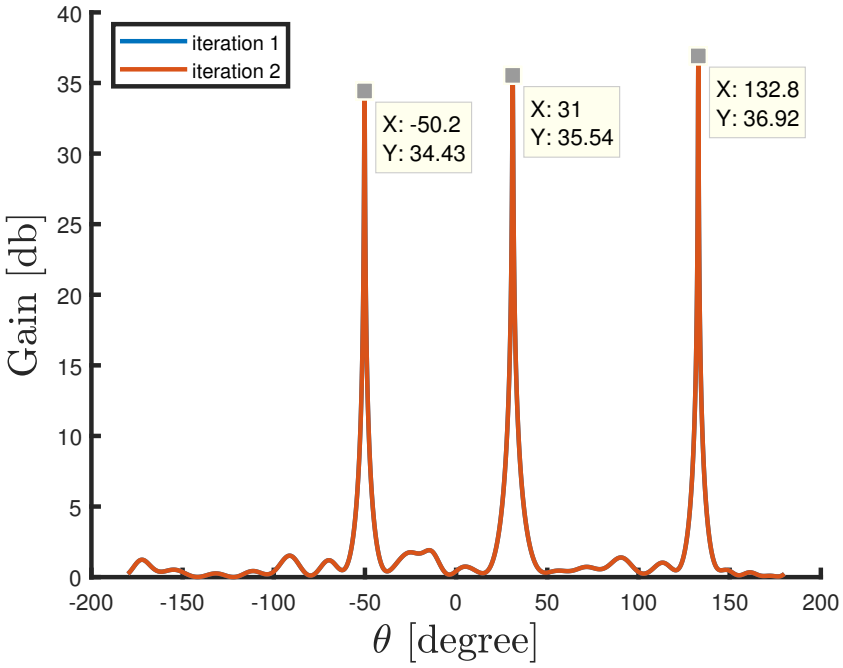


Figure 4.19: The spectrum obtained through the WF-method while using the peaks of Figure 4.18 as an input.

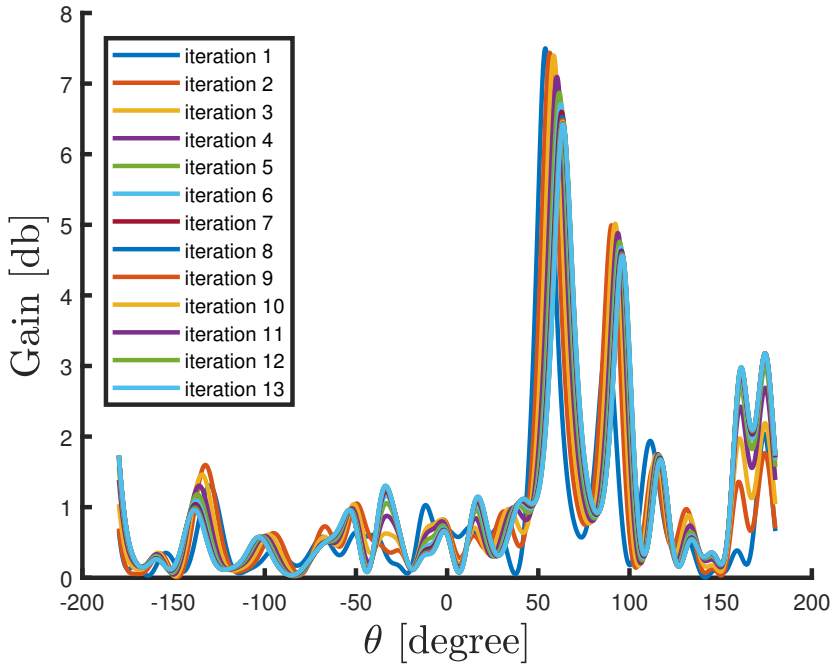


Figure 4.20: The spectrum obtained through the WF-method while using the assumption that there are no clock errors present as an initial guess.

5 Experimental Results

In this chapter, the experimental setup and the results that follow are discussed. In order to observe the difference of performance between a synchronized and inaccurate synchronized array, the experiments are conducted with both synchronized and unsynchronized sensors. The performance of the Liu-based method developed in Section 4.2 and the Cao-based method from Section 4.3 are both discussed.

Firstly, the experimental setup is discussed. Secondly simulations are carried out in order to have a reference result. Thirdly, the results of the experiment with the synchronized sensors are given and reviewed. Lastly, the results of the unsynchronized array are shown and discussed.

5.1 Experimental Setup

The experiments are conducted in the anechoic chamber of the Applied Physics faculty at the TU Delft. An anechoic chamber is a room where the walls are designed to absorb all sound waves impinging on them. These walls can be seen in Figure 5.1. The walls are made of pillions that are placed and constructed in such a way that almost

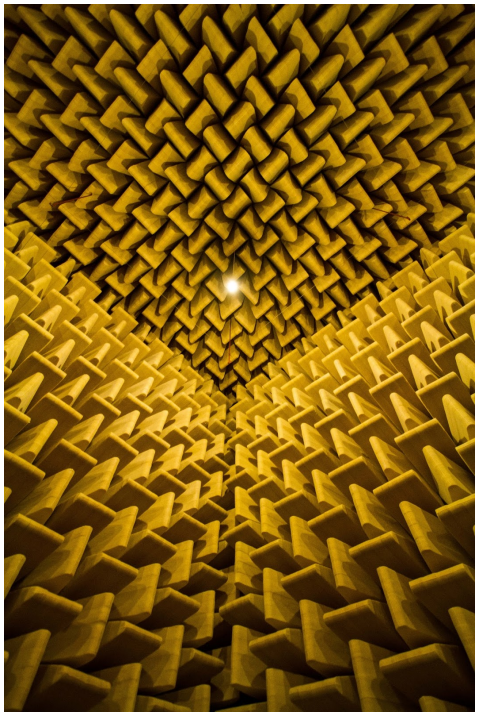


Figure 5.1: The walls in the anechoic chamber.

all acoustic waves impinging them will be absorbed.

In this room multiple AVSs and speakers are placed such that the new developed algorithm can be tested. The placement of the AVSs and speaker is shown schematically in Figure 5.2. In the setup five sensors are arranged in a cross-like setting with inter sensor distance $d = 52.5$ cm. Two speakers are placed at a distance of $l = 360$ cm from the center of the array. The angle that speaker one makes with the array will defined as $\theta_1 = 0°$ which implies that the angle of speaker 2 is at $\theta_2 = -45°$.

During the experiments the speakers are set to generate white Gaussian noise which will be measured and stored by the AVSs. The speakers are able to produce acoustic signals between 400 and 8000 Hz. Recall that the methods developed by Liu [18] and Cao [19] are only said to be unambiguous if the there are sensors placed on the vertices of a square with sides smaller than $\frac{\lambda}{4}$. Although there are sensors placed on the vertices of a square in this setup, the sides of the square are of length $52.5\sqrt{2}$ cm. So the frequencies for which the DOA estimation is unambiguous are given by:

$$f = \frac{c}{\lambda} \leq 115.5\text{Hz}. \tag{5.1}$$

As the used speakers are not able to produce this frequency we need to choose another. The signal frequency that is used during this chapter is $f = 460$Hz. This corresponds with a Nyquist under sampling rate of about a factor two, and an under sampling rate for the algorithms described by Liu and Cao of about a factor four. As this frequency does not align with the restrictions of both Liu's and Cao's method these methods will not be discussed in this chapter.

The measurements are carried out with two different types of sensors:

USP The Ultimate Sound Probe (USP) is a commercial acoustic vector sensor manufactured by Microflown Technologies. These sensors are developed in order to measure acoustic fields while the USPs are placed in an array setting. Because of this, there is a well working synchronization algorithm present in the sensors which enables us to test the algorithms on a synchronized array.

AMMS The Acoustic Multi-Mission Sensor (AMMS) is a sensor developed for situational awareness manufactured by Microflown Avisa. this sensor is designed to operate as a stand alone sensor. Currently, development is going on such that the sensors in the future are able to operate in array settings. By means of thi a non-accurate synchronization algorithm is implemented. The synchronization between these sensors is not accurate enough to perform the beamforming algorithms discussed in Section 2.2. This enables us to test the algorithm on inaccurate synchronized data.

By testing the algorithms on two different sensors we are able to show that the algorithms behave equally in a synchronized setting, as well as with unsynchronized data.

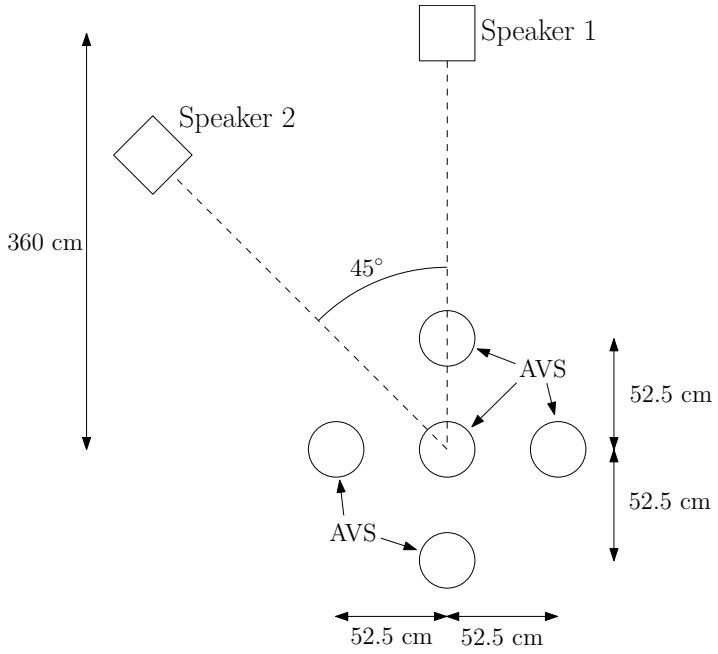


Figure 5.2: Schematic overview of the setup.

5.2 Simulation of the Experimental Setup

In this section simulations of the experimental setup are performed and evaluated. Simulating the results will give us insight in what results we expect, as the experiments are conducted in a under sampled setting it might happen that grating lobes appear. It would be interesting to see if the side lobes in the simulations have the same location as the side lobes in the measurement results.

For the simulations a SNR of 12dB and 16000 data point are used (i.e. one second of data). In the simulation no clock errors are assumed.
The simulation results for the method based on the new measurement vector (i.e. the Liu-based method) are shown in Figure 5.3. This plot shows that the four highest peaks are located at the correct positions. However there is a small side lobe located at $[\theta, \theta'] = [-45°, 45°]$ and $[\theta, \theta'] = [45°, -45°]$. This side lobe is probably due to the under sampling described above.
The simulation results of the Cao-based method are shown in Figure 5.4. This figure is similar to that of the simulation above. It is worth mentioning that the side lobe in this simulation is higher than in the previous. The side lobes in the previous simulation are about 28 dB lower then the peak at $[\theta, \theta'] = [-45, 0]$, while the side lobes in this simulation are only 20 dB lower.

