Creation of dynamic sound zones with adaptive filters

Master's Thesis
Group SPC-1073
SIGNAL PROCESSING AND COMPUTING
Aalborg University
04. JUNE 2020

Aalborg University
Department of Electronic systems
Signal Processing and Computing
Fredrik Bajers vej 7
9220 Aalborg Øst
www.es.aau.dk

Title:

Creation of dynamic sound zones with adaptive filters

Theme:

Signal Processing and Computing

Period of the project:

01. Febuary 2020 - 04. June 2020

Project group:

SPC-1073

Members:

Thomas Damgaard
Mads Hulegaard

Supervisor:

Jan Østergaard

Pages: 67
Appendix: A
Ended: 04-06-2020

Abstract:

In this Master's Thesis, frequency domain adaptive FIR filters that forms dynamic sound zones in spatially confined regions, when filtered playback signals are reproduced, are made. Based on the filtered signals, dynamic models have been proposed which models the acoustic pressure in dynamic spatial points with the assumptions of far-field and anechoic conditions.
By solving an optimization problem iteratively, the filters that minimizes the difference between the dynamic modeled pressure and a dynamic reference pressure are calculated. The filters are thereby adaptable to the dynamic sound zones.
Two dynamic models are proposed. One which includes the pressure contributions from other playback signals and one which models the pressure independent of other playback signals. The two methods had equal reproduction error and almost equal array effort.
The proposed adaptive filters require low memory and the updating procedure can be heavily parallelized.

Preface

This thesis has been compiled by project group SPC-1073 as the Master's Thesis under the main theme *Signal Processing and Computing*, at the Department of Electronic Systems, at Aalborg University, Spring 2020.

This thesis is indexed in chapters chronologically numbered after the order in which they appear. Sections and subsections in chapters are numbered likewise, while sub-subsections are without index numbers. Figures, tables and equations are also indexed in numbers equivalent to the chapter and chronological order in which they appear, appendixes are lettered in alphabetical order, in which they appear.

For the purpose of simulations conducted in this thesis, MATLAB R2020a has been used as the main simulation tool.

The authors would like to thank Business PhD at Department of Electronic Systems, Aalborg University, Martin Bo Møller for giving insight in different areas for designing and creating sound zones, being available for questions and giving feedback on material.

Thomas Damgaard
<tdamga15@student.aau.dk>

Mads Hulegaard
<madjen15@student.aau.dk>

Contents

Introduction 1

This thesis will focus on applying adaptive filters for reproduction of audio signals in dynamic sound zones.

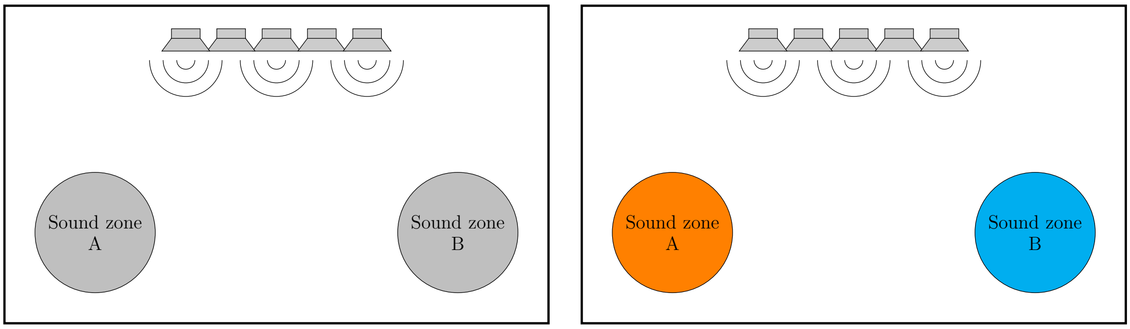
A reproduction system is considered to be a set of loudspeakers and the environment in which the loudspeakers are playing a controllable signal. The environment can be anechoic or a reflective environment such as a living room. The reproduction system is used to reproduce playback signals. A playback signal is considered to be a monophonic signal, which a user desires to listen to at his/her location. A specific confined region in the environment is referred to as a sound zone. When sound is produced by the reproduction system, a sound event is created. The sound event created is among other dependent on the controllable signal fed to the loudspeakers, the location of the loudspeakers and the reflective nature of the environment. The sound event in the sound zone will be a sum of the signals played by the individual loudspeakers but delayed with different time lags due to the difference in distance to the sound zone [1][2].

A desired sound event in a sound zone is referred to as a target signal or target sound. It is easy to think of a scenario with multiple sound zones, each with different target sounds. It could be one person in a living room, called person A, watching a movie in sound zone A, which is located in the couch. Another person, called person B, is listening to music at the dining table which is in sound zone B. Person A and B are located in the same environment, but in different sound zones. In this scenario, none of the people desires to wear headphones due to the strain of prolonged use. The target sound in sound zone A is the movie sound, and the target sound in zone B is the music. However, the sound event each person is experiencing is a mixture of the two target sounds. Person A is therefore distracted by the target sound from person B and vice versa.

To enhance the sound experience of person A and B, the sound event in each zone must be the target sound. This can be achieved by eliminating the cross talk between the sound zones, and thereby lessen the distraction [3]. This concept is referred to as sound zones, and it is applicable in many situations. An illustration of the target sounds in each sound zone, before and after sound zones is applied, is illustrated in Figure 1.1.

Sound zones can both be advantageous for home use and in the industry. It could be used at home to deliver different music to different people without interference. In the industry it could be used to lessen the general noise level from speaker systems, by only delivering the sound to the persons of interests. It could for example be used in hospitals to communicate with the hospital personnel over an intercom system without disturbing the patients.

The concept of sound zones have been studied through the years, and there exists multiple ways of achieving adequate separation between the sound zones [4][5][6][7]. Creating sound zones relies on the different loudspeakers responding to the same playback signal in different ways [5]. This could for



(a) Illustration of mixed target sounds in both sound zone A and B.

(b) Illustration of target sound A and B in each respectively sound zone.

Figure 1.1: Illustration of environment together with a loudspeaker array with two sounds and their sound event before(a) and after(b) sound zones are applied.

example be by delaying the playback signal slightly or adjusting the gain to some of the loudspeakers, such that cross talk is diminished. A way of achieving both gain adjustment and delaying of signals is by filtering the playback signal with FIR filters. Here each loudspeaker in the reproduction system will have an FIR filter, which is likely to be different across the loudspeakers. The difficult task in creating sound zones is to calculate these FIR filters such that sound zones are created.

Calculating the best FIR filters for creating sound zones can be done by solving an optimization problem [7]. How the optimization problem is formulated does however have an influence on the properties of the solution. Choosing a problem formulation such as pressure matching results in a low reproduction error but also a lower directivity of the sound event [3]. The opposite applies when choosing a method called acoustic contrast control [3].

When comparing the home and the industrial example above, different features may be desired. When listening to music in the home a low reproduction error of the playback signal in the sound zone might be desired. The opposite could be the case in the industrial environment where a high contrast might be of interest, rather than a low reproduction error, due to close placement of employees. As an example of this, a message regarding one employee at a production hall could be played with high acoustic contrast but with a higher reproduction error, such that the employees next to do not get disturbed, still ensuring that the first employee gets the message.

Defining stationary sound zones could be feasible in the case of watching movies or other activities which does not require movement. If a person moves outside the sound zone, the experienced sound event is presumably no longer the target sound. To improve the functionality of sound zones, the zones must be able to adapt to the location of a person. Such a functionality would make sound zones a useful tool in more scenarios, such as the hospital example, where the hospital personnel is required to move often.

The idea of sound zone which adapts to the listeners location have been studied in [5][8]. In this study a formulation based on the pressure matching have been presented, which allows controlling of the acoustic pressure at two sound zones, while adapting to the spatial position of the listeners. In addition to controlling the acoustic pressure at two sound zones, a implementation of binaural audio for one listener have been presented.

In the study a uniform linear array consisting of 28 loudspeakers, placed in a horizontal manner are used to form the sound zones. In addition to the speaker array, a tracking device are used such that the filter coefficients, that form the beam to the location of the listener, can be determined. As for the transition between filters from one listener position to another, various cross-fading techniques were used [5].

With the loudspeaker array and calculation of filter coefficients, a 30 dB cross-talk cancellation for two listeners at a frequencies between 300 Hz to 3 kHz were achieved, while one listener was moving [5]. For the binaural audio reproduction, where the listener was in motion, a cross-talk cancellation of 10 dB at 1 kHz growing to 20 dB at 8 kHz was achieved [5].

When using cross-fading techniques the fading ratio determines the speed of which the filter coefficients change [9]. Choosing a too fast transition between filter coefficients may result in audible artifacts in the sound when a listener is in motion. A slow transition would resolve the problem with the audible artifacts, but on the compromise of adaptability speed to the listeners position.

If the closed form filters must be calculated in real time and transitioning between them, it would require much computational power. This could be lessened by precalculating the filters for all possible positions in the environment and store these in memory. This approach can however require much memory.

1.1 Proposed solution

This thesis will work with creating sound zones which adapts to the location of users. The dynamic sound zones will be created with adaptive filters, instead of transitioning between closed form calculated filters. This is to avoid either excessive memory usage if closed form filters for all locations must be stored, or excessive computational power if closed form filters must be calculated and transitioned between in real time.

By using adaptive filters to create sound zones, the filters creating the sound zones approaches the closed form solution. The sound zones created with adaptive filters can therefore be as good as the closed form calculated filters, but do not require the need for precalculation and transitioning of multiple sets of filters per sound zone.

It can be assumed that an update of location will be in the vicinity of the previous location. When using the closed form filters, the necessary calculations are independent of the change of location. When using adaptive filters, adapting the filters from a close to optimal solution in one location to another location might not require many iterations. It thus might be possible to save calculation when using adaptive filters.

The adaptive filters must be applied to a reproduction system. In this thesis an environment with low reflection is used together with a uniform linear loudspeaker array. Furthermore two sound zones will be used which are able to move independent of each other. The sound zones specify the locations of two users who are equipped with a tracking device. The locations are therefore updated in real time. There is a target sound for each sound zone which is specified by two different playback signals, also illustrated in Figure 1.1b.

Furthermore it is decided to limit the creation of adaptable sound zones to the horizontal plane. This delimitation is made since most movement is assumed to be in the horizontal plane and not in three dimensions.

From the proposed solution the following research question is made:

> ***"How can adaptive filters be constructed and applied for creating dynamic sound zones?"***

2 Preliminary study of adaptive filter structure

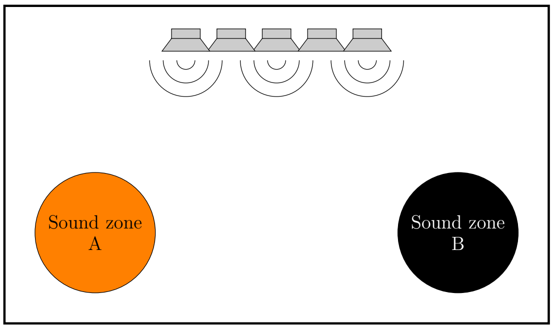
This chapter seeks to study how adaptive filters that fulfills the proposed solution can be made. First an approach for creating the dynamic sound zones, each with their own playback signal, will be explained. The approach will be used when forming a structure for the adaptive filters. The structure of the adaptive filters will be used to determine which elements to design in order to create dynamic sound zones.

2.1 Approach for creating the two sound zones

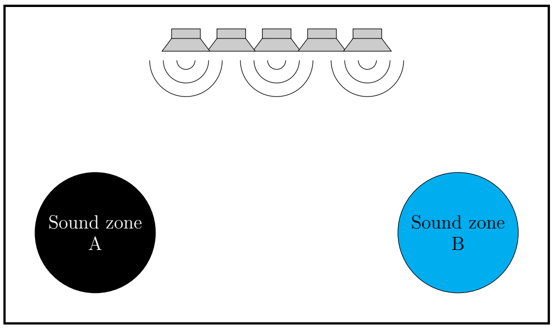
As written in section 1.1 the end goal is to create dynamic sound zones that moves along with users. It is here chosen to define two dynamic sound zones with a user in each zone. This section will describe an approach for creating these dynamic sound zones who each must have their own target sound.

First a scenario where playback signal A is considered along with two sound zones called sound zone A and B. In sound zone A, the target sound is playback signal A, while the target sound in sound zone B is silence. In a perfect scenario, a person located in sound zone A is able to hear playback signal A, while a person located in sound zone B hears nothing when the sound zones are created. Therefore there is no cross talk between sound zone A and sound zone B. This scenario is illustrated in Figure 2.1(a).

Similarly, a scenario considering playback signal B as the target signal in sound zone B while there must be silence in sound zone A is created. This is illustrated in Figure 2.1(b).



(a) Illustration of bright acoustic sound zone A with target sound A and the acoustical dark zone B.



(b) Illustration of bright acoustic sound zone B with target sound B and the acoustical dark zone A.

Figure 2.1: Illustration of the two scenarios where playback signal A is not present in sound zone B and vice versa.

To create two sound zones, each with their own playback signal as target sound, the two aforementioned scenarios can be combined. This is done by creating two sets of filters, where each set is used to filter their associated playback signal, before being played by the loudspeakers. The first set of filters is created such that the first scenario is realized and the second set of filters such that the second scenario is realized.

The two filtered signals are then added together and played by the loudspeaker, creating a mixture of the two scenarios. Ideally, since one set of filters is created such that the other sound zone is acoustically dark, there is no cross talk between the sound zones. The mixture of the two scenario thus corresponds to the original problematique with two sound zones, each with their own playback signal as target sound.

To make the sound zones dynamic, the filters must be created such that the acoustically bright zone in scenario A follows the location of user A, while the acoustically dark zone follows the location of user B. In scenario B, the location of the acoustically bright zone must follow user B, while the location of the acoustically dark zone follows user A. The two sets of filters must therefore be adaptable to the locations of the two users.

It is thus necessary to create two sets of adaptable filters, but the design procedure is identical except which zones must be acoustically bright or dark. The proposed solution from section 1.1 can therefore be narrowed down to creating a single set of adaptable filters, but for two scenarios. It must here be possible to specify an acoustical bright zone where the target signal is a playback signal, and an acoustical dark zone where the target signal is silence.

It is therefore desired to study how a set of filters can be created, such that dynamic acoustical bright and acoustical dark zones are formed.

2.2 Methods for creating sound zones

Calculating the filters for creating sound zones can be done by formulating and solving an optimization problem. The formulation of the optimization problem can be done in many ways, and in this section a discussion of selected methods for formulating the optimization problem will be made. The discussion of the methods will be used to determine which formulation method will be used when creating dynamic sound zones. The methods of forming the optimization problem will be described in the frequency domain and the variables in this section are therefore complex.

In order to choose a method to form the optimization problem to calculate a set of adaptable filters, it is assumed that knowledge of the pressure contributions from the filtered playback signals in a number of measuring points in the two sound zones are available. How this information is obtained will be elaborated later in the thesis.

For this section, the scenarios described in section 2.1 are considered, but to summarize, the ability to create acoustically bright and acoustically dark zones is a requirement. The pressure contributions at discrete frequencies in the acoustically bright zone and the acoustically dark zone are defined as $\mathbf{P}_B \in \mathbb{C}^{M_B \cdot K \times 1}$ and $\mathbf{P}_D \in \mathbb{C}^{M_D \cdot K \times 1}$ respectively. Here M_B is the number of measuring points in the bright zone, M_D is the number of measuring points in the dark zone and K is the number of discrete frequencies.

A simple and common method for creating sound zones, is the delay-and-sum (DAS), which relies on constructive/deconstructive acoustic interference of the reproduced field inside the sound zone [7]. When designing the filters with DAS, only the distances between the bright zone and the loudspeakers are considered. This means that the reproduction field can not be controlled in the acoustically dark zone [7].

As it is desired to control the reproduced field in both the acoustically bright and dark zone, the DAS method is not considered in this study.

The methods considered in this study are the acoustic contrast control and the pressure matching method.

2.2.1 Acoustic contrast control

The acoustic contrast control (ACC) seeks to maximize the contrast of acoustic energy between bright and dark zones.

The acoustic potential energy density in said zones can be described as [10][11][7]

$$E_B = \frac{1}{\rho_0 c^2} \frac{1}{M_B} \|\mathbf{P}_B\|_2^2 \tag{2.1}$$

for the energy in the bright zone and

$$E_D = \frac{1}{\rho_0 c^2} \frac{1}{M_D} \|\mathbf{P}_D\|_2^2 \tag{2.2}$$

for the energy in the dark zone, where ρ_0 is the density of the medium and c is the velocity of sound in the medium.

Now let AC denote be the acoustic contrast, which is the ratio between the energy in the bright (E_B) and dark zone (E_D), and is often used as a performance measure for the directivity of the array [7]. The AC can be formed as

$$AC = \frac{E_B}{E_D} = \frac{M_D}{M_B} \frac{\|\mathbf{P}_B\|_2^2}{\|\mathbf{P}_D\|_2^2}. \tag{2.3}$$

The acoustic contrast control method, introduced in [10], aims to find the set of filters that maximizes the cost function, $J_{ACC} = AC$. The maximization problem can be written as

$$\text{maximize } J_{\text{ACC}} = \text{maximize} \frac{M_D}{M_B} \frac{\|\mathbf{P}_B\|_2^2}{\|\mathbf{P}_D\|_2^2}. \tag{2.4}$$

As the ACC method seeks to maximize the acoustic contrast between bright and dark zones, it is likely that the cross talk between bright and dark zones are low, which is desirable [12]. On the other hand, the ACC do not consider the phase of the reproduced sound field in the optimization problem, and this may lead to reduced audio quality [12].

2.2.2 Pressure matching

The pressure matching method seeks to compute a set of filters such that the reproduced playback sound field becomes equal to a target sound field [3].

