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

CAS-2020-00

M.Sc. Thesis

Clock skew invariant beamforming

Laurens Buijs B.Sc.

Abstract

This thesis is focused on Wireless Acoustic Sensor Networks (WASNs) used for beamforming in a speech enhancement task. Since each node in a WASN has its own clock, clock offsets and clock skews between the nodes are inevitable. Clock offsets and clock skew can be detrimental to the beamformer performance. In this thesis we focus on the effect of clock skew on the beamformer performance. Existing methods for clock skew compensation for the speech enhancement application do this explicitly. In this thesis we investigate the possibility to formulate the beamformer such that explicit clock skew compensation is not necessary.

Instead, we propose an algorithm for implicit clock skew compensation, which takes advantage of the Generalized Eigenvalue Decomposition (GEVD) to construct beamformers (e.g. Minimum Variance Distortionless Response (MVDR)), recently proposed in the literature. Using the GEVD, no explicit compensation has to be applied to the received data. Compared to the state-of-the-art, where clock skew estimation/compensation algorithms are used, this reduces the computational complexity for beamformer processing.

The algorithm depends on exact knowledge of the noisy correlation matrix across the microphones. In practice, this matrix is unknown and estimation will reduce the performance of the proposed algorithm. We therefore quantify the error made in the estimation of the correlation matrix using the standard Welch method and also look at a recursive smoothing based method for correlation matrix estimation. Compared to a selected state-of-the-art algorithm, the proposed algorithm shows similar or better performance using this recursive smoothing method. For future work on this subject, more study can be done on correlation matrix estimation methods, as these play a key role in clock skew invariant beamforming.

Clock skew invariant beamforming
In wireless acoustic sensor networks

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Laurens Buijs B.Sc.
born in Delft, The Netherlands

This work was performed in:

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

Delft University of Technology
Department of
Microelectronics & Computer Engineering

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 **“Clock skew invariant beamforming”** by **Laurens Buijs B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 10/09/2020

Chairman: ______________________

dr.ir. R.C. Hendriks

Advisor: ______________________

dr. J. Martinez

Committee Members: ______________________

dr.ir. R.C. Hendriks

____

dr.ir. S. Vollebregt

____

dr. J. Martinez

Abstract

This thesis is focused on Wireless Acoustic Sensor Networks (WASNs) used for beamforming in a speech enhancement task. Since each node in a WASN has its own clock, clock offsets and clock skews between the nodes are inevitable. Clock offsets and clock skew can be detrimental to the beamformer performance. In this thesis we focus on the effect of clock skew on the beamformer performance. Existing methods for clock skew compensation for the speech enhancement application do this explicitly. In this thesis we investigate the possibility to formulate the beamformer such that explicit clock skew compensation is not necessary.

Instead, we propose an algorithm for implicit clock skew compensation, which takes advantage of the Generalized Eigenvalue Decomposition (GEVD) to construct beamformers (e.g. Minimum Variance Distortionless Response (MVDR)), recently proposed in the literature. Using the GEVD, no explicit compensation has to be applied to the received data. Compared to the state-of-the-art, where clock skew estimation/compensation algorithms are used, this reduces the computational complexity for beamformer processing.

The algorithm depends on exact knowledge of the noisy correlation matrix across the microphones. In practice, this matrix is unknown and estimation will reduce the performance of the proposed algorithm. We therefore quantify the error made in the estimation of the correlation matrix using the standard Welch method and also look at a recursive smoothing based method for correlation matrix estimation. Compared to a selected state-of-the-art algorithm, the proposed algorithm shows similar or better performance using this recursive smoothing method. For future work on this subject, more study can be done on correlation matrix estimation methods, as these play a key role in clock skew invariant beamforming.

Acknowledgments

I would like to thank my advisors dr.ir. Richard Hendriks and dr. Jorge Martinez for their continuous support during my time at CAS. Their assistance helped me set the direction for the thesis, throughout the whole process. Also, they made me see that the delivery of my results was just as important as the results themselves. During the writing phase Richard and Jorge allowed numerous rounds of feedback, which was very meticulous. They also managed to ask new questions at every feedback round, which I found very helpful.

I am also grateful for the feedback from Jurriaan, Stijn, Gijs and Jasper, who read through (parts of) the unfinished thesis. They each approached the problem from a different angle, which was insightful. Furthermore, their comments helped me find structure in the many pages.

I want to thank my girlfriend Paula for being there for me. She gave me more attention than I could ask for during this process. Also, I would like to thank her parents, Bea and Ed, for letting me stay at their place during the first month of the COVID-19 pandemic. I had a wonderful time and could also play drums in their studio.

Finally, I want to thank my parents, Willem-Jan and Monique, for their support with everything, including this study.

Laurens Buijs B.Sc.
Delft, The Netherlands
10/09/2020

Contents

List of Figures

List of Tables

1 Introduction

In the past decade, portable devices have become increasingly important in our daily lives. The computational power of devices such as mobile telephones and tablets has become greater over the years, to an extent that they take over tasks that were only performed on a personal computer (PC) before. In addition to their increased computational power, they are able to connect wirelessly to a network.

Another development of the past 15 years is the Internet of Things (IoT) [3]. The IoT is a general name for a network of “intelligent devices”, connected through the internet. This network can be used to perform e.g. a distributed measurement task, such as sensing temperatures at different locations in a room.

Portable and IoT devices are thus able to operate in a wireless network, allowing them to share data. This data may consist of e.g. a text message, location information or measurements.

1.1 Wireless Acoustic Sensor Networks and beamforming

The aforementioned devices are often equipped with an acoustic sensor, i.e. a microphone. Networks of devices with such sensors are referred to as Wireless Acoustic Sensor Networks (WASNs) [4] [5]. A WASN can be viewed as a network of wirelessly connected nodes, each equipped with a microphone. The nodes are capable of recording data from their respective microphones, and transmitting it to other nodes for further processing. WASNs can be used for multi-microphone tasks, such as beamforming, in order to estimate a particular target sound source in a noisy environment. With beamforming spatial selectivity is employed by manipulating the amplitude and phase differences of signals received at different microphones.

Beamforming is a technique that can be used for speech enhancement, by focusing on a target speech source in a noisy environment. Spatial and spectral properties of the received signals are used in order to focus on a desired target source, in the presence of interference and background noise.

Conventionally, microphones used for beamforming were all part of the same device. This means that the microphones all use the same internal clock. Having the same notion of time is important when employing beamforming techniques as one plays with phase (i.e. time) differences across the different microphones. However, with WASNs, each device potentially has its own clock. This will have a significant effect on the beamforming algorithms as will be detailed further in the next sections. See Figure 1.1 for a visual representation of a beamformer implemented on a WASN where clock skew is present.

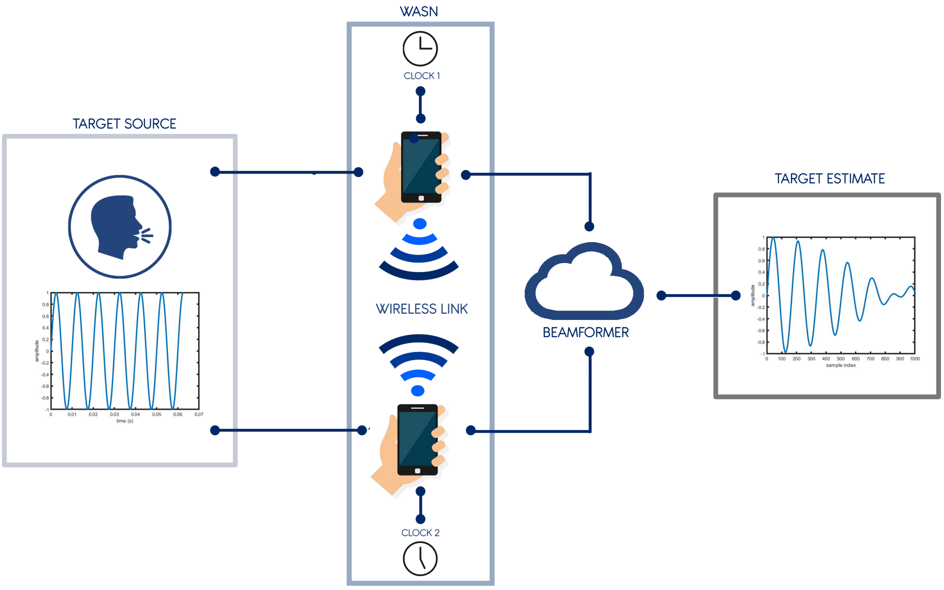


Figure 1.1: A depiction of the problem addressed in this work. Simple beamforming in a wireless acoustic sensor network (WASN) setup: two microphones in different devices (mobile phones) are used. The signals received at the microphones from the source signal (sine wave) are sampled. A clock skew between the devices causes the output from the beamformer (delay-and-sum) to be distorted with respect to the source signal.

1.2 Clock behaviour in Wireless Acoustic Sensor Networks

In a WASN each node is equipped with its own processing circuitry. These processing circuits can be assumed to operate digitally, which means they run off a clock to generate the discrete time instants at which the processing circuit performs a computation. This clock is usually embedded in the processing circuitry. Consequently each node has its own independent clock. Because the clocks are independent and there is no synchronization between them, they run asynchronously. This asynchronous behaviour can be separated into two phenomena: clock offset and clock skew; the asynchronous behaviour is usually a combination of both phenomena. Clock offset is when the ticks of the clocks occur at offset time instants, this is caused by different starting times of the clocks. Clock skew is when the clock frequency of one clock is (slightly) different than the frequency of another clock, this is caused by tiny manufacturing defects in the clock crystals [6] [7].

Beamforming explicitly uses the time differences between signals received by different microphones, to focus on a desired target source. However, since clock offset and clock skew affect the time instants at which samples are taken at microphones, the time differences between received signals are affected. An impact on the beamformer performance can thus be expected. Kotti et al. [8] has recently shown that for a certain group of beamformers the beamformer performance is invariant to clock offsets. For clock skew this invariance property has not been shown (yet) and most works focus on the compensation of the effect of clock skew. The works can be divided in protocol based synchronization and signal processing based algorithms. In protocol based synchronization, artifacts of the protocol are used to periodically transmit a specific sequence between nodes to maintain clock synchronization, as in Wang et al. [9]. In signal processing based algorithms, properties of the received signals are used, such as correlation between signals received by different microphones, as in Bahari et al. [1]. In that work, a signal model is derived to study the effects of clock skew on beamformers, implemented in a WASN. This model shows that a clock skew can be modeled as a linearly increasing clock offset. Because of this property, we study the applicability of the work of Kotti et al. [8] to a clock skew affected WASN beamforming setup in this thesis.

1.3 Contribution

The asynchronous clock behaviour in WASNs may impact the performance of beamformers. For a situation where only clock offset is present, Kotti et al. [8] has shown that a certain group of beamformers is invariant to clock offsets. In this thesis we apply the theory from [8] to a clock skew affected WASN beamforming setup to investigate to which extent beamforming algorithms can also be made clock skew invariant. We use a signal model similar to the one derived in [1].

1.4 Outline

The thesis is structured as follows: In Chapter 2 the general signal model to be used in this thesis will be defined. In Chapter 3 the clock synchronization problem will be discussed in more detail, as well as the application of beamforming in a WASN setup: a simple and intuitive example of a beamformer will given, to illustrate the problem of clock synchronization. Then the work of Kotti et al. [8] will be discussed. Finally, related works in literature regarding clock skew in a WASN setup will be summarized. In Chapter 4 a signal model is given for the multi microphone setup with clock skew, then the theory of Kotti et al. [8] is applied to this model. Chapter 5 discusses the practical side of the approach devised in Chapter 4: the estimation of correlation matrices is necessary for beamforming, which causes errors in the beamformer output in the presence of clock skew. These errors are quantified for two different estimation methods. In Chapter 6 a beamformer setup in a WASN environment is simulated, to obtain results for the proposed method and compare them with an existing method by Bahari et al. [1].

Preliminaries 2

2.1 Signal model

For the multi-microphone, multi-source setup we assume the following underlying model:

$$
\begin{aligned}
y_i(t) &= \sum_{p=1}^{N_s} h_{i,p}(t) * s_p(t) + v_i(t) \\
&= x_i(t) + v_i(t) \qquad (2.1)
\end{aligned}
$$

where $i = 1, 2, \ldots, M$ is the microphone index, N_s the number of target sources, t is the continuous time and $*$ is defined as linear convolution. Here y_i denotes the received signal at microphone i, s_p the target source signal p, $h_{i,p}$ the room impulse response (RIR) from target source p to microphone i and v_i is additive noise. The noise term consists of all signals that are not caused by the target (i.e. microphone self-noise and interfering sources).

After sampling with sampling frequency f_s we obtain:

$$
y_i[n] = y_i\left(t\right)\Bigg|_{t=\frac{n}{f_s}}. \qquad (2.2)
$$

We then define the discrete Fourier transform (DFT) as:

$$
X[k] = \sum_{n=0}^{K-1} x[n] e^{-j\frac{2\pi kn}{K}} \qquad (2.3)
$$

where k is the k-th frequency bin, n the sample index and K the number of samples used.

Approximating the linear convolution in (2.1) with a circular convolution we obtain the signal model in the frequency domain, where the linear convolution result is obtained by piece-wise multiplication of the DFT coefficients:

$$
\begin{aligned}
Y_i[k] &= \sum_{p=1}^{N_s} H_{i,p}[k] S_p[k] + V_i[k] \\
&= X_i[k] + V_i[k]. \qquad (2.4)
\end{aligned}
$$

As the speech signals are non-stationary, we consider in this work the short-time DFT (STFT), that is:

$$
X[k,l] = \sum_{n=0}^{K-1} x\left[lK_h + n\right] w[n] e^{-j\frac{2\pi kn}{K}} \qquad (2.5)
$$

where k is the k-th frequency bin, l the l-th STFT frame, K the STFT frame length, K_h the hop size and $w[n]$ a discrete time smoothing window.

As all the processing is done per frequency and time frame, we drop the frequency and time frame indices for readability. Using a stacked vector notation across the microphones we then get:

$$\begin{aligned} \mathbf{y} &= \sum_{p=1}^{N_s} \mathbf{d}_p S_p + \mathbf{v} && (2.6) \\ &= \mathbf{x} + \mathbf{v} && (2.7) \end{aligned}$$

where $\mathbf{d}_p$ is the steering vector towards the p-th source (see section 2.2). S_p is the p-th source in the frequency domain and $\mathbf{v}$ the vector with noise DFT coefficients for all microphones.

2.2 Steering vector

The steering vector is obtained when applying the DFT (2.3) to the RIR $h[n]$. This gives it a general form of:

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_M \end{pmatrix}. \tag{2.8}$$

Alternatively, the relative steering vector may be defined as:

$$\mathbf{d} = \begin{pmatrix} 1 \\ d_2/d_1 \\ \vdots \\ d_M/d_1 \end{pmatrix} \tag{2.9}$$

i.e. the steering vector is normalized with respect to the first element d_1. The relative steering vector (2.9) is usually found when using a *blind* estimation method for the steering vector, such as the GEVD. Refer to section 2.5 where different methods are discussed for estimating the steering vector. In this work, we use the definition in (2.9) and (2.6) is written as:

$$\mathbf{y} = \sum_{p=1}^{N_s} \mathbf{d}_p X_{1,p} + \mathbf{v} \tag{2.10}$$

where $\mathbf{d}_p$ is the relative steering vector towards the p-th source. $X_{1,p}$ is the signal received from the p-th target at microphone 1 in the frequency domain and $\mathbf{v}$ the vector with noise DFT coefficients for all microphones. The consequence of using the relative steering vector in (2.10) is that the received target signals at the microphones are with respect to the received target signal at microphone 1 instead of the target signal itself.

2.3 Stochastic processes

We assume that the signals $\mathbf{x}$ and $\mathbf{v}$ are realizations of the zero-mean wide-sense stationary processes $\mathbf{X}$ and $\mathbf{V}$, respectively. We assume mutually uncorrelated source and noise processes. That is: $\mathbb{E}[\mathbf{X}\mathbf{V}^{\mathrm{H}}] = \mathbb{E}[\mathbf{V}\mathbf{X}^{\mathrm{H}}] = \mathbf{0}_{M\times M}$. The spatial correlation matrix of the microphone signals is then given by:

$$\begin{aligned}\mathbf{R}_{\mathbf{Y}} &= \mathbb{E}[\mathbf{Y}\mathbf{Y}^{\mathrm{H}}] = \mathbb{E}[\mathbf{X}\mathbf{X}^{\mathrm{H}} + \mathbf{X}\mathbf{V}^{\mathrm{H}} + \mathbf{V}\mathbf{X}^{\mathrm{H}} + \mathbf{V}\mathbf{V}^{\mathrm{H}}] \\ &= \mathbb{E}[\mathbf{X}\mathbf{X}^{\mathrm{H}}] + \mathbb{E}[\mathbf{X}\mathbf{V}^{\mathrm{H}}] + \mathbb{E}[\mathbf{V}\mathbf{X}^{\mathrm{H}}] + \mathbb{E}[\mathbf{V}\mathbf{V}^{\mathrm{H}}] \\ &= \mathbb{E}[\mathbf{X}\mathbf{X}^{\mathrm{H}}] + \mathbb{E}[\mathbf{V}\mathbf{V}^{\mathrm{H}}] = \mathbf{R}_{\mathbf{X}} + \mathbf{R}_{\mathbf{V}}. \end{aligned} \tag{2.11}$$

2.4 Beamforming

The general expression for a beamformer output is given by:

$$\hat{X}_1 = \mathbf{w}^{\mathrm{H}}\mathbf{y} \tag{2.12}$$

where $\mathbf{w}$ is the vector with beamformer filter weights, $\mathbf{y}$ is the received microphone data and $\hat{X}_1$ is the estimated target signal at microphone 1. In Table 2.1 several commonly used beamformers are given, with their mathematical expressions. The single target source versions of the beamformers depend on the steering vector $\mathbf{d}$, either directly or via $\mathbf{R}_{\mathbf{X}}$ as we can write $\mathbf{R}_{\mathbf{X}} = \sigma^2_{\mathcal{X}_1}\mathbf{d}\mathbf{d}^{\mathrm{H}}$.

Beamformer	**Expression**
Delay-and-sum	$\mathbf{w} = \dfrac{\mathbf{d}}{\mathbf{d}^{\mathrm{H}}\mathbf{d}}$
Speech Distortion Weighted (SDW) Wiener filter	$\mathbf{w} = (\mathbf{R}_{\mathbf{X}} + \mu\mathbf{R}_{\mathbf{V}})^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{e}_1$
Classical Multichannel Wiener filter	$\mathbf{w} = \mathbf{R}_{\mathbf{Y}}^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{e}_1$
Classical Multichannel Wiener filter (single target source)	$\mathbf{w} = \sigma^2_{\mathcal{X}_1}\mathbf{R}_{\mathbf{Y}}^{-1}\mathbf{d}$
Minimum Variance Distortionless Response (MVDR) (single target source)	$\mathbf{w} = \dfrac{\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{d}}{\mathbf{d}^{\mathrm{H}}\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{d}}$

Table 2.1: Commonly used beamformers with their mathematical expressions, as derived in [2]. We have $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^M$ as the standard basis vector.