5.3 Experimental Results with Synchronized Data

For this experiment the already described USPs are used. However due to the measurement setup the far-field assumptions from section 2.1 does not hold anymore. In this section, firstly, a new datamodel is introduced which can be used for near-field signals.

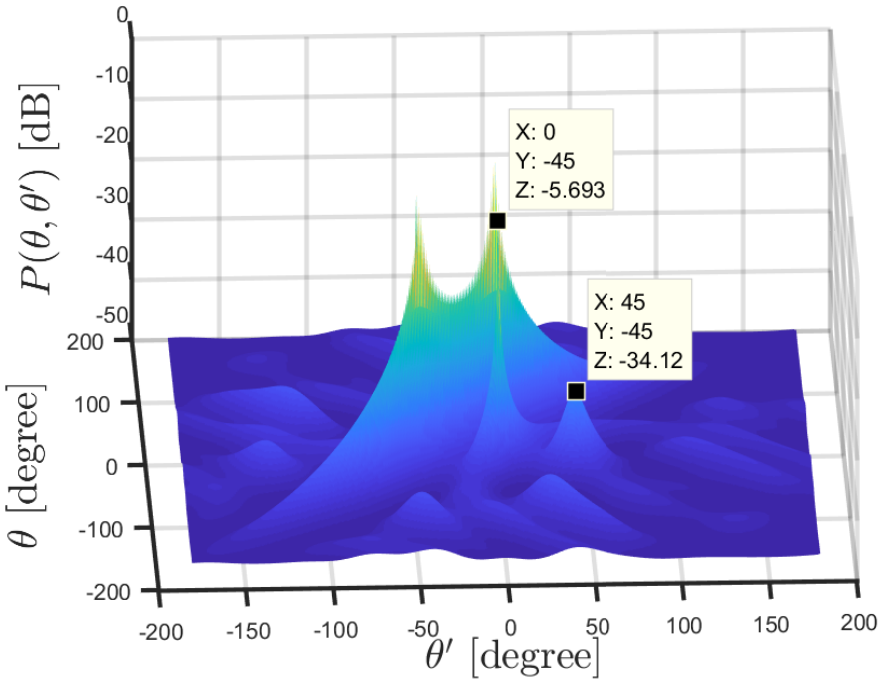


Figure 5.3: The simulation results for the method suggested in Section 4.2, the peaks in the half space $\theta' \geq \theta$ are located at $[\theta, \theta'] = [-45, -45]$; $[0, 0]$ $[-45, 0]$ and a side lobe appears at $[-45, 45]$.

Secondly the results of the experiment are shown and analyzed.

5.3.1 Near-Field Approximation

Due to that the sources are located in the near-field, the acoustic wave impinging the array can not be modeled as a plane wave anymore. Instead, circular waves are impinging the array. As a consequence θ is not equal for every sensor. It is clear from Figure 5.5 that the angles $\tilde{\theta}_1$ and $\tilde{\theta}_2$ are not the same as the angles observed by the other sensors. For simplicity we will use the center sensor as a reference. Due to near-field effects both the $\mathbf{a}_\text{p}$ and the $\mathbf{h}$-vectors change. In order to find the new pressure steering vector we first describe the new TDOA where we take the center of the array as a reference. It is easy to derive that the new TDOA becomes:

$$l_i(\theta) = l - \sqrt{l^2 + ||\mathbf{r}_i||_2^2 - 2l\left(r_{i,\text{y}} \cos\theta + r_{i,\text{x}} \sin\theta\right)}, \tag{5.2}$$

where l_i is the TDOA in meters. Define vector $\mathbf{l}(\theta)$ as the vector containing all the DOAs: $\mathbf{l}(\theta) = \begin{bmatrix} l_1(\theta) & l_2(\theta) & \dots & l_M(\theta) \end{bmatrix}^T$. Now, the new pressure steering vector $\tilde{\mathbf{a}}_\text{p}$ could be expressed as:

$$\tilde{\mathbf{a}}_\text{p}(\theta) = e^{\text{j}k\mathbf{l}(\theta)}. \tag{5.3}$$

Due to that the angle at which a signal impinges a sensor is different for every sensor,

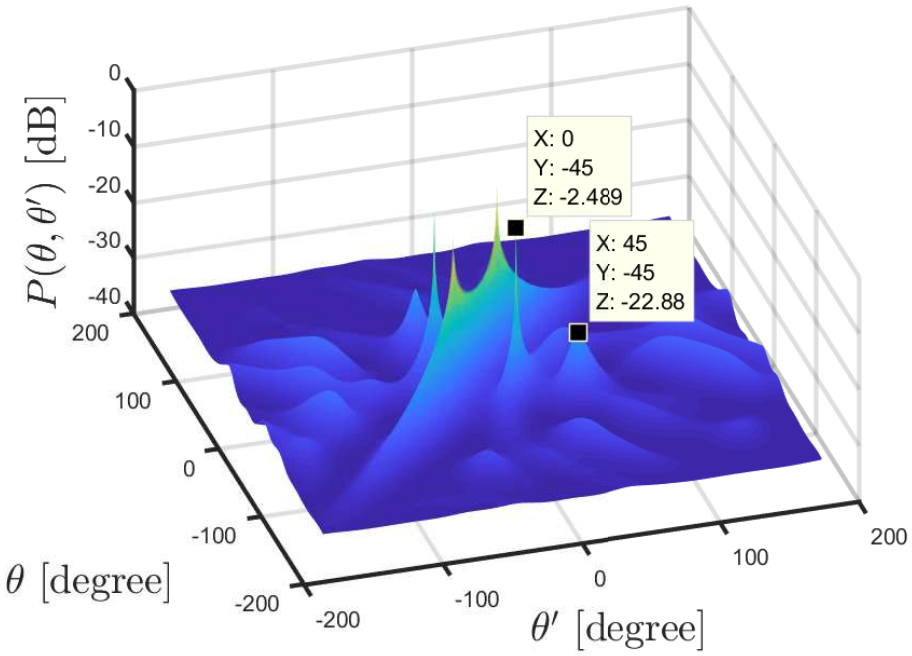


Figure 5.4: The simulation results for the method suggested in Section 4.3, the peaks in the half space $\theta' \geq \theta$ are located at $[\theta, \theta'] = [-45, -45]$; $[0, 0]$ $[-45, 0]$ and a side lobe appears at $[-45, 45]$.

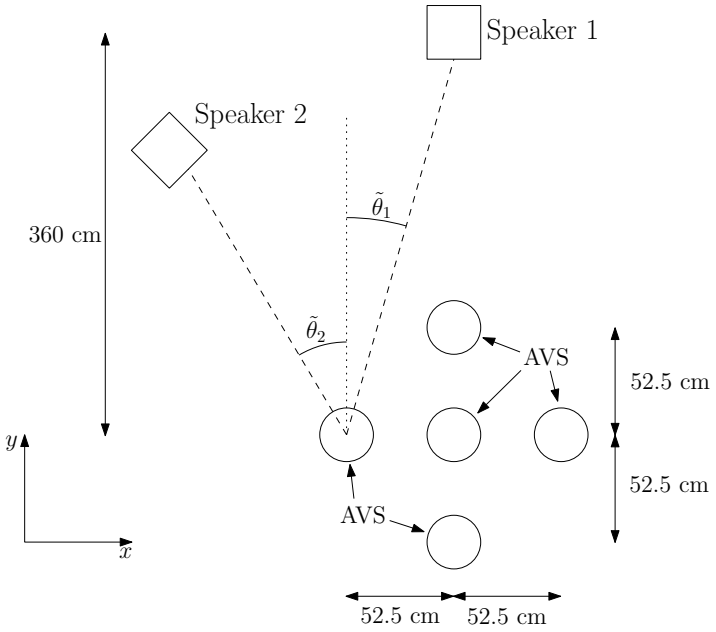


Figure 5.5: Near-Field approximation.

the $\mathbf{h}(\theta)$ will differ for all sensors. Note that $\bar{\theta}_i(\theta)$, the angle between a signal with DOA θ and sensor i, could be expressed as:

$$\bar{\theta}_i(\theta) = \text{atan2}\left(l\sin\theta - r_{i,\text{x}}, l\cos\theta - r_{i,\text{y}}\right), \tag{5.4}$$

where $\text{atan2}(x, y)$ denotes the four-quadrant inverse tangent. The new $\mathbf{h}$-vector for sensor i could be written as:

$$\tilde{\mathbf{h}}_i(\theta) = \begin{bmatrix} 1 & \cos\left(\bar{\theta}_i(\theta)\right) & \sin\left(\bar{\theta}_i(\theta)\right) \end{bmatrix}^T, \tag{5.5}$$

where $\tilde{\mathbf{h}}_i(\theta)$ is the near-field $\mathbf{h}$-vector corresponding to sensor i. Define $\tilde{\mathbf{h}}(\theta)$ the vector containing all $\mathbf{h}_i(\theta)$-vectors: $\tilde{\mathbf{h}}(\theta) = \begin{bmatrix} \mathbf{h}_1^T(\theta) & \mathbf{h}_2^T(\theta) & \dots & \mathbf{h}_M^T(\theta) \end{bmatrix}^T$. The steering vector can now be expressed as:

$$\tilde{\mathbf{a}}(\theta) = \left(\tilde{\mathbf{a}}_{\text{p}}(\theta) \otimes \mathbf{1}_3\right) \odot \tilde{\mathbf{h}}(\theta). \tag{5.6}$$

Thus the steering vector used by the new algorithms described in the previous chapter ($\boldsymbol{\alpha}(\theta_1, \theta_2)$) could be expressed as:

$$\tilde{\boldsymbol{\alpha}}(\theta_1, \theta_2) = \tilde{\mathbf{a}}(\theta_1) \odot \left(\tilde{\mathbf{a}}_{\text{p}}(\theta_2) \otimes \mathbf{1}_3\right) \tag{5.7}$$

where $\tilde{\boldsymbol{\alpha}}(\theta_1, \theta_2)$ is the near-field approximation of $\boldsymbol{\alpha}(\theta_1, \theta_2)$.

5.3.2 Results

In this section we will discuss the results of the measurement with the synchronized USPs. The setup is shown in Figure 5.6. The results of the Liu-based algorithm are shown in Figure 5.7 and for the Cao-based method in Figure 5.8.

Observe that the diagonal lines ($\theta = \theta'$) do not have two separable peaks, such that we can not estimate the DOAs by observing only this line. This is likely to be the result of calibration issues. Furthermore, we see in both plots the highest peaks at $(4°, -42°)$ and $(-42°, 4°)$, which is a few degrees from the expected angles. This can be due to calibration or placement issues.