Given a target sound field at discrete frequencies in all measuring points, $\mathbf{P}_r \in \mathbb{C}^{(M_B+M_D)\cdot K\times 1}$, the set of filters that allows for reproduction of the target sound field at the measuring points, can be designed with the pressure matching method [12].

Pressure matching seeks to minimize the complex error, $\mathbf{E}$, between the reproduced sound field in the measuring points, $\mathbf{P} \in \mathbb{C}^{(M_B+M_D)\cdot K\times 1}$, and the target sound field, namely $(\mathbf{P} - \mathbf{P}_r)$ [12]. The minimization can be done in a least squares sense and the cost function for the pressure matching method can be defined as [13]

$$J_{PM} = \sum_{m=1}^{M} |E_m|^2 = \mathbf{E}^H\mathbf{E} \tag{2.5}$$

$$= (\mathbf{P} - \mathbf{P}_r)^H(\mathbf{P} - \mathbf{P}_r). \tag{2.6}$$

The minimization problem is then defined as

$$\text{minimize } J_{\text{PM}} = \text{minimize}(\mathbf{P} - \mathbf{P}_r)^H(\mathbf{P} - \mathbf{P}_r) \tag{2.7}$$

$$= \text{minimize } \|\mathbf{P} - \mathbf{P}_r\|_2^2. \tag{2.8}$$

In [7] it is stated that J_{PM} is convex with respect to the real and imaginary parts of the set of filters in the frequency domain and therefore have a global minimum.

In the pressure matching method both the magnitude and phase of the reproduced field are considered as it tries to match the target sound field [12][14]. The reproduction error when using pressure matching is therefore lower than when using acoustic contrast control [12][14]. However, the separation of sound zones may be lower than with the ACC at the same array effort [12][14]. The array effort is the energy required by the array to control the sound field and is dependent on the playback signal strength and the filter coefficients [6].

It is desired to reproduce the playback signal in the bright sound zones with a low reproduction error even though it may result in more cross talk between sound zones. This is chosen to avoid changing the perceived reproduced signal, such that a user will experience a playback signal as it was intended by the creator of the playback signal. Therefore it is decided to formulate an optimization problem based on the pressure matching method.

2.3 Adaptive filter structure

This section seeks to define a structure for an adaptive filters such that dynamic sound zones are created. The adaptive filter structure will be used as a reference to which sub-elements that must be further studied or designed.

It is desired to calculate a set of filters, one filter for each loudspeaker, which filters a playback signal such that a target pressure is obtained in dynamic measuring locations. If the measuring points are static, it would be possible to calculate closed form filters by forming and solving a convex optimization problem [7]. However, since the measuring points are dynamic, such a method of calculating the filters is undesired due to lack of adaptability, which was mentioned in section 1.1. Instead of calculating the filters as a closed form solution, an iterative method of solving the optimization problem is considered. The filters will thus become adaptive if new iterations of the filters are continuously calculated as new location data of the sound zones becomes available.

In order to solve the pressure matching optimization problem, whether the measuring points are static or dynamic, it is necessary to obtain the pressure contribution from all the loudspeakers in the measuring points. This pressure contribution can be obtained in multiple ways.

One way is to measure the pressure in the measuring points with microphones. In a real life scenario, it would mean that a user is equipped with a microphone which is recording continuously. Such recordings would be likely to contain noise, which would have an impact on the calculated filters. Furthermore, unrelated pressure contributions, such as speech from the user, would also have an effect on the calculation of the filters, if not properly removed from the recording. Removing unrelated contribution from the recording is not necessarily an easy task, but using microphones could capture the reflections of the reproduced sound by objects, if the users are located in a reflective environment. This could be useful when eliminating cross talk between acoustically bright and dark zones.

Another method is to model the pressure contribution from the loudspeakers in the measuring points. Here the model parameters are determined by the dynamic locations of the measuring points. In a real life scenario, this method would only require the user to be equipped with a tracking device which transmits location data, such that the model can be updated. By using a model based method to calculate the pressure contribution, no unrelated contribution is present in the calculated pressure. This means that the resulting filters is not affected by other acoustical noise sources in the environment. The model can however be unreliable if the tracking device does not transmit the accurate location of the user.

For this thesis it is chosen to model the pressure contribution in the measuring points when calculating the filters for the loudspeakers. How the model is created will be further elaborated, but for the purpose of making an adaptive filter structure, the model is only considered conceptually. The model describes the pressure contribution in the measuring points based on the filtered playback signal and the room response of the environment in which the measuring points are located. Since the user is equipped with a tracking device, and may be moving, the measuring points, and therefore the room response, are dynamic. It is therefore assumed that the model is based on the latest user location data.

The model and the pressure matching optimization method is used to create an adaptive filter structure. An illustration of a time domain adaptive filter, when the pressure in the measuring points is modelled, is seen in Figure 2.2. Here both the time domain room response and the reference pressure are dynamically calculated based on the location of the users.

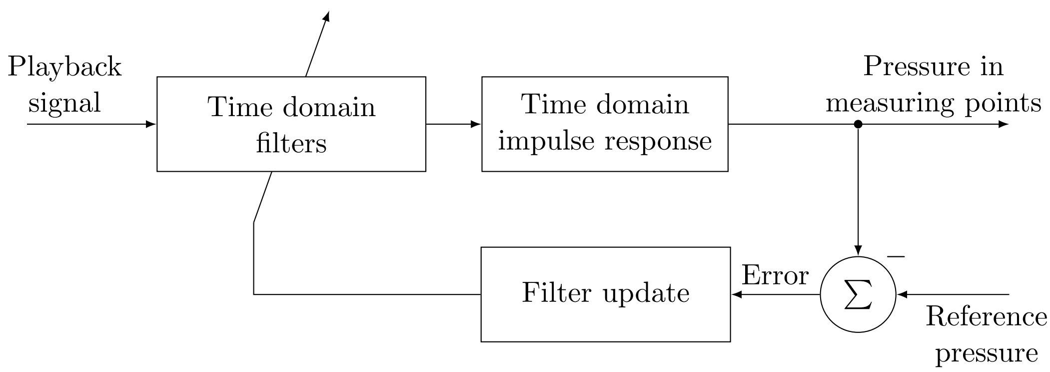


Figure 2.2: Structure of adaptive filters for time domain implementation.

The adaptive filter structure illustrated in Figure 2.2 can be implemented to work on a sample to sample basis, meaning that the filters are updated when a new sample of the playback signal becomes available. Such an implementation is desirable if little latency of the playback signal is desired. The iterative filter update can be made by the use of gradient based optimization. If a cost function of an optimization problem is convex, a global minimum will be in the direction of the negative gradient of the cost function [15]. By formulating a convex cost based on the error from Figure 2.2, which is the error between the modelled pressure and the reference pressure in all measuring points, and using gradient based optimization, the filters will approach the optimal filters even with a dynamic model [15]. The update step of the filters is given by [15]

$$q_{i+1} = q_i - \frac{\mu}{2} g(q_i), \tag{2.9}$$

where i is an iteration counter, q_i are the current time domain filters, $g(\cdot)$ is the gradient of the cost function and μ is a step size.

To calculate the gradient of the cost function, the modelled pressure in the measuring points must be calculated. This calculation requires filtering of the playback signal, which is done by performing a linear convolution between each of the filters and the playback signal. Afterwards another linear convolution with the room impulse response from a loudspeaker to all measuring points is made to calculate the pressure contribution from a single loudspeaker. The contribution from each loudspeaker to all measuring points must then be added together. To calculate the pressure in all measuring points many linear convolutions must be made. Calculating the linear convolution of two signals of length N will require N^2 multiplications [16]. Due to the complexity a more efficient method of convolution is desirable.

Linear convolution via the discrete Fourier transform

Convolution in the time domain corresponds to multiplication in the frequency domain and vice versa [15]. However, the playback signal and the filters are time discrete, and the convolution via the frequency domain will result in circular convolution and not linear convolution due to the underlying periodic nature of the discrete Fourier transform [15]. Under certain circumstances, a circular convolution via the frequency domain can correspond to a linear convolution in the time domain [15][17].

Let u be a discrete time playback signal of length N and let q be a discrete time filter of length K. A linear convolution between u and q will result in a discrete time signal of length $N + K - 1$. By zero-padding the time signal and filter, such that they both have length $N + K - 1$, a circular convolution via the discrete Fourier transform will correspond to a linear convolution [15][17]. Thus the following relation is true [15][17]

$$y = u * q = \text{IDFT}\left(\text{DFT}\left([u\ \mathbf{0}]\right) \odot \text{DFT}\left([q\ \mathbf{0}]\right)\right), \tag{2.10}$$

where $*$ denotes linear convolution, $\odot$ denotes elementwise multiplication, DFT() is the discrete Fourier transform and IDFT() is the inverse discrete Fourier transform.

Using a fast Fourier transform (FFT) to transform $[u\ \mathbf{0}]$ and $[q\ \mathbf{0}]$ can reduce the number of multiplications required for the linear convolution via the frequency domain. A FFT requires $(N + K - 1) \cdot \log_2((N + K - 1))/2$ multiplication when transforming a signal of length $(N + K - 1)$ [16]. The same applies when using a inverse FFT. In the frequency domain, $(N + K - 1)$ complex multiplications must be made to ”convolute” the signals. If this is performed by a computer, where the real and imaginary parts of a complex number are stored in separate registers, it will correspond to $4(N+K-1)$ multiplications. Therefore $(N+K-1)\cdot\log_2((N+K-1))+4(N+K-1)$ multiplications are required for the linear convolution via the frequency domain. If $N = K$, it is more efficient to use the FFT to calculate the linear convolution when $N \geq 18$, compared to the linear convolution in the time domain if only multiplications are considered.

In (2.10) a linear convolution between the playback signal and a single filter was made. For the adaptive filters there will be a filter for each loudspeaker, meaning that the procedure must be repeated. The transformation of u to the frequency domain is however only necessary once per segment since there is only a single playback signal.

2.3.1 Block implementation of the frequency domain adaptive filters

In an online algorithm the whole playback signal is likely to be unknown. This could for example be the case when using an intercommunication system to communicate between management and employees. Here the playback signal will be delivered to the employees as the playback signal is being recorded, making the whole signal unavailable. This causes problems when performing linear convolutions via the frequency domain due to the need of zero padding the whole playback signal, as seen in (2.10).

Therefore a block implementation of the adaptive filters is considered alongside with the overlap-save method. Here the samples of the playback signal will be stored in a buffer, when becoming available, until the buffer is full, and the updating procedure will begin. The filters will therefore be updated on a block by block basis, instead of a sample by sample basis. This consequently introduces a latency based on the length of the buffer, but if an application such as the intercommunication system is considered, the constant latency is not assumed be a problem, since the employee does not have a reference of when the playback signal should be played. The length of the buffer is also dependent on the length of the filters for the loudspeakers. As the necessary length of the filters is unknown for now, the length of the buffer is to remain undetermined until later in the thesis.

In the overlap-save method, the time domain playback signal is partitioned into blocks of N samples that overlap $K-1$ samples with each other [15]. Here K is the length of the time domain filters. The overlapping blocks are denoted as $u(b)$ for $b = 0,1,2,...$. Each block is then convoluted with the filters via the frequency domain and the first $K-1$ samples are ignored [15][17]. The remaining samples corresponds to the filtered block [15][17]. The filtered block is written as

$$y(b) = \text{Last } N-K+1 \text{ samples of IDFT}\left(\text{DFT}(u(b)) \odot \text{DFT}\left([q\ \mathbf{0}]\right)\right). \tag{2.11}$$

The filtered blocks can then be concatenated which will corresponds to filtering of the non-partitioned playback signal.

$$y = [y(0),\ y(1),\ y(2),\ ...]. \tag{2.12}$$

The overlap-save method is illustrated in Figure 2.3.

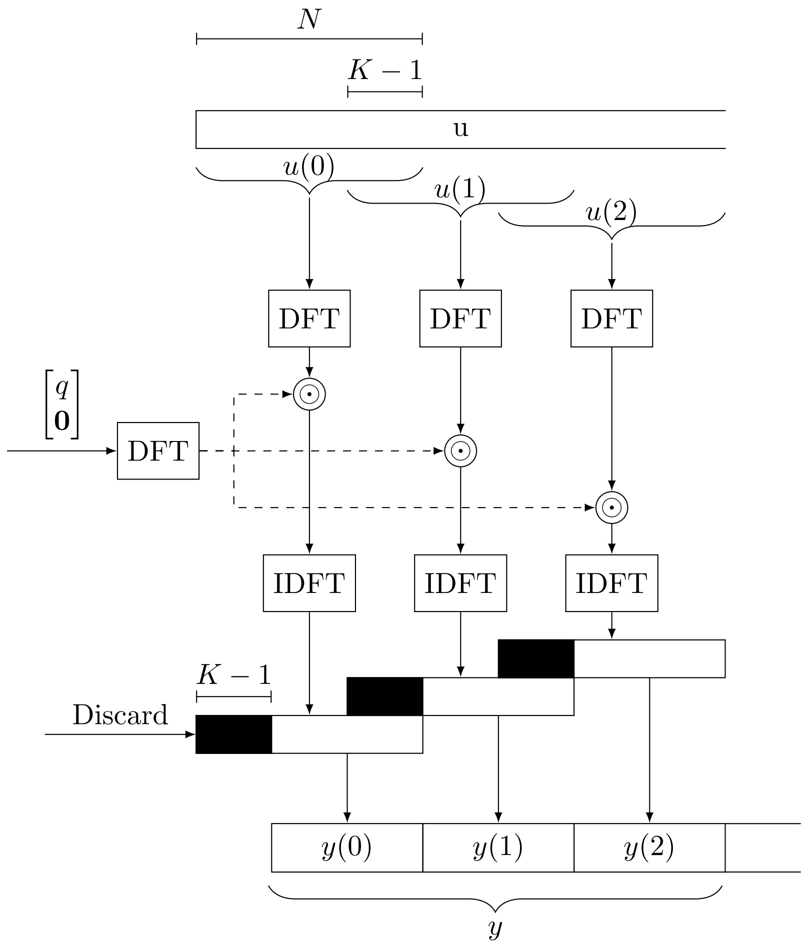


Figure 2.3: Illustration of the overlap-save method. Here ⊙ is the operation of elementwise multiplication.

It is emphasized that (2.11) and (2.12) only describes the linear convolution via the frequency domain with the overlap-save method between the playback signal and a *single* filter. The procedure must be repeated for all filters, and similar operations with the room impulse response must be performed to calculate the pressure in the measuring points. It is however not necessary to transform the time domain filters and the time domain room model to the frequency domain for every new block if the pressure in the measuring points is calculated in the frequency domain. With this procedure the room model can be updated with the latest user location data directly in the frequency domain. This implies that the corrections to the filters also can be made in the frequency domain, which further implies that it is not necessary to transform the time domain filters to the frequency domain for every new block. If the corrections to the filters are made in the frequency domain it is therefore only necessary to transform the newest block of the playback signal to the frequency domain.

To summarize the structure of the frequency domain adaptive filters, newly available samples of the playback signal are stored in a buffer until a whole block is available. Here a block of samples will contain samples from the previous block due to the necessary overlap when using the overlap-save method [15]. The block is then transformed to the frequency domain where it will be multiplied with each of the frequency domain filters, creating a filtered block for each loudspeaker. Each of the filtered blocks and the frequency domain room response is then used to calculate the pressure in each measuring point. The calculated pressure is compared with a reference pressure and their difference is used to make corrections in each of the filters. The correction of the filters are made based on the gradient based optimization, see (2.9), where some number of iterations are calculated in order for the filters to approach their optimal values for the current location of the users. When the corrections to the filters are made, the playback signal must be filtered anew with the newly corrected filters in order to be played by the loudspeakers. The filtering can be made in the time domain as a tap-delay line, which would require to transform the corrected filters to the time domain. It can also be made in the frequency domain by multiplying the frequency domain block of the playback signal with each of the newly corrected filters, and the result transformed back to the time domain. In this thesis, the reconstruction from the frequency domain is chosen due to the additional computational complexity involved in first transforming all the filters to the time domain, and then performing the convolution between the block of the playback signal and each of the filters. The reconstruction from the frequency domain is made in the same manner as (2.11) and (2.12). An illustration of the frequency domain adaptive filter structure and reconstruction of the filtered playback signal can be seen in Figure 2.4.