For the single target source case ($\operatorname{rank}(\mathbf{R}_{\mathbf{X}}) = 1$) we see from Table 2.1 that these beamformers depend on the steering vector $\mathbf{d}$. The mathematical expression for $\mathbf{d}$ and several methods to obtain it are discussed in the next sections.

2.5 Estimation of steering vector $\mathbf{d}$

The steering vector $\mathbf{d}$ can be obtained in different ways, as summarized below:

- Directly from the RIR. This can be done by transmitting a training sequence from the target source. From the received signals at the different microphones the respective RIRs are obtained through deconvolution, since the received signals are the result of convolution of the source with the RIRs, as defined in (2.1). The DFT is applied to the recorded RIRs and the steering vector is found.
- Based on microphone setup geometry. In this case only the direct path is considered. Given the distance from the target source to each microphone and the speed of sound ($v \approx 340\,\mathrm{m/s}$), the time delay between transmitting and receiving is known. These delays are visible in the steering vector as phase shifts, see (2.13).
- Through the Generalized Eigenvalue Decomposition (GEVD). The composition of the received signal $\mathbf{y}$ in (2.7) allows us to decompose it in a signal and noise subspace. From the signal subspace we can determine the steering vector, as presented in [10].

The direct path steering vector is given by:

$$\mathbf{d} = \begin{pmatrix} 1 \\ |a_2|e^{j\tau_2} \\ \vdots \\ |a_M|e^{j\tau_M} \end{pmatrix} \tag{2.13}$$

where $|a_i|$ is the damping of the signal at microphone i with respect to the reference microphone and τ_i is the delay of the signal at microphone i with respect to the reference microphone.

From the expression for the delay-and-sum beamformer in Table 2.1 it is easy to see that the beamformer compensates for the differences in delay and damping that the target source experienced when travelling to the different microphones, using the information in the steering vector. In the next chapter we will look in more detail at the effect of clock offset and clock skew on this capability.

Problem formulation and related work

3

3.1 Clock synchronization in Wireless Acoustic Sensor Networks

In Wireless Acoustic Sensor Networks (WASNs), the different devices (nodes) have their own processing circuit. Each node has its own clock to drive this processing circuit, and the nodes thus have independent, non synchronous clocks.

For this thesis, we will assume the devices are similar, as well as their clocks. This means that the clocks will have approximately the same clock frequency. However, in practice, the clock frequencies are never perfectly the same across nodes.

The frequency offset is often defined in parts per million (ppm) [11]. One ppm equals 10^{-6}. For example a maximum frequency offset of ± 100 ppm from a frequency of 16 kHz indicates a frequency range of 16000 ± 1.6 Hz.

As an example, the Bluetooth and Zigbee standards require ± 25 ppm frequency stability [12] and the GSM standard requires a tighter stability of ± 5 ppm [12]. The IEEE 802.11a Wireless Local Area Network (WLAN) standard specifies a carrier offset in the range of ± 40 ppm [13].

Typical quartz crystal oscillators may have frequency offsets in the range of 10 ppm-100 ppm [11] [14]. However, with the use of compensation techniques, oscillators with better frequency stability may be developed: Oven Controlled Crystal Oscillators (OXCOs) reach frequency stability as low as ± 1 ppb [15]. For this work, we assume the frequency offset may be up to 100 ppm when using state-of-the-art quartz oscillators.

Besides the frequency offset, the time instants when the clocks tick almost surely will have an offset as well. This is due to the electrical circuits, and thus the clocks, starting at different time instants. The devices are independent, therefore we cannot guarantee a perfectly synchronous start of the electrical circuits.

The non synchronous clock behaviour can thus be divided in two classes:

- Clock offset: the clocks tick at the same rate, but the tick times are offset by a constant amount. This is due to the different clock start times.
- Clock skew: the clocks tick at a different rate, due to the imperfection of the quartz crystals.

3.2 Application of clock synchronization to beamforming

This thesis focuses on the application of beamforming in WASNs, in particular for a speech enhancement task. The devices could be mobile phones, hearing aids or general distributed processors, whose microphones are used for the beamforming task. In this section we demonstrate the effect of clock offset and clock skew on beamforming. The target source (speech) in Figure 3.1 is in blue, and there is an interfering source in red. The aim of beamforming is to recover the target signal.

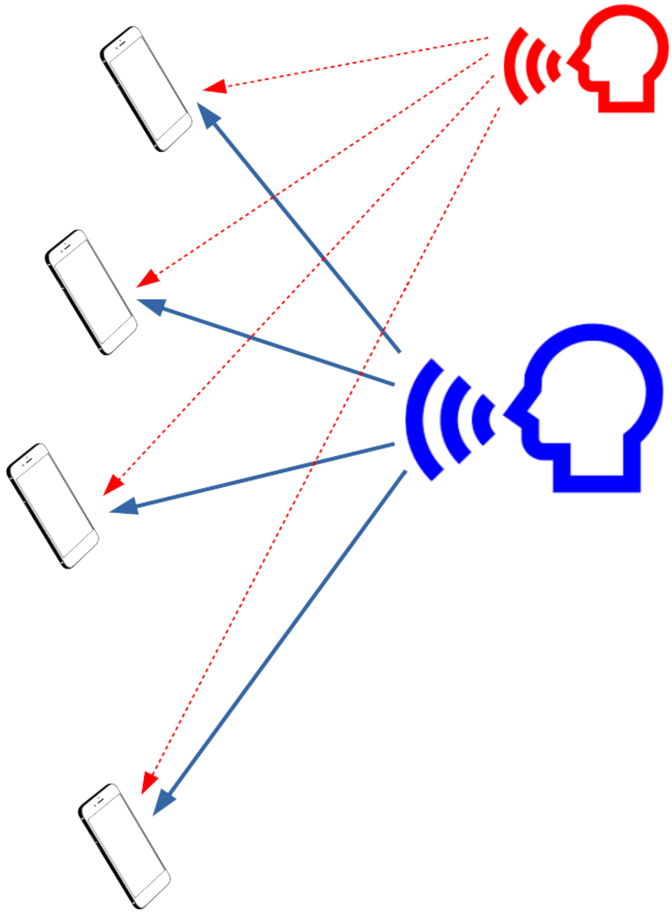


Figure 3.1: Wireless acoustic sensor network beamforming setup, the target source is in blue, the interference/noise is in red.

The general expression for a beamformer output is given by:

$$\hat{X}_1 = \mathbf{w}^{\mathrm{H}}\mathbf{y} \tag{3.1}$$

where $\mathbf{w}$ can be any beamformer, for example one of the beamformers from Table 2.1. Generally, $\mathbf{w}$ will estimate the target X_1 by properly compensating for the differences in delay and damping that the target source experienced when travelling to the different microphones. By doing so it reduces the effect of the interfering noise sources. Beamformers rely on the fact that the time delay and damping can properly be compensated. However, if the different clocks are not synchronized, additional time delays will be experienced which will lead to an incorrect time delay compensation by the beamformer.

The effect of asynchronous clocks on beamforming can best be illustrated with a simple example. Suppose the target source is a sine wave, and 2 microphones are set up at the same distance from the source. Let's for simplicity assume there is no noise and we are in free-field.

First, the ideal scenario of perfect synchronization is shown in Figure 3.2. The sample points are aligned in time for both microphones.

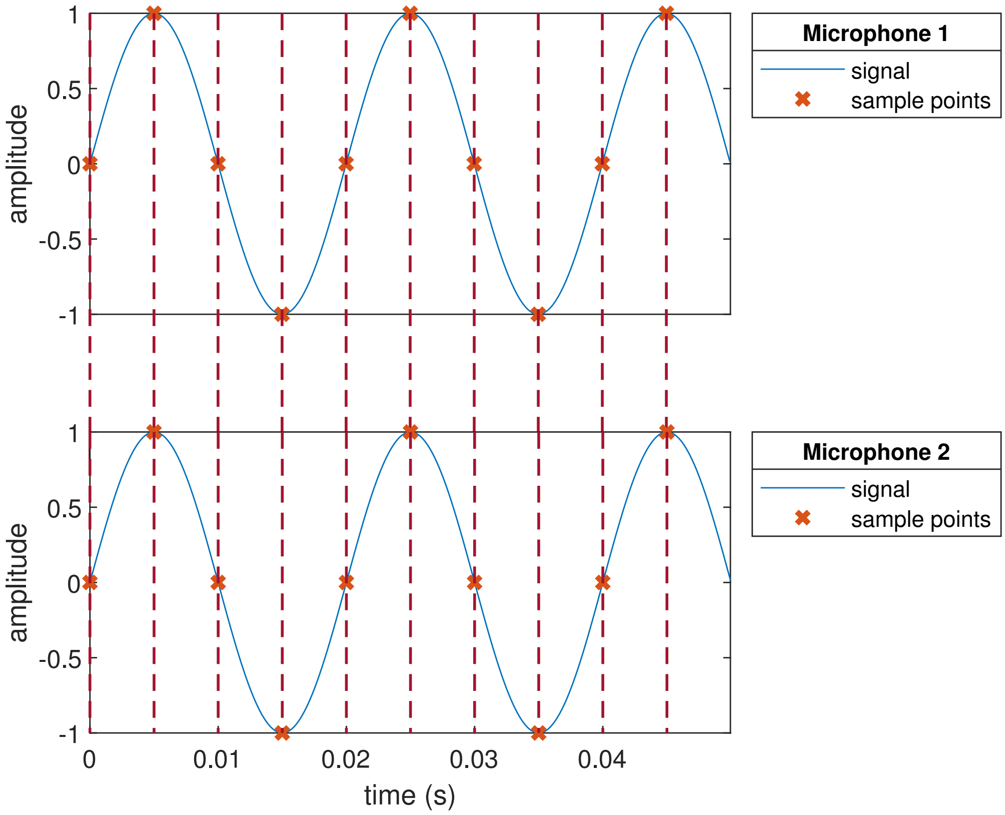


Figure 3.2: Received signals with perfectly synchronized clocks. Two microphones are used. The sine frequency is $f_c = 50\,\text{Hz}$. The sample rate is $f_s = 200\,\text{Hz}$.

The situation with clock offsets is shown in Figure 3.3. Microphone 2 samples at later time instants than microphone 1.

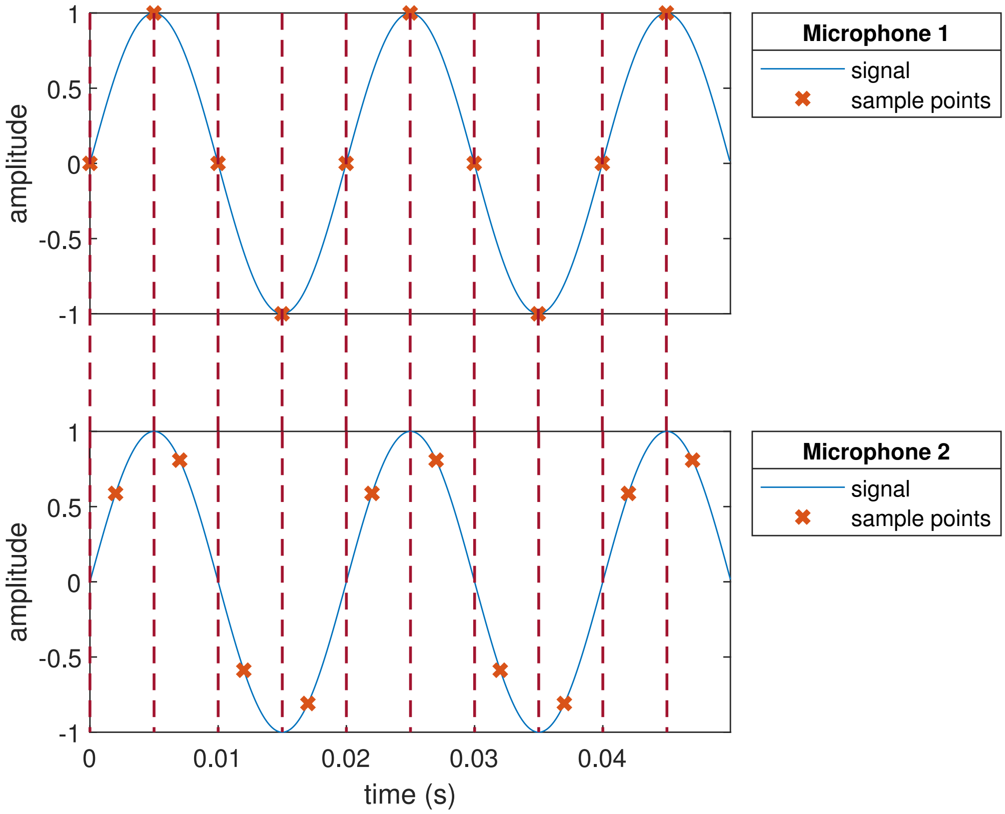


Figure 3.3: Received signals with offset clocks. Two microphones are used. The sine frequency is $f_c = 50\,\text{Hz}$. The sample rate is $f_s = 200\,\text{Hz}$. Microphone 2 samples 2 ms later than microphone 1.

The situation with clock skew is shown in Figure 3.4. Microphone 2 samples faster than microphone 1. Therefore the samples of microphone 2 are taken at increasingly earlier time instants than microphone 1.

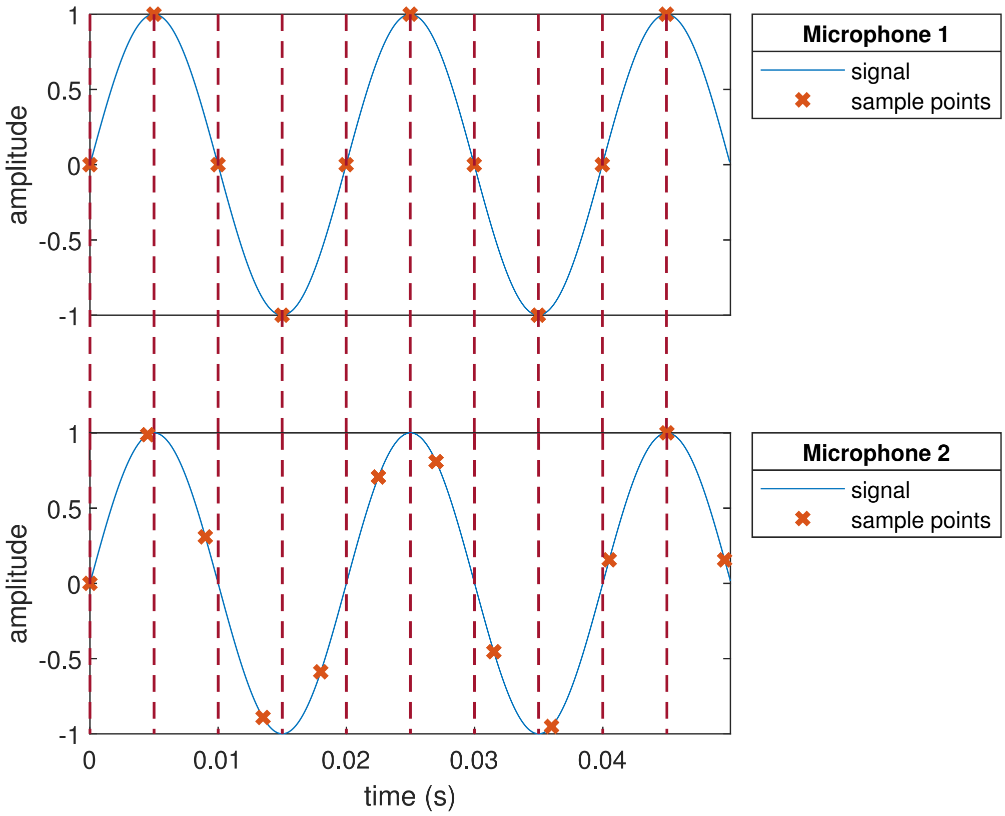
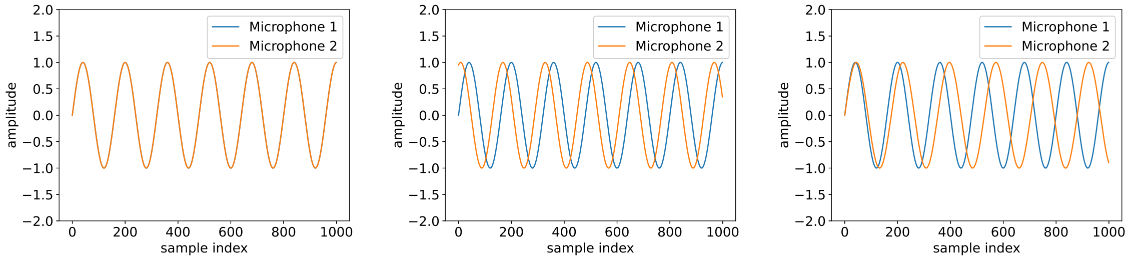


Figure 3.4: Received signals with offset clocks. Two microphones are used. The sine frequency is $f_c = 50\,\mathrm{Hz}$. Microphone 1 samples at $f_s = 200\,\mathrm{Hz}$ and microphone 2 samples $10\,\%$ faster at $f_s = 220\,\mathrm{Hz}$.

For this illustration, we consider a very simple type of beamformer where the two simulated microphone signals are simply averaged. This would correspond to the application of a delay and sum beamformer, which would be the optimal beamformer in the case of noise that is uncorrelated across the microphones.

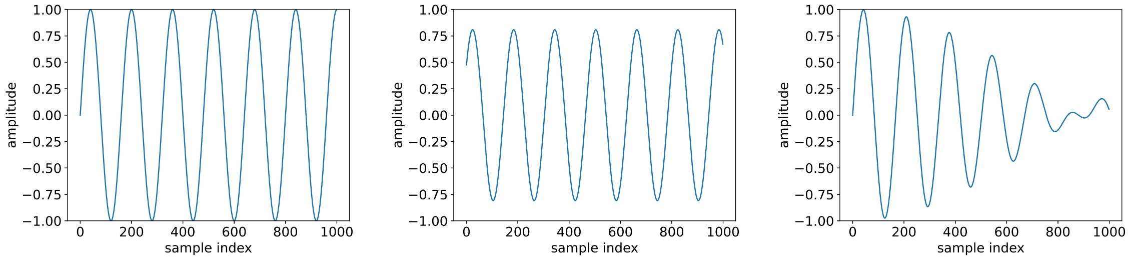
Let us now consider the received signals vs. the sample index, with the following setup: sampling rate $f_s = 16\,\mathrm{kHz}$, sine wave with frequency $f_c = 1\,\mathrm{kHz}$, see Figure 3.5, under the condition that the two clocks are synchronized (Figure 3.5a), have a clock offset of 2 ms (Figure 3.5b) and for the case that the two clocks have a clock skew of 10 %.



(a) Perfect synchronization (b) Clock offset $\tau = 2\,\mathrm{ms}$ (c) Clock skew of 10 %

Figure 3.5: Received signals for different clock synchronization situations.

Applying the aforementioned delay and sum beamformer to the signals in Figure 3.5a, 3.5b and 3.5c without any synchronization of the clocks then leads to estimated source signals as depicted in Figure 3.6a, 3.6b and 3.6c, respectively.



(a) Perfect synchronization (b) Clock offset $\tau = 2\,\mathrm{ms}$ (c) Clock skew of 10 %

Figure 3.6: Source estimate after beamforming (averaging).