Moreover, we observe that the side lobes in the experiments are higher than the side lobes we had in the simulation. In the simulation the DOAs where at $(0°, -45°)$ exactly, while in the experiment the DOAs are somewhere around these angles. As the DOAs in the simulation are exactly on the grid, the correct peaks will be higher. Due to that the pseudo spectra are relative, the peaks of the side lobes will be lower in the simulation. Another reason could be that the assumption of co-located sensor components is not valid in this measurement setup.

In Figure 5.9 the results are shown when the data from this measurement is applied to the classical beamformer discussed in Section 2.2. In this figure we see that the classical beamformer is capable of finding the two DOAs at this frequency. However, there is a big side lobe around $-140°$.

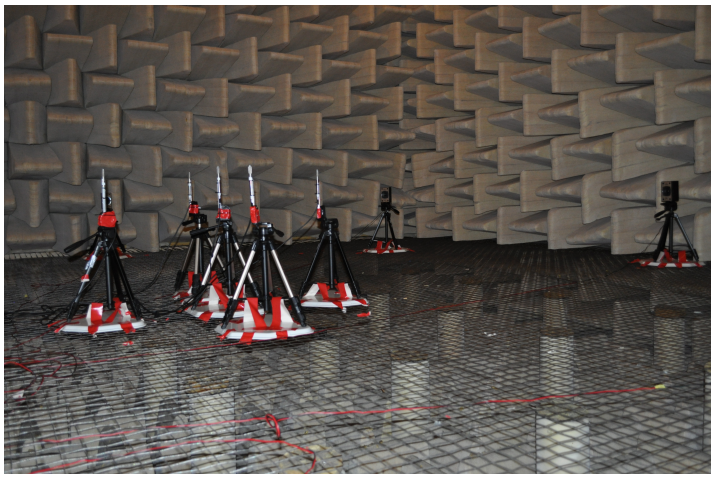


Figure 5.6: The experimental setup with USPs.

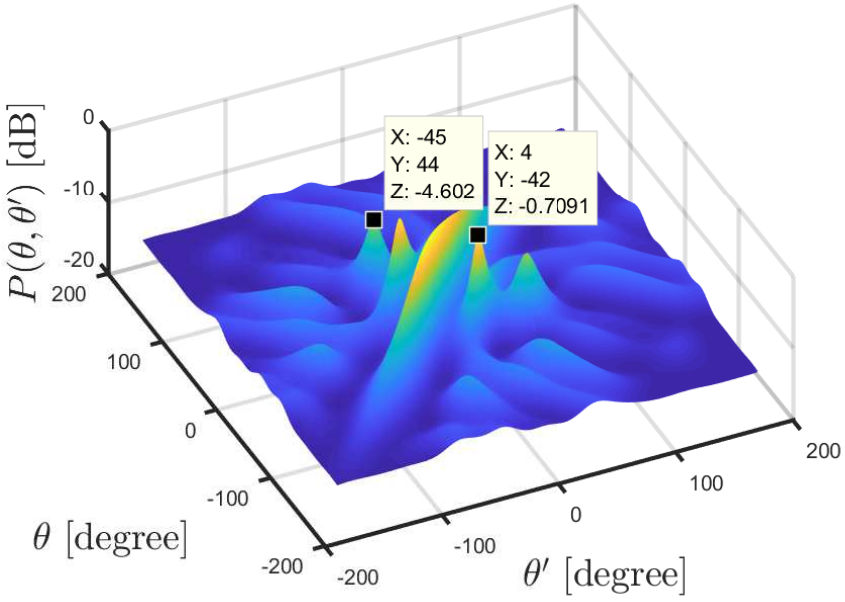


Figure 5.7: The results of the experiment with synchronized sensors with the Liu-based method. Peaks are located at $(-42°, 4°)$ and $(4°, -42°)$. Side lobes are located at $(-45°, 44°)$ and $(44°, -45°)$.

5.4 Experimental Results with Badly Synchronized Data

In this section the results of the measurements with inaccurate synchronized sensors are shown and reviewed. This setup is shown in Figure 5.10. However due to technical issues the data of the most left sensor in Figure 5.2 is not available. Therefore the results shown here are based on the four other sensors only.

The pseudo spectra of the Liu-based method and the Cao-based method are shown

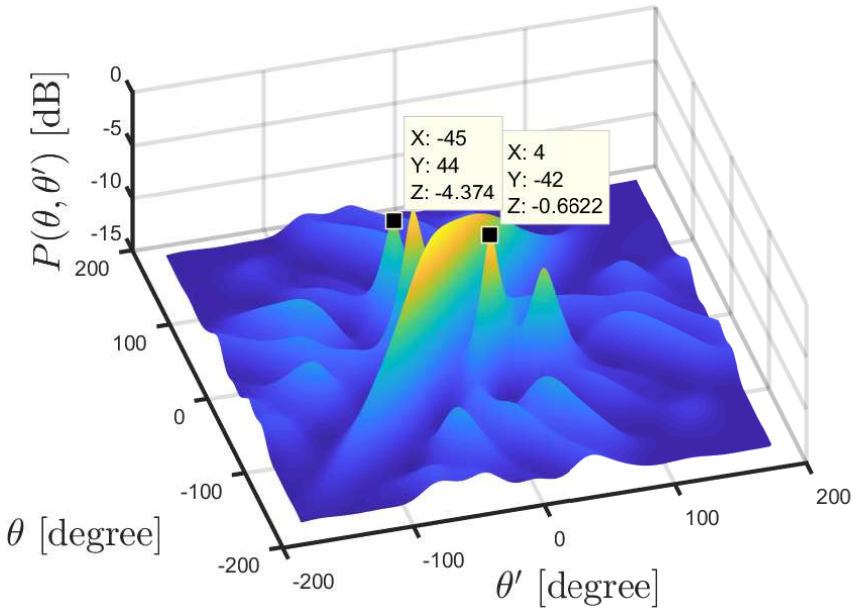


Figure 5.8: The results of the experiment with synchronized sensors with the Cao-based method. Peaks are located at $(-42°, 4°)$ and $(4°, -42°)$. Side lobes are located at $(-45°, 44°)$ and $(44°, -45°)$.

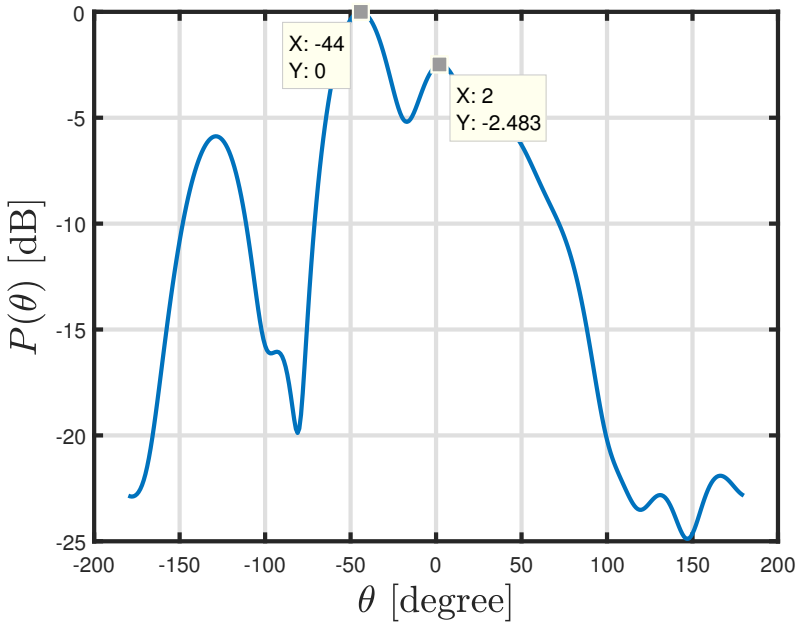


Figure 5.9: The spectrum of the experiment with synchronized data when the classical beamformer is used.

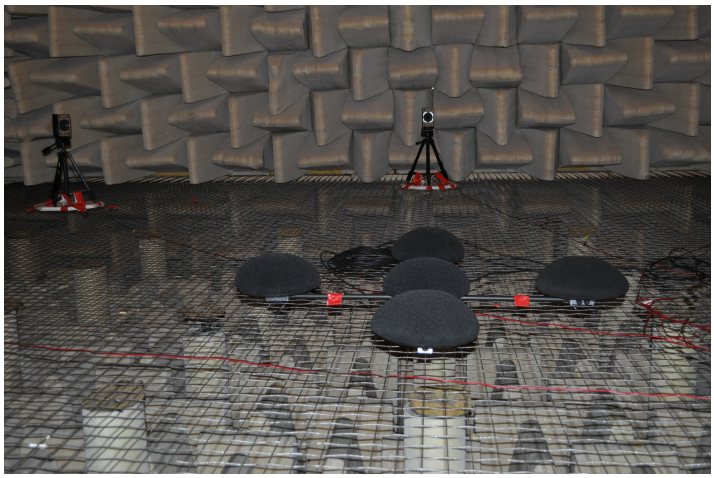


Figure 5.10: The experimental setup with AMMSs.

in Figure 5.11 and Figure 5.12 respectively. The peaks of this setup are less sharp due to that the most left sensor is missing. However, the obtained results are similar to the results of the experiment in the previous section. Thus we conclude that the algorithms derived in Section 4.2 and Section 4.3 both lead to reliable results in this experiment, whether the sensors are well synchronized or not.

For completion the results of the classical beamformer on the unsynchronized data is shown in Figure 5.13. From this figure it is clear that this approach is not suitable for the given dataset.

In Figure 5.14a the results of the WF-method with the previously obtained DOAs as initial guess (i.e. the combined method of Section 4.4) are shown. Figure 5.14b uses the same method with as initial guess that there are no clock errors present. Although both initial guesses eventually converge to the same DOAs, it is clear that the first iteration of the latter is quite far from the final spectrum. If the clock errors become bigger it might occur that the spectrum obtained in the first iteration of this method gets worse. This can lead towards that the optimization will converge towards the wrong DOAs.

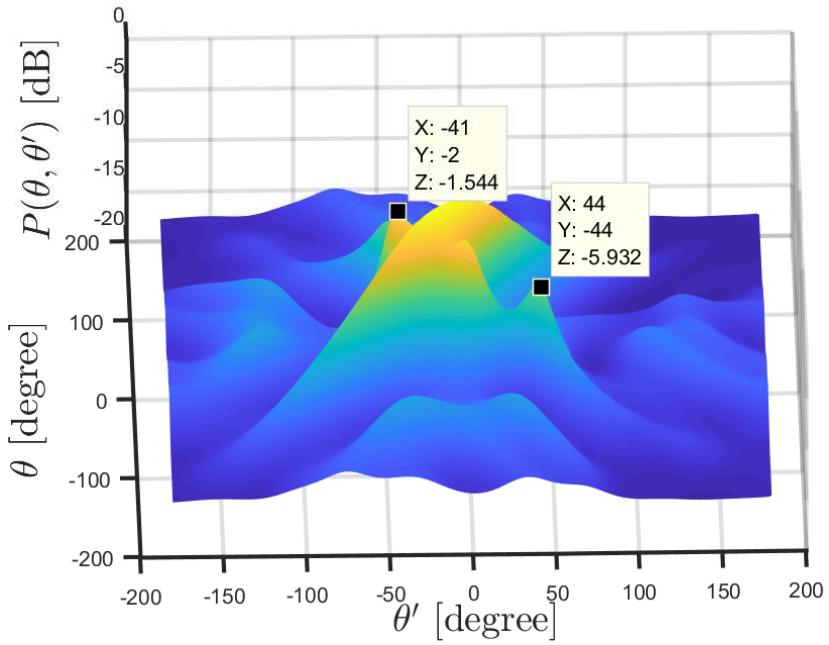


Figure 5.11: The results of the experiment using inaccurate synchronized sensors with the Liu-based method. Peaks are located at $(-2°, -41°)$ and $(-41°, -2°)$. Side lobes are located at $(-44°, 45°)$ and $(44°, -45°)$.