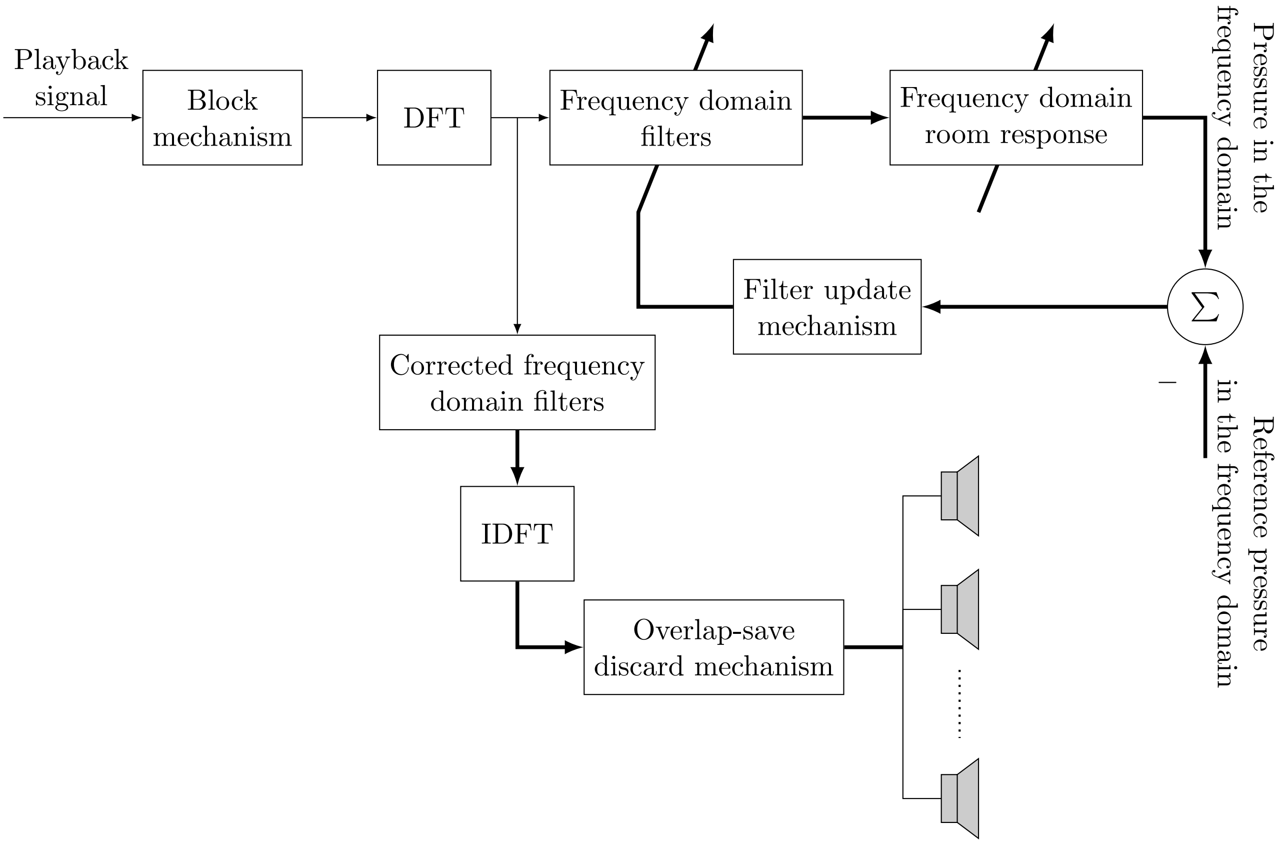


Figure 2.4: Conceptual structure of the frequency domain adaptive filters together with the reconstruction of the filtered playback signal. Here wide lines illustrate that the operation is done for all loudspeakers/measuring points.

Before the "Filter update mechanism" from Figure 2.4 can be designed, it is first and foremost necessary to create the model of the reproduction system describing the pressure in the measuring points, and elaborate how the reference pressure is created.

Modelling the reproduction system 3

In this chapter, a model describing the reproduction system will be derived. The reproduction system consists of a number of loudspeakers in a known constellation. Each loudspeaker is equipped with a controllable filter, which filters the playback signal before being played. It is assumed that all loudspeaker signals are synchronized. The reproduction system furthermore consists of the environment in which the loudspeakers are emitting sound.

3.1 Room acoustics

As the first step in modelling the reproduction system, this section seeks to create an expression describing the reproduction of a signal with a single loudspeaker and a single location in a specified environment.

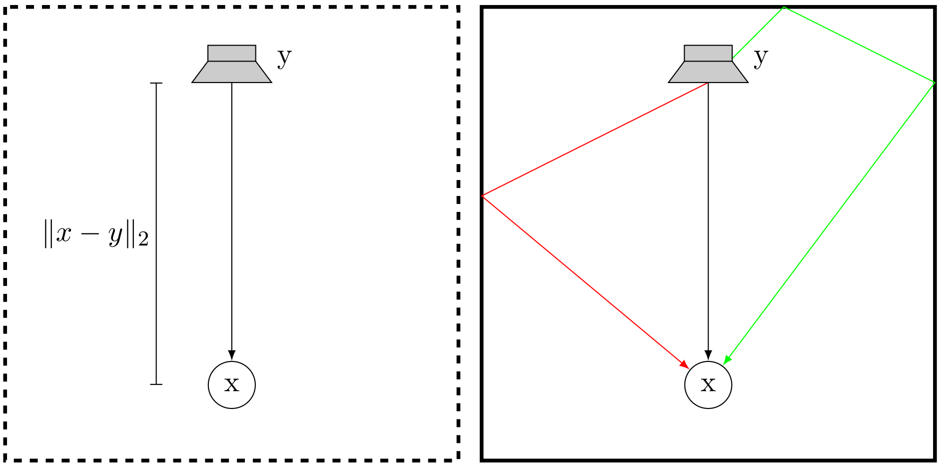
Essentially, a loudspeaker is a transducer converting electrical energy to mechanical energy [18]. When a signal is fed to a loudspeaker, the diaphragm in the loudspeaker is moved according to the signal [18]. This movement of the diaphragm creates a difference in pressure of the fluid in which the loudspeaker is placed, and that pressure difference creates sound waves [19]. The movement of the diaphragm to an input signal is approximately linear, assuming that amplitude and the frequency of the signal is limited [20]. This limit is determined by mechanical constraints of the loudspeaker [20].

In order to create a simple model of the sound radiated from a loudspeaker, far-field conditions must be met. If this condition is met, the radiated sound can be assumed a spherical wave radiated from a point source, and the pressure of the wave follows the inverse distance law [20]. As a rule of thumb, the far-field condition is met if the distance from the source is greater than two wave lengths [21]. The far-field condition is therefore dependent on the frequency of the sound wave.

Furthermore, two methods of modelling the environment are considered. One method is to modelled the environment as an anechoic environment, and the other method is to model it as a reflective environment. An illustration of these two methods is seen in Figure 3.1. In the anechoic case, only the direct path between the source and the measuring point is considered in the model. If the loudspeaker have an omnidirectional radiation pattern, the model is simple to create, since the only variable in the transfer function is the distance.

If the reflections of the sound in the environment is included, the complexity of the model increases. The transfer function describing the relationship between a loudspeaker and a microphone is dependent on the attenuation and delay of all the reflective paths [18]. This transfer function may only be accurate for the specific locations of the loudspeaker and measuring point in the specific environment, since a change of loudspeaker, measuring point or environment could change the reflective paths [18].

Obtaining accurate delays and attenuations in an environment might not be an easy task, making the transfer function hard to obtain. To simplify the transfer function, only the most dominant reflections can be used. Since a reflection and the increased traveling distance of the reflective path attenuates the sound wave [18], reflective paths of higher order can be assumed to contribute little to the pressure in a measuring point. How many reflective paths to include in the model will depend on the environment, since different materials reflects the sound differently [18]. An illustration of a 1^{st} and 2^{nd} order reflection is seen in Figure 3.1(b). Even if few reflections are included in the model, the model parameters are still hard to obtain, especially with a dynamic location of the measuring point. Using this complex model could however result in better separation between sound zones, since disturbance from another sound zone could be caused from reflections by the wall instead of only from the direct path.



(a) Illustration of an anechoic environment.

(b) Illustration of a reflective environment.

Figure 3.1: Illustration of a room with anechoic conditions(a) and reflective conditions(b).

For this thesis it is desired to create adaptive filters that only relies on the location of users relative to the loudspeakers in the reproduction system, thereby creating a general model which can be used in multiple environments. If reflections are included in the model it is necessary to define the positions of walls, objects etc. relative to the loudspeakers, which means that the model is likely to be unique for the environment. This would make the adaptable filters difficult to apply in real life, since a new model must be created for every different environment. It is therefore chosen to assume anechoic conditions and creating a simple model of the environment, even though this might cause poorer acoustical separation between the sound zones due to pressure contribution from reflections. How this choice affects the sound zones will be examined in section 5.5.

Assuming that the loudspeaker have an omnidirectional radiation pattern and that anechoic and far-field conditions are met, an acoustical transfer function from a loudspeaker to a measurement point can be described as a delay and an attenuation [7].

Let y denote the location coordinates of the loudspeaker and x denote the location coordinates of a measurement point. Here Cartesian coordinates are used. The attenuation of the sound pressure caused by the loudspeaker to the measurement point, can be calculated as [7]

$$\alpha = \frac{G}{4\pi\|x - y\|_2}, \tag{3.1}$$

where G is a transformation factor between the signal strength of an input signal to the loudspeaker and the pressure created by the loudspeaker.

The delay of a signal, when played by a loudspeaker, to a measuring point can be calculated as

$$\tau = \frac{\|x - y\|_2}{c}, \tag{3.2}$$

where c is the propagation speed which for this thesis is the speed of sound in air and is assumed to be 343 ms^{-1}.

If the loudspeaker is placed in an anechoic environment, no reflections of the sound radiated from the speaker will be present. Therefore, by combining (3.2) and (3.1) an acoustical transfer function between a loudspeaker and a measuring point for an omnidirectional loudspeaker in anechoic conditions can be described as

$$z(t) = \alpha\delta(t - \tau), \tag{3.3}$$

where t is continuous time and $\delta(\cdot)$ is defined as

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{3.4}$$

Creating the acoustic transfer function in anechoic conditions is rather simple, as the only variable is the distance between the loudspeaker and the measuring point. It is therefore easy to update the transfer function when the measuring point is relocated.

If a signal, denoted as $s(t)$, satisfies a loudspeakers linearity constraints and is played by said loudspeaker, then the pressure contribution in the measuring point, $p(t)$, can be modeled as a delayed and attenuated version of $s(t)$. This is calculated as

$$p(t) = s(t) * z(t), \tag{3.5}$$

where $*$ denotes linear convolution. A graphical interpretation of this model can be seen in Figure 3.2.

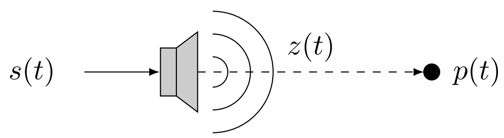


Figure 3.2: Illustration of modeled pressure $p(t)$ of the delayed and attenuated signal $s(t)$.

3.2 Frequency domain model of the reproduction system

In section 3.1 an expression describing the pressure in a given location from a single loudspeaker in an anechoic environment was established, namely (3.5). This section aims to extend the findings from section 3.1 and create a model describing the pressure contribution from multiple loudspeakers to multiple measuring points. The multiple measuring points are necessary in order to include both acoustical bright and acoustical dark points as mentioned in section 2.1.

In an online algorithm the whole playback signal is usually unknown in which case a block implementation of the algorithm can be used, as written in subsection 2.3.1. For the purpose of establishing a model of the reproduction system, a block of the playback signal A is considered. Let b = 0,1,2... denote block time and let $u_A(\mathrm{b})$ be a block of the discrete time playback signal A consisting of N samples. The block is made according to the overlap-save method, also described in subsection 2.3.1. The length of the time domain filters for the loudspeakers are of length K with the relation between the block length of the filter length being $N = 2K - 1$.

Let L denote the total number of loudspeakers used in the reproduction system, and let their spatial coordinates in the plane be denoted as $y_l \in \mathbb{R}^2$ for $l = \{1,2,...,L\}$. Through this thesis a uniform linear loudspeaker array with 20 loudspeakers and 5 cm between the center of the loudspeakers is considered. It is furthermore chosen to use the center of the loudspeaker array as origin of the Cartesian coordinate system. The locations of the loudspeakers are thus defined as

$$y_l = \begin{bmatrix} 0.05 \cdot (l - 10.5) \\ 0 \end{bmatrix} \text{ for } l = \{1,2,...,L\}. \tag{3.6}$$

Now M measuring points are defined in the environment and their spatial coordinates in the plane are denoted as

$$x_m(\mathrm{b}) \in \mathbb{R}^2 \text{ for } m = \{1,2,...,M\}. \tag{3.7}$$

Here the measuring points can be moved according to the locations of the users and the measuring points are therefore a function of the block time.

When $u_A(\mathrm{b})$ has been filtered with each of the L filters and being played by the loudspeakers, the total pressure contribution in a measuring point is calculated as a sum of the pressure contribution from all loudspeakers [1]. The end goal is therefore to formulate an expression of the pressure contribution in all measuring points, when the contribution from all the loudspeakers are summed.

The acoustical transfer function from the l'th loudspeaker to the m'th measuring point is made by transforming (3.3) to the discrete frequency domain and is calculated as

$$Z_{m,l}(\omega,\mathrm{b}) = \frac{G}{4\pi \|x_m(\mathrm{b}) - y_l\|_2} e^{-j\omega\tau_{l,m}(\mathrm{b})}, \tag{3.8}$$

where ω denotes discrete radial frequency, $\tau_{l,m}(\mathrm{b})$ is the propagation delay of sound in air from the l'th loudspeaker to the m'th measuring point at block time b.

Now let $H(\omega,\mathrm{b}) \in \mathbb{C}^{M \times L}$ denote a matrix containing the acoustical transfer function from all loudspeakers to all measuring points for a given frequency

$$H(\omega,\mathrm{b}) = \begin{bmatrix} Z_{1,1}(\omega,\mathrm{b}) & \dots & Z_{1,L}(\omega,\mathrm{b}) \\ \vdots & \ddots & \vdots \\ Z_{M,1}(\omega,\mathrm{b}) & \dots & Z_{M,L}(\omega,\mathrm{b}) \end{bmatrix}. \tag{3.9}$$

For a discrete N point uniform Fourier transform, a transfer function matrix for all frequency bins is denoted as $\mathbf{H}(\mathrm{b}) \in \mathbb{C}^{M\cdot N\times L\cdot N}$ and formed by

$$\mathbf{H}(\mathrm{b}) = \begin{bmatrix} H(\omega_0,\mathrm{b}) & & & \\ & H(\omega_1,\mathrm{b}) & & \Large 0 \\ & & \ddots & \\ \Large 0 & & & H(\omega_{N-1},\mathrm{b}) \end{bmatrix}. \tag{3.10}$$

It is worth noting, that every time new location data of the measuring points becomes available, the transfer function matrix $\mathbf{H}(\mathrm{b})$ must be updated.

Let $U_A(\mathrm{b})$ be the N point discrete Fourier transform of the block of the playback signal, $u_A(\mathrm{b})$, and let $Q_{A,l}(\mathrm{b})$ be the N point discrete Fourier transform of the l^{th} time domain filter associated with playback signal A where $K-1$ zeros have been padded to the end.

The playback signal must be filtered with the filters for all L loudspeakers, which happens in the form of multiplications in the frequency domain. It is desired to form an expression where $U_A(\mathrm{b})$ is elementwise multiplied with $Q_{A,l}(\mathrm{b})$ for all l, in order to calculate the filtered playback signal to be played by all loudspeakers.

Now let $\mathbf{U}_A(\mathrm{b}) \in \mathbb{C}^{L\cdot N\times L\cdot N}$ be an expression for the replication of all frequency components which is defined as

$$\mathbf{U}_A(\mathrm{b}) = \mathrm{diag}\left(U_A(\omega_0,\mathrm{b}),\, U_A(\omega_1,\mathrm{b}),\, ...,\, U_A(\omega_{N-1},\mathrm{b})\right) \otimes I, \tag{3.11}$$

where I is an L-by-L identity matrix and $\otimes$ is the Kronecker product.

For the frequency domain filters, let

$$Q_A(\omega,\mathrm{b}) = [Q_{A,1}(\omega,\mathrm{b}),\, Q_{A,2}(\omega,\mathrm{b}),\, ...,\, Q_{A,L}(\omega,\mathrm{b})], \tag{3.12}$$

and let $\mathbf{Q}_A(\mathrm{b}) \in \mathbb{C}^{L\cdot N\times 1}$ be a vector containing the filter coefficients for all loudspeakers for all frequencies.

$$\mathbf{Q}_A(\mathrm{b}) = [Q_A(\omega_0,\mathrm{b}),\, Q_A(\omega_1,\mathrm{b}),\, ...,\, Q_A(\omega_{N-1},\mathrm{b})]^T. \tag{3.13}$$

The filtered playback signal A for all loudspeakers can be calculated as $\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})$, and the modelled pressure in all measuring points, contributed by all loudspeakers from playback signal A, which is denoted as $\mathbf{P}_A(\mathrm{b}) \in \mathbb{C}^{M\cdot N\times 1}$, is calculated as

$$\mathbf{P}_A(\mathrm{b}) = \mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}). \tag{3.14}$$

The expression in (3.14) thus models the pressure in all measuring points in the frequency domain under anechoic and far-field conditions when a block of the playback signal A is filtered and played by all loudspeakers. An illustration of the model is shown in Figure 3.3. The model will be used when optimizing the filters $\mathbf{Q}_A(\mathrm{b})$ such that the reference pressure is obtained.

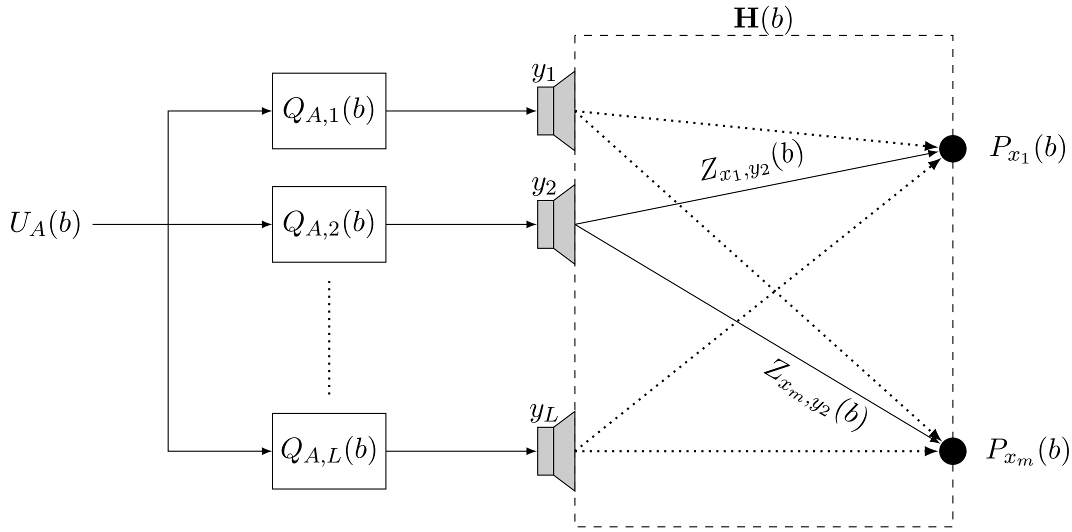


Figure 3.3: Illustration of (3.14), with shown transfer functions from loudspeaker y_2 to the measuring points x_1 and x_m.