For the perfect synchronization case, the source signal is perfectly reconstructed. For the case with clock offset a sine is generated, but with a lower amplitude and with a phase offset. For the case with a clock skew, the sine amplitude is modulated with the difference frequency (100 Hz). To summarize, in both cases where the clocks are not synchronized, the reconstruction is inaccurate. This emphasizes that unsynchronized clocks can have a detrimental effect in the final performance. The effects of clock offset and clock skew are evaluated separately in this simple example, however, in reality both clock offset and clock skew may affect a system.

3.3 Clock offset invariant beamforming (for Generalized Eigenvalue Decomposition based beamformers)

In [8] it was shown based on the Generalized Eigenvalue Decomposition (GEVD) that beamformers from the general SDW class as given in Table 2.1 are invariant to clock offsets, under the condition that the GEVD is used to estimate $\mathbf{d}$ and not an *a priori* estimate of $\mathbf{d}$ is used.

Given the general SDW class, for different settings of μ (and with some assumptions on the signal model) the beamformers from Table 2.1 can be obtained.

The GEVD yields a continuously updated estimate of $\mathbf{d}$, because it is based on the received data $\mathbf{y}$. As seen in Figure 3.5b a clock offset introduces a phase shift in the received signal (sine wave). For the simplified situation in Figure 3.6b we can assume the following model:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{d}} X_1 \tag{3.2}$$

where $\tilde{\mathbf{d}}$ and $\tilde{\mathbf{y}}$ are respectively the relative steering vector and received data including the phase shift caused by the clock offset and X_1 is the target signal at microphone 1. In (3.2) the phase shift is incorporated in the steering vector as follows:

$$\tilde{\mathbf{d}} = \mathbf{G}\mathbf{d} \tag{3.3}$$

which for this example then becomes:

$$\tilde{\mathbf{d}} = \mathbf{G}\mathbf{d} = \begin{pmatrix} 1 & 0 \\ 0 & e^{j\tau_2} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ e^{j\tau_2} \end{pmatrix} \tag{3.4}$$

where τ_2 is the phase shift introduced by the clock offset for microphone 2. It is clear that an *a priori* estimate of $\mathbf{d}$ based on the geometry does not include this phase shift (refer to section 2.5). When the *a priori* estimate is based on the RIR, this phase shift may be included at the time of measurement. However, since the clock offset is a result of different starting times of electrical circuits, the phase shift can be different for each restart of the system. Therefore such an estimate would not be satisfactory in most cases. The GEVD allows us to decompose the received signal at runtime and correctly estimate $\tilde{\mathbf{d}}$ as seen in (3.4).

We can show for the delay-and-sum beamformer that, given (3.4), the source signal is preserved perfectly without any explicit clock synchronization. Substituting (3.2) and (3.4) in (2.12):

$$\begin{aligned} \hat{X}_1 = \tilde{\mathbf{w}}^{\mathrm{H}} \tilde{\mathbf{y}} &= \frac{\tilde{\mathbf{d}}^{\mathrm{H}}}{\tilde{\mathbf{d}}^{\mathrm{H}} \tilde{\mathbf{d}}} \tilde{\mathbf{d}} X_1 \\ &= \frac{\tilde{\mathbf{d}}^{\mathrm{H}} \tilde{\mathbf{d}}}{\tilde{\mathbf{d}}^{\mathrm{H}} \tilde{\mathbf{d}}} X_1 = X_1 \end{aligned} \tag{3.5}$$

i.e. the target source signal is perfectly preserved. Obviously, using an *a priori* estimate of the steering vector, i.e. $\mathbf{d} = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ would give an erroneous result with a phase shifted version of the signal at microphone 2 averaged with the signal at microphone 1, as in Figure 3.6b. Therefore, in [8] the GEVD is presented as a key component to clock offset invariant beamforming, and in this thesis we investigate if this theory can be applied to a clock skew affected WASN beamforming setup as well.

3.4 Related work in literature

For a moment, we broaden our scope to the more general case of clock synchronization in Wireless Sensor Networks (WSNs). In the field of clock synchronization for WSNs, a distinction can be made in protocol based synchronization and signal processing based synchronization techniques. Protocol based synchronization is discussed in works by Wobschall [16], Wang [9] and Zhao [17]. These protocol based synchronization techniques apply to WSNs, but not specifically to WASNs or a beamforming task. Since our proposed algorithm will be signal processing based, signal processing based synchronization techniques are of interest. In the following section different works will be summarized on signal processing based synchronization techniques for clock skew, applied to WASNs.

3.4.1 Signal processing based synchronization

Markovich-Golan et al. [18] approximates the effect of clock skew by a linearly increasing phase shift in the STFT domain. The coherence between two microphones is calculated, which allows for an estimate of the sample rate offset (a SRO introduces a linearly increasing phase shift in the coherence). The result from all frequency bins and time domain frames is averaged for all microphone pairs. Then the results from all microphone pairs are averaged which yields a SRO estimate. To compensate for the SRO, the received data is resampled to the sample rate of the fusion center (i.e. one of the nodes in the network) using Lagrange polynomials and the earlier obtained SRO estimates.

For the test setup two microphone arrays comprised of 6 microphones each are used. One microphone array is set to a sampling rate of $f_s = 8\,\text{kHz}$ and the other array has SRO values in the range of $(-300, -250, \ldots, 300)$ ppm. The generalized sidelobe canceler is used as beamformer. One desired target source is deployed and a number of interfering sources at random locations. The SRO estimation variance is lower than 3.2 ppm in all tested scenarios. For the scenario with one interfering source, when using the presented compensation algorithm with respect to no compensation, the increase in signal-to-distortion ratio is 11.2 dB and the increase in signal-to-noise ratio is 7.2 dB.

Cherkassky et al. [19] use the continuous wavelet transform (CWT) to estimate the SRO, and resampling in the time domain is used to synchronize the nodes. Four microphones, set to a sampling rate of $f_s = 16\,\text{kHz}$, are used in a test setup with manually manipulated SROs. The SDW multichannel Wiener filter is used as beamformer. At an average array SRO of 100 ppm an increase in array gain of 7 dB is reached with the proposed algorithm, at an average array SRO of 300 ppm this increase is 9 dB. The array gain is defined as the ratio of signal-to-interference ratio (SIR) at the input of the array to SIR at the array output.

In [20], Cherkassky et al. use the recursive band-limited interpolation (RBI) algorithm for SRO estimation. The estimation is applied sequentially in the time domain, therefore it can track a time-varying SRO. Six microphones, set to a sampling rate of $f_s = 16\,\text{kHz}$ are used in a test setup with manually manipulated SROs. The SROs are set to $(-100, -50, 0, 50, 100, 150)$ ppm. With the proposed algorithm the SIR gain improvement is 40 dB and the SNR gain improvement is 25 dB.

In [21] and [1] Bahari et al. approximate the effect of clock skew by a linearly increasing phase shift across time frames in the STFT domain, similar to [18]. The SRO is estimated using a linear coherence drift (LCD) method, followed by a least-squares algorithm to use multiple time frames. In [1] a weighted least-squares algorithm is used, for applicability of the work in [21] to multiple target sources.

A hybrid compensation framework is then devised which applies a sample synchronization in the time domain, and a compensation for the linearly increasing phase shift across time frames in the STFT domain. The time domain synchronization is applied because signals drift apart over time due to the SRO, which would cause them to be unrelated eventually. The frequency domain compensation allows for precise compensation of the phase shift caused by the SRO (up to the approximation accuracy of that particular SRO model).

For the experiments, the multichannel SDW Wiener filter is used as beamformer. The algorithm is tested for SROs up to 100 ppm. A SNR gain improvement of 3 dB is attained for a SRO of 100 ppm, with respect to using no compensation algorithm.

Zeng et al. [22] evaluate 3 different synchronization algorithms for WASNs subject to clock skew. Two algorithms are time-stamp based and one algorithm is signal based, using the algorithm described in [18]. The time-stamp based algorithms degrade the performance of the MVDR beamformer in the presence of measurement uncertainty or noise, whereas the signal based algorithm corresponds to the ideal centralized MVDR beamformer. The signal based algorithm needs more data however, i.e. more data has to be transferred between the nodes.

The discussed signal processing based algorithms are referred to as *blind synchronization* algorithms in the literature, using an SRO estimation algorithm based on the available data and then applying a compensation based on the estimated SROs. A performance comparison of three *blind synchronization* algorithms discussed here is shown in Table 3.1.

In this thesis, the approach used by Bahari et al. [1] is chosen as the reference *blind synchronization* algorithm. Our proposed algorithm will approximate the effect of clock skew by a linearly increasing phase shift across time frames in the STFT domain, thus the same signal model is used as in the reference algorithm. The difference between the reference algorithm and our proposed algorithm, is that the reference algorithm applies explicit compensation for the SRO whereas our proposed algorithm applies implicit compensation. Because of the similarities between the selected reference algorithm and our proposed algorithm, we argue that it is a logical choice. Also, the reference algorithm yields performance similar to using the true SRO for compensation, for SROs up to 100 ppm.

Author	**SRO estimation**	**SRO compensation**	**Perfomance measure(s)**	**Result(s)**
Markovich-Golan et. al. [18]	LCD	Lagrange polynomial time domain resampling	SDR gain difference SNR gain difference	$< 0.8\,\text{dB}$ $< 0.2\,\text{dB}$
Cherkassky et. al. [20]	RBI	Time domain resampling	SIR gain difference SNR gain difference	$< 5\,\text{dB}$ $\approx 5\,\text{dB}$
Bahari et. al. [1]	LCD	Time domain sample drop, frequency domain phase shift	SNR gain difference (includes interference)	$< 0.01\,\text{dB}$

Table 3.1: Comparison of the performance of three blind synchronization algorithms discussed in this section. The performance measure(s) column compares the ideal synchronized performance with the performance under the proposed algorithm from the respective paper. SDR denotes the signal-to-distortion ratio, SNR denotes the signal-to-noise ratio and SIR denotes the signal-to-interference ratio.

3.5 Research question

The effect of clock skew on beamforming in a WASN can be detrimental to the final performance of the beamformer. In the previous section several algorithms have been discussed which can compensate for the effect of a SRO. It has been shown by Kotti et al. in [8] that GEVD based beamformers are invariant to clock offset. From [8] and [1] we can establish that clock skew is the same as a linearly increasing clock offset. Given this similarity, we state our research question.

Is the theory for clock offset invariant beamforming applicable to clock skew affected beamformers in a WASN?

To answer the research question, the thesis will follow this structure: in Chapter 4 a signal model is given for the multi microphone setup with clock skew, then the theory of Kotti et al. [8] is applied to this model. Chapter 5 discusses the practical side of the approach devised in Chapter 4: the estimation of data matrices is necessary for beamforming, which causes errors in the beamformer output in the presence of clock skew. These errors are quantified for two different estimation methods. In Chapter 6 a beamformer setup in a WASN environment is simulated, to obtain results for the proposed algorithm and compare them with the *blind synchronization* algorithm presented by Bahari et al. [1].

Generalized Eigenvalue Decomposition for clock skew model 4

4.1 Signal model with clock skew

Suppose we have a multi microphone setup with M microphones, where $M-1$ microphones experience each a different clock skew – sample rate offset (SRO) – with respect to a reference microphone, say microphone 1. Without loss of generalization, we assume that this is the fastest sampling microphone. We now have for the sampling rate $f_{s,i}$ at microphone i:

$$f_{s,i} = f_{s,1}(1+\epsilon_i) = f_{s,\mathrm{ref}}(1+\epsilon_i) \tag{4.1}$$

where $i = 1, 2, \dots M$ and $|\epsilon_i| \ll 1$ is the SRO for the i-th microphone. Note that $\epsilon_1 = 0$ and $\epsilon_i \leq 0$ for $i \neq 1$. Denoting the continuous-time signal at microphone i as $y_i(t)$ and its sampled counterpart as $y_i[n]$, both without clock skew, we can write:

$$y_i[n] = y_i\left(\frac{n}{f_{s,\mathrm{ref}}}\right) \tag{4.2}$$

where $n = 0, 1, \dots, N-1$. For the signals for the same microphone with clock skew, we have:

$$\tilde{y}_i[n] = y_i\left(\frac{n}{f_{s,\mathrm{ref}}(1+\epsilon_i)}\right) \tag{4.3}$$

where $\tilde{y}_i[n]$ is the sampled signal with clock skew. Any microphone i with a positive clock skew samples at increasingly earlier time instants than without clock skew. Therefore the signal stored in $\tilde{y}_i[n]$ becomes increasingly delayed from the signal in $y_i[n]$. We now define a sample difference which indicates how many samples we need to delay $y_i[n]$ at sample n, to obtain $\tilde{y}_i[n]$. Recalling (4.2) and (4.3) we may now write the sample difference ΔN between the situation with clock skew and without clock skew as follows:

$$\Delta N = f_{s,\mathrm{ref}}\left(\frac{n}{f_{s,\mathrm{ref}}(1+\epsilon_i)} - \frac{n}{f_{s,\mathrm{ref}}}\right) = \frac{n - n(1+\epsilon_i)}{1+\epsilon_i} = \frac{-n\epsilon_i}{1+\epsilon_i}. \tag{4.4}$$

In practice, the SRO values are in the range of ppm, so we can assume $\epsilon_i \ll 1$. In that case we can approximate (4.4) by:

$$\Delta N \approx -n\epsilon_i. \tag{4.5}$$

This means that the effect of the SRO can be approximated by a linearly increasing sample delay in the time domain. With beamforming, we are working in the frequency domain. We can use the following property here:

$$x[n-p] \leftrightarrow X[k]e^{\frac{-j2\pi kp}{K}} \tag{4.6}$$

i.e. a sample delay of p samples translates into a phase shift in the Fourier domain. We can write:

$$\tilde{y}_i[n] = y_i[n - n\epsilon_i] \leftrightarrow \tilde{Y}_i(k) = Y_i(k)e^{\frac{-j2\pi k n\epsilon_i}{K}}. \tag{4.7}$$

A problem arises, as the phase shift is different for each time domain index n, and obviously our Fourier window length will never be equal to just 1 sample. In fact, in practice we divide the time domain signal into multiple frames and apply a windowing function and Fourier transform for each frame. As introduced in Chapter 2, this method is called the Short Time Fourier Transform (STFT). The reason for splitting the time domain signal in frames is because the signals under consideration are non-stationary. By working with the STFT we aim to obtain wide-sense stationary data per time frame.

Using the STFT we have frames of length K, with a hop size of $K_h = \gamma K$ where γ is the frame overlap factor. To compensate for the clock skew in the frequency domain, we approximate the linearly-increasing time shift $-n\epsilon_i$ by a constant time shift across the time frame. A logical choice would be to let the constant time shift be determined by the time shift experienced by the mid-frame sample, since this gives on average the lowest error. With the parameters given before, the mid-frame sample index is: $m_l = K_h l + K/2 = \gamma K l + K/2 = K(\gamma l + 1/2)$, where $l = 0, 1, \ldots, \mathcal{L} - 1$ is the frame index. Referring to the STFT signal model in Chapter 2 and recalling (4.7), we can now write:

$$\tilde{Y}_i(k,l) = Y_i(k,l)e^{-j2\pi k(\gamma l + 1/2)\epsilon_i}. \tag{4.8}$$

This model thus approximates the clock skew effect with a linearly-increasing phase shift across the STFT frames, although in reality the phase shift increases linearly across the samples. Therefore the approximation is linearly-increasing, piecewise-constant with respect to the real situation. We assume however that the maximum sample shift inside a single STFT frame is much smaller than 1 sample, i.e. $K\epsilon_i \ll 1$. In that case the linearly-increasing, piecewise-constant approximation is accurate enough, because all samples in a single STFT frame experience almost the same phase shift. In Figure 4.1 the sample shift is plotted for the real situation vs. the approximation by the model.

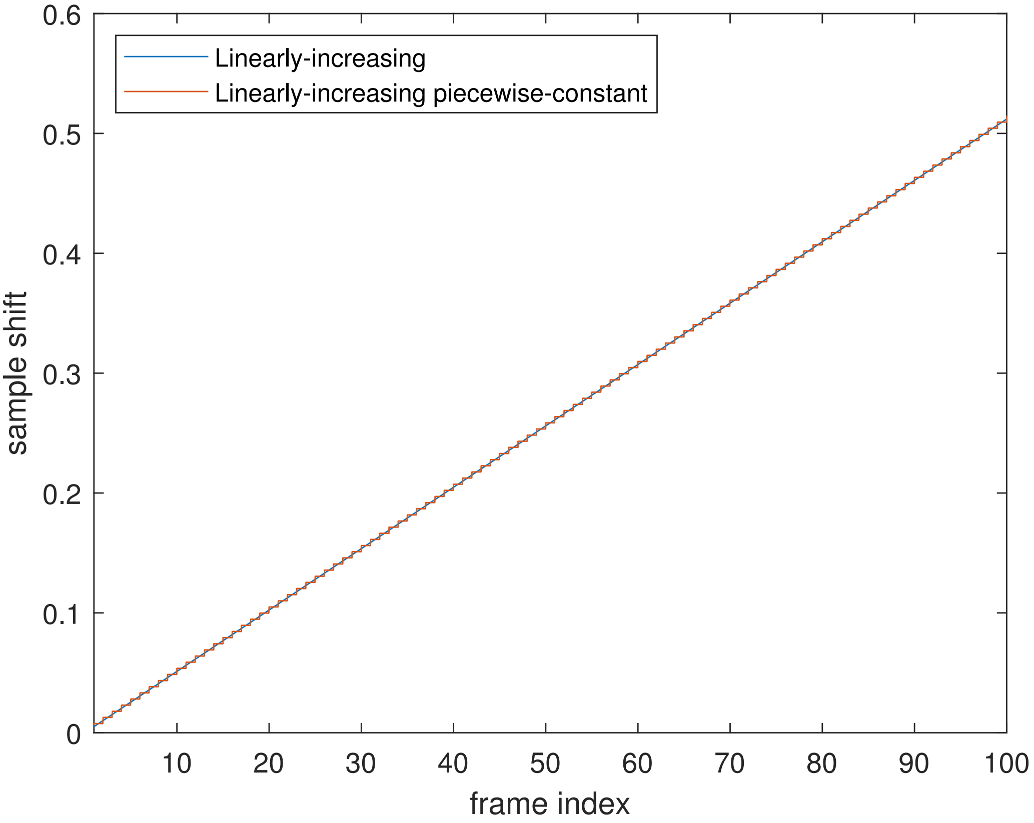


Figure 4.1: Plot of the linearly increasing phase shift experienced by each consecutive frame, versus the linearly increasing, piecewise constant phase shift described by the model. The phase shift is expressed in number of samples. In this example we have for the sample rate offset (SRO) of microphone two with respect to microphone one $\epsilon_2 = -10^{-5}$, frame length $K = 512$ and the frequency bin selected corresponds to $1\,\mathrm{kHz}$.

4.2 Vector form of received data

We use a vector notation and stack the STFT coefficients per frequency for all M microphones in a vector $\mathbf{y}$ as done in (2.7), where $\mathbf{y}$ would denote the data for a synchronized system. For the clock skew affected data we write $\tilde{\mathbf{y}}$. The relation between the clock skew affected data $\tilde{\mathbf{y}}$ and the synchronized data $\mathbf{y}$ is then given by:

$$\tilde{\mathbf{y}} = \mathbf{T}\mathbf{y} \tag{4.9}$$

where we have $\mathbf{T} = \mathrm{diag}\left(1, e^{-j2\pi k(\gamma l+1/2)\epsilon_2}, \ldots, e^{-j2\pi k(\gamma l+1/2)\epsilon_M}\right)$. Notice that $\mathbf{y}$ and $\mathbf{T}$ are given for each frequency bin k and frame index l. This expression is similar to the expression found for clock offset affected data, as found in [8], except now the matrix is varying with time frame l.