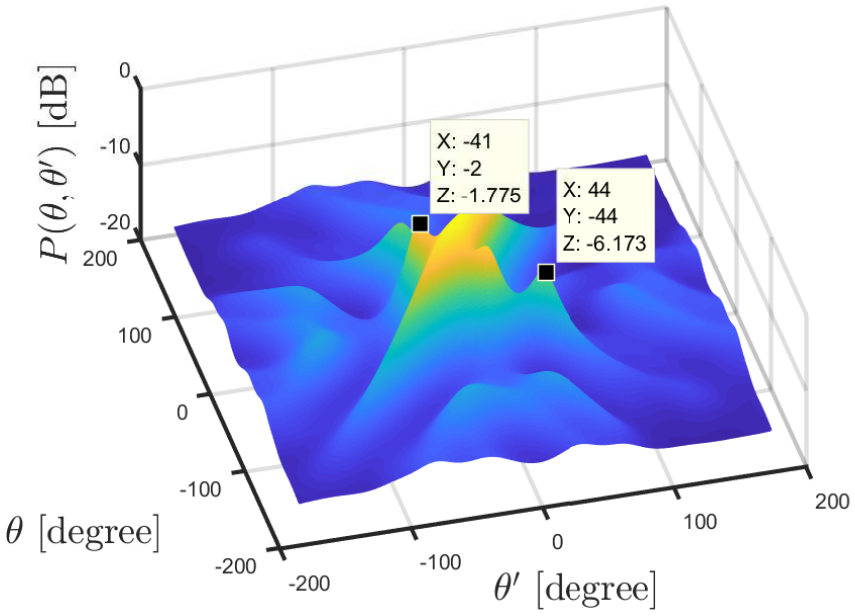


Figure 5.12: The results of the experiment using inaccurate synchronized sensors with the Liu-based method. Peaks are located at $(-2°, -41°)$ and $(-41°, -2°)$. Side lobes are located at $(-44°, 45°)$ and $(44°, -45°)$.

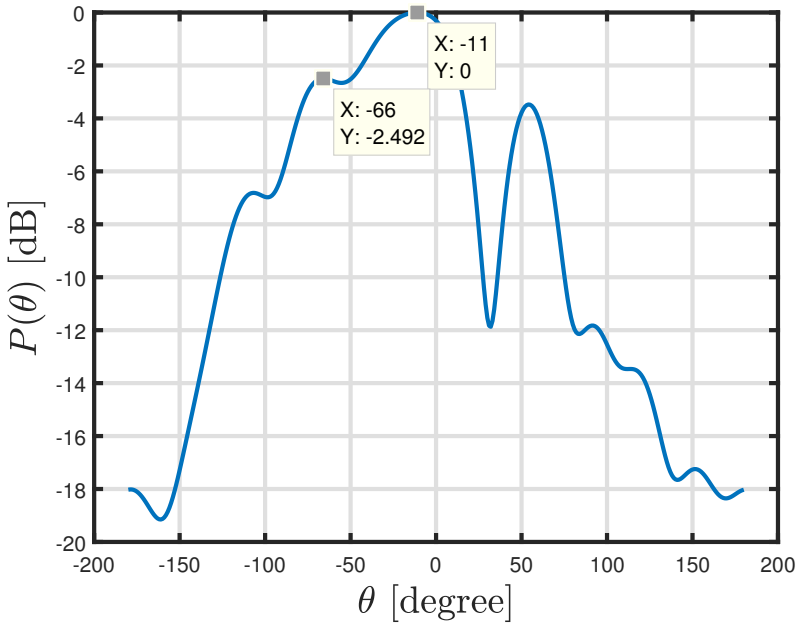
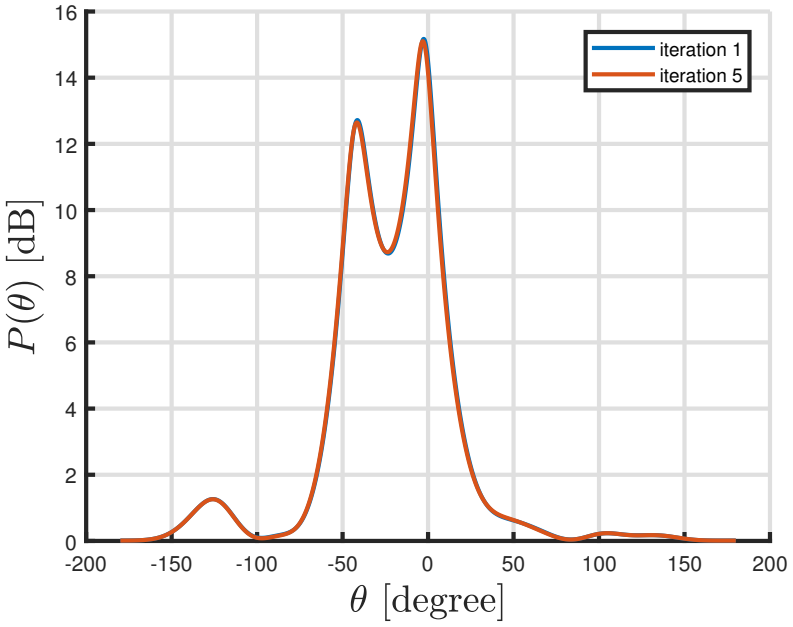
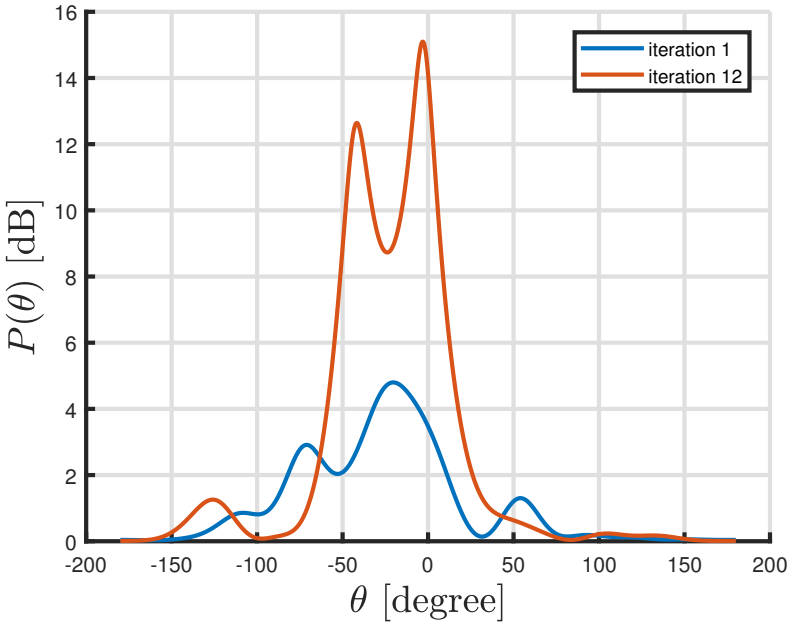


Figure 5.13: The spectrum of the experiment with inaccurate synchronized data when the classical beamformer is used.



(a) Results of the combined method. Peaks of the final iteration are located at -3.1° and -41.7°



(b) Results of the WF-method. Peaks of the final iteration are located at -3.1° and -41.7°

Figure 5.14: Comparison of the combined method and the WF-algorithm.

6 Conclusion and Future Work

6.1 Conclusion

Acoustic Vector Sensors (AVSs) can be applied in an array setup to find the Direction-Of-Arrivals (DOAs) of multiple sources. These AVSs outperform the conventional acoustic pressure sensors in terms of direction finding. Traditional DOA estimation for AVS-array require perfect synchronization between the different sensors. Due to environmental limitations this is not always possible and clock effects should be taken into account when an accurate DOA needs to be found. From the introduction we have two main questions in this thesis, these questions are repeated and briefly answered below:

- How can we model an AVS-array that is subjected to clock errors?

In Chapter 3 we found that the clocks in the sensors can be modeled as an affine transformation of time. This affine error can then be merged with the conventional model for AVS-arrays by adding a time dependent phase error. By simulation it is then found that if the crystal applied in the sensor has certain properties these phase errors can be modeled as time independent.

- How can we perform DOA estimation on an AVS-array that is subjected to clock errors?

In Chapter 4, two novel algorithms are developed, that are independent of clock errors and are capable of finding the DOAs. However these algorithms require a two dimensional search, such that these techniques become computational expensive. Therefore a third algorithm, that is not completely clock independent, is proposed. This method can be combined with the previous methods to reduce its computational cost.

In Chapter 5, measurements are conducted in a controlled environment. These measurements are applied to the clock independent algorithms that where developed in the previous chapter. The outcomes of the experiments show that the developed methods are applicable to synchronized data and inaccurate synchronized data.

6.2 Future Work

Optimization of the two dimensional search - The two clock independent algorithms proposed in this work suffer from high computational cost due to the two dimensional search applied in these methods. It would be interesting to see if it is possible to optimize this search.

Optimization of the array aperture - Although the proposed method is applicable for all array apertures. Performance might differ between different apertures. An interesting research topic would be to find the optimum aperture for a given set of conditions.

Space-time equalization - In this thesis, it is assumed that phase errors in the data are induced in the model due to clock errors. If the sensors are able to move, the movement will also induce phase errors. In [26] Rajan and van der Veen exploit the affine time model of the clocks to jointly estimate the ranging and synchronization of the sensors. An interesting research topic would be to merge the joint DOA and synchronization of this thesis with the method developed by Rajan and van der Veen such that the DOA, synchronization and position of the sensors are jointly estimated.

Nested arrays - In [27], Pal and Vaidyanathan show that if two ore more uniform linear arrays are nested ths can increase the amount of DOAs that can be found. This geometry is extended for AVSs by means of multi-linear algebra by Han and Nehorai in [28]. As one the disadvantages of this method is the low amount of DOAs that can be estimated, it would be interesting to see if the model used in this thesis could be fitted in that of Han and Nehorai. This is likely to increase the amount of DOAs that can be estimated.