In (3.14) the pressure contribution from the filtered playback signal A is modelled. When modelling the pressure contribution from the filtered playback signal B, a similar expression is formed. Their difference is that playback signal B and its associated filters are used instead of $\mathbf{U}_A$(b) and $\mathbf{Q}_A$(b).

3.2.1 Model of reproduction system with signal compensation

The pressure model in (3.14) models the pressure contributed from playback signal A independently of playback signal B. The filters associated with playback signal A should therefore ideally remove all pressure contribution from playback signal A in measuring points belonging to sound zone B. If the two playback signals contains some of the same frequency components removing these should not be necessary since these frequency components should be available in both sound zones. By including both filtered playback signals when calculating the pressure contribution, it might be possible to compensate for contributions from the filtered playback signal B when calculating the filters associated with playback signal A. This compensation might lower the array effort while maintaining the same level of reproduction error compared to the model from (3.14).

In the frequency domain, the b'th block of playback signal B and its associated filters are denoted as $\mathbf{U}_B$(b) and $\mathbf{Q}_B$(b) respectively, and they are defined in the same manner as with playback signal A and its associated filters, which is seen in (3.11) and (3.13). The pressure contribution in the measuring points contributed from both playback signals is calculated as

$$\mathbf{P}_C(\mathrm{b}) = \mathbf{H}(\mathrm{b})\Big(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})\Big). \tag{3.15}$$

Compared to the expression in (3.14), the expression in (3.15) can be used when making corrections to the filters associated with both playback signal A and B. It will be tested in section 5.2 if there is any benefit in compensating for the filtered playback signal B when correcting the filters associated with playback signal A and vice versa. An illustration of the pressure contribution in the measuring point for two playback signals, can be seen in Figure 3.4.

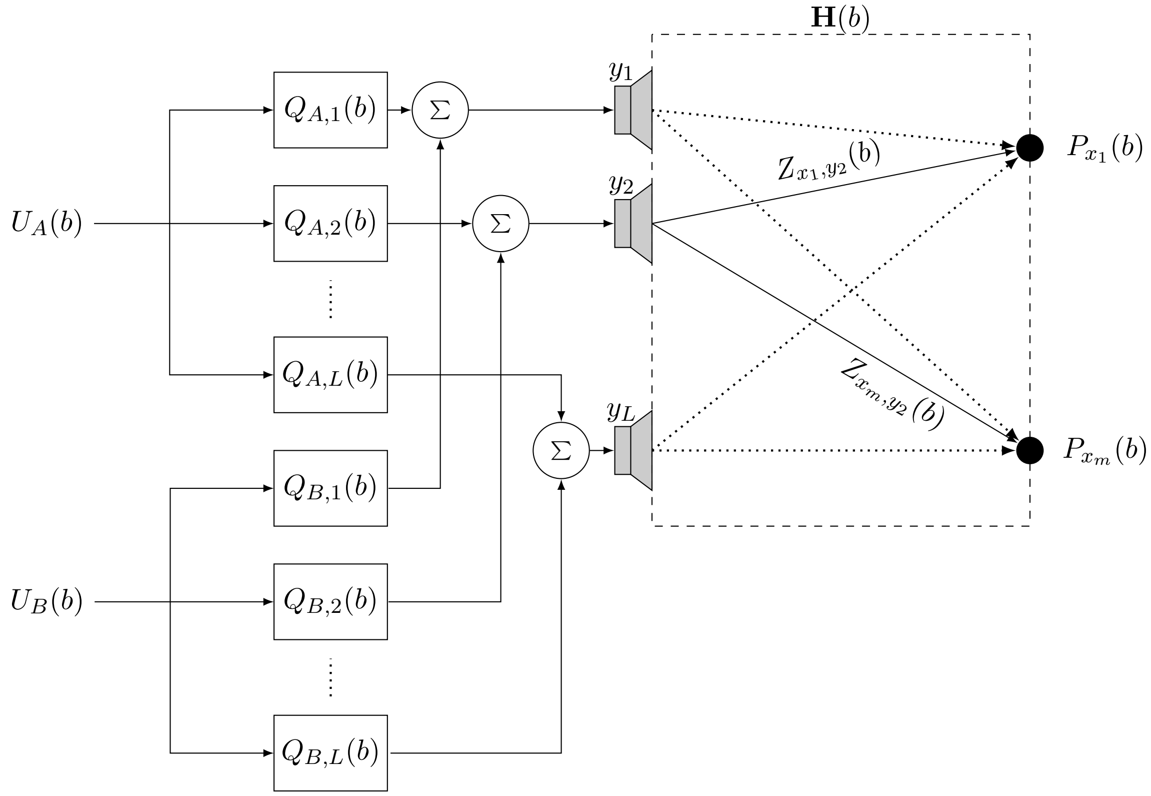


Figure 3.4: Illustration of (3.15), with shown transfer functions from loudspeaker y_2 to the measuring points x_1 and x_m.

3.3 Formulation of reference pressure

As written in section 2.2 it is decided to formulate an optimization problem based on pressure matching. This method evaluates the filters ability to create the target signal, or reference pressure, in the measuring points. This section aims to formulate said reference pressure.

3.3.1 Constellation of measuring points

The reference pressure describes an intended or target pressure in all M measuring points, which makes it dependent on the location of the measuring points. In section 2.1 it was chosen to define two sound zones, and it is desired to control the reproduced sound field in these two zones. How the sound field behaves outside of the two sound zones is of no interest, and a measuring point will therefore either belong to sound zone A or sound zone B. As to not indirectly weight the importance of the reproduced sound field when defining the optimization problem, $M/2$ measuring points are allocated to sound zone A, and $M/2$ measuring points are allocated to sound zone B.

It is now necessary to define the constellation of the measuring points inside a sound zone. As the location of the sound zones are based on the latest location data of the two users, so must the measuring points be, such that the measuring points follows the user. It is therefore decided to define a measuring point in the coordinates of the latest user location data. The rest $M/2-1$ points belonging to a sound zone are then uniformly placed on the circumference of a circle with center in the coordinate of the latest user location data and a radius of 25 cm. The choice of the radius is based on the choice of a single user inside each sound zone. There is therefore no benefit in controlling the sound field in a too large area. If the radius is increased it will also be easier for the two sound zones to overlap if the users are located close to each other. An illustration of the placement of the measuring points in sound zone A and B for $M = 22$ is seen in Figure 3.5.

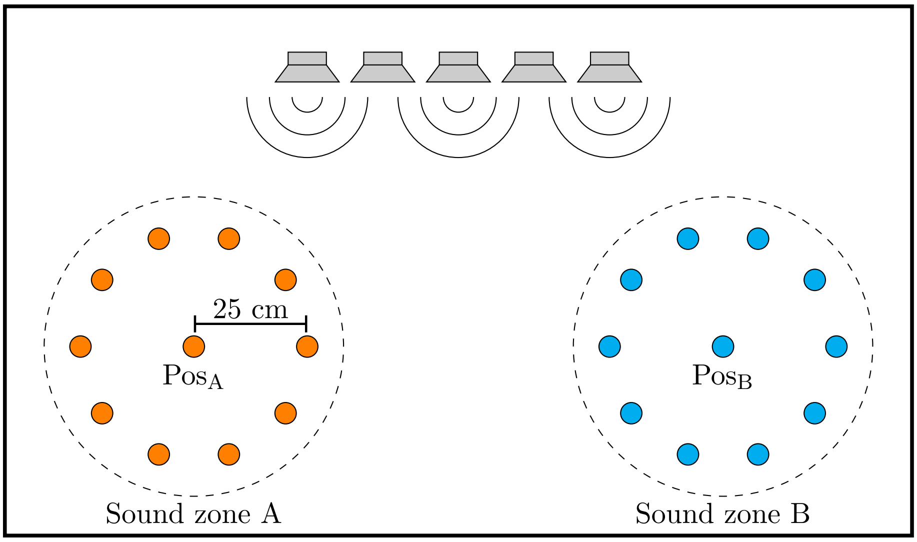


Figure 3.5: Choice of measuring points in the two sound zones based on the location data of user A and B with $M = 22$. Here bright measuring points associated with playback signal A and B are illustrated as ■ and ■ respectively.

3.3.2 Reference pressure without signal compensation

This section will formulate the reference pressure for the signal model without signal compensation, which is written in (3.14). In section 2.1 the approach for creating the two sound zones was, that for playback signal A, sound zone A must be acoustically bright and sound zone B must be acoustically dark. The opposite applies for playback signal B. Based on the chosen placement of the measuring points written above, a measuring point is therefore either acoustically bright or dark.

Ideally the target signal in an acoustically bright measuring point is the playback signal with no alterations. This is however an optimistic goal because moving the measuring point further away from the loudspeakers would require for the filters to increase the gain of the signal fed to the loudspeakers. There will thus be a point where the loudspeakers becomes nonlinear [20] or simply unable to play loud enough. The reference pressure is therefore attenuated according to the inverse distance law, based on the distance from the loudspeaker array. The location of the loudspeaker array was specified in (3.6), and the center of this array is used as a global reference point for calculating the distance to a measuring point when creating the reference pressure.

There will furthermore exists a propagation delay from the global reference point to the measuring points. By combining the attenuation and the delay, the acoustical transfer function from (3.8) is again obtained, but with minor changes since the delay and attenuation is calculated based on the distance to the global reference point and not the loudspeakers. This is written as

$$Z_{r,m}(\omega,\mathrm{b}) = \frac{1}{4\pi\|x_m(\mathrm{b})\|_2} e^{-j\omega\|x_m(\mathrm{b})\|_2/343}. \tag{3.16}$$

Thus the reference pressure at a given frequency becomes an attenuated and delayed version of the playback signal if a measuring point is acoustically bright. If a measuring point is acoustically dark, then there will ideally be no pressure contribution from the loudspeakers in said measuring point. The reference pressure associated with playback signal A is therefore defined as

$$P_{r,A}(\omega,\mathrm{b}) \in \mathbb{C}^{M\times 1} = \begin{cases} U_A(\omega,\mathrm{b})Z_{r,m}(\omega,\mathrm{b}) & \text{if } x_m(\mathrm{b}) \text{ is a bright point} \\ 0 & \text{if } x_m(\mathrm{b}) \text{ is a dark point} \end{cases}, \text{ for } m = \{1,2,...,M\} \quad (3.17)$$

for a given angular frequency of the b'th block of playback signal A.

The reference pressure for all N angular frequencies are now concatenated such that there is compliance between the modelled pressure, described in (3.14), and the reference pressure associated with playback signal A which is denoted as $\mathbf{P}_{r,A}(\mathrm{b}) \in \mathbb{C}^{M\cdot N\times 1}$. The reference pressure is structured as

$$\mathbf{P}_{r,A}(\mathrm{b}) = [P_{r,A}(\omega_0,\mathrm{b})^T\ , P_{r,A}(\omega_1,\mathrm{b})^T\ ,..., P_{r,A}(\omega_{N-1},\mathrm{b})^T]^T. \quad (3.18)$$

Just as with the acoustical transfer function, $\mathbf{H}(\mathrm{b})$ from (3.10), the reference pressure must be updated when new location data of the measuring points becomes available. It is furthermore necessary to update the reference pressure when a new block of the playback signal becomes available.

The reference pressure for playback signal A is illustrated in Figure 3.6(a). Creating the reference pressure associated with playback signal B, denoted as $\mathbf{P}_{r,B}(\mathrm{b})$, follows the same procedure as for playback signal A, but the bright measuring points are now dark and vice versa. This is illustrated in Figure 3.6(b). When playback signal A and B has been filtered by their associated filters, the two filtered signals must be added together and be played by the loudspeakers. The sum of the reference pressure associated with playback signal A and B should therefore be completely separated with respect to which playback signal is available in the measuring points. This is illustrated in Figure 3.6(c), and will be the case since no measuring point can belong to both zone A and B by definition.

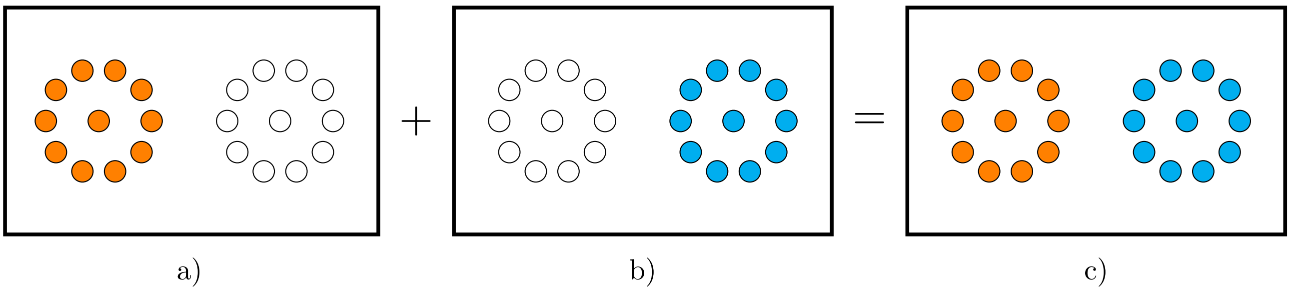


Figure 3.6: a) Illustration of reference pressure associated with playback signal A, b) Illustration of reference pressure associated with playback signal B, c) Sum of the two reference pressures. The white circles illustrate acoustically dark measuring points.

3.3.3 Reference pressure with signal compensation

The reference pressure for the signal model with signal compensation, described in (3.15), is formulated a little different than the reference pressure without signal compensation. The placement of the measuring points and whether a measuring point belongs to sound zone A or B is identical to the reference pressure without signal compensation. So is the method of defining the target signal in a measuring point for the two playback signals.

Since both playback signals are present in (3.15), which models the pressure contribution in all measuring points regardless of them belonging to zone A or B, it will also be necessary to include both playback signals in the reference pressure. The sound zones must still be separated, meaning that playback signal A only is available in the measuring points belonging to zone A and vice versa. This is illustrated in Figure 3.6(c). For a given angular frequency and block of the two playback signals, the reference pressure in all measuring points is defined as

$$P_{r,C}(\omega,\mathrm{b}) \in \mathbb{C}^{M\times 1} = \begin{cases} U_A(\omega,\mathrm{b})Z_{r,m}(\omega,\mathrm{b}) & \text{if } x_m(\mathrm{b}) \text{ belongs to zone A} \\ U_B(\omega,\mathrm{b})Z_{r,m}(\omega,\mathrm{b}) & \text{if } x_m(\mathrm{b}) \text{ belongs to zone B} \end{cases}, \text{ for } m = \{1,2,...,M\}. \quad (3.19)$$

The reference pressure for the signal model with signal compensation for all N angular frequencies in all M measuring points is now denoted as $\mathbf{P}_{r,C}(\mathrm{b}) \in \mathbb{C}^{M\cdot N\times 1}$ and defined as

$$\mathbf{P}_{r,C}(\mathrm{b}) = [P_{r,C}(\omega_0,\mathrm{b})^T,\ P_{r,C}(\omega_1,\mathrm{b})^T,\ ...\ P_{r,C}(\omega_{N-1},\mathrm{b})^T]^T. \quad (3.20)$$

Thus a reference pressure for the signal model without signal compensation has been defined both for playback signal A and B, which makes it possible to define an optimization problem without signal compensation. Similarly a reference pressure has been created for the signal model with signal compensation, and it is therefore possible to define two different optimization problems.

3.4 Practical limitation regarding the frequency range of the playback signals

Before the definition of the two optimization problems, and how they are used to create an adaptive filter algorithm, it is however necessary to address some practical limitations.

So far in the modelling of the reproduction system, the playback signal has been assumed to fulfill the Nyquist sampling frequency criterion, as to not be aliased.

The modelling of the playback signal in the measuring points might suffer from spatial aliasing if the playback signal is not further limited in its frequency range [22][23]. For the purpose of reproducing a sound field in the measuring points, spatial aliasing can cause coloration to the perceived signal [24]. The spatial aliasing could cause the user to hear ”echos” of the playback signal [24], which is desired to avoid.

The upper frequency limit before spatial aliasing occurs is determined by the angle between the loudspeaker array and a measuring point and the formation of the loudspeakers in the array [22].

As written in the beginning of section 3.2, a uniform linear loudspeaker array with 20 loudspeakers and 5 cm between the center of each loudspeaker is used. Here the angle perpendicular to the array is defined as 0 degrees, and a measuring point can be located in the interval $[-90°; 90°]$ as illustrated in Figure 3.7.

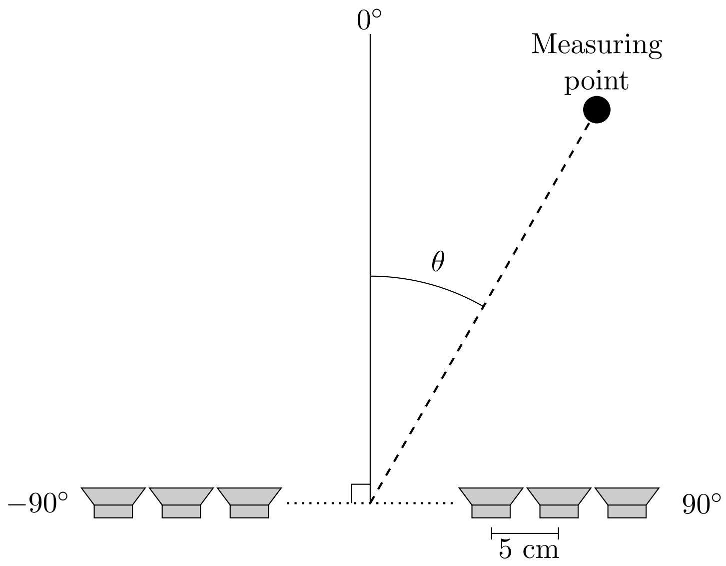


Figure 3.7: Illustration of angle between a measuring point and the linear loudspeaker array.