As mentioned in [8], $\mathbf{T}$ has the significant property that it is unitary, which will be exploited in the following sections:

$$\mathbf{T}\mathbf{T}^{\mathrm{H}} = \mathbf{T}^{\mathrm{H}}\mathbf{T} = \mathbf{I} \tag{4.10}$$

4.3 Generalized eigenvalue decomposition based beamforming

In this section, we will write out the equations necessary for the generalized eigenvalue decomposition (GEVD) based beamformers. We define the generalized eigenvalue problem as in [8], [23] and [24]. For the Hermitian matrix pencil $(\mathbf{R}_{\mathbf{X}}, \mathbf{R}_{\mathbf{V}})$ with $\mathbf{R}_{\mathbf{V}} \succ 0$ – referred to as a *Hermitian definite pencil* in the literature – the GEVD is given as:

$$\mathbf{U}^{\mathrm{H}}\mathbf{R}_{\mathbf{X}}\mathbf{U} = \mathbf{\Lambda} \qquad \mathbf{U}^{\mathrm{H}}\mathbf{R}_{\mathbf{V}}\mathbf{U} = \mathbf{I}_M \tag{4.11}$$

where we have $\mathbf{R}_{\mathbf{X}}, \mathbf{R}_{\mathbf{V}} \in \mathbb{C}^{M\times M}$, $\mathbf{U} \in \mathbb{C}^{M\times M}$ are the (right) generalized eigenvectors, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M) \in \mathbb{R}^{M\times M}$ the generalized eigenvalues and $\mathbf{I}_M$ is the identity matrix. The generalized eigenvalues λ_i are real because the matrix pencil $(\mathbf{R}_{\mathbf{X}}, \mathbf{R}_{\mathbf{V}})$ is Hermitian. Generally we also have $\mathbf{R}_{\mathbf{X}} \succeq 0$, then $\lambda_i \geq 0$ for all i.

The GEVD problem can be written as an EVD as follows:

$$\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}} = (\mathbf{U}^{-\mathrm{H}}\mathbf{U}^{-1})^{-1}\mathbf{U}^{-\mathrm{H}}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{U}^{\mathrm{H}}\mathbf{U}^{-\mathrm{H}}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \tag{4.12}$$

$$\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{U} = \mathbf{\Lambda}\mathbf{U} \tag{4.13}$$

where we used that $\mathbf{R}_{\mathbf{V}} \succ 0$, therefore $\mathbf{R}_{\mathbf{V}}^{-1}$ exists. Note that the condition $\mathbf{R}_{\mathbf{V}} \succ 0$ can be expected for common types of additive noise, but certainly not for matrices constructed of noise estimations (which is the case in a practical situation). However, due to estimation disturbances, we have $\mathrm{rank}(\mathbf{R}_{\mathbf{V}}) = M$, i.e. $\mathbf{R}_{\mathbf{V}}$ has full rank. This means that $\mathbf{R}_{\mathbf{V}}^{-1}$ exists nonetheless and (4.12) holds. From (4.13) we see that indeed $\mathbf{U}$ contains the right eigenvectors of $\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}$.

Since $\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}$ is not generally Hermitian, $\mathbf{U}$ is not necessarily unitary; $\mathbf{U}^{\mathrm{H}} \neq \mathbf{U}^{-1}$. Then the right eigenvectors $\mathbf{u}_i$ do constitute a basis for $\mathbb{C}^M$, albeit a non-orthogonal one. We can make use of the following property of $\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}$ however:

$$\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}} = \mathbf{R}_{\mathbf{V}}^{-1/2}(\mathbf{R}_{\mathbf{V}}^{-1/2}\mathbf{R}_{\mathbf{X}}\mathbf{R}_{\mathbf{V}}^{-1/2})\mathbf{R}_{\mathbf{V}}^{1/2} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \tag{4.14}$$

where $\mathbf{P} = \mathbf{R}_\mathbf{V}^{-1/2}$ is the unique Hermitian square-root of $\mathbf{R}_\mathbf{V}$ and $\mathbf{A} = \mathbf{R}_\mathbf{V}^{-1/2}\mathbf{R}_\mathbf{X}\mathbf{R}_\mathbf{V}^{-1/2}$ is a Hermitian positive semi-definite matrix. From (4.14) we see that $\mathbf{R}_\mathbf{V}^{-1}\mathbf{R}_\mathbf{X}$ is similar to a Hermitian matrix $\mathbf{A}$ and thus shares the same, real eigenvalues.

Defining $\mathbf{Q} = \mathbf{U}^{-\mathrm{H}}$, we can write (4.11) as:

$$\mathbf{R}_\mathbf{X} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathrm{H} \qquad \mathbf{R}_\mathbf{V} = \mathbf{Q}\mathbf{I}_M\mathbf{Q}^\mathrm{H} \tag{4.15}$$

where we have $\mathbf{Q} \in \mathbb{C}^{M\times M}$ as the (left) generalized eigenvectors of the matrix pencil $(\mathbf{R}_\mathbf{X}, \mathbf{R}_\mathbf{V})$. We can also rewrite the EVD form of the GEVD problem using (4.15):

$$\mathbf{R}_\mathbf{V}^{-1}\mathbf{R}_\mathbf{X} = (\mathbf{Q}\mathbf{I}_M\mathbf{Q}^\mathrm{H})^{-1}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathrm{H} = \mathbf{Q}^{-\mathrm{H}}\mathbf{I}_M\mathbf{Q}^{-1}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathrm{H} = \mathbf{Q}^{-\mathrm{H}}\boldsymbol{\Lambda}\mathbf{Q}^\mathrm{H} \tag{4.16}$$

$$\mathbf{Q}^\mathrm{H}\mathbf{R}_\mathbf{V}^{-1}\mathbf{R}_\mathbf{X} = \boldsymbol{\Lambda}\mathbf{Q}^\mathrm{H} \tag{4.17}$$

where we used again $\mathbf{R}_\mathbf{V} \succ 0$, so that $\mathbf{R}_\mathbf{V}^{-1}$ exists. From (4.17) we see that indeed $\mathbf{Q}$ contains the left eigenvectors of $\mathbf{R}_\mathbf{V}^{-1}\mathbf{R}_\mathbf{X}$. The left and right eigenvectors are related to each other as:

$$\mathbf{Q}^\mathrm{H}\mathbf{U} = \mathbf{U}^{-1}\mathbf{U} = \mathbf{I}_M \tag{4.18}$$

i.e. they are bi-orthogonal. This property will be used later on in the derivation of the GEVD based beamformers. If we add up the two equations in (4.11), we find:

$$\mathbf{U}^\mathrm{H}\mathbf{R}_\mathbf{X}\mathbf{U} + \mathbf{U}^\mathrm{H}\mathbf{R}_\mathbf{V}\mathbf{U} = \mathbf{U}^\mathrm{H}(\mathbf{R}_\mathbf{X} + \mathbf{R}_\mathbf{V})\mathbf{U} = \boldsymbol{\Lambda} + \mathbf{I}_M. \tag{4.19}$$

Substitution of (2.11) into (4.19) then gives:

$$\mathbf{U}^\mathrm{H}\mathbf{R}_\mathbf{Y}\mathbf{U} = \boldsymbol{\Lambda} + \mathbf{I}_M. \tag{4.20}$$

This means that if $(\lambda_i, \mathbf{u}_i)$ is a right generalized eigenpair of the pencil $(\mathbf{R}_\mathbf{X}, \mathbf{R}_\mathbf{V})$, then $(\lambda_i + 1, \mathbf{u}_i)$ is a right eigenpair of $(\mathbf{R}_\mathbf{Y}, \mathbf{R}_\mathbf{V})$. This result is important, because generally we do not have direct access to $\mathbf{R}_\mathbf{X}$. The generalized eigenvectors of both matrix pencils are similar however, and the generalized eigenvalues of the pair $(\mathbf{R}_\mathbf{X}, \mathbf{R}_\mathbf{V})$ can be obtained by subtracting 1 from the generalized eigenvalues of the pair $(\mathbf{R}_\mathbf{Y}, \mathbf{R}_\mathbf{V})$. Similarly, for the left eigenvectors we can write using (4.20) and $\mathbf{Q} = \mathbf{U}^{-\mathrm{H}}$:

$$\mathbf{R}_\mathbf{Y} = \mathbf{Q}(\boldsymbol{\Lambda} + \mathbf{I}_M)\mathbf{Q}^\mathrm{H}. \tag{4.21}$$

We shall now proceed to partition (4.21) into block matrices. To do this, we take note of the ranks of the correlation matrices. The rank R of the signal matrix $\mathbf{R}_\mathbf{X}$ is determined by the number of sources N_s. The EVD of $\mathbf{R}_\mathbf{X}$ therefore has only $R = N_s$ nonzero eigenvalues. The rank of the noise matrix $\mathbf{R}_\mathbf{V}$ is generally equal to the number of microphones M due to microphone self-noise, or estimation disturbances. Therefore the EVD of $\mathbf{R}_\mathbf{V}$ contains M nonzero eigenvalues. Recalling (4.21), the EVD of $\mathbf{R}_\mathbf{Y}$ can now be partitioned as follows:

$$\begin{aligned}\mathbf{R}_\mathbf{Y} &= \mathbf{Q}(\boldsymbol{\Lambda} + \mathbf{I}_M)\mathbf{Q}^\mathrm{H} \\ &= \begin{pmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{\Lambda}_R + \mathbf{I}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M-R}\end{pmatrix}\begin{pmatrix}\mathbf{Q}_1^\mathrm{H} \\ \mathbf{Q}_2^\mathrm{H}\end{pmatrix} \\ &= \mathbf{Q}_1(\boldsymbol{\Lambda}_R + \mathbf{I}_R)\mathbf{Q}_1^\mathrm{H} + \mathbf{Q}_2\mathbf{I}_{M-R}\mathbf{Q}_2^\mathrm{H}\end{aligned} \tag{4.22}$$

where $\mathbf{Q}_1 \in \mathbb{C}^{M\times R}$ contains the first R eigenvectors of $\mathbf{R}_\mathbf{Y}$ as its columns, $\mathbf{Q}_2 \in \mathbb{C}^{M\times(M-R)}$ contains the last $M-R$ eigenvectors of $\mathbf{R}_\mathbf{Y}$ as its columns, $\mathbf{\Lambda}_R = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_R) \in \mathbb{R}^R$ with $\lambda_i \leq 0$ are the R eigenvalues belonging to $\mathbf{R}_\mathbf{X}$ and $\mathbf{I}_R$ and $\mathbf{I}_{M-R}$ are identity matrices.

From (4.22) we can conclude the following: $\mathbf{Q}_1(\mathbf{\Lambda}_R + \mathbf{I}_R)\mathbf{Q}_1^\text{H}$ spans a speech + noise subspace, because all the information of the speech signal is contained in the first R eigenvalues/eigenvectors of $\mathbf{R}_\mathbf{Y}$. Moreover, $\mathbf{Q}_2\mathbf{I}_{M-R}\mathbf{Q}_2^\text{H}$ spans a noise-only subspace. From the property $\mathbf{Q}\mathbf{U}^\text{H} = \mathbf{I}_M$ we have that $\mathbf{Q}_1\mathbf{U}_2^\text{H} = \mathbf{0}_{R\times(M-R)}$, i.e. the speech + noise subspace and noise-only subspace are orthogonal.

As mentioned in Chapter 3 a beamformer is generally tasked with estimating a target source. From the discussion in the previous paragraph we may conclude that it is based on a linear combination of the column vectors in the speech subspace, i.e. $\mathbf{w} = \mathbf{U_1 a}$. With this notion in mind, optimal GEVD based beamformers can be constructed as in [8]:

$$\mathbf{w} = (\mathbf{R}_\mathbf{X} + \mu\mathbf{R}_\mathbf{V})^{-1}\mathbf{R}_\mathbf{X}\mathbf{e}_1 = \sum_{i=1}^{M} \frac{\mathbf{u}_i\mathbf{u}_i^\text{H}}{\lambda_i + \mu}\mathbf{R}_\mathbf{X}\mathbf{e}_1 \tag{4.23}$$

where $\mathbf{u}_i$ is the i-th column of $\mathbf{U}$, $\mathbf{e}_1$ is the standard basis vector defined as $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^M$ and λ_i is the i-th eigenvalue. The result found in (4.23) is the classic SDW Wiener filter [25] [2]. The parameter μ is a tradeoff parameter between noise reduction and speech distortion. For different settings of this parameter, we find different well-known beamformers.

Another group of beamformers can be obtained by restricting (4.23) to use only the first P eigenvectors, that is:

$$\mathbf{w}(P) = \sum_{i=1}^{P} \frac{\mathbf{u}_i\mathbf{u}_i^\text{H}}{\lambda_i + \mu}\mathbf{R}_\mathbf{X}\mathbf{e}_1 \tag{4.24}$$

where P is the number of eigenvectors used in the beamformer. These beamformers are known as variable span beamformers. For example, the MVDR beamformer may be obtained by setting $P = R = \text{rank}(\mathbf{R}_\mathbf{X})$. With this expression, the beamformers are constructed indeed using only the column vectors in the speech subspace.

4.4 Generalized eigenvalue decomposition based beamforming for the signal model with clock skew

We can write out the spatial correlation matrices for the system affected by clock skew using (4.9):

$$\tilde{\mathbf{R}}_\mathbf{Y} = \mathbb{E}[\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\text{H}] = \mathbf{T}\mathbb{E}[\mathbf{Y}\mathbf{Y}^\text{H}]\mathbf{T}^\text{H} = \mathbf{T}\mathbf{R}_\mathbf{Y}\mathbf{T}^\text{H}. \tag{4.25}$$

Recalling (2.11) and using (4.25), we can write:

$$\tilde{\mathbf{R}}_\mathbf{X} = \mathbb{E}[\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\text{H}] = \mathbf{T}\mathbb{E}[\mathbf{X}\mathbf{X}^\text{H}]\mathbf{T}^\text{H} = \mathbf{T}\mathbf{R}_\mathbf{X}\mathbf{T}^\text{H} \tag{4.26}$$

$$\tilde{\mathbf{R}}_\mathbf{V} = \mathbb{E}[\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\text{H}] = \mathbf{T}\mathbb{E}[\mathbf{V}\mathbf{V}^\text{H}]\mathbf{T}^\text{H} = \mathbf{T}\mathbf{R}_\mathbf{V}\mathbf{T}^\text{H}. \tag{4.27}$$

The expressions found in (4.26) and (4.27) can be used to write:

$$\begin{aligned}\tilde{\mathbf{R}}_{\mathbf{V}}^{-1}\tilde{\mathbf{R}}_{\mathbf{X}} &= (\mathbf{T}\mathbf{R}_{\mathbf{V}}\mathbf{T}^{\mathrm{H}})^{-1}\mathbf{T}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}} \\ &= \mathbf{T}^{-\mathrm{H}}\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{T}^{-1}\mathbf{T}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}} \\ &= \mathbf{T}^{-\mathrm{H}}\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}} \\ &= \mathbf{T}\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}}. \qquad (4.28)\end{aligned}$$

Substituting (4.12) in (4.28) we find:

$$\begin{aligned}\tilde{\mathbf{R}}_{\mathbf{V}}^{-1}\tilde{\mathbf{R}}_{\mathbf{X}} &= \mathbf{T}\mathbf{R}_{\mathbf{V}}^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}} \qquad (4.29) \\ &= \mathbf{T}\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}\mathbf{T}^{\mathrm{H}} \qquad (4.30) \\ &= (\mathbf{T}\mathbf{U})\boldsymbol{\Lambda}(\mathbf{T}\mathbf{U})^{-1} \qquad (4.31)\end{aligned}$$

the eigenvalues of the skewed problem remain the same, whereas the new right eigenvectors are given by $\tilde{\mathbf{U}} = \mathbf{T}\mathbf{U}$. Substituting this result in (4.23), we find:

$$\begin{aligned}\tilde{\mathbf{w}} &= (\tilde{\mathbf{R}}_{\mathbf{X}} + \mu\tilde{\mathbf{R}}_{\mathbf{V}})^{-1}\tilde{\mathbf{R}}_{\mathbf{X}}\mathbf{e}_1 \\ &= (\mathbf{T}(\mathbf{R}_{\mathbf{X}} + \mu\mathbf{R}_{\mathbf{V}})\mathbf{T}^{\mathrm{H}})^{-1}\mathbf{T}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}}\mathbf{e}_1 \\ &= \mathbf{T}^{-\mathrm{H}}(\mathbf{R}_{\mathbf{X}} + \mu\mathbf{R}_{\mathbf{V}})^{-1}\mathbf{T}^{-1}\mathbf{T}\mathbf{R}_{\mathbf{X}}\mathbf{T}^{\mathrm{H}}\mathbf{e}_1 \\ &= \mathbf{T}(\mathbf{R}_{\mathbf{X}} + \mu\mathbf{R}_{\mathbf{V}})^{-1}\mathbf{R}_{\mathbf{X}}\mathbf{e}_1 \qquad (4.32) \\ &= \mathbf{T}\mathbf{w} \qquad (4.33)\end{aligned}$$

where we used $\mathbf{T}^{\mathrm{H}}\mathbf{e}_1 = \mathbf{e}_1$ in (4.32). For the beamformer output we find:

$$\begin{aligned}\hat{s} &= \tilde{\mathbf{w}}^{\mathrm{H}}\tilde{\mathbf{y}} \\ &= (\mathbf{T}\mathbf{w})^{\mathrm{H}}\mathbf{T}\mathbf{y} \\ &= \mathbf{w}^{\mathrm{H}}\mathbf{T}^{\mathrm{H}}\mathbf{T}\mathbf{y} \\ &= \mathbf{w}^{\mathrm{H}}\mathbf{y}. \qquad (4.34)\end{aligned}$$

Thus we conclude that the clock skew affected (GEVD based) beamformer gives the same beamformer output as the original beamformer and is therefore invariant to the clock skew presented here. This is expected, as the model is constructed to approximate clock skew as a clock offset that increases linearly per STFT frame.

4.5 Low-rank approximation of $\mathbf{R_X}$

In the beamformer expression (4.23) the correlation matrix $\mathbf{R_X}$ is used. Usually we have access to $\mathbf{R_Y}$ from the microphones and $\mathbf{R_V}$ from a noise PSD estimation framework, or an observation of $\mathbf{R_Y}$ during a 'quiet' period. Naturally, we have:

$$\mathbf{R_X} = \mathbf{R_Y} - \mathbf{R_V} \tag{4.35}$$

and $\mathbf{R_X}$ can be obtained perfectly. However, due to estimation errors in $\mathbf{R_Y}$ and $\mathbf{R_V}$ a perfect estimate of $\mathbf{R_X}$ can often not be found through (4.35), i.e. for the estimation we may find $\text{rank}(\hat{\mathbf{R}}_\mathbf{X}) \neq R = N_s$. In that case a low-rank approximation of $\mathbf{R_X}$ is desirable. Recalling (4.22), a low-rank approximation of $\mathbf{R_X}$ can be found by subtracting 1 from the eigenvalues of $\mathbf{R_Y}$ and only using the first R eigenvectors, that is:

$$\hat{\mathbf{R}}_\mathbf{X} = \mathbf{Q_1}\mathbf{\Lambda_R}\mathbf{Q_1}^\mathrm{H} = \sum_{r=1}^{R} \lambda_r \mathbf{q}_r \mathbf{q}_r^\mathrm{H} \tag{4.36}$$

where $\mathbf{q}_r$ is the r-th column of $\mathbf{Q_1}$ and λ_r is the r-th eigenvalue. In this way we force $\text{rank}(\mathbf{R_X}) = R = N_s$.