Clock Characterization

A

In this appendix, we show by means of measurement that the clock can can be modeled as an affine system. In order to find the clocks, the sensor inputs all are connected electrically to the same source. The source is set to generate white Gaussian noise such that the expected cross correlation between sensor i and j ($\mathbb{E}[r_{ij}(n)]$) is zero for every sampling number n but has a peak at the Time Difference Of Arrival (TDOA) which can be described as:

$$\tau_{ij} = \tau_j - \tau_i, \tag{A.1}$$

where τ_i denotes the time delay between the source and sensor i and τ_{ij} denotes the TDOA between sensor i and j.

In order to compute the TDOAs τ_{ij}, simply computing $r_{ij}(n)$ and searching for the index that corresponds to the maximum will leave us with TDOAs at sample accuracy (1 sample $= 1/f_s = 62.5\mu s$). However, for the clock characterization higher accuracy is necessary. the higher resolution could be achieved by convolving $r_{ij}(n)$ with the sinc-function $\mathrm{sinc}(t)$ to find the autocorrelation as a function of full time t: $r_{ij}(t)$. However, as this is computational expensive we rather use Newton's method (also known as Newton-Raphson method) as described in [29] to find the solution of the following:

$$\hat{\tau}_{ij} = \arg\max_{\tau'_{ij}} \left(\mathrm{sinc}(\tau'_{ij} - n) * r_{ij}(n)\right), \tag{A.2}$$

where $\hat{\tau}_{ij}$ is the estimate of the TDOA between sensor i and j and $*$ denotes the convolution operator. This method does not require to compute the full time convolution given before. Alternatively the cost function is evaluated iteratively in order to find the maximum of $r_{ij}(t)$. Note that it is depending on the data if this cost function is concave. Fortunately, due to the nature of our measurement and its expected crosscorrelation, we can safely assume that the cost function is quasi-concave around its maximum. It follows that if we take the index corresponding to the maximum of $r_{ij}(n)$ as initial guess for τ'_{ij} we will always end up at the desired maximum.

From [29] we learn that the step size of Newton's method is equal to:

$$\Delta\tau'_{ij} = \frac{\nabla_{\tau'_{ij}} \left(\mathrm{sinc}(\tau'_{ij} - n) * r_{ij}(n)\right)}{\nabla^2_{\tau'_{ij}} \left(\mathrm{sinc}(\tau'_{ij} - n) * r_{ij}(n)\right)}. \tag{A.3}$$

In order to find the derivatives necessary to compute the step. Note the property of the convolution that the derivative of a convolution is the derivative of one side of the convolution convolved with the other side. This property leads to:

$$\nabla_{\tau'_{ij}} \left(\mathrm{sinc}(\tau'_{ij} - n) * r_{ij}(n)\right) = \left(\nabla_{\tau'_{ij}} \mathrm{sinc}(\tau'_{ij} - n)\right) * r_{ij}(n). \tag{A.4}$$

By denoting

$$\nabla_{\tau'_{ij}} \text{sinc}(\tau'_{ij} - n) = \frac{\cos\left(\pi(\tau'_{ij} - n)\right) - \text{sinc}(\tau'_{ij} - n)}{\tau'_{ij} - n} \tag{A.5}$$

and

$$\nabla^2_{\tau'_{ij}} \text{sinc}(\tau'_{ij} - n) = \frac{\left(2 - \pi(\tau'_{ij} - n)^2\right) \text{sinc}(\tau'_{ij} - n) - 2\cos\left(\pi(\tau'_{ij} - n)\right)}{(\tau'_{ij} - n)^2}. \tag{A.6}$$

We can write the updating algorithm as:

$$\tau'^{(k+1)}_{ij} = \tau'^{(k)}_{ij} - \Delta\tau'_{ij}. \tag{A.7}$$

Now, by windowing the signal in time-domain by a rectangular window, taking the crosscorrelation with respect to a reference sensor and all other sensors we find the delays in the signal with respect to the clocks in the sensor. In order to delete any lowband noises we will use a highpass filter on the unwindowed data. The full algorithm is shown in Algorithm 2.

Algorithm 2 Time Delays

```
procedure FIND TIME DELAYS
    load data into s(i,t) with sensor number i and time t
    s(i,t) = highpass(s(i,t))
    for m windows do
        s(i,n) = window(s(i,t))
        for j = 2 until 6 do
            xCross = crosscorrelation((s(1,n), s(j,n))
            t_max(j,n) = Newton-Step(xCross) ← equation A.7 till stopping requirement
        end
    end
```

The above mentioned method computes the TDOAs with one reference sensor (i.e. $\tau_{i,1}$). Note that it is straightforward to compute the TDOAs with respect to the average. The above described measurement is carried out over a time of ten minutes using five sensors. There is a time synchronization at the first sample point. The expectation is that the above method leads to estimated delays at $t = 0$ of zero and to diverge afterwards. The result of this experiment is give in Figure A.1. This figure shows linear TDOAs as expected before. Such that we conclude that the affine model mentioned in Chapter 3 is correct.

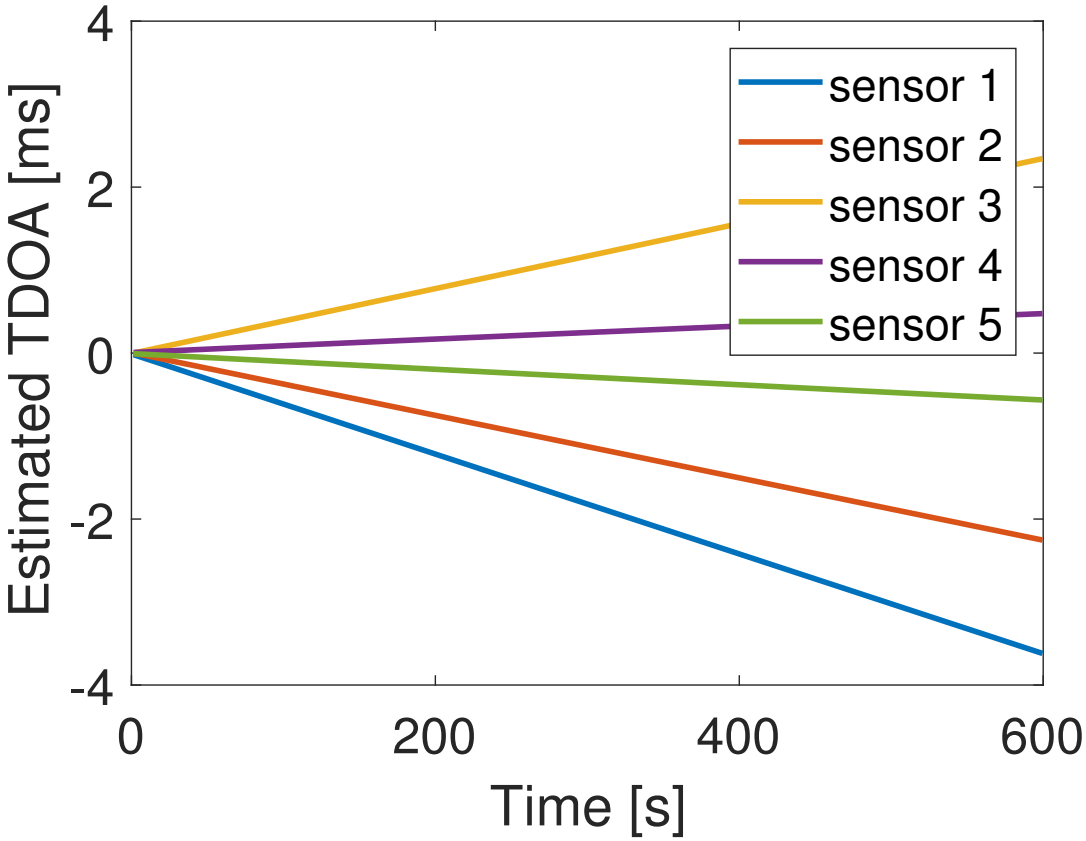


Figure A.1: The estimated time difference of arrival.

Simulations on the approximation of (3.22)

B

In this appendix we show that the affine clock errors can approximated by a model in where the clock errors are constant. We show, by means of a Monte Carlo simulation, the effects of the time dependency on the MUSIC DOA estimation. For simplicity, the clock offset in the model will be neglected. So or time vector containing the times according to sensor i at time t is given by:

$$t_i = \omega_i t \tag{B.1}$$

where $\omega_i = (1 + \alpha_i)$ and α_i is the clock drift in s/s.

The simulations are executed with varying α_is. To show that the clock drift do not have a big influence on the DOA estimation, DOA estimation is performend using the MUSIC method trying to recover a single source at $\theta = 0°$. The found DOA through this method will be denoted as $\hat{\theta}$. As an measurement of performance of our estimate the RMSE is chosen which is defined as:

$$\text{RMSE} = \left(N^{-1} \sum_{n=1}^{N} \left(\hat{\theta}(n) - \theta \right)^2 \right)^{1/2}, \tag{B.2}$$

where N is the number of iterations and $\hat{\theta}(n)$ the DOA estimate of the n^{th} iteration. The used α_is are drawn from a zero-mean Gaussian distribution with standard deviation σ_α.

Note that the phase errors induced by the α are frequency dependent, higher frequency will lead to a bigger error. The sensors that are used throughout this work have a sampling frequency of 16kHz which means that the nyquist rate lies at 8kHz because this will be the worst case scenario, we will use this frequency within the simulation. Furthermore an ULA is used with sensor spacing $\frac{\lambda}{2}$. 0.5 Seconds worth of samples are used to perform the DOA estimation and the SNR is set at 10dB. For every σ_α $N = 100$ simulations are performed.

The results of the simulation are shown in Figure B.1. Observe that the RMSE stays close to zero till about 10^{-5}s/s after which the RMSE increases. This aligns with a clock drift of 10 ppm. The sensors currently used have a clock that operates at 30 ppm but, based on these simulations, it is suggested to use different clocks with a drift that is lower than 1 ppm.