For this configuration of the loudspeaker array, the upper frequency limit before spatial aliasing can be calculated as [22]

$$f_{upper}(\theta) = \frac{343}{0.05(1 + |\cos(\theta)|)}. \tag{3.21}$$

As seen in (3.21) the upper frequency limit before spatial aliasing is dependent on the angle, θ, meaning that the upper frequency limit dependents on the location of the measuring point. In Figure 3.8 the upper frequency limit before aliasing as a function of θ is illustrated.

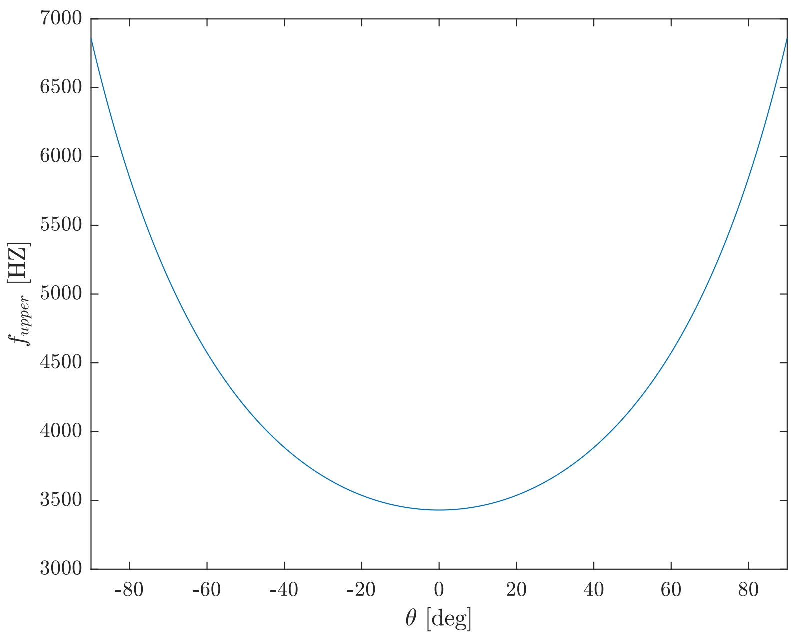


Figure 3.8: Illustration of the upper frequency limit before spatial aliasing as a function of θ.

As seen in Figure 3.8, to avoid spatial aliasing independently of the location of the measuring points, the playback signal must not contain frequencies above 3430 Hz.

In order to fulfill the assumption of far-field conditions, which was mentioned in section 3.1, a measuring point must be located at least two wave lengths from the loudspeaker. The wave length of a sound wave is calculated as

$$\lambda = \frac{343}{f}, \tag{3.22}$$

where f is the frequency of the sound wave. The inverse relation between the wave length and the frequency would require that a measuring point should be located far from the loudspeaker if the playback signal contains low frequencies. Therefore the playback signal is limited such that it do not contain frequencies below 350 Hz. The far-field condition is then fulfilled if the distance between the loudspeakers and a measuring point is greater than two meters.

In a real life scenario the physical limitation of the loudspeakers will also have an influence on the reproducible frequencies of the playback signal [25]. Here the size of the diaphragm of a loudspeaker puts a lower limit on the reproducible frequencies [25]. Similarly, a loudspeaker tends to become more directional when producing high frequencies [25]. These limitations regarding the frequency range unfortunately limits the applicable scenarios if the adaptive sound zones is applied as a stand alone solution, since few people are assumed to be willing to listen to music with a frequency range between 350-3430 Hz. Fortunately the hospital example mentioned in chapter 1, where messages are delivered to hospital personnel over an intercommunication system, still seems possible due to the limited frequency range of speech [26].

3.5 Choice of filter length

Another consideration to make, is regarding the length of the filters, which was denoted as N for the frequency domain filters. The variable, N, determines the frequency resolution of the filtering, but also the amount of calculations required to update the filters. A high frequency resolution is desired since it allows for much finer adjustment of the playback signal, which could result in more accurate reproduction of the playback signal in an acoustically bright zone, or higher separation between acoustically bright and dark zones.

If the modeling of the pressure in the measuring points included reflection, i.e approximated a room response, the length of the time domain filters should be at least as long as the room impulse response in order to negate the pressure contribution of the reflections. Since the model does not include reflections the minimum requirement of the filter length becomes more lax. The room impulse response in this case, is only the direct path from a loudspeaker to a measuring point. The length of the filter therefore depends on the greatest distance between a loudspeaker and a measuring point. Furthermore the filter length will depend on the sampling frequency of the playback signal.

If the playback signal is limited such that spatial aliasing does not occur, it is in principle possible to use a sampling frequency of $2 \cdot 3430 = 6860$ Hz. This is however an odd choice of sampling frequency, which is not likely to be supported in many sound cards, which is why a sampling frequency of 8000 Hz is chosen. With a sampling frequency of 8000 Hz, and requiring that the time domain filter length is minimum as long as the length of the room impulse response, the minimum time domain filter length can be calculated as

$$K_{min} = \frac{d}{c} fs, \tag{3.23}$$

where d is the distance from a loudspeaker to a measuring point, $c = 343$ is the speed of sound in air and $fs = 8000$ is the sampling frequency. For a time domain filter length of 256, a measuring point must be no further than 10.98 meters from any loudspeaker in the loudspeaker array. This is deemed as an adequate maximum distance for simulation purposes in this thesis. With the relation between the length of the time domain filters and the frequency domain filters being $N = 2K - 1$, and a sampling frequency of 8000 Hz, the frequency resolution becomes $\Delta f = 8000/511 = 15.66$ Hz.

3.6 Choice of window function

So far a rectangular window has been assumed when transforming the playback signal to the frequency domain. The window length is here equal to the block size, N. Using a rectangular window can however result in the windowed playback signal not being periodic, which in turn can add high frequency material to its spectrum [17]. When looking at the chosen frequency range of the playback signal it is relatively close to half of the chosen sampling frequency. The added frequency material from the rectangular windowing may therefore be present in the operating frequency range. It is therefore chosen to use a Hann window function instead of the rectangular window since it has great dampening of side lobes close to half the sampling frequency, and a relatively narrow main lobe [17].

The Hann window function is calculated as [17]

$$w(n) = \sin^2\left(\frac{\pi n}{N-1}\right),, \tag{3.24}$$

where $0 \leq n \leq N - 1$. Using the Hann window consequently means that it is not possible to reconstruct the playback signal, since the amplitude of some time indexes of the window being zero. It is therefore necessary to transform the playback signal with the rectangular window for the purpose of reconstruction of the filtered playback signal, and transform the playback signal with the Hann window for the purpose of correction of the filters. The later was previously denoted as $\mathbf{U}_A(\text{b})$ for playback signal A, and it will now be redefined.

Let $w \in \mathbb{R}^{N\times 1}$ be a vector containing the Hann window and let its individual elements be calculated according to (3.24). Before transforming the current block of the playback signal, $u_A(\text{b})$, to the frequency domain, its elements are multiplied with the elements of w. Now

$$U_A(\text{b}) = \text{DFT}(w \odot u_A(\text{b})), \tag{3.25}$$

where $\odot$ denoted elementwise multiplication and DFT is an N point uniform discrete Fourier transform. As before

$$\mathbf{U}_A(\text{b}) = \text{diag}\left(U_A(\omega_0,\text{b}),\, U_A(\omega_1,\text{b}),\, ...,\, U_A(\omega_{N-1},\text{b})\right) \otimes I, \tag{3.26}$$

where I is an L-by-L identity matrix and $\otimes$ is the Kronecker product, which is done such that the filtered playback signal A for all loudspeakers can be calculated as $\mathbf{U}_A(\text{b})\mathbf{Q}_A(\text{b})$.

The Hann window is also used in the same manner on playback signal B.

This concludes the modelling of the reproduction system. A model describing the pressure contribution in the measuring points when a block of the playback signal is filtered and played by the loudspeakers have been formulated, alongside with a reference pressure for the same block. It is therefore possible to formulate an optimization problem and design an adaptive filter algorithm such that dynamic sound zones are created.

Design of the adaptive filters 4

Now that an expression for the pressure in all measuring points and the reference pressure has been determined, it is possible to design the update procedure of the filters in an adaptive filter algorithm. This will first be made for the signal model without signal compensation described in (3.14), with its associated reference pressure from (3.18), and afterwards for the signal model with signal compensation described in (3.15) and its reference pressure from (3.20).

To help the reader understand the variable names and their relation to each other, the adaptive filter structure without signal compensation is illustrated in Figure 4.1 with variable names. The figure illustrates the filter structure with respect to both playback signal A and B to give the reader an understanding of the work flow from two playback signals as input to a single signal to be played by the loudspeakers. This chapter will focus on formulating the ”Filter update mechanism” from Figure 4.1.

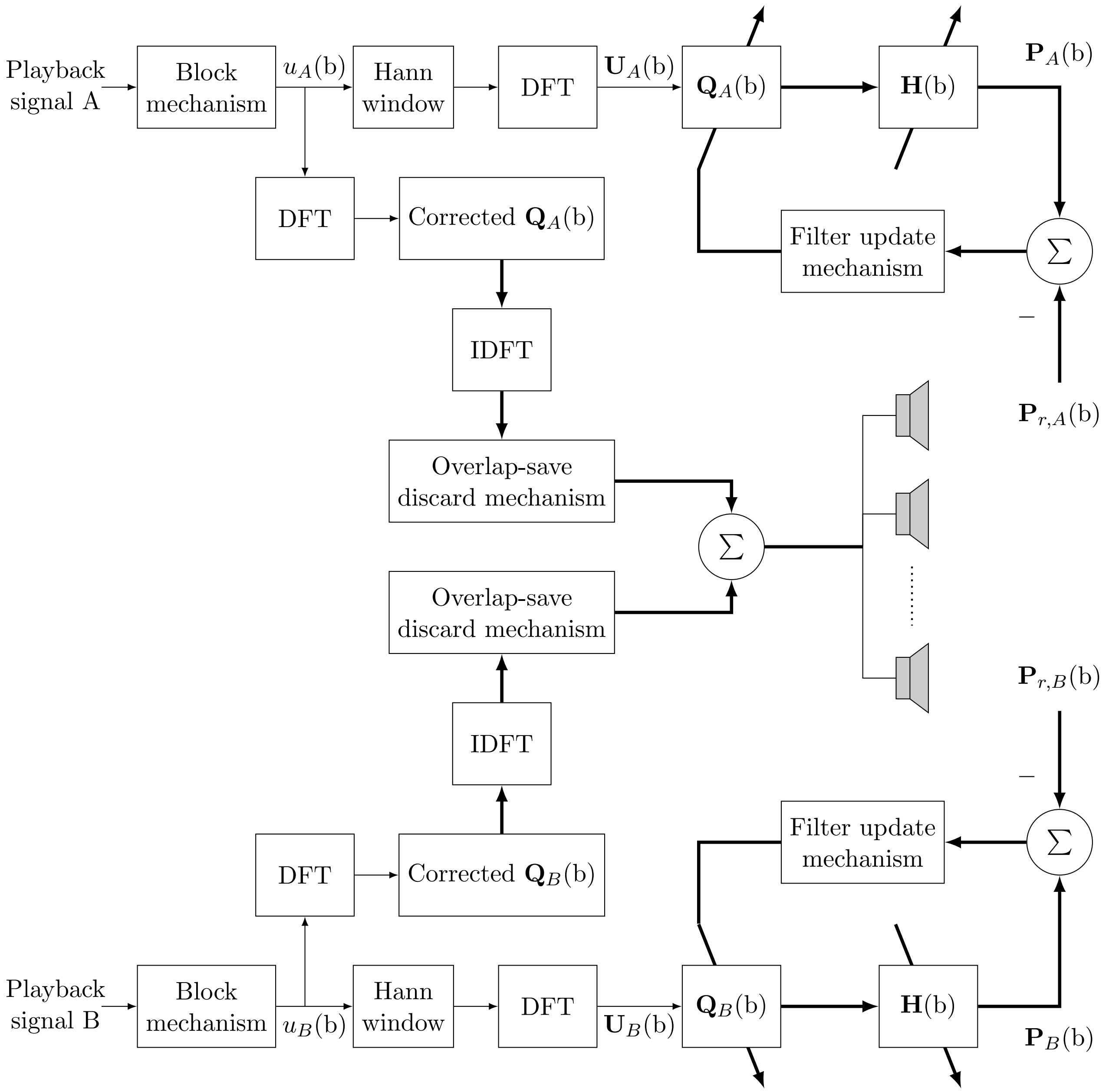


Figure 4.1: Adaptive filter structure with variable names for two playback signals as input, to an output signal to the loudspeaker array.

4.1 Adaptive filter algorithm without signal compensation

This section aims to create an adaptive filter algorithm based on the signal model without signal compensation from (3.14).

The first step in creating the adaptive filter algorithm, is to formulate the optimization problem. It is here chosen only to show the derivations for the filters associated with playback signal A, namely $\mathbf{Q}_A(\text{b})$, since the derivation for $\mathbf{Q}_B(\text{b})$ is nearly identical. The difference is which filters and which playback signal is used when modeling the pressure in the measuring points, and how the reference pressure is created.

Defining the optimization problem requires the modelled pressure from (3.14) and the reference pressure from (3.18), where both are calculated based on playback signal A and its associated filters. It is furthermore decided to introduce regularization with regards to the input power of the filtered playback signal to the loudspeaker array. The cost function is now defined as

$$\begin{aligned}
J_A(\mathrm{b}) &= \|\mathbf{P}_A(\mathrm{b}) - \mathbf{P}_{r,A}(\mathrm{b})\|_2^2 + \beta\|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\|_2^2 && (4.1)\\
&= \mathbf{P}_A^H(\mathrm{b})\mathbf{P}_A(\mathrm{b}) - 2\mathbf{P}_A^H(\mathrm{b})\mathbf{P}_{r,A}(\mathrm{b}) + \mathbf{P}_{r,A}^H(\mathrm{b})\mathbf{P}_{r,A}(\mathrm{b}) + \beta\|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\|_2^2 && (4.2)\\
&= \mathbf{Q}_A^H(\mathrm{b})\mathbf{U}_A^H(\mathrm{b})\mathbf{H}^H(\mathrm{b})\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) - 2\mathbf{Q}_A^H(\mathrm{b})\mathbf{U}_A^H(\mathrm{b})\mathbf{H}^H(\mathrm{b})\mathbf{P}_{r,A}(\mathrm{b})\\
&\quad + \mathbf{P}_{r,A}^H(\mathrm{b})\mathbf{P}_{r,A}(\mathrm{b}) + \beta\mathbf{Q}_A^H(\mathrm{b})\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) && (4.3)\\
&= \mathbf{Q}_A^H(\mathrm{b})\mathbf{R}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) - 2\mathbf{Q}_A^H(\mathrm{b})\mathbf{r}_A(\mathrm{b}) + \mathbf{P}_{r,A}^H(\mathrm{b})\mathbf{P}_{r,A}(\mathrm{b})\\
&\quad + \beta\mathbf{Q}_A^H(\mathrm{b})\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}), && (4.4)
\end{aligned}$$

where $\mathbf{R}_A(\mathrm{b})$ is the correlations matrix of $\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})$, and $\mathbf{r}_A(\mathrm{b})$ is the cross-correlation vector between $\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})$ and $\mathbf{P}_{r,A}(\mathrm{b})$, and $\beta \geq 0$ is the regularization parameter.

If the cost function $J_A(\mathrm{b})$ is convex, then a global minimum can be found by using gradient based optimization methods [27]. To simplify things $\beta = 0$, such that the regularization term becomes zero. For the cost to be convex the hessian of the cost function must be positive semi-definite [27]. The hessian of $J_A(\mathrm{b})$ w.r.t $\mathbf{Q}_A(\mathrm{b})$ is calculated as

$$\frac{\partial^2 J_A(\mathrm{b})}{\partial^2\ \mathbf{Q}_A(\mathrm{b})} = 2\mathbf{R}_A(\mathrm{b}) = 2\mathbf{U}_A^H(\mathrm{b})\mathbf{H}^H(\mathrm{b})\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b}). \tag{4.5}$$

The hessian is positive semi-definite if and only if [27]

$$z^H\Big(\mathbf{U}_A^H(\mathrm{b})\mathbf{H}^H(\mathrm{b})\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})\Big)z \geq 0, \tag{4.6}$$

where z is some vector. A matrix is positive semi-definite if and only if its eigenvalues are non negative [27]. To show that the hessian contains only non negative eigenvalues, a singular value decomposition (SVD) of $\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})$ is made [28]

$$\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b}) = U\Sigma V^H, \tag{4.7}$$

where U and V^H are unitary and Σ contains real singular values [28]. By using the SVD the hessian can be calculated as

$$\begin{aligned}
\mathbf{U}_A^H(\mathrm{b})\mathbf{H}^H(\mathrm{b})\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b}) &= V\Sigma^T U^H U\Sigma V^H && (4.8)\\
&= V\Sigma^T\Sigma V^H. && (4.9)
\end{aligned}$$

As the singular values are the square root of the non negative eigenvalues, and since the singular values are real [28], the term $\Sigma^T\Sigma$ contains only non negative eigenvalues. The hessian is therefore positive semi-definite, and the cost function is therefore convex. Applying the same procedure on the regularization term, $\beta\|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\|_2^2$ for $\beta > 0$, will show that this term is also convex. As the sum of two convex functions is also a convex function [27], the cost function from (4.1) is convex.