Correlation matrix estimation 5

As we have seen in the previous section, the effects of clock skew can be compensated automatically when using the form (4.33). Given that we know the true correlation matrices, this compensation is exact, up to the approximation of the linearly-increasing phase difference per time sample by a linearly-increasing phase difference per time frame. However, the beamformers depend on the correlation matrices $\mathbf{R}_\mathbf{V}$ and $\mathbf{R}_\mathbf{Y}$, which denote the noise and received signal correlation matrices, respectively. In practice these are unknown and have to be estimated. Typically this is done by time averaging the received data. However, due to the sampling rate mismatch these estimates become biased and an error is introduced. In this section we will investigate the effect of correlation matrix estimation on the algorithm presented in the previous chapter.

5.1 Welch method

One common way to estimate the correlation matrix $\mathbf{R}_\mathbf{Y}$ is by employing the Welch method [26], that is:

$$\hat{\mathbf{R}}_\mathbf{Y} = \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \mathbf{y}(l)\mathbf{y}^\mathrm{H}(l) \tag{5.1}$$

where we average the single-frame estimate of $\mathbf{R}_\mathbf{Y}$ over the most recent L STFT frames, up to STFT frame $\mathcal{L}-1 \geq L$. Here $\hat{\mathbf{R}}_\mathbf{Y}$ denotes the Welch correlation matrix estimate without clock skew. Let $\mathcal{R}_\mathbf{Y}$ denote the estimate for the skewed data, recalling (4.9) we then have:

$$\begin{aligned}\mathcal{R}_\mathbf{Y} &= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \tilde{\mathbf{y}}(l)\tilde{\mathbf{y}}^\mathrm{H}(l) \\ &= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \mathbf{T}(l)\tilde{\mathbf{y}}(l)(\mathbf{T}(l)\tilde{\mathbf{y}}(l))^\mathrm{H} \\ &= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \mathbf{T}(l)\mathbf{y}(l)\mathbf{y}^\mathrm{H}(l)\mathbf{T}^\mathrm{H}(l). \end{aligned}\tag{5.2}$$

For each estimation frame l, the phase offsets for the skewed microphones increase linearly, as per the approximation in the previous section. If summation (averaging) is applied to these increasingly offset frames, the estimated phase offset for each frame becomes blurred. This can be illustrated by further examination of (5.2).

The expression in (5.2) can be expanded by writing out $\mathbf{T}$. Denote $\tau_i(l) = e^{-j2\pi k(\gamma l+1/2)\epsilon_i}$, we can now write:

$$
\begin{aligned}
\mathcal{R}_{\mathbf{Y}} &= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \mathbf{T}(l)\mathbf{y}(l)\mathbf{y}^{\mathrm{H}}(l)\mathbf{T}^{\mathrm{H}}(l) \\
&= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \tau_2(l) & 0 & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & \tau_M(l) \end{pmatrix} \begin{pmatrix} Y_1(l) \\ Y_2(l) \\ \vdots \\ Y_M(l) \end{pmatrix} \begin{pmatrix} Y_1(l) \\ Y_2(l) \\ \vdots \\ Y_M(l) \end{pmatrix}^{\mathrm{H}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \tau_2^*(l) & 0 & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & \tau_M^*(l) \end{pmatrix} \\
&= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \begin{pmatrix} Y_1(l) \\ Y_2(l)\tau_2(l) \\ \vdots \\ Y_M(l)\tau_M(l) \end{pmatrix} \begin{pmatrix} Y_1^*(l) & Y_2^*(l)\tau_2^*(l) & \dots & Y_M^*(l)\tau_M^*(l) \end{pmatrix} \\
&= \frac{1}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \begin{pmatrix} Y_1(l)Y_1^*(l) & Y_1(l)Y_2^*(l)\tau_2^*(l) & \dots & Y_1(l)Y_M^*(l)\tau_M^*(l) \\ Y_2(l)Y_1^*(l)\tau_2(l) & Y_2(l)Y_2^*(l) & \dots & Y_2(l)Y_M^*(l)\tau_M^*(l) \\ \vdots & \vdots & \ddots & \vdots \\ Y_M(l)Y_1^*(l)\tau_M(l) & Y_M(l)Y_2^*(l)\tau_M(l)\tau_2^*(l) & \dots & Y_M(l)Y_M^*(l) \end{pmatrix}. \qquad (5.3)
\end{aligned}
$$

The off-diagonal terms in (5.3) give information on the correlation between different microphones. Indeed, we observe the presence of the exponential phase shift terms $\tau_i(l)$. At this point, we can establish that a phase term estimation error will accumulate due to the summation, i.e. the estimator $\mathcal{R}_{\mathbf{Y}}$ is biased.

5.1.1 Effect of sample rate offset on estimate using Welch method

The reason for using a method like Welch, is to reduce the variance in the estimate for $\mathcal{R}_{\mathbf{Y}}$. We have established however, that a phase term estimation error is introduced by the SRO. In this section we quantify the error made in $\mathcal{R}_{\mathbf{Y}}$ due to the SRO. To this end, we study a noiseless scenario. This scenario allows us to focus on the SRO related error only. From (2.11) we have that if $\mathbf{R}_{\mathbf{V}} = \mathbf{0}$ then $\mathbf{R}_{\mathbf{Y}} = \mathbf{R}_{\mathbf{X}}$. Therefore, we study the effect caused by a SRO on $\mathbf{R}_{\mathbf{X}}$ equivalently.

Denote the Welch estimate of $\mathbf{R}_{\mathbf{X}}$ based on skewed data as $\mathcal{R}_{\mathbf{X}}$. The effect of the SRO on $\mathcal{R}_{\mathbf{X}}$ can be quantified by inserting (4.9) into (5.3). Assume that we have a single wide-sense stationary (WSS) target source signal over the estimation period for $\mathcal{R}_{\mathbf{X}}$. Then the PSD of the target source process is constant, with variance $\mathbb{E}|\mathcal{X}_1|^2 = \sigma^2_{\mathcal{X}_1}$, where $\mathcal{X}_1$ denotes the stochastic process that realizes X_1. Now (5.3) can be written as:

$$
\begin{aligned}
\mathbb{E}[\mathcal{R}_{\mathbf{X}}] &= \frac{\sigma^2_{\mathcal{X}_1}}{L}\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \begin{pmatrix} 1 & d_2^*\tau_2^*(l) & \dots & d_M^*\tau_M^*(l) \\ d_2\tau_2(l) & d_2d_2^* & \dots & d_2d_M^*\tau_2(l)\tau_M^*(l) \\ \vdots & \vdots & \ddots & \vdots \\ d_M\tau_M(l) & d_Md_2^*\tau_M(l)\tau_2^*(l) & \dots & d_Md_M^* \end{pmatrix} \\
&= \frac{\sigma^2_{\mathcal{X}_1}}{L} \begin{pmatrix} L & d_2^*\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_2^*(l) & \dots & d_M^*\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_M^*(l) \\ d_2\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_2(l) & Ld_2d_2^* & \dots & d_2d_M^*\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_2(l)\tau_M^*(l) \\ \vdots & \vdots & \ddots & \vdots \\ d_M\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_M(l) & d_Md_2^*\sum_{l=\mathcal{L}-L}^{\mathcal{L}-1}\tau_M(l)\tau_2^*(l) & \dots & Ld_Md_M^* \end{pmatrix}. \qquad (5.4)
\end{aligned}
$$

On the last line of (5.4) we recognize the relative steering vector $\hat{\mathbf{d}}_s$ (based on the skewed data) in the first column of the matrix, recalling section 2.2. $\hat{\mathbf{d}}_s$ can be obtained by a normalization with respect to $\mathcal{R}_{\mathbf{X}[1,1]}$: $\hat{\mathbf{d}}_s = \mathcal{R}_{\mathbf{X}[:,1]}/\sigma^2_{\mathcal{X}_1}$. This justifies to study the effect of the additional (skew originating) phase terms on $\mathcal{R}_{\mathbf{X}[:,1]}$, because it directly reveals the skewed steering vector $\hat{\mathbf{d}}_s$, which is involved in most beamformer expressions. We thus conclude that the relevant information is located in the first column of $\mathcal{R}_{\mathbf{X}}$. Let ∇_i denote the i-th entry of the first column of $\mathcal{R}_{\mathbf{X}}$:

$$\nabla_i = \mathcal{R}_{\mathbf{X}[i,1]}. \tag{5.5}$$

Below, starting with (5.4) an entry-wise closed form expression for the first column of $\mathcal{R}_{\mathbf{X}}$ is derived:

$$\begin{aligned}\mathbb{E}[\nabla_i] &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i \sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} \tau_i(l) = \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i \sum_{l=\mathcal{L}-L}^{\mathcal{L}-1} e^{-j2\pi k(\gamma l + 1/2)\epsilon_i} \\ &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i \sum_{l=0}^{L-1} e^{-j2\pi k(\gamma(l+\mathcal{L}-L)+1/2)\epsilon_i}.\end{aligned} \tag{5.6}$$

We recognize the summation term in (5.6) as a geometric series with $a = e^{-j\pi k(1+2\gamma(\mathcal{L}-L))\epsilon_i}$ and $r = e^{-j2\pi k\gamma\epsilon_i}$:

$$\begin{aligned}\mathbb{E}[\nabla_i] &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i \sum_{l=0}^{L-1} ar^l = \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i a \frac{1-r^L}{1-r} \\ &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k(1+2\gamma(\mathcal{L}-L))\epsilon_i} \frac{1-e^{-j2\pi k\gamma L\epsilon_i}}{1-e^{-j2\pi k\gamma\epsilon_i}} \\ &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k(1+2\gamma(\mathcal{L}-L))\epsilon_i} \frac{e^{-j\pi k\gamma L\epsilon_i}}{e^{-j\pi k\gamma\epsilon_i}} \frac{e^{j\pi k\gamma L\epsilon_i} - e^{-j\pi k\gamma L\epsilon_i}}{e^{j\pi k\gamma\epsilon_i} - e^{-j\pi k\gamma\epsilon_i}} \\ &= \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k(1+2\gamma(\mathcal{L}-L))\epsilon_i} e^{-j\pi k\gamma(L-1)\epsilon_i} \frac{\sin(\pi k\gamma L\epsilon_i)}{\sin(\pi k\gamma\epsilon_i)}\end{aligned} \tag{5.7}$$

where we have the condition $r \neq 1$, which generally holds given a finite SRO, and $k \neq 0$. Note that the Dirichlet kernel [27] (Chapter 10) appears in this expression, so the estimated phase offset shows periodic behavior as a function of L or ϵ_i. In this thesis, we will make use of 50 % STFT frame overlap, i.e. $K_h = K/2 \rightarrow \gamma = 1/2$. Then the expression in (5.7) becomes:

$$\mathbb{E}[\nabla_i] = \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k\mathcal{L}\epsilon_i} e^{j\pi k(L-1)\epsilon_i/2} \frac{\sin(\pi kL\epsilon_i/2)}{\sin(\pi k\epsilon_i/2)}. \tag{5.8}$$

This expression is useful to evaluate the beamforming error caused by the SRO. In the ideal case, $\mathcal{R}_{\mathbf{X}}$ would be estimated based on the current STFT frame only ($L = 1$), because then there is no phase offset accumulation. In that case, the correct phase for each frame is included in $\hat{\mathbf{d}}_s$ and we have perfect SRO compensation (assuming the piece-wise constant approximation derived in section 4.1 is accurate).

Using $L = 1$ in (5.8) we find:

$$\mathbb{E}[\nabla_i]\Big|_{L=1} = \frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k\mathcal{L}\epsilon_i}. \tag{5.9}$$

We can devise an error measure which compares the averaging estimate found in (5.8) with the estimate found using the current STFT frame only in (5.9). A proper error measure would take the expected value of the square of the difference between (5.8) and (5.9), that is, the mean squared error (MSE), as follows:

$$\begin{aligned} \mathrm{C}_{\mathrm{W},i} &= \mathbb{E}\left[\left|\nabla_i - \nabla_i\Big|_{L=1}\right|^2\right] && (5.10)\\ &\geq \left|\mathbb{E}\left[\nabla_i - \nabla_i\Big|_{L=1}\right]\right|^2 && (5.11)\\ &= \left|\mathbb{E}\left[\nabla_i\right] - \mathbb{E}[\nabla_i]\Big|_{L=1}\right|^2 && (5.12)\end{aligned}$$

where (5.11) follows from Jensen's inequality [28]. For Jensen's inequality to hold, we need the MSE function $g(x) = |x|^2$ to be strictly convex, which is indeed true. We proceed with substitution of (5.8) and (5.9) into (5.12):

$$\begin{aligned} \mathrm{C}_{\mathrm{W},i} &\geq \left|\frac{\sigma^2_{\mathcal{X}_1}}{L} d_i e^{-j\pi k \mathcal{L}\epsilon_i} e^{j\pi k(L-1)\epsilon_i/2} \frac{\sin(\pi k L \epsilon_i/2)}{\sin(\pi k \epsilon_i/2)} - \sigma^2_{\mathcal{X}_1} d_i e^{-j\pi k \mathcal{L}\epsilon_i}\right|^2 \\ &= \left|\sigma^2_{\mathcal{X}_1} d_i \left(\frac{e^{j\pi k(L-1)\epsilon_i/2}}{L} \frac{\sin(\pi k L \epsilon_i/2)}{\sin(\pi k \epsilon_i/2)} - 1\right)\right|^2. && (5.13)\end{aligned}$$

We conclude that using the Welch method, the larger the value of L or ϵ_i, the larger the error in $\mathcal{R}_{\mathbf{X}[i,1]}$ introduced by a SRO.

5.2 Recursive smoothing method

In the previous section, it was shown that estimating $\mathcal{R}_{\mathbf{Y}}$ using the Welch method in the presence of clock skew, is subject to an error due to averaging of frames. However, estimating $\mathcal{R}_{\mathbf{Y}}$ using only the current frame data leads to a high variance. The averaging of frames is therefore necessary in estimating the correlation matrices. However, there exist methods which can emphasize either the previous time frames or the current time frames. In [29] a variable forgetting factor recursive smoothing method is used to estimate the spectrum of a non-stationary signal. In [30] and [31] a recursive smoothing method is used in the estimation/tracking of the noise PSD. Such methods can be used instead of the Welch method in the estimation of $\mathcal{R}_{\mathbf{Y}}$. We shall proceed to define such a recursive smoothing method.

We define the recursive smoothing with a forgetting factor α as:

$$\mathcal{R}_{\mathbf{Y}}(l) = \alpha \mathcal{R}_{\mathbf{Y}}(l-1) + (1-\alpha)\tilde{\mathbf{y}}(l)\tilde{\mathbf{y}}^{\mathrm{H}}(l) \tag{5.14}$$

where we have $0 \leq \alpha < 1$. This equation can be written explicitly, for recursive smoothing up to STFT frame $\mathcal{L}-1$. We can write (5.14) as:

$$\mathcal{R}_{\mathbf{Y}} = (1-\alpha)\sum_{l=0}^{\mathcal{L}-1} \alpha^{\mathcal{L}-1-l}\tilde{\mathbf{y}}(l)\tilde{\mathbf{y}}^{\mathrm{H}}(l). \tag{5.15}$$

This expression is useful for deriving the error introduced by a SRO, as derived in the previous section for the Welch method. Note that recursive smoothing as defined here uses all $\mathcal{L}$ available STFT frames at current frame $\mathcal{L}-1$ with appropriate weights applied to them. In contrast, the Welch method uses the L previous, equally weighted STFT frames, which may be less frames than available at current frame $\mathcal{L}-1$.

5.2.1 Effect of sample rate offset on estimate using recursive smoothing method

The effect of the SRO on $\mathcal{R}_{\mathbf{Y}}$ can be evaluated similar to the previous section. Again we assume a noiseless scenario, thus we study $\mathcal{R}_{\mathbf{X}}$. Assume again that we have a single WSS target source signal over the estimation period for $\mathcal{R}_{\mathbf{X}}$. Then the PSD of the target source process is constant, with variance $\mathbb{E}|\mathcal{X}_1|^2 = \sigma^2_{\mathcal{X}_1}$, where $\mathcal{X}_1$ denotes the stochastic process that realizes X_1. We look at entry $[i, 1]$ of $\mathcal{R}_{\mathbf{X}}$ in (5.15), because the relevant information is located in the first column of $\mathcal{R}_{\mathbf{X}}$ as discussed before. Then, recalling (4.9) we find:

$$\begin{aligned}
\mathbb{E}[\nabla_i] &= \sigma^2_{\mathcal{X}_1} d_i (1-\alpha) e^{-j\pi k\epsilon_i} \sum_{l=0}^{\mathcal{L}-1} \alpha^{\mathcal{L}-1-l} e^{-j2\pi k\gamma l\epsilon_i} \\
&= \sigma^2_{\mathcal{X}_1} d_i (1-\alpha) \alpha^{\mathcal{L}-1} e^{-j\pi k\epsilon_i} \sum_{l=0}^{\mathcal{L}-1} \left(\frac{e^{-j2\pi k\gamma\epsilon_i}}{\alpha} \right)^l \\
&= \sigma^2_{\mathcal{X}_1} d_i (1-\alpha) \alpha^{\mathcal{L}-1} e^{-j\pi k\epsilon_i} \frac{1 - \left(\frac{e^{-j2\pi k\gamma\epsilon_i}}{\alpha} \right)^{\mathcal{L}}}{1 - \left(\frac{e^{-j2\pi k\gamma\epsilon_i}}{\alpha} \right)} \\
&= \sigma^2_{\mathcal{X}_1} d_i (1-\alpha) e^{-j\pi k\epsilon_i} \frac{\alpha^{\mathcal{L}} - e^{-j2\pi k\gamma\mathcal{L}\epsilon_i}}{\alpha - e^{-j2\pi k\gamma\epsilon_i}}. \qquad (5.16)
\end{aligned}$$

We can generally assume the system is in a running state, which means $\mathcal{L} \gg 1$. Then (5.16) can be reduced to:

$$\mathbb{E}[\nabla_i] = \sigma^2_{\mathcal{X}_1} d_i (\alpha-1) e^{-j\pi k\epsilon_i} \frac{e^{-j2\pi k\gamma\mathcal{L}\epsilon_i}}{\alpha - e^{-j2\pi k\gamma\epsilon_i}}. \qquad (5.17)$$

Using again $K_h = K/2 \rightarrow \gamma = 1/2$, we find:

$$\mathbb{E}[\nabla_i] = \sigma^2_{\mathcal{X}_1} d_i (\alpha-1) \frac{e^{-j\pi k(\mathcal{L}+1)\epsilon_i}}{\alpha - e^{-j\pi k\epsilon_i}}. \qquad (5.18)$$

We define the error similar to (5.12), by taking the MSE between (5.18) and (5.9) as follows:

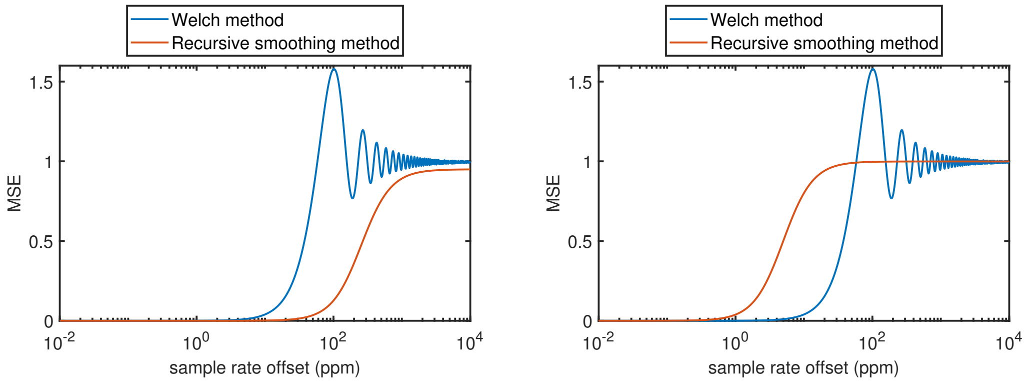
$$\mathrm{C}_{\mathrm{R},i} \geq \left| \mathbb{E}\left[\nabla_i\right] - \mathbb{E}[\nabla_i] \Big|_{L=1} \right|^2 . \qquad (5.19)$$

Substitution of (5.18) and (5.9) into (5.19) then gives:

$$\begin{aligned}
\mathrm{C}_{\mathrm{R},i} &\geq \left| \sigma^2_{\mathcal{X}_1} d_i (\alpha-1) \frac{e^{-j\pi k(\mathcal{L}+1)\epsilon_i}}{\alpha - e^{-j\pi k\epsilon_i}} - \sigma^2_{\mathcal{X}_1} d_i e^{-j\pi k\mathcal{L}\epsilon_i} \right|^2 \\
&= \left| \sigma^2_{\mathcal{X}_1} d_i \left(\frac{\alpha-1}{\alpha e^{j\pi k\epsilon_i} - 1} - 1 \right) \right|^2 . \qquad (5.20)
\end{aligned}$$

We conclude that using the recursive smoothing method, the larger the value of α or ϵ_i, the larger the error in $\mathcal{R}_{\mathbf{X}[i,1]}$ introduced by a SRO.