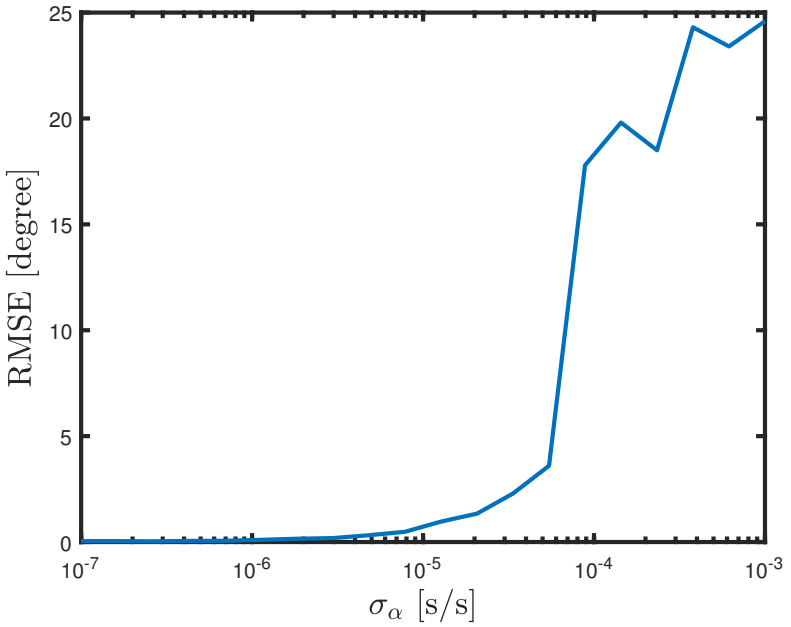


Figure B.1: Monte carlo simulation of the approximation

C Proof of Lemma 3.1

Goal of this appendix is to proof Lemma 3.1.

First rewrite$\mathbf{a}_{\mathrm{p}}(\theta')^H\mathbf{a}_{\mathrm{p}}(\theta)\mathbf{a}_{\mathrm{p}}(\theta)^H\mathbf{a}_{\mathrm{p}}(\theta')$:

$$\mathbf{a}_{\mathrm{p}}(\theta')^H\mathbf{a}_{\mathrm{p}}(\theta)\mathbf{a}_{\mathrm{p}}(\theta)^H\mathbf{a}_{\mathrm{p}}(\theta') = \left|\mathbf{a}_{\mathrm{p}}(\theta')^H\mathbf{a}_{\mathrm{p}}(\theta)\right|^2 \tag{C.1}$$

$$= \left|\sum_{i=0}^{M-1} e^{\mathrm{j}iq(\theta,\theta')}\right|^2 \tag{C.2}$$

$$= \left|\sum_{i=0}^{M-1} e^{\mathrm{j}i\phi}\right|^2, \tag{C.3}$$

where we defined $\phi = q(\theta, \theta')$. Now, note that we could write the inner part of the absolute value as:

$$\sum_{i=0}^{M-1} e^{\mathrm{j}i\phi} = \sum_{i=0}^{M-1} \cos(i\phi) + \mathrm{j}\sum_{i=0}^{M-1} \sin(i\phi). \tag{C.4}$$

Substituting this in (C.3):

$$\left|\sum_{i=0}^{M-1} e^{\mathrm{j}i\phi}\right|^2 = \left|\sum_{i=0}^{M-1} \cos(i\phi) + j\sum_{i=0}^{M-1} \sin(i\phi)\right|^2 \tag{C.5}$$

$$= \left(\sum_{i=0}^{M-1} \cos(i\phi)\right)^2 + \left(\sum_{i=0}^{M-1} \sin(i\phi)\right)^2. \tag{C.6}$$

Now consider the case where $M = 3$:

$$\left|\sum_{i=0}^{M-1} e^{\mathrm{j}i\phi}\right|^2 = (1 + \cos(\phi) + \cos(2\phi))^2 + (\sin(\phi) + \sin(2\phi))^2 \tag{C.7}$$

$$= 1 + 2\cos(\phi) + 2\cos(2\phi) + \cos^2(\phi) + \cos^2(2\phi) + 2\cos(\phi)\cos(2\phi) \tag{C.8}$$

$$+ \sin^2(\phi) + \sin^2(2\phi) + 2\sin(\phi)\sin(2\phi) \tag{C.9}$$

$$= 3 + 2\left(\cos(\phi) + \cos(2\phi) + \cos(\phi)\cos(2\phi) + 2\sin(\phi)\sin(2\phi)\right), \tag{C.10}$$

where we made use of the identity $\cos^2(a) + \sin^2(a) = 1$. Note that the double angle formulas for the sine and cosine could be written as:

$$\sin(2a) = 2\sin(a)\cos(a), \tag{C.11}$$

$$\cos(2a) = 1 - 2\sin^2(a). \tag{C.12}$$

Substituting these into (C.10):

$$\left|\sum_{i=0}^{2} e^{\mathrm{j}i\phi}\right|^2 = 3 + 2\left(\cos(\phi) + \cos(2\phi) + \cos(\phi)\left(1 - 2\sin^2(\phi)\right) + \sin(\phi)\left(2\sin(\phi)\cos(\phi)\right)\right) \tag{C.13}$$

$$= 3 + 2\left(2\cos(\phi) + \cos(2\phi)\right). \tag{C.14}$$

Repeating the same steps from (C.7) till (C.14) for $M = 4$ while using the triple angle formulas leads to:

$$\left|\sum_{i=0}^{3} e^{ji\phi}\right|^2 = 4 + 2\left(3\cos(\phi) + 2\cos(2\phi) + \cos(3\phi)\right). \tag{C.15}$$

Expanding this even further leads to the result:

$$\left|\sum_{i=0}^{M-1} e^{\mathrm{j}i\phi}\right|^2 = M + 2\left(\sum_{i=1}^{M-1}(M-i)\cos\left(i\phi\right)\right). \tag{C.16}$$

Which completes the proof.

Simulations on the approximation of (3.39)

D

In this appendix we will give some simulations supporting the approximation in (3.39) i.e.

$$\arg\max_{\theta'} P_{MVDR} \approx \arg\min_{\theta',C} \sum_{i=0}^{M-1} \left| iq(\theta,\theta') + kc\beta_{i+1} - C \right|, \tag{D.1}$$

$$\hat{\theta} \approx \hat{\theta}' \tag{D.2}$$

where we defined:

$$\hat{\theta} = \arg\max_{\theta'} P_{MVDR}, \tag{D.3}$$

$$\hat{\theta}' = \arg\min_{\theta',C} \sum_{i=0}^{M-1} \left| iq(\theta,\theta') + kc\beta_{i+1} - C \right|. \tag{D.4}$$

In the following we will simulate this for varying β, the βs that will be used are drawn from a Gaussian set with given standard deviation σ_β. In order to analyze the performance of the approximation we define the error as the MSE:

$$N^{-1} \sum_{i=1}^{N} \epsilon_i^2 \tag{D.5}$$

where we defined $\epsilon_i = \hat{\theta}_i - \hat{\theta}'_i$ the error of the i^{th} iteration, $\hat{\theta}'_i$ and $\hat{\theta}_i$ as the θs belonging to that iteration. N is the total amount of iterations.

The above has been simulated for varying σ_β and frequency f with $N = 1000$ iterations. The actual DOA is in this simulation held at $\theta = 45°$ The results are shown in Figure D.1. In this figure we see that when $f = 100$ Hz the approximation stays valid until at least $\sigma_\beta = 100\mu$s. For $f = 500$Hz an error starts appearing for $\sigma_\beta = 30\mu$s and for $f = 1000$Hz this error already start at $\sigma_\beta = 10\mu$s. Note that these simulations are for $\theta = 45°$, the error will increase if $\theta \to 0°$ or $\theta \to 180°$ and will decrease for $\theta \to 45°$.

Although the simulations show that $\hat{\theta} \neq \hat{\theta}'$ we are still able to conclude that $\hat{\theta} \approx \hat{\theta}'$ when the clock errors are small and the frequency is sufficient low.

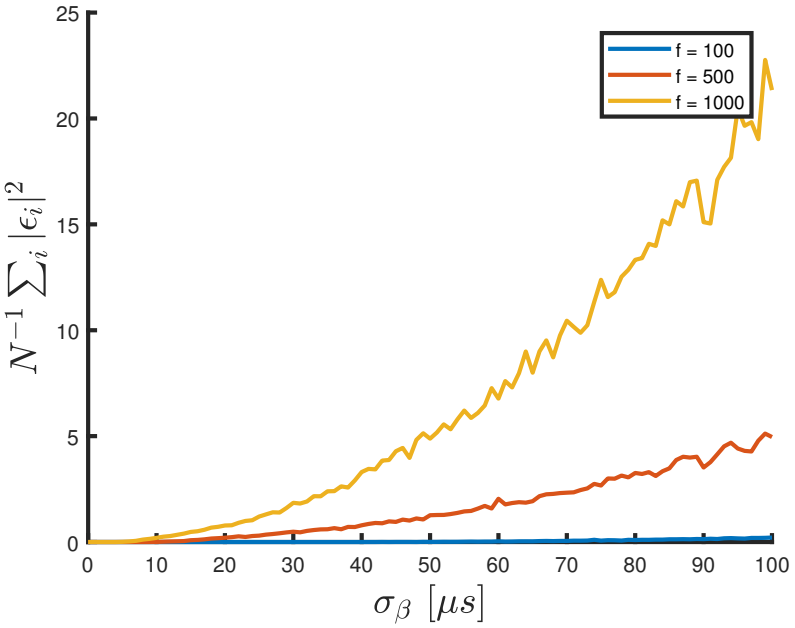


Figure D.1: Monte carlo simulations of the approximation

Proof of Theorem 4.2

E

The goal of this appendix is to proof the identity given in (4.41). Therefore we first expand the definition of $\tilde{\boldsymbol{R}}$:

$$\begin{aligned}\tilde{\boldsymbol{R}} =& \mathbb{E}\left[\tilde{\mathbf{y}}\tilde{\mathbf{y}}^H\right] && \text{(E.1)}\\ =& \mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] + \mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{v}}^H\right]\\ &+ \mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{p}}^H\right]\\ &+ \mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{n}}^H\right]. && \text{(E.2)}\end{aligned}$$

Note that we can write the first term of (E.2) as follows:

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] = \boldsymbol{C}\boldsymbol{R}_{\mathbf{s}_d}\boldsymbol{C}^H \tag{E.3}$$

where we defined $\boldsymbol{R}_{\tilde{\mathbf{s}}} = \mathbb{E}\left[\tilde{\mathbf{s}}\tilde{\mathbf{s}}^H\right]$ which will be expanded later in this appendix.

Lemma E.1.

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] = \mathbf{0}_{3M\times 3M} \tag{E.4}$$

Proof. By observing the term $\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{p}}^H\right]$ we find that all terms in the expected value have three signal factors and one noise factor. This enables us to use the fact that the noise is uncorrelated with the signals and therefore we could split the expected value for index i, j as $\mathbb{E}\left[\bar{s}_{i,j}\right]\mathbb{E}\left[\bar{n}_{i,j}\right]$ where $\bar{s}_{i,j}$ is a scalar containing the appropriate signal information of that index and $\bar{n}_{i,j}$ the weighted noise factor of the corresponding index. Now, as $\bar{n}_{i,j}$ contains only one random variable (i.e. noise), which is zero mean, it is easy to see that $\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] = \mathbf{0}_{3M\times 3M}$. ■

From the proof of Lemma E.1 we observe in a similar way the following equalities:

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] = \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] = \mathbf{0}_{3M\times 3M}, \tag{E.5}$$

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] = \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] = \mathbf{0}_{3M\times 3M}, \tag{E.6}$$

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] = \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] = \mathbf{0}_{3M\times 3M}. \tag{E.7}$$

Furthermore we have that the noise is white Gaussian distributed which implies:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\right] = \sigma_\eta^2\left(\mathbf{1}_M \otimes \mathbf{e}_1\right) = \sigma_\eta^2\mathbf{p} \in \mathbb{R}^{3M} \tag{E.8}$$

where we defined $\mathbf{p} = \mathbf{1}_M \otimes \mathbf{e}_1$.