Since the cost function is convex, the global minimum is in the direction of the negative gradient of the cost [27]. The derivative of the cost function w.r.t $\mathbf{Q}_A(\mathrm{b})$ is therefore computed, such that the gradient can be used to correct the filters.

$$\begin{aligned}
\nabla J_A(\mathrm{b}) &= 2\mathbf{R}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) - 2\mathbf{r}_A(\mathrm{b}) + 2\beta\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) && (4.10)\\
&= 2\Big(\mathbf{R}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) - \mathbf{r}_A(\mathrm{b}) + \beta\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\Big). && (4.11)
\end{aligned}$$

Note that the correlation, $\mathbf{R}_A(\mathrm{b})$, and cross-correlation, $\mathbf{r}_A(\mathrm{b})$, from (4.11), in this case are circular correlations, due to the periodic nature of the discrete frequency domain [15]. As a consequence, the N frequency domain filter weights do not corresponds to the $K = N/2$ time domain weights [15]. To have equivalence between the time- and frequency domain, a linear correlation and linear cross-correlation must be used instead of a circular [15]. The gradient is therefore constrained such that the circular correlation and cross-correlation becomes equivalent to the linear correlation and cross-correlation. At the same time a notation to increase the readability is introduced.

$$\sigma(\mathbf{X}) = \mathrm{FFT}\left(\begin{bmatrix}\text{first K elements of IFFT}(\mathbf{X}) \\ \mathbf{0}\end{bmatrix}\right), \tag{4.12}$$

where $\mathbf{X} \in \mathbb{C}^{L \cdot N \times 1}$ is a vector of the same shape as the gradient from (4.11), and FFT() and IFFT() calculates the discrete- and inverse discrete Fourier transform on all frequency bins belonging to the individual loudspeakers. Since 20 loudspeakers are used, 20 FFTs and 20 IFFTs are calculated when constraining the gradient.

The constrained gradient is written as

$$\mathbf{\Phi}_A(\mathrm{b}) = \sigma(\nabla J_A(\mathrm{b})). \tag{4.13}$$

Using the unconstrained gradient could result in faster converges, but it will result in a larger misadjustement [15]. It will be necessary to calculate twice as many iterations for the unconstrained gradient to obtain the same level of misadjustment as the constrained gradient [15].

If the equivalence between the time- and frequency domain filters is ignored, it is now examined if constraining the gradient is more computational efficient than not constraining it, if a specific level of misadjustment is required. Later, in (4.15) to (4.18) it will be shown how the filter update step can be calculated with only matrix-vector products. If a specific misadjustment is desired, every iteration for the constrained gradient requires two iterations with the unconstrained gradient [15]. Assuming that all matrices are of size N by N and all vectors are of size N, using the unconstrained gradient will result in $4N^2$ [29] extra multiplications per iteration compared to the constrained gradient. Constraining the gradient requires 20 FFTs and 20 IFFTs, which each requires $N/2 \cdot \log_2(N)$ multiplications [30]. The constraining of the gradient therefore requires $20N\log_2(N)$ multiplications per iteration. For $N > 22$ it will be more efficient to constraint the gradient. For this application, $N = 512$ is the number of frequency bins used, so even without including the number of loudspeakers and measuring points, constraining the gradient is more computational efficient.

The filters can now be updated iteratively according to

$$\mathbf{Q}_A(\mathrm{b},i+1) = \mathbf{Q}_A(\mathrm{b},i) - \frac{\mu}{2}\mathbf{\Phi}_A(\mathrm{b},i), \tag{4.14}$$

where $i = 0,1,...$ is an iteration counter and $\mu > 0$ is a step size. For the filters to converge, the step size can not be chosen arbitrarily large, and the requirements for convergence regarding the step size will be discussed in subsection 4.1.1. First it is however desired to rearrange the order of calculations when calculating the gradient such that products of two matrices can be avoided.

The gradient decent filter update step can also be written in an alternative form, with the following relations.

$$\mathbf{Y}_A(\mathrm{b},i) = \mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b},i) \tag{4.15}$$

$$\mathbf{E}_A(\mathrm{b},i) = \mathbf{Y}_A(\mathrm{b},i) - \mathbf{P}_{r,A}(\mathrm{b}) \tag{4.16}$$

$$\mathbf{\Phi}_A(\mathrm{b},i) = \sigma\Big(\mathbf{U}_A^H(\mathrm{b})\Big(\mathbf{H}^H(\mathrm{b})\mathbf{E}_A(\mathrm{b},i) + \beta\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b},i)\Big)\Big) \tag{4.17}$$

$$\mathbf{Q}_A(\mathrm{b},i+1) = \mathbf{Q}_A(\mathrm{b},i) - \mu\mathbf{\Phi}_A(\mathrm{b},i). \tag{4.18}$$

Here $\mathbf{Y}_A(\mathrm{b},i)$ is the modelled pressure in the M measuring points, and $\mathbf{E}_A(\mathrm{b},i)$ is the error between the modelled pressure and the reference pressure. By rearranging it is possible to calculate (4.15) to (4.18) with only matrix-vector products instead of matrix-matrix products, which reduces the computational complexity.

4.1.1 Convergence criterion of the adaptive filter algorithm without signal compensation

The value of the step size, $\mu > 0$, determines how quickly the filters converges to their optimal values, and how numerically close to closed form solution the filters can come [15]. Here a large step size results in few iteration until convergence, but the numerical values of the filters can also be further from the closed form filters [15]. This is due to the negative gradient ”pointing” in the direction of the minimum and the step size determining how far to ”walk” in the direction of the negative gradient. When close to the minimum, a large step size can result in ”missing” the minimum, and calculating more iterations can thus result in continuing to miss the minimum without improving the solution, given that the step size is constant. If a constant step size is used it is likely that there will be difference between the filters found iteratively and the filters found as a closed form solution [15]. The difference is also known as the excessive cost. Furthermore, if the step size is too large, the algorithm will diverge [15]. This imposes a requirement upon the step size since the algorithm must be stable.

The requirements for μ to ensure convergence will now be shown. To simplify things, the unconstrained gradient is used and rewritten as

$$\nabla J_A(\mathrm{b}) = 2\Big((\mathbf{R}_A(\mathrm{b}) + \beta\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b}))\mathbf{Q}_A(\mathrm{b}) - \mathbf{r}_A(\mathrm{b})\Big) \tag{4.19}$$

$$= 2\Big(\Upsilon\mathbf{Q}_A(\mathrm{b}) - \upsilon\Big), \tag{4.20}$$

where Υ is the hessian of the cost function. The filter update step can thus be written as

$$\mathbf{Q}_A(\mathrm{b},i+1) = \mathbf{Q}_A(\mathrm{b},i) - \mu\Big(\Upsilon\mathbf{Q}_A(\mathrm{b},i) - \upsilon\Big) \tag{4.21}$$

$$= (I - \mu\Upsilon)\mathbf{Q}_A(\mathrm{b},i) + \mu\upsilon, \tag{4.22}$$

where I is an identity matrix. Previously it was shown that the cost function is convex and the hessian is positive semi-definite.

Υ is hermitian and can be unitarily diagonalized [28], such that $\Upsilon = \Gamma\Lambda\Gamma^H$, where Γ is unitary and Λ contains the non negative eigenvalues of the hessian in the diagonal. The update step can now be written as

$$\mathbf{Q}_A(\mathrm{b},i+1) = (I - \mu\Gamma\Lambda\Gamma^H)\mathbf{Q}_A(\mathrm{b},i) + \mu\upsilon. \tag{4.23}$$

Since $\Gamma\Gamma^H = I$, the update step can be written as

$$\mathbf{Q}_A(\mathrm{b},i+1) = \Gamma(I - \mu\Lambda)\Gamma^H\mathbf{Q}_A(\mathrm{b},i) + \mu\upsilon. \tag{4.24}$$

Calculating the filters is an iterative process, and by expressing $\mathbf{Q}_A(\text{b},i)$ in terms of $\mathbf{Q}_A(\text{b},i-1)$ the following is obtained.

$$\mathbf{Q}_A(\text{b},i+1) = \Gamma(I-\mu\Lambda)\Gamma^H\Big(\Gamma(I-\mu\Lambda)\Gamma^H\mathbf{Q}_A(\text{b},i-1)+\mu\upsilon\Big)+\mu\upsilon \tag{4.25}$$

$$= \Gamma(I-\mu\Lambda)(I-\mu\Lambda)\Gamma^H\mathbf{Q}_A(\text{b},i-1)+\Gamma(I-\mu\Lambda)\Gamma^H\mu\upsilon+\mu\upsilon. \tag{4.26}$$

If such an expansion is made until $\mathbf{Q}_A(\text{b},i+1)$ is expressed in terms of $\mathbf{Q}_A(\text{b},0)$, the update step will include the expression

$$(I-\mu\Lambda)^n \tag{4.27}$$

where n is the number of iterations. In order for (4.18) to converge, (4.27) must converge as n increases. This will require that

$$-I < (I-\mu\Lambda) < I. \tag{4.28}$$

As the eigenvalues in Λ are non-negative, it is necessary that $|\mu|$ is smaller than the inverse of the largest eigenvalue, such that scaling all the eigenvalues yields a positive number less than one. Therefore

$$-1 < (1-\mu\lambda_{max}) < 1, \tag{4.29}$$

where λ_{max} is the largest eigenvalue. Solving for μ gives the requirement for the step size in order for the filters to converge. This requirement is that

$$0 < \mu < \frac{2}{\lambda_{max}}. \tag{4.30}$$

4.1.2 Improvements to the convergence rate

As seen in (4.30) the value of the step size, and therefore the convergence rate, is proportional to the inverse of the largest eigenvalue of the hessian of the cost function. If there is a large spread of the eigenvalues it will consequently mean that the convergence in the direction with small eigenvalues will be slow [15]. To improve the convergence rate, it is therefore desired to normalize the gradient and to decrease the spread of the eigenvalues.

It is possible to improve the convergence rate by computing a step size for each individual frequency bin, where the step sizes are adapted individual from each other based on the power in the individual frequency bins [15].

A step size for each individual frequency bin can be defined as

$$\mu_k = \frac{\alpha}{P_{A,k}}, \quad k = 0,1,\dots,N-1, \tag{4.31}$$

where $P_{A,k}$ is an estimate of the average power in the k^{th} frequency bin of playback signal A and $\alpha > 0$ is a constant that scales the step sizes.

However when the algorithm operates in a non stationary environment, or if an estimate of the average input power in each frequency bin is not available, it is possible to estimate it by the use of a recursion [15]. In an online algorithm the whole playback signal will not be available, and it is therefore not possible to calculate the average power in the frequency bins. The recursion is based on the idea of convex combination and defined as [15]

$$P_{A,k}(\text{b}) = \gamma P_{A,k}(\text{b}-1) + (1-\gamma)\left|U_{A,k}(\text{b})\right|^2, \quad k = 0,1,\dots,\text{N}-1, \tag{4.32}$$

where $P_{A,k}(\text{b})$ is the estimated input power in the k^{th} frequency bin at time block b. $U_{A,k}(\text{b})$ is the k^{th} frequency bin of playback signal A. The parameter γ is a constant, chosen in the range $0 \leq \gamma \leq 1$, that controls the effective "memory" in the recursion. γ is also referred to as the forgetting factor, as the choice of γ determines how much of the previous estimated power are forgotten or kept.

Now let $\mathbf{D}_A(\text{b}) \in \mathbb{R}^{L \cdot N \times L \cdot N}$ be an expression for the replication of the power estimate for all loudspeakers expressed for all frequencies.

$$\mathbf{D}_A(\text{b}) = \text{diag}((P_{A,0} + \epsilon)^{-1}(\text{b}), (P_{A,1} + \epsilon)^{-1}(\text{b}), \ldots, (P_{A,N-1} + \epsilon)^{-1}(\text{b})) \otimes I, \tag{4.33}$$

where $\epsilon > 0$ is a small number in case $P_k(\text{b}) = 0$ and $\otimes$ is the Kronecker product.

By the calculated estimate of the input power, individual step sizes for all frequency bins for all filters can be calculated. The new step size are defined as

$$\boldsymbol{\mu}(\text{b}) = \alpha \mathbf{D}(\text{b}). \tag{4.34}$$

When applying these changes to the algorithm, the constrained gradient defined in (4.17) becomes

$$\mathbf{\Phi}_{A,imp}(\text{b},i) = \sigma\Big(\mathbf{D}_A(\text{b})\mathbf{U}_A^H(\text{b})\Big(\mathbf{H}^H(\text{b})\mathbf{E}_A(\text{b},i) + \beta\mathbf{U}_A(\text{b})\mathbf{Q}_A(\text{b},i)\Big)\Big). \tag{4.35}$$

As a linear correlation and cross-correlation is still desired when applying the inverse powers in each individual frequency bin, $\mathbf{D}_A(\text{b})$ is applied in the gradient constraint. Furthermore the old step size parameter μ have been replaced by α, which scales the new individual step size for all bins accordingly. By normalizing the gradient with the inverse input power estimate, the contribution from $\mathbf{U}_A(\text{b})$ to the eigenvalues of the hessian has been diminished, which can result in the spread of the eigenvalues of the normalized hessian to also be diminished due to the individual adjustment. The scaling parameter, α, must still be chosen such that the algorithm converges, and its maximum value is determined by the inverse of the greatest eigenvalue of the now normalized hessian, but the eigenvalues are now smaller due to the normalization. The filter update step from (4.18), is now modified as follows

$$\mathbf{Q}_A(\text{b},i+1) = \mathbf{Q}_A(\text{b},i) - \alpha\mathbf{\Phi}_{A,imp}(\text{b},i). \tag{4.36}$$

Making an expression for updating the filters associated with playback signal B follows the same procedure for playback signal A, and will therefore not be shown. The update steps are however included in algorithm 1.

4.1.3 Algorithm with no signal compensation

In algorithm 1, pseudo code of the adaptive filter algorithm with no signal compensation is seen. To emphasise that the derivations so far only have been for the filters associated with playback signal A, the pseudo code for computing the filters associated with playback signal B is also included.

Initialization:
$\mathbf{Q}_A(0,0)$ = LN by 1 zero vector;
$\mathbf{Q}_B(0,0)$ = LN by 1 zero vector;
$\alpha > 0$;
$\beta \geq 0$;
Computation:
for $b = 1$ **to** *last block* **do**
  Calculate $\mathbf{U}_A(\text{b})$, as written in (3.26), with latest block of playback signal A;
  Calculate $\mathbf{U}_B(\text{b})$, as written in (3.26), with latest block of playback signal B;
  Compute desired response in zone A, $\mathbf{P}_{r,A}(\text{b})$, with $\mathbf{U}_A(\text{b})$, as written in (3.18);
  Compute desired response in zone B, $\mathbf{P}_{r,B}(\text{b})$, with $\mathbf{U}_B(\text{b})$, as written in (3.18);
  Update the inverse power estimate, $\mathbf{D}_A(\text{b})$, with $\mathbf{U}_A(\text{b})$, as written in (4.33);
  Update the inverse power estimate, $\mathbf{D}_B(\text{b})$, with $\mathbf{U}_B(\text{b})$, as written in (4.33);
  for $i = 0$ **to** *iterations per block* **do**
    $\mathbf{Y}_A(\text{b},i) = \mathbf{H}(\text{b})\mathbf{U}_A(\text{b})\mathbf{Q}_A(\text{b},i)$;
    $\mathbf{Y}_B(\text{b},i) = \mathbf{H}(\text{b})\mathbf{U}_B(\text{b})\mathbf{Q}_B(\text{b},i)$;
    $\mathbf{E}_A(\text{b},i) = \mathbf{Y}_A(\text{b},i) - \mathbf{P}_{r,A}(\text{b})$;
    $\mathbf{E}_B(\text{b},i) = \mathbf{Y}_B(\text{b},i) - \mathbf{P}_{r,B}(\text{b})$;
    $\mathbf{\Phi}_A(\text{b},i) = \sigma\Big(\mathbf{D}_A(\text{b})\mathbf{U}_A^H(\text{b})\Big(\mathbf{H}^H(\text{b})\mathbf{E}_A(\text{b},i) + \beta\mathbf{U}_A(\text{b})\mathbf{Q}_A(\text{b},i)\Big)\Big)$;
    $\mathbf{\Phi}_B(\text{b},i) = \sigma\Big(\mathbf{D}_B(\text{b})\mathbf{U}_B^H(\text{b})\Big(\mathbf{H}^H(\text{b})\mathbf{E}_B(\text{b},i) + \beta\mathbf{U}_B(\text{b})\mathbf{Q}_B(\text{b},i)\Big)\Big)$;
    $\mathbf{Q}_A(\text{b},i+1) = \mathbf{Q}_A(\text{b},i) - \alpha\mathbf{\Phi}_A(\text{b},i)$;
    $\mathbf{Q}_B(\text{b},i+1) = \mathbf{Q}_B(\text{b},i) - \alpha\mathbf{\Phi}_B(\text{b},i)$;
  end
  $\mathbf{Q}_A(\text{b}+1,0) = \mathbf{Q}_A(\text{b},i)$;
  $\mathbf{Q}_B(\text{b}+1,0) = \mathbf{Q}_B(\text{b},i)$;
end

Algorithm 1: Pseudo code for the adaptive filter algorithm with no signal compensation.

4.1.4 Closed form expressing for signal model without signal compensation

For comparison reasons, an expression for calculating the filters as a closed form solution will now be derived.

As it was found that the pressure matching cost function with no signal compensation is convex, the optimal filters can be computed with the closed form expression of the cost. Using the gradient (4.11), set it equal to zero and isolating the filters $\mathbf{Q}_A(\text{b})$, the closed form filters for playback signal A, $\mathbf{Q}_{A,cf}(\text{b})$, are found and are defined as

$$\mathbf{Q}_{A,cf}(\text{b}) = (\mathbf{U}_A(\text{b})^H\mathbf{H}(\text{b})^H\mathbf{H}(\text{b})\mathbf{U}_A(\text{b}) + \beta\mathbf{U}_A^H(\text{b})\mathbf{U}_A(\text{b}))^{-1}\mathbf{U}_A(\text{b})^H\mathbf{H}(\text{b})^H\mathbf{P}_{r,A}(\text{b}). \tag{4.37}$$

Note that the closed form filters are a function of the block time, b, which means that the closed form filters must be computed for every block if the parameters changes, since the closed from filters no longer is guaranteed to be optimal.