In Figure 5.1 the error measures (5.13) and (5.20) are plotted vs. the sample rate offset. For the settings in Figure 5.1a the recursive smoothing method performs better than the Welch method. The “ringing” of the Welch method error originating from the Dirichlet kernel appearing in (5.8) can also be seen.



(a) For the Welch method $L = 200$, for the recursive smoothing method $\alpha = 0.95$.

(b) For the Welch method $L = 200$, for the recursive smoothing method $\alpha = 0.9990$.

Figure 5.1: Mean squared error (MSE) vs. sample rate offset (SRO) for different correlation matrix estimators. Welch method and recursive smoothing method error measures as defined in (5.13) and (5.20) respectively. The SRO is between microphone i and microphone 1. A direct transfer is assumed, i.e. $|d_i| = 1$. The frequency bin selected corresponds to 2 kHz.

Simulation of Wireless Acoustic Sensor Network 6

In Chapter 4 we concluded that in theory the GEVD based beamformers are invariant to clock skew, up to the linearly-increasing, piecewise-constant approximation used for the phase shift caused by a SRO. In Chapter 5 we have seen that a SRO introduces an error in the estimated correlation matrices, which increases with the SRO and the estimation length. In this chapter we will investigate the performance of the GEVD based clock skew invariant beamformers using either Welch or recursive smoothing as correlation matrix estimation methods. We shall denote the combination of these GEVD based beamformers and correlation matrix estimation methods as the *proposed algorithms* in the following discussion. The performance of the proposed algorithms is compared with a *blind synchronization* algorithm by Bahari et al. [1], we shall denote this algorithm as the *reference algorithm* in the following discussion. Note that the reference algorithm uses the Welch method for correlation matrix estimation.

The algorithms are compared using a simulated WASN setup, to be detailed in the next sections. The parameters for the simulation are then defined and the results are presented. At the end of this chapter a conclusion is made using the obtained results and recommendations are made on future work.

6.1 Wireless Acoustic Sensor Network setup

The setup used in the experiments is depicted in Figure 6.1. The network consists of nodes, each comprised of a microphone and a wireless communication mechanism. One of the nodes, the fusion node, collects data from the other, slave nodes. The fusion node is also tasked with performing computations on the received data; it receives buffers of sampled time domain data from the slave nodes. These buffers are converted to the frequency domain through the STFT transform as defined in (2.5). Then, the fusion node calculates the beamformer and applies it to the received data.

6.1.1 Buffering scheme

The buffering scheme inherent to the WASN setup provides a periodic time domain synchronization. For the proposed algorithm, the data for each node is stored in a buffer of length Q, and the fusion node collects these buffers when its buffer is full. Assume now without loss of generality that the fusion node is the fastest sampling node, so we have for the SRO $\epsilon_i \leq 0$ for $i \neq 1$. Due to this condition, the buffers from the slave nodes contain less samples than the buffer from the fusion node at the time when the buffers are taken. Therefore, automatically zero-padding is applied to the slave nodes buffers, which leads to time domain synchronization (up to one sample precision) every Q samples.

For the reference algorithm, the buffering length is dependent on the estimated SRO value. The algorithm takes a buffer when the signals from the fusion node and a slave node have drifted apart further than 1 sample. This happens each $1/|\epsilon_i|$ samples, where ϵ_i is the SRO of microphone i. At that time instant, a zero is padded to the buffer of a slave node. This means time domain synchronization (up to one sample precision) happens each $1/|\epsilon_i|$ samples.

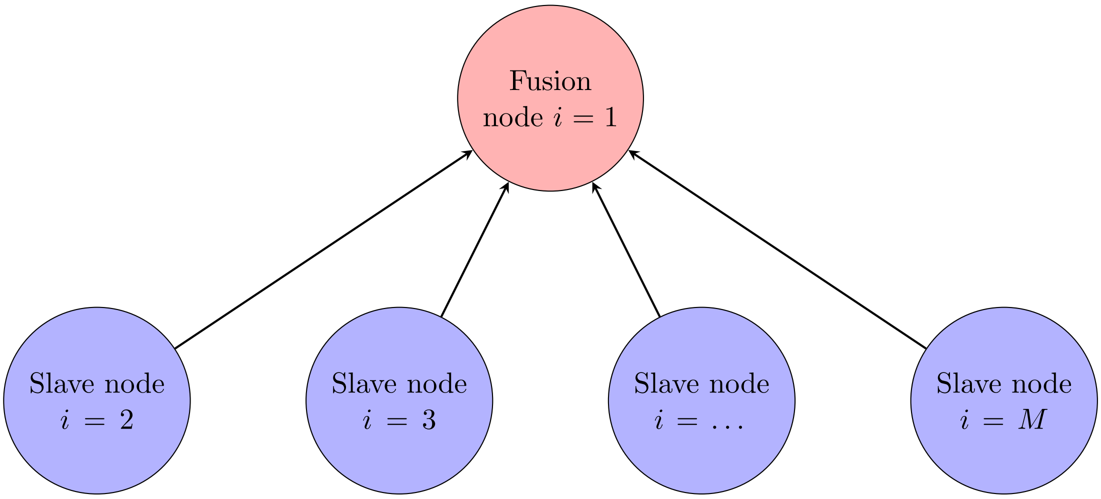


Figure 6.1: Wireless acoustic sensor network setup as used in the experiments. All nodes have microphones with indices as depicted in this figure; $i = 1, 2, \dots, M$.

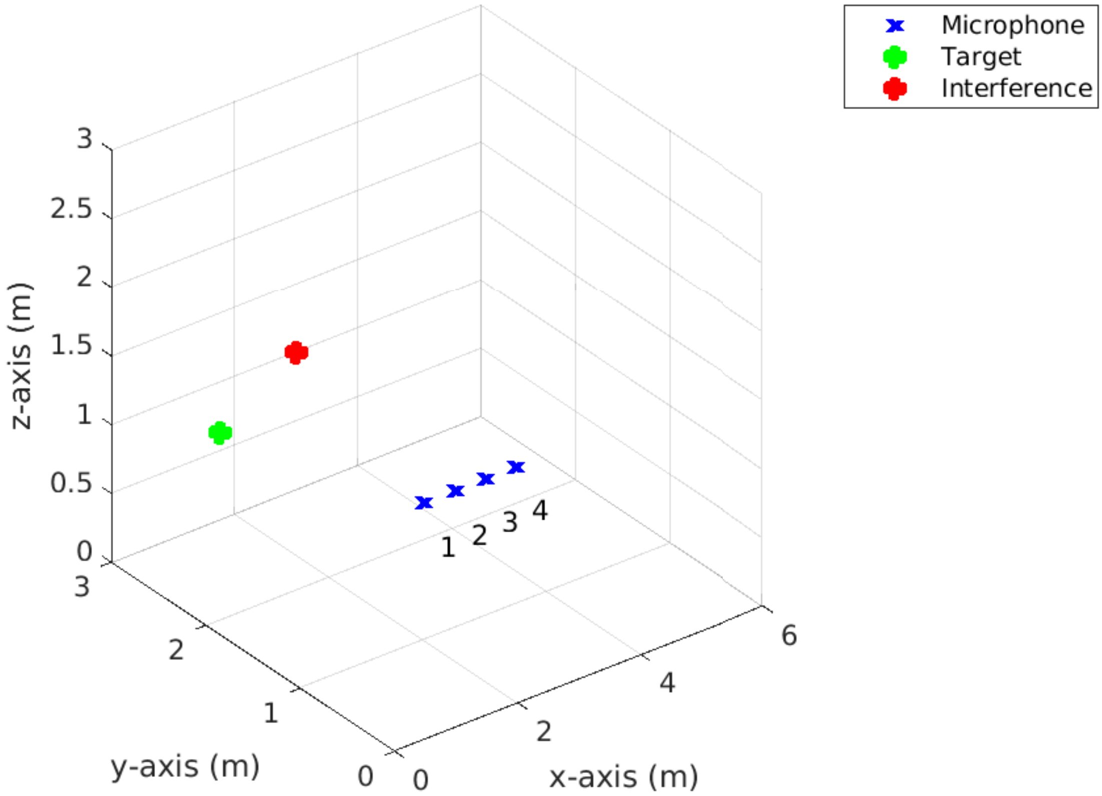


Figure 6.2: The geometry of the setup used in the simulations is depicted here, with the locations of the microphones and signal sources. A room with dimensions 6 m × 3 m × 3 m is used.

6.2 Simulation setup

The WASN setup discussed in the previous section and shown in Figure 6.1 was simulated in Matlab. See Figure 6.2 for the geometry of the setup. The basic signal flow for the simulation can be seen in Figure 6.3, with a clarification of the symbols used in Table 6.1. The setup consists of a single target (speech) and an interfering signal (i.e. background noise). The speech recordings are sourced from [32]. Additionally, the microphones are subject to spatially white self noise. To estimate the noise correlation matrix $\mathbf{R_V}$, it is assumed that there is a 'quiet' period where only the self noise and interference are present.

From the received data $\mathbf{y}$ a blind estimate of the (relative) steering vector $\hat{\mathbf{d}}$ is constructed. This is done using the low-rank approximation $\hat{\mathbf{R}}_{\mathbf{X}}$ as defined in (4.36). Using $\hat{\mathbf{R}}_{\mathbf{X}}$ and the earlier estimated noise correlation matrix $\hat{\mathbf{R}}_{\mathbf{V}}$ we can construct the MVDR beamformer, as given in Table 2.1. The beamformer can then be used to estimate the target signal (at microphone 1).

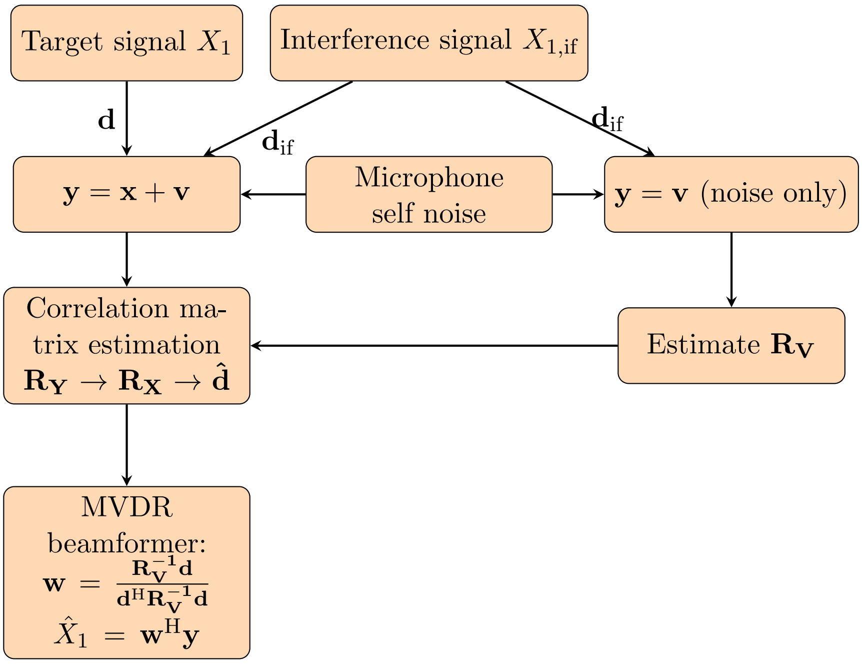
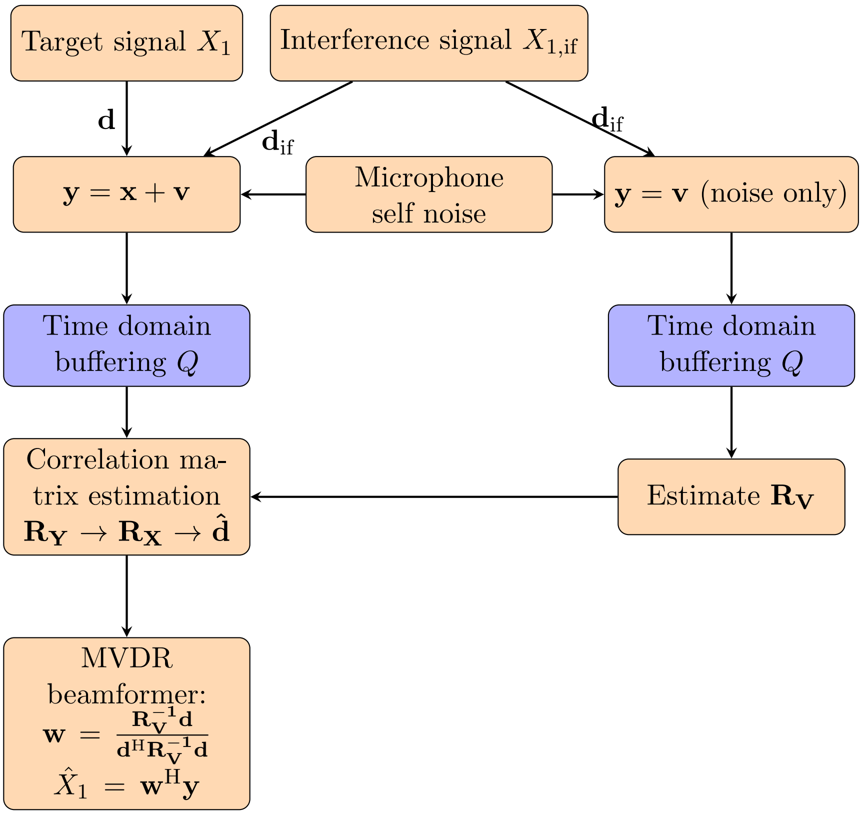

Target signal X_1
Interference signal $X_{1,\mathrm{if}}$
$\mathbf{d}$
$\mathbf{d}_{\mathrm{if}}$
$\mathbf{d}_{\mathrm{if}}$
$\mathbf{y} = \mathbf{x} + \mathbf{v}$
Microphone self noise
$\mathbf{y} = \mathbf{v}$ (noise only)
Correlation matrix estimation $\mathbf{R_Y} \to \mathbf{R_X} \to \hat{\mathbf{d}}$
Estimate $\mathbf{R_V}$
MVDR beamformer: $\mathbf{w} = \frac{\mathbf{R_V^{-1}d}}{\mathbf{d^H R_V^{-1} d}}$
$\hat{X}_1 = \mathbf{w^H y}$

Figure 6.3: Basic signal flow for obtaining beamformed signal.

Symbol	**Definition**
X_1	target source signal at microphone 1
$\mathbf{R_Y}$	received data correlation matrix
$\mathbf{R_X}$	target source correlation matrix
$\mathbf{R_V}$	noise and interference correlation matrix
$\mathbf{d}$	steering vector to target source
$\mathbf{d}_{\mathrm{if}}$	steering vector to interfering source
$\mathbf{w}$	beamformer
$\hat{X}_1$	beamformer target source estimate
ϵ_i	sample rate offset (SRO) of microphone i w.r.t. microphone 1 (the reference microphone)
L	number of averaging STFT frames used with the Welch method
α	recursive smoothing window forgetting factor

Table 6.1: Symbols used in the wireless acoustic sensor network (WASN) simulation

We shall now proceed to extend the basic signal flow shown in Figure 6.3 to the specific implementation for the proposed and reference algorithms. The signals flows for the proposed algorithm and reference algorithm are depicted in Figure 6.4 and Figure 6.5, respectively.



Target signal X_1
Interference signal $X_{1,\mathrm{if}}$
$\mathbf{d}$
$\mathbf{d}_{\mathrm{if}}$
$\mathbf{d}_{\mathrm{if}}$
$\mathbf{y} = \mathbf{x} + \mathbf{v}$
Microphone self noise
$\mathbf{y} = \mathbf{v}$ (noise only)
Time domain buffering Q
Time domain buffering Q
Correlation matrix estimation $\mathbf{R_Y} \rightarrow \mathbf{R_X} \rightarrow \hat{\mathbf{d}}$
Estimate $\mathbf{R_V}$
MVDR beamformer: $\mathbf{w} = \frac{\mathbf{R_V^{-1}d}}{\mathbf{d}^{\mathrm{H}}\mathbf{R_V^{-1}d}}$ $\hat{X}_1 = \mathbf{w}^{\mathrm{H}}\mathbf{y}$

Figure 6.4: Signal flow for obtaining beamformed signal with the proposed algorithm. A time domain buffer is implemented, which synchronizes the data from different nodes every Q samples.