Lemma E.2.

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] = \sigma_\eta^2\boldsymbol{C}\mathbb{E}[\tilde{\mathbf{s}}]\mathbf{p}^T \tag{E.9}$$

Proof. It is easy to see that $\boldsymbol{C}$ can be written out of the expected value due to that it is deterministic. Observe that the $\tilde{\mathbf{s}}$-vector only contains signal information and the $\boldsymbol{\nu}_{\mathrm{n}}^H$-vector only contains noise. By the assumption that the noise is uncorrelated with the signal we are able to write: $\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] = \boldsymbol{C}\mathbb{E}\left[\tilde{\mathbf{s}}\right]\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}^H\right]$. Substituting (E.8) leads to the desired result. ■

From Lemma E.2 we obtain the following identities:

$$\mathbb{E}\left[\boldsymbol{C}\tilde{\mathbf{s}}\boldsymbol{\nu}_{\mathrm{n}}^H\right] = \sigma_\eta^2\boldsymbol{C}\mathbb{E}[\tilde{\mathbf{s}}]\mathbf{p}^T, \tag{E.10}$$

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\tilde{\mathbf{s}}^H\boldsymbol{C}^H\right] = \sigma_\eta^2\mathbf{p}\mathbb{E}[\tilde{\mathbf{s}}^H]\boldsymbol{C}^H. \tag{E.11}$$

Lemma E.3.

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] = \mathbf{0}. \tag{E.12}$$

Proof. Note that we could expand $\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{v}}^H\right]$ such that all terms are of the form $\mathbb{E}\left[s_i^* s_j \boldsymbol{\nu}_{\mathrm{p},i}\boldsymbol{\nu}_{\mathrm{v},j}^H\right] \forall i,j \in \{1,2\}$. Expanding these terms further leads to

$$\mathbb{E}\left[s_i^* s_j \boldsymbol{\nu}_{\mathrm{p},i}\boldsymbol{\nu}_{\mathrm{v},j}^H\right] = \mathbb{E}\left[s_i^* s_j \left(\boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^*\mathbf{a}_{\mathrm{p},i}^* \otimes \mathbf{1}_3\right)\right)\left(\left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},j} \odot \boldsymbol{\eta}_{\mathrm{p}}^*\right) \otimes \mathbf{h}_j\right)^H\right] \tag{E.13}$$

$$= \mathbb{E}\left[s_i^* s_j\right]\mathbb{E}\left[\left(\boldsymbol{\eta} \odot \left(\boldsymbol{\Gamma}^*\mathbf{a}_{\mathrm{p},i}^* \otimes \mathbf{1}_3\right)\right)\left(\left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},j} \odot \boldsymbol{\eta}_{\mathrm{p}}^*\right) \otimes \mathbf{h}_j\right)^H\right], \tag{E.14}$$

where we used the assumption that the signals are uncorrelated with the noise. If we split the resulting matrix in blocks of 3×3 then we could write block k,l as follows:

$$\left\{\mathbb{E}\left[s_i^* s_j \boldsymbol{\nu}_{\mathrm{p},i}\boldsymbol{\nu}_{\mathrm{v},j}^H\right]\right\}_{k,l} = \mathbb{E}\left[s_i^* s_j\right]\mathbb{E}\left[\left(\boldsymbol{\eta}_k \odot \mathbf{d}_k\right)\left(\mathbf{f}_l \odot \left(\eta_{\mathrm{p},l}^* \otimes \mathbf{h}_j\right)\right)^H\right], \tag{E.15}$$

where $\boldsymbol{\eta}_k$ contains the elements $\boldsymbol{\eta}$ that agree with the indices of the k^{th} block, $\eta_{\mathrm{p},l}$ the l^{th} entry of $\boldsymbol{\eta}_{\mathrm{p}}$ and the vectors $\mathbf{d}_k$ and $\mathbf{f}_l$ are chosen such that the equation holds. From here it is easy to see that with the assumption that the noise is complex circular distributed (i.e. $\mathbb{E}\left[\Re\left\{\boldsymbol{\eta}\right\}\Re\left\{\boldsymbol{\eta}\right\}^T\right] = \mathbb{E}\left[\Im\left\{\boldsymbol{\eta}\right\}\Im\left\{\boldsymbol{\eta}\right\}^T\right]$) that the equation above goes to zero. ■

From Lemma E.3 we obtain:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] = \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] = \mathbf{0}. \tag{E.16}$$

Substituting (E.5)-(E.7), (E.10), (E.11) and (E.16) into (E.2) leads to:

$$\tilde{\boldsymbol{R}} = \boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^T + \sigma_\eta^2\boldsymbol{C}\mathbb{E}[\tilde{\mathbf{s}}]\mathbf{p}^T + \sigma_\eta^2\mathbf{p}\mathbb{E}[\tilde{\mathbf{s}}^H]\boldsymbol{C}^H + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{v}}^T\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{p}}^H\right] + \mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{n}}^H\right]. \tag{E.17}$$

As the signals are uncorrelated to each other. It follows that the cross terms of $\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{v}}^H\right]$ are equal to zero and we can expand this as:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{v}}^H\right] = \sigma_{\mathrm{s},1}^2\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v},1}\boldsymbol{\nu}_{\mathrm{v},1}^H\right] + \sigma_{\mathrm{s},2}^2\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v},2}\boldsymbol{\nu}_{\mathrm{v},2}^H\right], \tag{E.18}$$

where $\sigma_{\mathrm{s},i}^2$ denotes the power of the i^{th} source. Now, evaluate:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v},1}\boldsymbol{\nu}_{\mathrm{v},1}^H\right] = \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},1}\mathbf{a}_{\mathrm{p},1}^H\boldsymbol{\Gamma}^H \odot \mathbb{E}\left[\mathbf{n}_{\mathrm{p}}^*\mathbf{n}_{\mathrm{p}}^H\right]\right) \otimes \mathbf{h}_1\mathbf{h}_1^T \tag{E.19}$$

$$= \left(\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},1}\mathbf{a}_{\mathrm{p},1}^H\boldsymbol{\Gamma}^H \odot \sigma_\eta^2\boldsymbol{I}\right) \otimes \mathbf{h}_1\mathbf{h}_1^T. \tag{E.20}$$

Now notice the equality $\operatorname{diag}\left\{\boldsymbol{\Gamma}\mathbf{a}_{\mathrm{p},1}\mathbf{a}_{\mathrm{p},1}^{H}\boldsymbol{\Gamma}^{H}\right\} = \mathbf{1}$. As we are taking the Hadamard product between a matrix that has only ones on the diagonal matrix and a matrix that is zero on the non-diagonal entries we completely lose the first matrix. So:

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v},1}\boldsymbol{\nu}_{\mathrm{v},1}^{H}\right] =&\sigma_{\eta}^{2}\boldsymbol{I}\otimes\mathbf{h}_{1}\mathbf{h}_{1}^{T} &\text{(E.21)}\\ =&\sigma_{\eta}^{2}\boldsymbol{H}_{1}, &\text{(E.22)}\end{aligned}$$

where $\boldsymbol{H}_i = \boldsymbol{I}\otimes\mathbf{h}_i\mathbf{h}_i^T$ is a blockdiagonal matrix. Similar:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v},2}\boldsymbol{\nu}_{\mathrm{v},2}^{H}\right] = \sigma_{\eta}^{2}\boldsymbol{H}_{2}. \tag{E.23}$$

Now, substituting (E.22) and (E.23) into (E.18) gives:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{v}}\boldsymbol{\nu}_{\mathrm{v}}^{H}\right] = \sigma_{\eta}^{2}\left(\sigma_{\mathrm{s},1}^{2}\boldsymbol{H}_{1}+\sigma_{\mathrm{s},2}^{2}\boldsymbol{H}_{2}\right). \tag{E.24}$$

Similar to (E.18) we have:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{p}}^{H}\right] = \sigma_{\mathrm{s},1}^{2}\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p},1}\boldsymbol{\nu}_{\mathrm{p},1}^{H}\right]+\sigma_{s2}^{2}\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p},2}\boldsymbol{\nu}_{\mathrm{p},2}^{H}\right]. \tag{E.25}$$

By evaluating $\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p},1}\boldsymbol{\nu}_{\mathrm{p},1}^{H}\right]$ we find:

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p},1}\boldsymbol{\nu}_{\mathrm{p},1}^{H}\right] =&\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^{H}\right]\odot\left(\boldsymbol{\Gamma}^{*}\mathbf{a}_{\mathrm{p},1}^{*}\mathbf{a}_{\mathrm{p},1}^{T}\boldsymbol{\Gamma}^{T}\otimes\mathbf{1}_{3}\mathbf{1}_{3}^{T}\right) &\text{(E.26)}\\ =&\sigma_{\eta}^{2}\boldsymbol{I} &\text{(E.27)}\end{aligned}$$

and similar

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p},2}\boldsymbol{\nu}_{\mathrm{p},2}^{H}\right] = \sigma_{\eta}^{2}\boldsymbol{I}. \tag{E.28}$$

Substituting (E.27) and (E.28) into (E.25) leads to:

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{p}}\boldsymbol{\nu}_{\mathrm{p}}^{H}\right] = \sigma_{\eta}^{2}\left(\sigma_{\mathrm{s},1}^{2}+\sigma_{\mathrm{s},2}^{2}\right)\boldsymbol{I}. \tag{E.29}$$

Note that we could write $\mathbb{E}\left[\boldsymbol{\nu}_{n}\boldsymbol{\nu}_{n}^{H}\right]$ as

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{n}}^{H}\right] = \mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^{H}\odot\left(\boldsymbol{\eta}_{\mathrm{p}}^{*}\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}\mathbf{1}_{3}^{T}\right)\right]. \tag{E.30}$$

From [21], we have that the product of four complex Gaussian variables $\{x_i\}_{i=1}^{4}$ of which at least one is zero mean has the following relation:

$$\mathbb{E}\left[x_1x_2x_3x_4\right] = \mathbb{E}\left[x_1x_2\right]\mathbb{E}\left[x_3x_4\right]+\mathbb{E}\left[x_1x_3\right]\mathbb{E}\left[x_2x_4\right]+\mathbb{E}\left[x_1x_4\right]\mathbb{E}\left[x_2x_3\right]. \tag{E.31}$$

Note that this implies:

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{n}}^{H}\right] =&\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^{H}\right]\odot\mathbb{E}\left[\boldsymbol{\eta}_{\mathrm{p}}^{*}\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}\mathbf{1}_{3}^{T}\right]\\ &+\mathbb{E}\left[\boldsymbol{\eta}\odot\left(\boldsymbol{\eta}_{\mathrm{p}}^{*}\otimes\mathbf{1}_{3}\right)\right]\mathbb{E}\left[\boldsymbol{\eta}^{H}\odot\left(\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}^{T}\right)\right]\\ &+\mathbb{E}\left[\boldsymbol{\eta}\left(\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}^{T}\right)\right]\odot\mathbb{E}\left[\left(\boldsymbol{\eta}_{\mathrm{p}}^{*}\otimes\mathbf{1}_{3}\right)\boldsymbol{\eta}^{H}\right]. &\text{(E.32)}\end{aligned}$$

Note that it is easy to see that due to the assumption of complex circular noise the third term in the equation above reduces to zero:

$$\mathbb{E}\left[\boldsymbol{\eta}\left(\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}^{T}\right)\right]\odot\mathbb{E}\left[\left(\boldsymbol{\eta}_{\mathrm{p}}^{*}\otimes\mathbf{1}_{3}\right)\boldsymbol{\eta}^{H}\right] = \mathbf{0}. \tag{E.33}$$

Now, by substituting (E.33) into (E.32):

$$\mathbb{E}\left[\boldsymbol{\nu}_{\mathrm{n}}\boldsymbol{\nu}_{\mathrm{n}}^{H}\right] =\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^{H}\right]\mathbb{E}\left[\boldsymbol{\eta}_{\mathrm{p}}^{*}\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}\mathbf{1}_{3}^{T}\right]+\mathbb{E}\left[\boldsymbol{\eta}\odot\left(\boldsymbol{\eta}_{\mathrm{p}}^{*}\otimes\mathbf{1}_{3}\right)\right]\mathbb{E}\left[\boldsymbol{\eta}^{H}\odot\left(\boldsymbol{\eta}_{\mathrm{p}}^{T}\otimes\mathbf{1}_{3}^{T}\right)\right] \tag{E.34}$$

$$=\sigma_{\eta}^{4}\boldsymbol{I}+\sigma_{\eta}^{4}\mathbf{p}\mathbf{p}^{T}. \tag{E.35}$$

Substitution of (E.24), (E.29) and (E.35) into (E.17) gives

$$\begin{aligned}\tilde{\boldsymbol{R}} =&\boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^{H}+\sigma_{n}^{2}\boldsymbol{C}\mathbb{E}[\tilde{\mathbf{s}}]\mathbf{p}^{T}+\sigma_{n}^{2}\mathbf{p}\mathbb{E}[\tilde{\mathbf{s}}^{H}]\boldsymbol{C}^{H}\\&+\sigma_{n}^{2}\left(\sigma_{\mathrm{s},1}^{2}\left(\boldsymbol{H}_{1}+\boldsymbol{I}\right)+\sigma_{\mathrm{s},2}^{2}\left(\boldsymbol{H}_{2}+\boldsymbol{I}\right)\right)+\sigma_{\eta}^{4}\left(\boldsymbol{I}+\mathbf{p}\mathbf{p}^{T}\right)\end{aligned} \tag{E.36}$$

Due to that the signals are independent of each other we find:

$$\mathbb{E}[\mathbf{s}_{d}]=\begin{bmatrix}\sigma_{\mathrm{s},1}^{2} & \sigma_{\mathrm{s},2}^{2} & 0 & 0\end{bmatrix}^{T}. \tag{E.37}$$

Thus by substituting (E.37) into (E.36):

$$\begin{aligned}\tilde{\boldsymbol{R}} =&\boldsymbol{C}\boldsymbol{R}_{\tilde{\mathbf{s}}}\boldsymbol{C}^{H}+\sigma_{\eta}^{2}\left(\sigma_{\mathrm{s},1}^{2}\boldsymbol{\xi}_{1}+\sigma_{\mathrm{s},2}^{2}\boldsymbol{\xi}_{2}\right)\mathbf{p}^{T}+\sigma_{\eta}^{2}\mathbf{p}\left(\sigma_{\mathrm{s},1}^{2}\boldsymbol{\xi}_{1}+\sigma_{\mathrm{s},2}^{2}\boldsymbol{\xi}_{2}\right)^{T}\\&+\sigma_{\eta}^{2}\left(\sigma_{\mathrm{s},1}^{2}\left(\boldsymbol{H}_{1}+\boldsymbol{I}\right)+\sigma_{\mathrm{s},2}^{2}\left(\boldsymbol{H}_{2}+\boldsymbol{I}\right)\right)+\sigma_{\eta}^{4}\left(\boldsymbol{I}+\mathbf{p}\mathbf{p}^{T}\right)\end{aligned} \tag{E.38}$$

Note that we can find $\boldsymbol{R}_{\tilde{\mathbf{s}}}$ while using (E.31) as follows:

$$\boldsymbol{R}_{\tilde{\mathbf{s}}}=\mathbb{E}\left[\tilde{\mathbf{s}}\tilde{\mathbf{s}}^{H}\right]=\begin{bmatrix}2\sigma_{\mathrm{s},1}^{4} & \sigma_{s1}^{2}\sigma_{\mathrm{s},2}^{2} & 0 & 0\\ \sigma_{\mathrm{s},1}^{2}\sigma_{\mathrm{s},2}^{2} & 2\sigma_{\mathrm{s},2}^{4} & 0 & 0\\ 0 & 0 & \sigma_{\mathrm{s},1}^{2}\sigma_{\mathrm{s},2}^{2} & 0\\ 0 & 0 & 0 & \sigma_{\mathrm{s},1}^{2}\sigma_{\mathrm{s},2}^{2}\end{bmatrix}. \tag{E.39}$$

Bibliography

[1] H.-E. de Bree, P. Leussink, T. Korthorst, H. Jansen, T. S. Lammerink, and M. Elwenspoek, "The μ-flown: a novel device for measuring acoustic flows," *Sensors and Actuators*, vol. A, no. 54, pp. 552–557, 1996.

[2] H.-E. de Bree, "An overview of microflown technologies," *Acta Acustica united with Acustica*, vol. 89, no. 1, pp. 163–172, 2003.

[3] Krishnaprasad, "Acoustic vector sensor based source localization," Master's thesis, Delft University of Technology, 2016.

[4] S. Najeem, K. Kiran, A. Malarkodi, and G. Latha, "Open lake experiment for direction of arrival estimation using acoustic vector sensor array," *Applied Acoustics*, vol. 119, pp. 94–100, apr 2017.

[5] Y. Zou, R. Gu, D. Wang, A. Jiang, and C. H. Ritz, "Learning a robust DOA estimation model with acoustic vector sensor cues," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, dec 2017.

[6] D. F. Comesana, S. Steltenpool, G. C. Pousa, H.-E. de Bree, and K. R. Holland, "Scan and paint: Theory and practice of a sound field visualization method," *ISRN Mechanical Engineering*, vol. 2013, pp. 1–11, 2013.

[7] D. F. Comesana, "Scan-based sound visualisation methods using sound pressure and particle velocity," Ph.D. dissertation, University of Southamptom, 2014.

[8] H.-E. de Bree, "Acoustic vector sensors increasing uav's situational awereness," *SAE Technical Paper*, 2009.

[9] H.-E. de Bree, J. Wind, and P. de Theije, "Detection, localization and tracking of aircraft using acoustic vector sensors," *Inter-Noise*, Sep. 2011.

[10] A. Nehorai and E. Paldi, "Acoustic vector-sensor array processing," *IEEE Transaction on Signal Processing*, vol. 42, no. 9, pp. 2481–2491, 1994.

[11] G. Leus, "Signal processing for communications," 2005.

[12] J. A. Barnes and et al., "Characterization of frequency stability," *IEEE Transactions on Instrumentation and Measurement*, vol. IM-20, no. 2, pp. 105–120, 1971.

[13] R. T. Rajan, "Relative space-time kinematic of an anchorless network," Ph.D. dissertation, Delft University of Technology, 2016.

[14] W. Riley, *Handbook of Frequency Stability Analysis*. 325 Broadway, Boulder: NIST, 2008.

[15] G. S. Levy, R. P. Linfield, J. S. Ulvestad, C. D. Edwards, J. F. Jordan, S. J. D. Nardo, C. S. Christensen, R. A. Preston, L. J. Skjerve, L. R. Stavert, B. F. Burke, A. R. Whitney, R. J. Ccappallo, A. E. E. Rogers, K. B. Blaney, M. J. Maher, C. H. Ottenhoff, D. L. Jauncey, W. L. Peters, T. Nishimura, T. Hayashi, T. Takano, T. Yamada, H. Hirabayashi, M. Morimoto, M. Inoue, T. Shiomi, N. Kawaguchi, and H. Kunimori, "Very long baseline interferometric observations made with an orbiting radio telescope," *Science*, vol. 234, no. 4773, pp. 187–189, oct 1986.

[16] M. Lombardi, *The Mechatronics Handbook*. CRC Press, 2002.

[17] S. Vorobyov, A. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: a solution to the signal mismatch problem," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 313–324, feb 2003.

[18] A. Liu, G. Liao, C. Zeng, Z. Yang, and Q. Xu, "An eigenstructure method for estimating DOA and sensor gain-phase errors," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5944–5956, dec 2011.

[19] S. Cao, Z. Ye, D. Xu, and X. Xu, "A hadamard product based method for doa estimation and gain-phase error calibration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1224–1233, apr 2013.

[20] Y. Bresler, "Maximum likelihood estimation of a linearly structured covariance with application to antenna array processing," in *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling*. IEEE, 1988.

[21] P. Janssen and P. Stoica, "On the expectation of the product of four matrix-valued gaussian random variables," *IEEE Transactions on Automatic Control*, vol. 33, no. 9, pp. 867–870, 1988.

[22] S. D. Blunt, T. Chan, and K. Gerlach, "Robust DOA estimation: The reiterative superresolution (RISR) algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 332–346, jan 2011.

[23] W.-J. Zeng, H. C. So, and L. Huang, "ℓ_{p}-MUSIC: Robust direction-of-arrival estimator for impulsive noise environments," *IEEE Transactions on Signal Processing*, vol. 61, no. 17, pp. 4296–4308, sep 2013.

[24] B. Friedlander and A. Weiss, "Eigenstructure methods for direction finding with sensor gain and phase uncertainties," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1988.

[25] K. N. Ramamohan, S. P. Chepuri, D. F. Comesana, G. C. Pousa, and G. Leus, "Blind calibration for acoustic vector sensor arrays." *IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2018.

[26] R. T. Rajan and A.-J. van der Veen, "Joint ranging and synchronization for an anchorless network of mobile nodes," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 1925–1940, Apr. 2015.

[27] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, aug 2010.

[28] K. Han and A. Nehorai, "Nested vector-sensor array processing via tensor modeling," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2542–2553, May 2014.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.