Furthermore it is seen that the closed form expression uses a matrix inversion, which means that the matrix to be inverted must be non singular [28]. This means that the hessian of the cost function must be positive definite, instead of just positive semi-definite. The hessian is hermitian and will be positive definite if it has full rank [28]. For the hessian to have full rank both $\mathbf{H}(\mathrm{b})$ and $\mathbf{U}_A(\mathrm{b})$ must have full rank [28]. This will as a minimum require that $M \geq L$, where M is the number of measuring points and L is the number of loudspeakers, and that the diagonal of $\mathbf{U}_A(\mathrm{b})$ is not zero.

4.2 Adaptive filter algorithm with signal compensation

This section aims to create an adaptive filter algorithm based on the pressure model were both playback signals are included from (3.15) and the reference pressure from (3.20). The explanation of the various steps involved in creating the filter update step is shortened compared to section 4.1, due to the procedure of creating the filter update step being nearly identical.

The cost function of the pressure matching optimization problem with signal compensation and regularization with regards to the filtered playback signals can now be defined as

$$\begin{aligned} J_C(\mathrm{b}) &= \|\mathbf{P}_C(\mathrm{b}) - \mathbf{P}_{r,C}(\mathrm{b})\|_2^2 + \beta\|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\|_2^2 + \beta\|\mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})\|_2^2 \qquad (4.38)\\ &= \|\mathbf{H}(\mathrm{b})(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})) - \mathbf{P}_{r,C}(\mathrm{b})\|_2^2 \\ &\quad + \beta\|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})\|_2^2 + \beta\|\mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})\|_2^2. \qquad (4.39)\end{aligned}$$

It is chosen to update $\mathbf{Q}_A(\mathrm{b})$ while assuming that $\mathbf{Q}_B(\mathrm{b})$ is a constant and vice versa. By doing so, the hessian with respect to only $\mathbf{Q}_A(\mathrm{b})$ is calculated as

$$\frac{\partial^2 J_C(\mathrm{b})}{\partial^2 \mathbf{Q}_A(\mathrm{b})} = 2\mathbf{U}_A(\mathrm{b})^H\mathbf{H}(\mathrm{b})^H\mathbf{H}(\mathrm{b})\mathbf{U}_A(\mathrm{b}) + 2\beta\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b}). \qquad (4.40)$$

It was shown that the cost function without signal compensation from (4.1) is convex. Since the hessian of (4.1) is identical to the hessian in (4.40) when assuming $\mathbf{Q}_B(\mathrm{b})$ is constant, then the cost function in (4.39) is also convex. Finding the hessian of (4.39) with respect to $\mathbf{Q}_B(\mathrm{b})$ while assuming $\mathbf{Q}_A(\mathrm{b})$ is constant will show that (4.39) is still convex.

The gradient of (4.39) with respect to $\mathbf{Q}_A(\mathrm{b})$ is calculated as

$$\begin{aligned}\nabla_{\mathbf{Q}_A(\mathrm{b})} J_C(\mathrm{b}) &= 2\mathbf{U}_A(\mathrm{b})^H\mathbf{H}(\mathrm{b})^H\Big(\mathbf{H}(\mathrm{b})\Big(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})\Big) - \mathbf{P}_{r,C}(\mathrm{b})\Big) \\ &\quad + 2\beta\mathbf{U}_A^H(\mathrm{b})\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b})). \qquad (4.41)\end{aligned}$$

As before, to improve the rate of convergence, the gradient in (4.41) is normalize with respect to an estimate of the input power of the block of the playback signal. Let $P_{A,k}(\mathrm{b})$ be the power estimate of the k'th frequency bin of playback signal A and let it be calculated as

$$P_{A,k}(\mathrm{b}) = \gamma P_{A,k}(\mathrm{b}-1) + (1-\gamma)|U_{A,k}(\mathrm{b})|^2, \qquad (4.42)$$

where $0 \leq \gamma \leq 1$, and $U_{A,k}(\mathrm{b})$ is the k'th frequency bin of the b'th block of playback signal A. Now let

$$\mathbf{D}_A(\mathrm{b}) = \mathrm{diag}((P_{A,0}+\epsilon)^{-1}(\mathrm{b}), (P_{A,1}+\epsilon)^{-1}(\mathrm{b}), \ldots, (P_{A,N-1}+\epsilon)^{-1}(\mathrm{b})) \otimes I, \qquad (4.43)$$

where $\epsilon > 0$ is a small number in case $P_{A,k}(\mathrm{b}) = 0$ and I is an L-by-L identity matrix.

As before, when no signal compensation was used, the gradient is constrained such the circular correlation and cross-correlation corresponds to linear correlation and cross-correlation. Furthermore the constrained gradient is normalized by the input power estimate. The constrained normalized gradient is calculated as

$$\mathbf{\Phi}_A(\mathrm{b}) = \sigma(\mathbf{D}_A(\mathrm{b})\nabla_{Q_A(\mathrm{b})} J_C(\mathrm{b})). \tag{4.44}$$

The filter update steps becomes

$$\mathbf{Q}_A(\mathrm{b},i+1) = \mathbf{Q}_A(\mathrm{b},i) - \frac{\alpha}{2}\mathbf{\Phi}_A(\mathrm{b},i), \tag{4.45}$$

where $\alpha > 0$ is the step size whose maximum values still is determined by the inverse of the largest eigenvalue of the normalized hessian, with respect to $\mathbf{Q}_A(\mathrm{b})$, of $J_C(\mathrm{b})$.

Making an update step for $\mathbf{Q}_B(\mathrm{b})$ will not be shown since the procedure is the same as for $\mathbf{Q}_A(\mathrm{b})$. The difference is which variable the gradient is calculated with respect to. The update will however be shown in algorithm 2.

4.2.1 Algorithm with signal compensation

Now the algorithm for the adaptive filter with signal compensation is written as pseudo code, which is seen in algorithm 2. Here the calculations of the gradient from (4.41) is rearranged such that only matrix-vector products are needed instead of matrix-matrix products.

Initialization:
$\mathbf{Q}_A(0{,}0)$ = LN by 1 zero vector;
$\mathbf{Q}_B(0{,}0)$ = LN by 1 zero vector;
$\alpha > 0$;
$\beta \geq 0$;
Computation:
for $b = 1$ **to** *last block* **do**
  Calculate $\mathbf{U}_A(\mathrm{b})$, as written in (3.26), with latest block of playback signal A;
  Calculate $\mathbf{U}_B(\mathrm{b})$, as written in (3.26), with latest block of playback signal B;
  Compute desired response, $\mathbf{P}_{r,C}(\mathrm{b})$, as written in (3.20);
  Update the inverse power estimate, $\mathbf{D}_A(\mathrm{b})$, with $\mathbf{U}_A(\mathrm{b})$ as written in (4.43);
  Update the inverse power estimate, $\mathbf{D}_B(\mathrm{b})$, with $\mathbf{U}_B(\mathrm{b})$ as written in (4.43);
  for $i = 0$ **to** *iterations per blocks* **do**
    $\mathbf{Y}_C(\mathrm{b},i) = \mathbf{H}(\mathrm{b})(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b},i) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b},i))$;
    $\mathbf{E}_C(\mathrm{b},i) = \mathbf{Y}_C(\mathrm{b},i) - \mathbf{P}_{r,C}(\mathrm{b})$;
    $\mathbf{\Phi}_A(\mathrm{b},i) = \sigma\Big(\mathbf{D}_A(\mathrm{b})\mathbf{U}_A^H(\mathrm{b})\Big(\mathbf{H}^H(\mathrm{b})\mathbf{E}_C(\mathrm{b},i) + \beta\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b},i)\Big)\Big)$;
    $\mathbf{\Phi}_B(\mathrm{b},i) = \sigma\Big(\mathbf{D}_B(\mathrm{b})\mathbf{U}_B^H(\mathrm{b})\Big(\mathbf{H}^H(\mathrm{b})\mathbf{E}_C(\mathrm{b},i) + \beta\mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b},i)\Big)\Big)$;
    $\mathbf{Q}_A(\mathrm{b},i+1) = \mathbf{Q}_A(\mathrm{b},i) - \alpha\mathbf{\Phi}_A(\mathrm{b},i)$;
    $\mathbf{Q}_B(\mathrm{b},i+1) = \mathbf{Q}_B(\mathrm{b},i) - \alpha\mathbf{\Phi}_B(\mathrm{b},i)$;
  end
  $\mathbf{Q}_A(\mathrm{b}+1{,}0) = \mathbf{Q}_A(\mathrm{b},i)$;
  $\mathbf{Q}_B(\mathrm{b}+1{,}0) = \mathbf{Q}_B(\mathrm{b},i)$;
end

Algorithm 2: Pseudo code for adaptive filter algorithm with signal compensation.

This concludes the design of the adaptive filter algorithms and it is now possible to test how they perform with dynamic playback signals and sound zones.

Test and comparison of adaptive algorithms 5

This chapter will focus on testing the adaptive filter algorithms. First the effect of the number of iterations per block will be tested on the adaptive filter algorithm without signal compensation. Second the two adaptive filter algorithms will be compared to each other, where their reproduced sound field and array effort will be compared. This is followed by an estimate of the necessary memory and computational complexity of using closed form filters compared to adaptive filters when creating the two sound zones. Lastly the influence of the anechoic assumptions of the room model will be tested by comparing simulations with real measurements in a room.

In all of the following tests the measuring points will be defined according to the description in subsection 3.3.1, where Pos_A defines the center of sound zone A and Pos_B defines the center of sound zone B. Since the adaptive filters will be compared to the closed form filters, both solutions must have equally many measuring points to make it a fair comparison. The closed form filters require that there are at least as many measuring points as there are loudspeakers, as described in subsection 4.1.4, and therefore $M = 22$ measuring points will be used in all tests.

5.1 Effect of the number of iterations per block

So far the stopping criterion for correction of the filters for a specific block has not been discussed. Either the correction for the block can be stopped when the difference in cost between two subsequent iterations of the filters becomes less than some threshold, or a fixed number of iterations per block can be calculated.

In the context of an online algorithm, there will exists a time limit for when the filtered playback signal must be available. Due to a fixed time limit, it will therefore make sense to calculate a fixed number of iterations per block, and this section aims to evaluate the effect on the solution of how many iterations are calculated.

For this test, algorithm 1 is used where it is chosen that sound zone A is bright and sound zone B is dark. This means that $\mathbf{U}_B(\mathrm{b}) = 0\ \forall\ \mathrm{b}$. Now $\mathbf{U}_A(\mathrm{b})\ \forall\ \mathrm{b}$ is assumed to be an $L \cdot N \times L \cdot N$ identity matrix, and the measuring points are defined according to the description in subsection 3.3.1 with $Pos_A = [2.4,\ 3.3]^T$ and $Pos_B = [-2.4,\ 3.3]^T$. The regularization parameter is set to $\beta = 0.001$, the step size is defined as $\alpha = 2$ and the forgetting factor is set to $\gamma = 0.5$. The filters are initialized as 0, but "warm start" is used every block. For every block the cost, $J_A(b)$ from (4.1) is computed. Now let

$$\Delta J_A(b) = J_A(b-1) - J_A(b) \tag{5.1}$$

be the difference between the cost of the current and previous block. The adaptive filter algorithm is stopped when $\Delta J(b) < 0.01$. The number of blocks calculated to reach the threshold for different number of iterations per block is listed in Table 5.1. This is listed along with the corresponding run time of the algorithm in real life, with an output of 256 samples per block and a sampling frequency of 8000 Hz.

Iteration per block	**Blocks until $\Delta J < 0.01$**	**Seconds until $\Delta J < 0.01$**
1	182	5.82
2	119	3.81
5	66	2.11
10	40	1.28

Table 5.1: The number of computed blocks for different number of iterations per block until $\Delta J_A(b) < 0.01$.

It is no surprise that calculating more iterations per block results in faster convergence. What needs to be considered, is what additional iterations per block would mean for a user who has to listen to the reproduced sound field. Therefore the closed form filters from (4.37) is used as comparison. The closed form filters are calculated with the same configuration as the adaptive filters, and the cost when using the closed form filters is denoted as J_{cf}. The difference in cost between the closed form filters and the adaptive filters is denoted as $J_{ex}(\text{b}) = J_A(\text{b}) - J_{cf}$, and is now calculated for the adaptive filters with one iteration per block after the 182 blocks. This gives

$$J_{ex}(182) = J(182) - J_{cf} = 1.03 \tag{5.2}$$

which is not considered to be much since the setup contains 22 measuring points, 20 loudspeakers and the length of the frequency domain filters being 512.

The test results from Table 5.1 is however for a static playback signal. In an online algorithm the change in $\mathbf{U}_A(\text{b})$ from block to block is likely to have an influence on the necessary number of iterations per block to obtain a steady convergence.

The effect of different numbers of iterations per block with dynamic playback signals is now tested. The test specifications are identical to previously, except that the current block of the playback signal is used to calculate $\mathbf{U}_A(\text{b})$ for the adaptive filters and the cost of both the adaptive filters and the closed form filters. For each block the excessive cost, $J_{ex}(\text{b})$ is calculated and saved. The results are illustrated in Figure 5.1 and Figure 5.2, where the playback signals are a selected time interval of "Serve the Servant" by Nirvana and "You Spin Me Round" by Dead or Alive respectively.

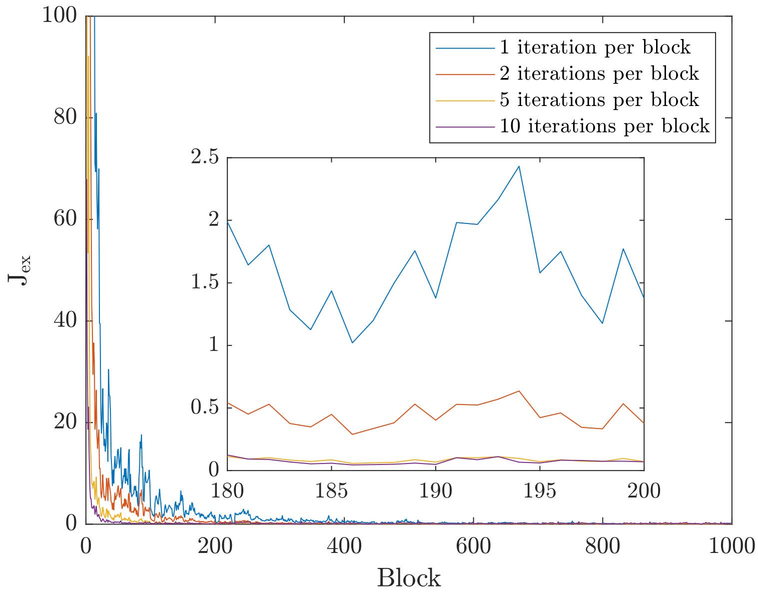


Figure 5.1: Illustration of the excessive cost for the playback signal "Serve the Servant" by Nirvana, with different number of iterations per block.

It can be seen in Figure 5.1, that the adaptive filters still converges towards the closed form filters for the chosen number of iterations per block. It can however also be seen that when only using one iteration per block, the excessive cost varies relative much from block to block. It can also be seen that using 10 iterations per block yields little decrease in the excessive cost compared to only using 5 iterations per block. Using twice as many iterations per block therefore seems wastefully for this choice of playback signal, as there is little difference in the excessive cost, and since the excessive cost varies little from block to block.

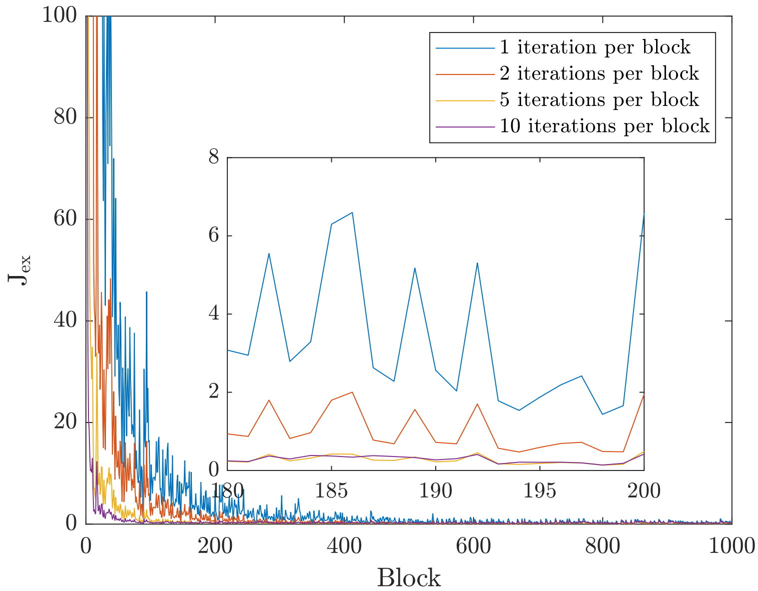


Figure 5.2: Illustration of the excessive cost for the playback signal "You Spin Me Round" by Dead or Alive, with different number of iterations per block.

Figure 5.2 tells the same story as Figure 5.1, but it is here more apparent that using only 1 and 2 iterations per block results in the excessive cost being more influenced by the playback signal.

5.1.1 Influence in cost with dynamic sound zones

From the results illustrated in Figure 5.1 and Figure 5.2, it seems that 5 iterations per block is a balanced choice between a steady and low excessive cost and few iterations per block, when the sound zones are static. However, for dynamic sound zones, the cost is also influenced by the movement of the sound zones from block to block. The cost associated with a specific constellation of measuring points is likely to change if the constellation is changed. It should however be ensured that the cost of the adaptive filters is not increased due to the speed of the movement of the sound zones. If this is not ensured, the correction of the filters are too slow to follow the movement of the user, resulting in sub-optimal dynamic sound zones.