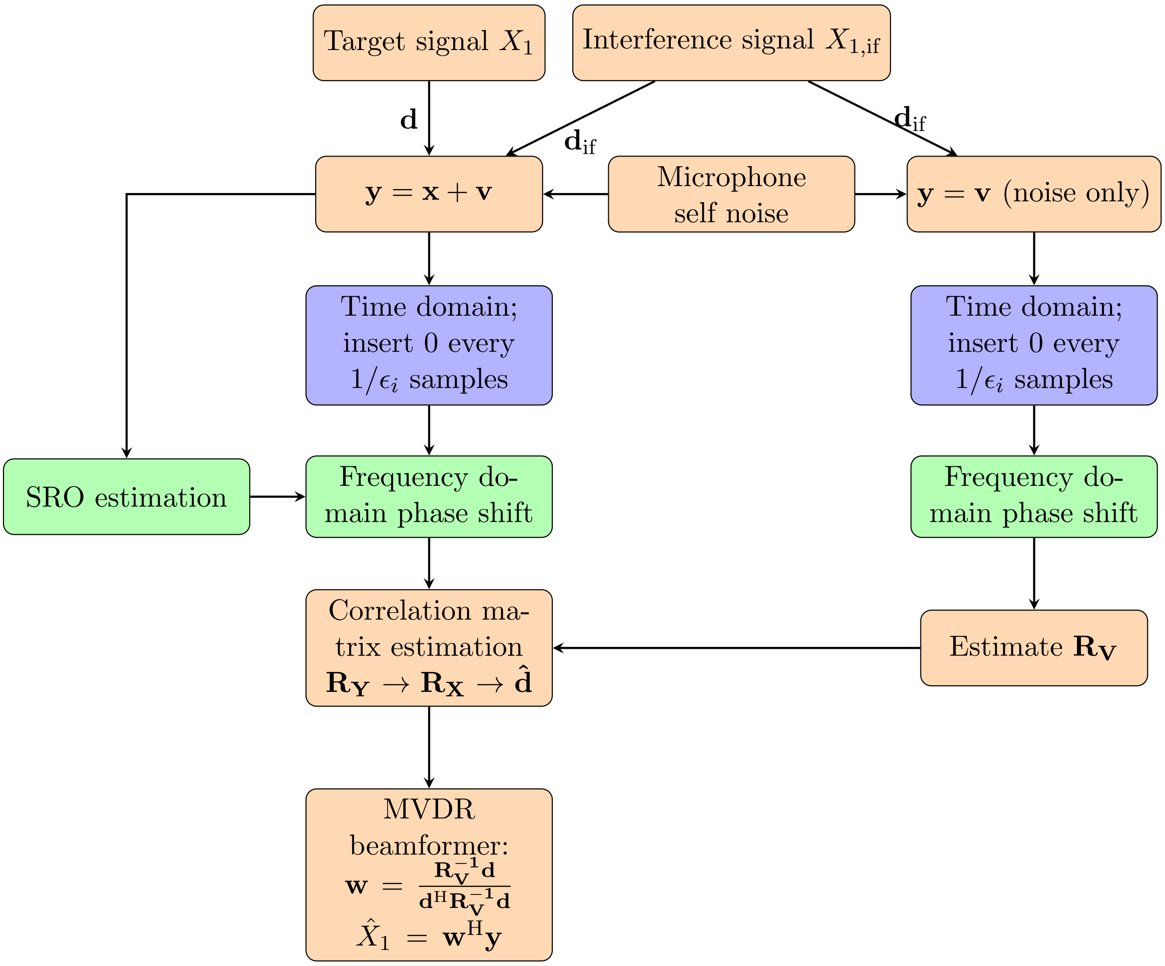

Target signal X_1
Interference signal $X_{1,\text{if}}$
$\mathbf{d}$
$\mathbf{d}_{\text{if}}$
$\mathbf{d}_{\text{if}}$
$\mathbf{y} = \mathbf{x} + \mathbf{v}$
Microphone self noise
$\mathbf{y} = \mathbf{v}$ (noise only)
Time domain; insert 0 every $1/\epsilon_i$ samples
Time domain; insert 0 every $1/\epsilon_i$ samples
SRO estimation
Frequency domain phase shift
Frequency domain phase shift
Correlation matrix estimation $\mathbf{R_Y} \rightarrow \mathbf{R_X} \rightarrow \hat{\mathbf{d}}$
Estimate $\mathbf{R_V}$
MVDR beamformer: $\mathbf{w} = \frac{\mathbf{R_V^{-1}d}}{\mathbf{d^H R_V^{-1} d}}$
$\hat{X}_1 = \mathbf{w^H y}$

Figure 6.5: Signal flow for obtaining beamformed signal with the reference algorithm [1]. The algorithm uses a time domain compensation which synchronizes data from different nodes every $1/|\epsilon_i|$ samples (where ϵ_i is the sample rate offset (SRO) of microphone i w.r.t. microphone 1). The SRO is estimated using linear coherence drift (LCD) [1] and compensated for in the frequency domain using a phase shift. For the SRO estimation parameters used, see Table B.1.

In Table 6.2 simulation parameters are summarized which are constant across the different scenarios used later on.

Parameter	**Symbol**	**Value**
Sample rate at microphone 1	$f_{s,1} = f_{s,\text{ref}}$	16 kHz
Simulation realizations	N_runs	10
Simulation length	T_sim	30 s
STFT parameters frame size hop size window	 K $K_h = K/2$ w	 512 samples 256 samples square root Hann (length K)
Microphone self noise signal-to-noise ratio (at each microphone)	SNR_i	30 dB
Signal-to-interference ratio at microphone 1	SIR_1	0 dB
Room impulse response (RIR) T60 decay time	T_{60}	< 14 ms

Table 6.2: Simulation parameters for the wireless acoustic sensor network (WASN) setup.

We shall proceed to define the signal-to-noise ratio (SNR) and signal-to-interference ratio (SIR) as used in Table 6.2. We define the microphone self noise SNR at microphone i as:

$$\text{SNR}_i = \frac{\|x_i\|_2^2}{\|v_{i,\text{sn}}\|_2^2} \tag{6.1}$$

where x_i is the target signal received at microphone i and $v_{i,\text{sn}}$ is the microphone self noise signal at microphone i.

We define the SNR at the beamformer output as:

$$\text{SNR}_\text{out} = \frac{\|x_i\|_2^2}{\|\hat{x}_i - x_i\|_2^2} \tag{6.2}$$

where x_i is the target signal received at microphone i and $\hat{x}_i$ is the target signal estimated by the beamformer. This equation is used for performance evaluation in the next section.

We define the SIR at microphone 1 as:

$$\text{SIR}_1 = \frac{\|x_1\|_2^2}{\|v_{1,\text{if}}\|_2^2} \tag{6.3}$$

where x_1 is the target signal received at microphone 1 and $v_{1,\text{inter}}$ the received signal at microphone 1 when only the interfering signals are present. Note that the SNR and

SIR are calculated in the time domain. The above definitions assume we have access to the target signals x_i at each microphone i, which is possible in the simulation. In practice, we may only have access to the target signal s and x_i can be found using the estimated steering vector.

6.2.1 Room impulse response synthesis

We used the code for synthesizing the room impulse responses (RIRs) from [33], which makes use of the method presented in [34]. The room depicted in Figure 6.2 was used to generate the RIRs and the microphones were set to an omnidirectional characteristic. The T_{60} time, that is, the time it takes for the magnitude of the RIR to decay to 60 dB below its peak value, is limited to half the length of an STFT frame, by choosing appropriate reflection coefficients for the walls. The T_{60} limit is given by: $K/(2f_{s,\text{ref}}) = 16\,\text{ms}$. Choosing a RIR with a length below 16 ms assures that the room impulse response can be fully contained in the steering vector. Refer to Table 6.2 for the realized T_{60} time of the RIRs.

6.3 Results using babble noise as interfering source

A speech signal is used as the target source and babble noise is used as the interfering source. The babble noise signal is a recording of multiple people speaking at the same time. See Figure 6.6 for time domain plots of the used signals. These signals are used throughout this section for the simulations.

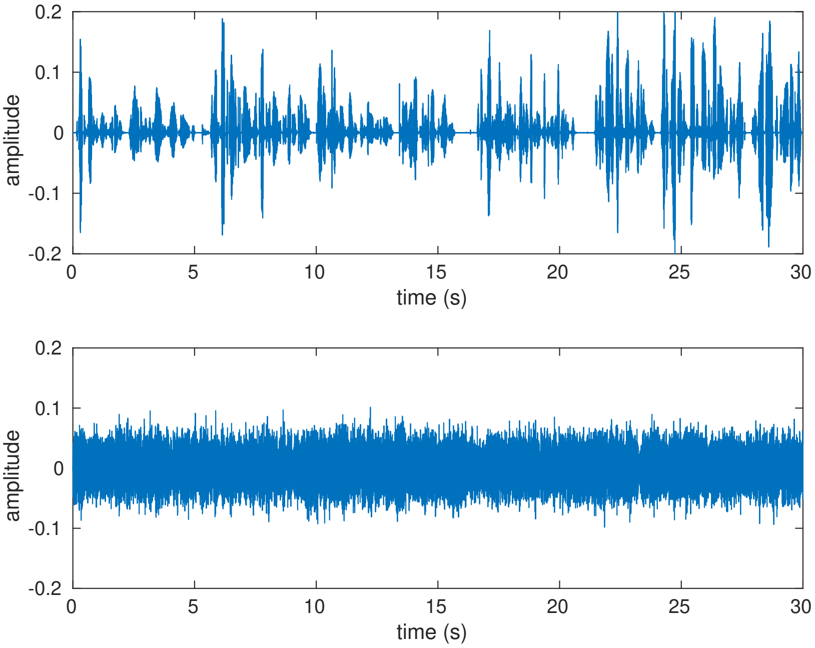
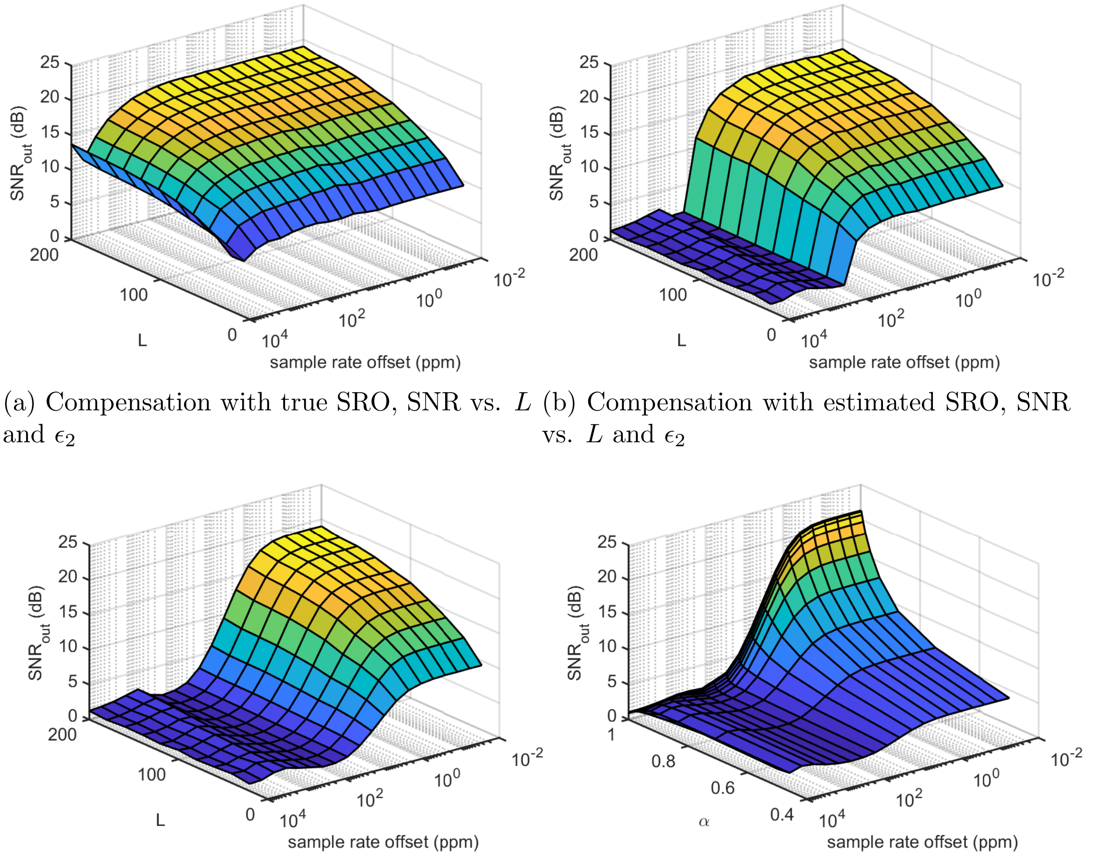


Figure 6.6: Plots of the speech signal (top) and babble noise interference signal (bottom).

6.3.1 2 microphones

In this section we evaluate the performance of the different algorithms, using only microphones 1 and 2 as displayed in Figure 6.2. From (5.13) we know that using the Welch method, the error caused by the estimation of the correlation matrices is a function of the SRO and the estimation length L. From (5.20) we know that using recursive smoothing, the error caused by the estimation is a function of the SRO and the forgetting factor α. Therefore it is useful to evaluate the performance of the beamformers as a function of two parameters. In Figure 6.7a and 6.7b the performance of the reference algorithm is plotted using either the true or estimated SRO value for compensation. In Figure 6.7c and Figure 6.7d the performance of the proposed algorithm is plotted using the Welch and recursive smoothing estimation methods, respectively.

Comparing Figure 6.7a and 6.7b with Figure 6.7c and Figure 6.7d we see that indeed the compensation performed by the reference method is effective, especially in the SRO region $> 10\,\text{ppm}$. From Figure 6.7d we see that the proposed method using recursive smoothing performs well only for values of α close to 1. When looking specifically at Figure 5.1b, which has the same α and L values as used in Figure 6.7, we may expect that recursive smoothing performs worse than the Welch method at those settings. The contrary is shown however in Figure 6.7d. We conclude that although recursive smoothing will give a lower estimation error due to a SRO at lower values of α, a high value of α should be selected. This is necessary to keep the variance of the estimate low enough, which otherwise dominates the estimation error. The same can be said for the Welch method, where sufficient estimation frames need to be used to reduce the variance of the estimate.



(a) Compensation with true SRO, SNR vs. L and ϵ_2

(b) Compensation with estimated SRO, SNR vs. L and ϵ_2

(c) Proposed algorithm with Welch method, SNR vs. L and ϵ_2, no time domain synchronization ($Q = \infty$)

(d) Proposed algorithm with recursive smoothing, SNR vs. α and ϵ_2, no time domain synchronization ($Q = \infty$)

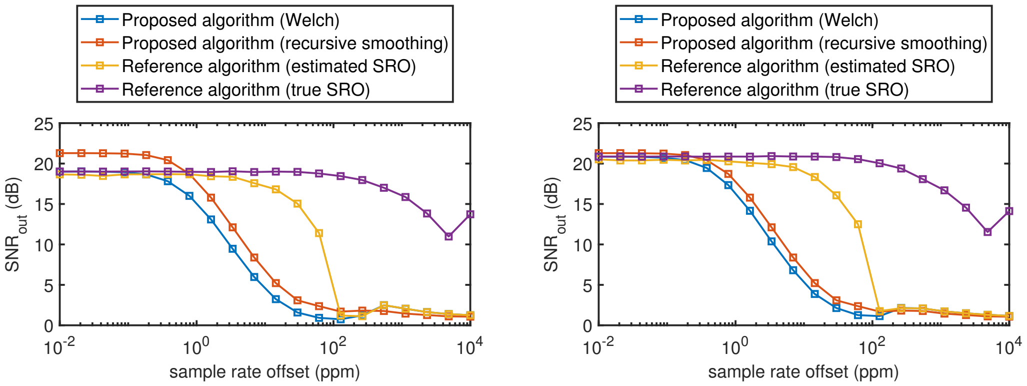
Figure 6.7: Plot of the SNR vs. L, α and the sample rate offset (SRO) for different algorithms, using $M = 2$ microphones. The larger the value of L, the more past frames are used to estimate $\mathbf{R_Y}$. The larger the value of α, the higher the emphasis on past frames used to estimate $\mathbf{R_Y}$. The reference algorithm uses the Welch method as presented in [1].

To compare the different algorithms, we 'slice' the plots in Figure 6.7 at high values of L and α, since there the performance is the best. Note that the performance of the methods using Welch can be improved by selecting higher values of L than shown in Figure 6.7. The comparison of the algorithms for $L = 200$ and $\alpha = 0.9990$ can be seen in Figure 6.8a. A separate simulation for $L = 500$ and $\alpha = 0.9990$ is shown in Figure 6.8b.

Using the information from Figure 6.8, we see that in the range of 10^{-2} - 10^{0} ppm the proposed algorithm with recursive smoothing estimation method performs the best. In the range of 10^{0} - 10^{2} ppm the reference algorithm performs better than the proposed algorithm. The region from 10^{2} ppm and higher is unrealistic for most scenarios, however the algorithms show similar results. Note that compensation with true SRO is not practically realizable, it is merely used as a reference.

Comparing the two implementations of the proposed algorithm, we see that the implementation using recursive smoothing performs better over most of the SRO range. The difference between the two implementations is smaller in Figure 6.8b than in Figure 6.8a, however. This is due to the larger number of frames used by the Welch method in Figure 6.8b.

In the region of higher SRO values (> 100 ppm), the performance of the SRO estimation decreases for the reference algorithm, as we see the performance of the two implementations of the reference algorithm diverging. The STFT frame size used for SRO estimation is 8192 samples (see Table B.1). For higher SRO values than $1/8192 \approx 122$ ppm, inside a single STFT frame, the signals from the two microphones drift apart more than 1 sample. The constant phase shift across a single STFT frame approximation will not be accurate anymore. In fact, the approximation becomes less accurate as the SRO becomes larger. This is likely the cause of the SRO estimation performance degradation at higher SRO values.



(a) Slices of the plots in Figure 6.7 are taken at $L = 200$ and $\alpha = 0.9990$.

(b) Simulation with $L = 500$ and $\alpha = 0.9990$.

Figure 6.8: A comparison of the different algorithms, using $M = 2$ microphones. No time domain synchronization is used for the proposed algorithm, therefore $Q = \infty$.

The performance loss for the proposed algorithm at high SROs in Figure 6.8 is caused in part by the error due to estimation of the correlation matrices. However, the

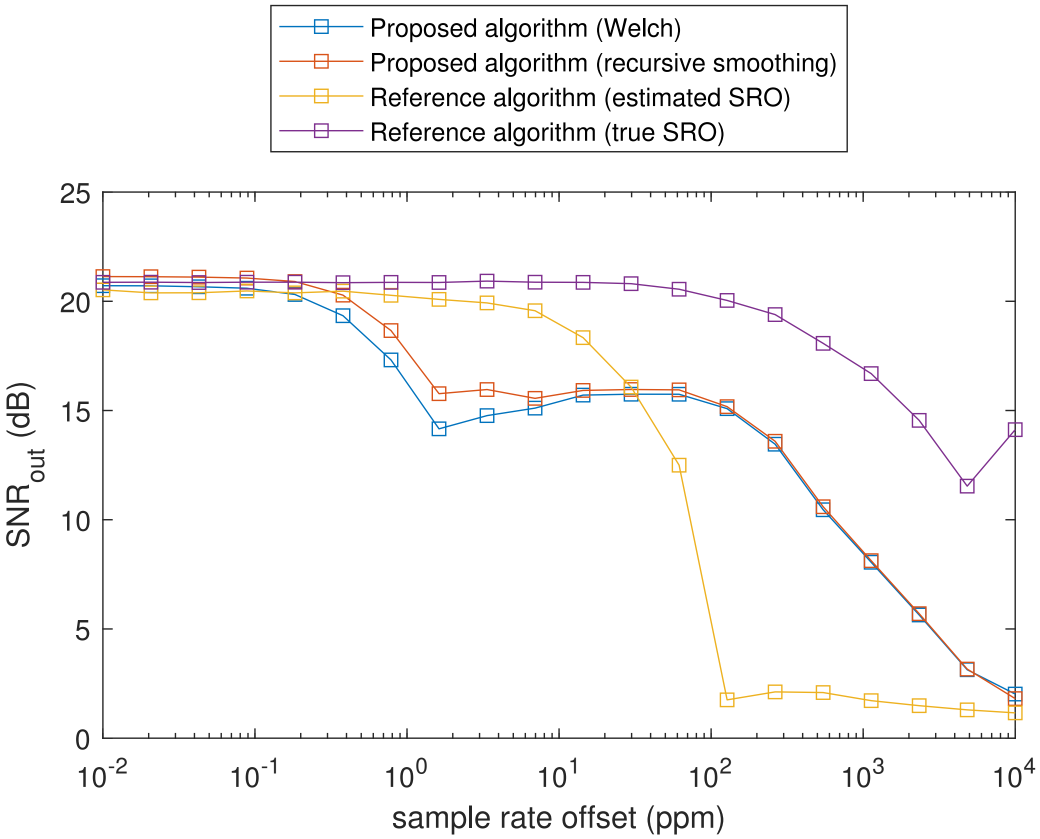


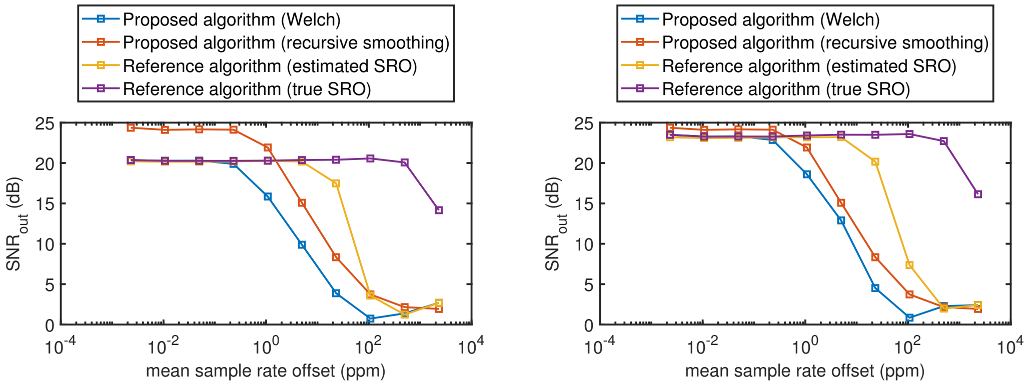
Figure 6.9: Simulation using $M = 2$ microphones with $L = 500$ and $\alpha = 0.9990$. For the proposed method, we use time domain synchronization with $Q = 5120$, i.e. every 20 short-time Fourier transform (STFT) frames the buffers are taken.

coherence between samples is lost over time due to SRO and this becomes apparent especially at high SROs. Up to this point we have looked at the most basic implementation of the proposed algorithm without any time domain synchronization. In Figure 6.9 the proposed algorithm performance is shown with time domain synchronization for every 20 STFT frames. This synchronization is applied using the buffering scheme described in subsection 6.1.1.

The time domain synchronization clearly boosts the performance in the SRO range of 1 up to 10 000 ppm.

6.3.2 4 microphones

To further study the application to a WASN, the microphone array is extended to using four microphones, as displayed in Figure 6.2. Refer to Figure 6.10: we observe from both Figure 6.10a and Figure 6.10b that the proposed method using recursive smoothing performs better than using the Welch method. Furthermore, the recursive smoothing method has the better performance up to a SRO of 1 ppm. From Figure 6.10 we also clearly see that the SNR performance has increased over Figure 6.8, which is expected for a larger microphone array.



(a) Simulation with $L = 200$ and $\alpha = 0.9990$. (b) Simulation with $L = 500$ and $\alpha = 0.9990$.

Figure 6.10: A comparison of the different algorithms, using $M = 4$ microphones. Different settings of L are used. No time domain synchronization is used for the proposed algorithm, therefore $Q = \infty$.

Similar to the setup with two microphones, we shall proceed to implement time domain synchronization in the proposed algorithm for the four microphone setup. See Figure 6.11 for a performance comparison. Again, we see performance from the proposed algorithm that is more comparable with the reference algorithm in the high SRO range. The proposed algorithm using recursive smoothing has a slight advantage over the proposed algorithm using the Welch method.

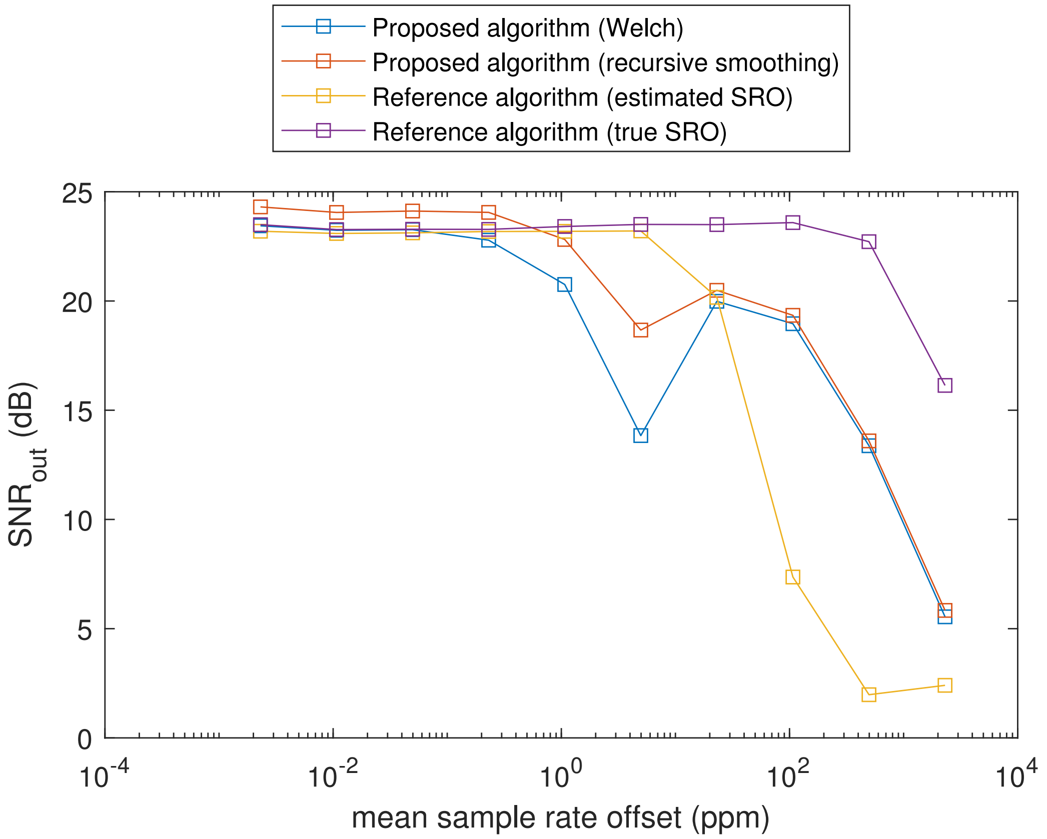


Figure 6.11: Simulation using $M = 4$ microphones with $L = 500$ and $\alpha = 0.9990$. For the proposed method, we use time domain synchronization with $Q = 5120$, i.e. every 20 short-time Fourier transform (STFT) frames the buffers are taken.

6.4 Conclusion

We recall the research question that was stated at the end of Chapter 3:

Is the theory for clock offset invariant beamforming applicable to clock skew affected beamformers in a WASN?

In Chapter 4 we showed that using GEVD based beamformers, under the assumption of a linearly increasing phase shift across the STFT frames, this is possible. The practical side of the problem was addressed in Chapter 5, where we discussed the estimation of correlation matrices that are necessary for the GEVD based beamformers. The “standard” Welch method was used to estimate the correlation matrices. It was shown that the estimated correlation matrices where biased, due to the clock skew. This introduced an error in the estimated correlation matrices and consequently in the beamformers. In addition to the Welch method, a recursive smoothing method was studied, and a possibly improved estimation performance for the latter method was predicted.

In this chapter we compared the proposed algorithms with a reference algorithm. From all the experiments we saw that the performance of the proposed algorithm using recursive smoothing was equal to or better than the proposed algorithm using the Welch method. Furthermore, for SRO values up to 1 ppm the performance of the proposed algorithm using recursive smoothing was equal to or better than the other algorithms. When the buffering scheme presented in subsection 6.1.1 is used, the performance of the proposed algorithm using recursive smoothing was comparable to the reference algorithm even for high SRO values.

Looking at hardware implementation of the proposed algorithm, we note that using the recursive smoothing method less memory resources have to be used than using the Welch method. With the recursive smoothing method we store a single correlation matrix, whereas the Welch method needs L correlation matrices which are stored in a first in, first out (FIFO) buffer. Compared with the reference algorithm, where the SRO estimation and compensation causes additional computational load for the fusion node, the proposed algorithm requires no additional computations.

From this study we conclude that for a wide range of SROs the theory of clock offset invariant beamforming is applicable to clock skew affected beamformers in a WASN, when used with periodic time domain synchronization. The periodic time domain synchronization is inherent to the WASN setup used in this thesis, because it utilizes a buffering scheme to collect data from the nodes.

6.5 Future work

This thesis focused on WASNs tasked with beamforming, subject to clock skew. Clock skew invariant beamforming is not perfectly realizable, however it appears that for a wide range of SRO values the proposed algorithm can be used. The material discussed in this thesis can benefit from further study. In Chapter 5 the main contribution from this thesis was presented. However, the extent to which the theory on correlation matrix estimation can be used is limited. In Chapter 6 a WASN setup was simulated, which was limited to a setup with up to four microphones and a single target and interference source. To address the limitations of the work done in this thesis various recommendations are stated below:

- *Developing (faster) correlation matrix estimation methods.* From the results we see that for the proposed algorithm, using recursive smoothing gives better performance than the Welch method. In general, it can be expected that "faster" estimation methods will perform better in the presence of clock skew (here "faster" means estimating over a shorter period, i.e. using less STFT frames). That is, with a "faster" method the accumulated error due to clock skew is lower. However, a tradeoff still exists between the variance of the estimate and the clock skew induced error. We know from Chapter 5 that error measures can be constructed for estimation methods and thus the methods can be compared analytically. The correlation matrix estimation methods need further study, because they play an important role in beamforming with clock skew.

  The Welch method makes use of periodograms and averages them. Therefore it belongs to the class of non-parametric methods for estimation of the PSD. It may be of interest to use a parametric method based on an autoregressive (AR), moving average (MA) or autoregressive moving average (ARMA) model, such as presented in Hayes [35].

- *Studying a noisy case for the correlation matrix estimation error.* An error measure for the estimated correlation matrices was devised in Chapter 5. A noiseless case was studied, to simplify the equations. This gives a good prediction for the estimation error made using many STFT frames. However, using a limited amount of STFT frames, the influence of noise on the estimate will become significant. A trade-off exists between error due to clock skew and noise. The expectation from Figure 5.1 that lower values of α are beneficial to estimator performance was contrasted by the result in Figure 6.7d where we saw that values of α close to 1 where desirable, which is due to this trade-off. It is therefore interesting to investigate a noisy case to study the trade-off between noise performance and clock skew induced error.

- *Simulating for a larger number of sources/microphones.* The setup used is not worked out for a large number of scenarios: only a single target, single interference scenario has been simulated. For the four microphone scenario we saw a significantly different outcome than the two microphone scenario, therefore larger sizes of the microphone array need to be simulated.

- *Using the results of many different source signals.* All of the simulations were carried out using a single target and interference signal. If we view the speech/interfering signal as a realization of a particular stochastic process, the setup used takes a very small number of samples, which generates errors in the results. We decided to focus on a comparison of the performance of our proposed method relative to a reference method, which should still be possible given this limited setup. To get a more realistic estimate of the absolute performance however, an extended setup should be used which uses more test signals.

- *Carrying out a Monte-Carlo analysis with varying source locations.* In the simulation conducted by [18] the locations of the sources are changed randomly for each different SRO value simulation. This reduces possible artifacts introduced in the results by the specific source setup in the room.
- *Realizing a physical WASN setup with real SROs.* Realizing a physical setup can be very useful, since the SROs present in actual devices can be used. Such a study is performed in [19]. This increases the connection to reality of the study.

A Glossary

Abbreviation	**Full form**
DFT	Discrete Fourier Transform
(G)EVD	(Generalized) Eigenvalue Decomposition
MSE	Mean Squared Error
ppm	parts per million
PSD	Power Spectral Density
RIR	Room Impulse Response
SIR	Signal-to-interference ratio
SNR	Signal-to-noise ratio
SRO	Sample Rate Offset
STFT	Short Time Fourier Transform
WASN	Wireless Acoustic Sensor Network
WSS	Wide-Sense Stationary

Table A.1: A list of commonly used abbreviations throughout the thesis.

Extra figures and tables B

B.1 Parameters used for sample rate offset estimation

Parameter	**Value**
Coherence frame size	16 384 samples
Coherence frame overlap	8192 samples
Number of coherence frames used	6
Periodogram frame size	8192 samples
Periodogram hop size	2048 samples
Periodogram window	square root Hamming

Table B.1: Parameters used for the sample rate offset (SRO) estimation, taken from [1] and adapted to a sample rate of $f_s = 16\,\text{kHz}$.

Bibliography

[1] M. H. Bahari, A. Bertrand, and M. Moonen, "Blind sampling rate offset estimation for wireless acoustic sensor networks through weighted least-squares coherence drift estimation," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 25, p. 674–686, Mar. 2017.

[2] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*. Springer, 2011.

[3] R. A. Dolin, "Deploying the "internet of things"," in *International Symposium on Applications and the Internet (SAINT'06)*, pp. 4–219, 2006.

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 4, pp. 2033–2036 vol.4, 2001.

[5] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[6] R. K. Karlquist and H. M. Stephanian, "Manufacturing issues for a high performance crystal oscillator," in *Proceedings of the 2000 IEEE/EIA International Frequency Control Symposium and Exhibition (Cat. No.00CH37052)*, pp. 238–246, 2000.

[7] M. Bloch, J. Ho, and O. Mancini, "Highly reproducible state-of-the-art quartz oscillators," in *Proceedings of the 2002 IEEE International Frequency Control Symposium and PDA Exhibition (Cat. No.02CH37234)*, pp. 615–618, 2002.

[8] S. E. Kotti, R. Heusdens, and R. C. Hendriks, "Clock-offset and microphone gain mismatch invariant beamforming," Available at https://cas.tudelft.nl/Education/courses/in4182/slides/2020022490627_390736_1307.pdf (visited on 25/08/20).

[9] H. Wang, H. Zeng, and P. Wang, "Linear estimation of clock frequency offset for time synchronization based on overhearing in wireless sensor networks," *IEEE Communications Letters*, vol. 20, no. 2, pp. 288–291, 2016.

[10] X. Sun, Z. Wang, R. Xia, J. Li, and Y. Yan, "Effect of steering vector estimation on mvdr beamformer for noisy speech recognition," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, 2018.

[11] J. Main, "PPM Clock accuracy examples." Available at https://www.best-microcontroller-projects.com/ppm.html (visited on 08/08/20).

[12] M. S. McCorquodale, N. Gaskin, and V. Gupta, *Frequency Generation and Control with Self-Referenced CMOS Oscillators*, ch. 9, pp. 207–238. John Wiley & Sons, Ltd, 2012.

[13] C. Harish, R. Mohan, and R. Shashank, "Sampling clock offset estimation and correction in frequency domain for ofdm receivers," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 1583–1587, 2017.

[14] V. Kaajakari, A. Pangaro, Y. Goto, T. Nishimura, T. Okawa, H. Seki, A. Suzuki, and K. Umeda, "A 32.768 khz mems resonator with +/-20 ppm tolerance in 0.9 mm x 0.6 mm chip scale package," in *2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC)*, pp. 1–4, 2019.

[15] IQD Frequency Products, "Oven-controlled crystal oscillator (OXCO) product datasheet." Available at `https://www.iqdfrequencyproducts.com/products/details/iqov-114-1-01.pdf` (visited on 08/08/2020).

[16] D. Wobschall and Y. Ma, "Synchronization of wireless sensor networks using a modified ieee 1588 protocol," in *2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pp. 67–70, 2010.

[17] G. Zhao, H. Ma, H. Luo, and Y. Sun, "Adaptive audio synchronization scheme based on feedback loop with local clock in wireless audio sensor networks," in *2010 IEEE 16th International Conference on Parallel and Distributed Systems*, pp. 609–616, 2010.

[18] S. Markovich-Golan, S. Gannot, and I. Cohen, "Blind sampling rate offset estimation and compensation in wireless acoustic sensor networks with application to beamforming," in *IWAENC 2012; International Workshop on Acoustic Signal Enhancement*, pp. 1–4, 2012.

[19] D. Cherkassky and S. Gannot, "Blind synchronization in wireless sensor networks with application to speech enhancement," in *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 183–187, 2014.

[20] D. Cherkassky and S. Gannot, "Blind synchronization in wireless acoustic sensor networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 651–661, 2017.

[21] M. H. Bahari, A. Bertrand, and M. Moonen, "Blind sampling rate offset estimation based on coherence drift in wireless acoustic sensor networks," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2281–2285, 2015.

[22] Y. Zeng, R. C. Hendriks, and N. D. Gaubitch, "On clock synchronization for multi-microphone speech processing in wireless acoustic sensor networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 231–235, 2015.

[23] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel wiener filter algorithms for noise reduction with application in cochlear implants," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 785–799, 2014.

[24] A. Hassani, A. Bertrand, and M. Moonen, "Gevd-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2557–2572, 2016.

[25] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.

[26] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70–73, June 1967.

[27] V. Serov, *Fourier Series, Fourier Transform and Their Applications to Mathematical Physics*. Springer International Publishing, 2017.

[28] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégualités entre les valeurs Moyennes," Nov. 1906.

[29] Y. S. Cho, S. B. Kim, and E. J. Powers, "Time-frequency analysis using ar models with variable forgetting factors," in *International Conference on Acoustics, Speech, and Signal Processing*, pp. 2479–2482 vol.5, 1990.

[30] R. C. Hendriks, R. Heusdens, J. Jensen, and U. Kjems, "Fast noise psd estimation with low complexity," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3881–3884, 2009.

[31] M. Parchami, W. Zhu, and B. Champagne, "A new algorithm for noise psd matrix estimation in multi-microphone speech enhancement based on recursive smoothing," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 429–432, 2015.

[32] J. S. Garofolo, National Institute of Standards and Technology (U.S.), United States, and Linguistic Data Consortium, "Timit: acoustic-phonetic continuous speech corpus," 1993.

[33] E. A. Habets, "Rir generator." Available at International Audio Laboratories Erlangen https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator (visited on 18/08/2020.

[34] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[35] M. H. Hayes, *Statistical digital signal processing and modeling*, p. 440–451. Wiley, 2006.