For this test, algorithm 1 is used where $\mathbf{U}_A(\mathrm{b})$ is again set to be an $L \cdot N \times L \cdot N$ identity matrix and $\mathbf{U}_B(\mathrm{b}) = 0$ for all b, such that dynamic playback signals do not influence the cost. Sound zone B is static, and has center in the location $Pos_B = [-2.83,\ 2.83]^T$. Sound zone A is bright and initialized in $Pos_A = [2.83,\ 2.83]^T$. After 500 blocks it is moved clockwise on a circle circumference with center in the center of the loudspeaker array and a radius of $\|Pos_A\|_2 = 4$. Sound zone A is moved with 1.4 meters per second, which is a fast walking speed [31], until it has been moved one meter. As reference this is achieved after 22 blocks when 256 new samples are processed per block and with a sampling frequency of 8000 Hz. For this test it has been pre-examined that the cost at the start position of Pos_A is greater than the last position of Pos_A with the closed from solution of the same configuration. This means that enough iterations per block should be calculated such that the cost only decreases while moving.

The result is illustrated in Figure 5.3. It is here seen that 50 iterations per block is required for the cost not to be influenced by the movement speed.

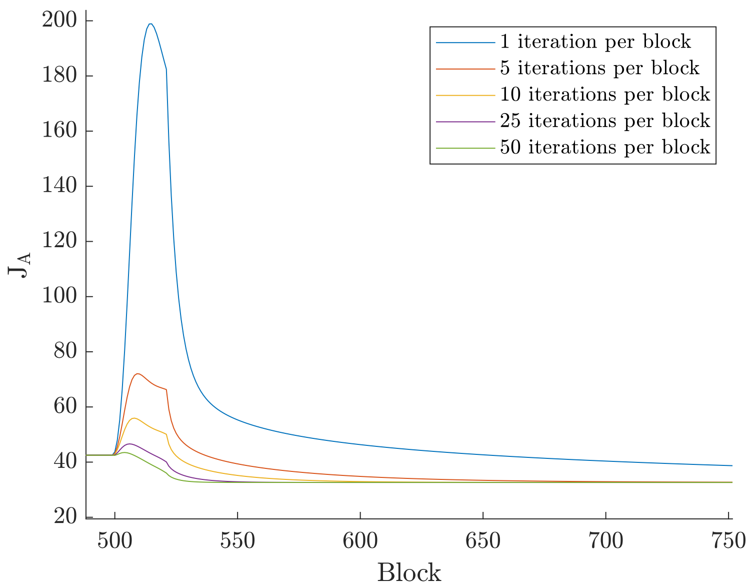


Figure 5.3: Illustration of the cost for the adaptive filters, where the spatial position of sound zone A is changed with a movement speed of 1.4 m/s.

The test is repeated, but now the movement speed of Pos_A is changed to 2.8 meters per second until Pos_A has traveled one meter. The result is illustrated in Figure 5.4, where it is seen that 75 iterations per block is necessary.

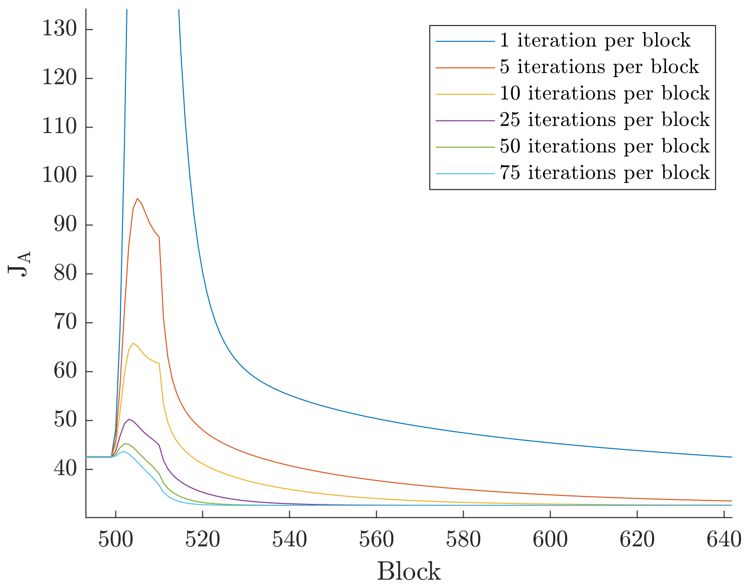


Figure 5.4: Illustration of the cost for the adaptive filters, where the spatial position of sound zone A is changed with a movement speed of 2.8 m/s.

It thus seems like the movement speed of the sound zones has a far greater influence on the number of iterations necessary per block, than the tested playback signals. It is no surprise that by moving with a greater speed, a greater number of iterations per block is needed, if the cost should not be influenced by the moving speed. It does however also mean that it is likely that the number of iterations per block can be calculated based on the movement speed of the user. However, the convergence of the adaptive filters also depends on the step size, which for these tests were chosen as $\alpha = 2$. If α is decreased it would naturally imply that the convergence rate decreases, and more iterations per block will be required. The opposite implies if α is increased. As mentioned in subsection 4.1.1, the maximum step size is determined by the eigenvalues of the hessian of the cost function, and it can therefore be hard to determine a large step size in an online algorithm since the hessian is not constant. Therefore determining the maximum number of iterations to calculate per block is also difficult.

5.2 Comparison of the two adaptive filter algorithms

So far two methods of creating the dynamic sound zones have been made. One which includes both playback signals in the calculation of the filters of the two sound zones, and one where the filters for one sound zone is calculated independent of the other playback signal. The two methods can be seen in algorithm 2 and algorithm 1 respectively.

The two algorithms will be evaluated by comparing their ability to produce the reference sound field in the measuring points, and compare it to the array effort required for producing the sound fields. To do this a common measure of the ability to produce the sound field and calculate the array effort must be established.

5.2.1 Measure of array effort and ability to create the sound field

As written in section 2.1, the idea behind the creation of the filters with no signal compensation, was that when the filtered playback signals are summed and played by the loudspeakers, the pressure contribution in the measuring points should be equal to the sum of the reference pressure of zone A and zone B. This is illustrated in Figure 5.5. The reference pressure for the method with no signal compensation is created according to (3.18), and by summing the reference from playback signal A and B, the reference pressure used in the method with signal compensation is obtained. This reference is written in (3.20), and it will be used as the reference for both algorithms when evaluating their ability to produce the sound field. The reference is denoted as $\mathbf{P}_r(\mathrm{b})$ where b is the block time.

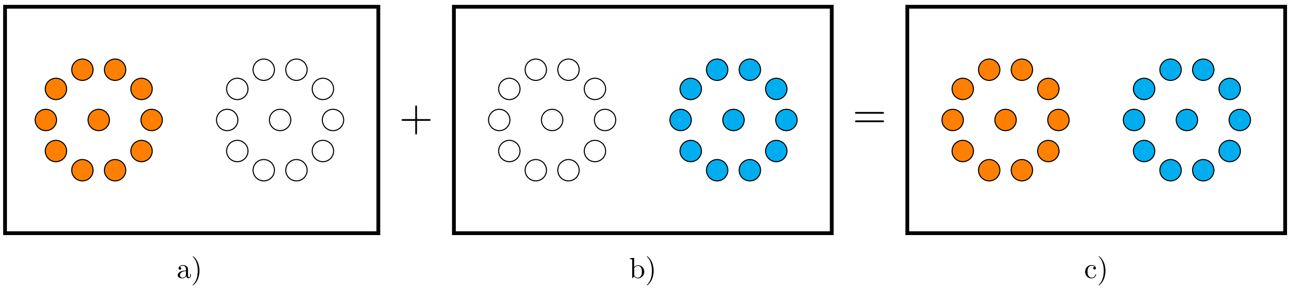


Figure 5.5: a) Illustration of reference pressure associated with playback signal A, b) Illustration of reference pressure associated with playback signal B, c) Sum of the two reference pressures. The white circles illustrates acoustically dark measuring points.

The signal to be played by a loudspeaker is a sum of the filtered playback signal A and the filtered playback signal B. To simulate the pressure contribution in the measuring points, the model of the room from (3.10) is used. The pressure contribution from all loudspeakers in the measuring points when simulating the filtered playback signals can be calculated as

$$\mathbf{H}(\mathrm{b})(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})). \tag{5.3}$$

A cost based on the difference between the simulated pressure contribution and the reference pressure for a given block of the playback signals can then be formulated as

$$J_{sim}(b) = \|\mathbf{H}(\mathrm{b})(\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})) - \mathbf{P}_r(\mathrm{b})\|_2^2, \tag{5.4}$$

which will be used as a measure of how close the simulated sound field is to the reference. As seen, (5.4) is nearly identical to the cost function used for formulating the adaptive filters with signal compensation, which is written in (4.39), but the simulation cost do not include the regularization terms.

The array effort is a measure of the energy required to produce the sound field. The signals to be played by the loudspeakers is calculated as

$$\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b}), \tag{5.5}$$

and the energy required to produce this sound field is calculated as

$$AE(\mathrm{b}) = \|\mathbf{U}_A(\mathrm{b})\mathbf{Q}_A(\mathrm{b}) + \mathbf{U}_B(\mathrm{b})\mathbf{Q}_B(\mathrm{b})\|_2^2. \tag{5.6}$$

5.2.2 Test and results

It is decided that for this test, no regularization will be used when running the algorithms, i.e $\beta = 0$, in order to not influence either the reproduced sound field or array effort for the two algorithms with the regularization terms.

Position A, which defines the location of sound zone A, is now defined as $Pos_A = [2.4,\ 3.3]^T$ and position B is defined as $Pos_B = [-2.4,\ 3.3]^T$. Both sound zones are static. At these coordinates the center of the sound zones are located 4.08 meters from the center of the loudspeaker array.

Now Algorithm 1 and Algorithm 2 is run for 1000 blocks with 5 iterations per block. The iterations per block is based on the findings from section 5.1. After each block, the current block of the playback signals are filtered by the newly updated filters, and the cost from (5.4) and the array effort from (5.6) is calculated and saved. This is done with $\alpha = 2$ and $\gamma = 0.5$.

As for the choice of playback signal A and B, different combinations are used. With playback signal A being a selected time interval of the song ”Smooth Criminal” by Michael Jackson and playback signal B being a selected time interval of ”You Spin Me Round” by the band Dead or Alive, the cost when using the two algorithms becomes as illustrated in Figure 5.6.

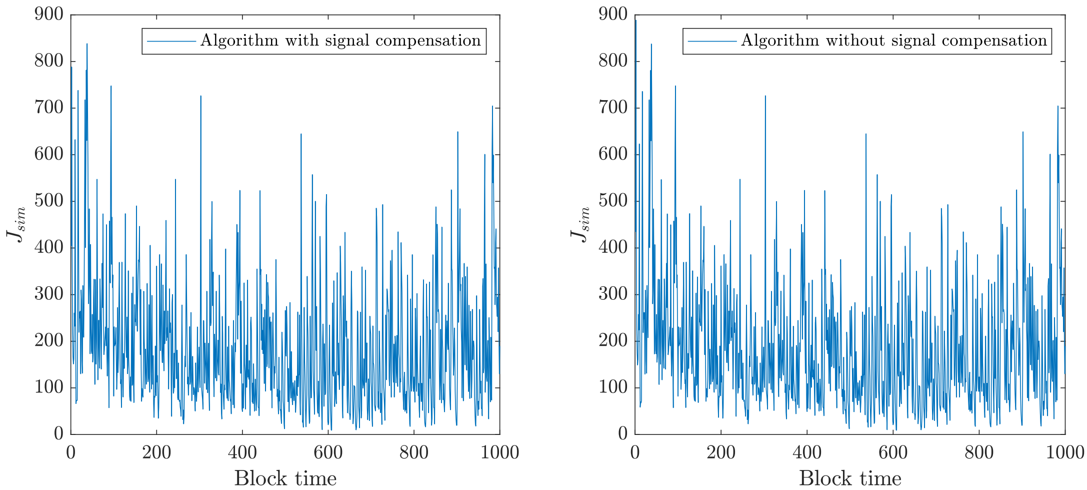


Figure 5.6: Cost of the two adaptive filter algorithms as a function of blocks with playback signal A being ”Smooth Criminal” by Michael Jackson and playback signal B being ”You Spin Me Round” by the band Dead or Alive.

It can be seen in Figure 5.6, that the simulation cost fluctuates much depending on the current block of the playback signals. This makes it hard to tell which algorithm performs best. Therefore the ratio between the cost of the two methods as a function of the blocks is calculated and is defined as

$$\text{Ratio of cost (b)} = \log_{10}\left(\frac{J_{sim}(\text{b})\text{ with compensation}}{J_{sim}(\text{b})\text{ without compensation}}\right). \tag{5.7}$$

Similarly, the ratio of the array effort is calculated as

$$\text{Ratio of array effort (b)} = \log_{10}\left(\frac{AE(\text{b})\text{ with compensation}}{AE(\text{b})\text{ without compensation}}\right). \tag{5.8}$$

The ratio of the cost and the array effort is illustrated in Figure 5.7.

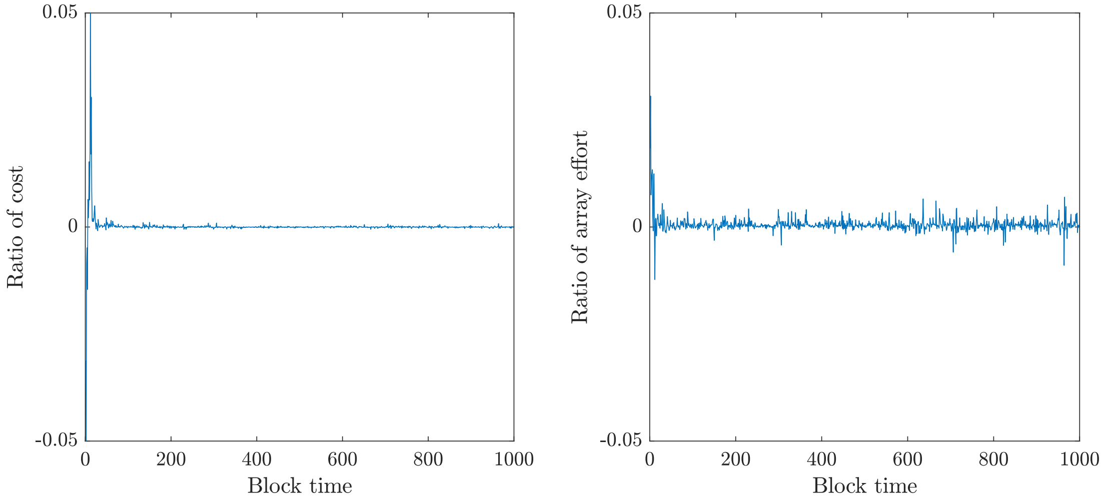


Figure 5.7: Ratio of cost and ratio of array effort between the two algorithms with playback signal A being ”Smooth Criminal” by Michael Jackson and playback signal B being ”You Spin Me Round” by the band Dead or Alive.

Since the ratio of cost is mostly positive until 100 blocks, it seems that the algorithm with signal compensation converges slower than the algorithm without signal compensation. It can also be seen that when converged, there is no great benefit of using one algorithm over the other with regards to cost and array effort. Interestingly enough the ratio of the array effort indicates no clear benefit of using the algorithm with signal compensation. It was expected that by using the signal compensation, the same level of simulation cost could be achieved but with lower array effort, since one set of filters would not have to remove pressure contribution of specific frequencies, if the frequencies should be present in both sound zones. The results in Figure 5.7 however indicates that there is no clear benefit of choosing one algorithm over the other.

By using a selected time interval of ”Serve the Servants” by Nirvana as playback signal A and a selected time interval of ”Living on the Edge” by Aerosmith as playback signal B, the cost ratio and ratio of array effort, illustrated in Figure 5.8, is obtained.

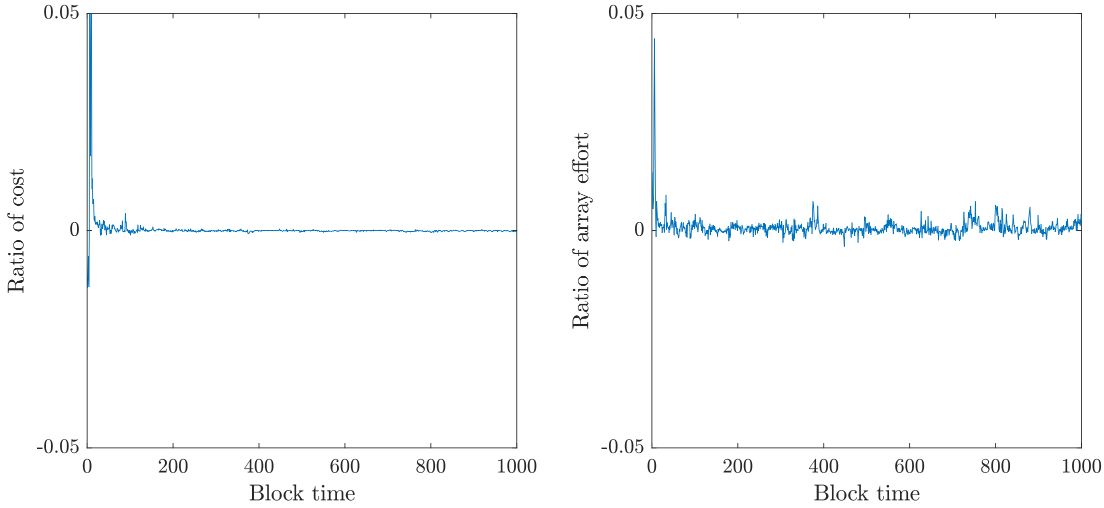


Figure 5.8: Ratio of cost and ratio of array effort between the two algorithms with playback signal A being "Serve the Servants" by Nirvana and playback signal B being "Living on the Edge" by Aerosmith.

It can be seen in Figure 5.8 that the tendency of the ratios are similar. The algorithm with no signal compensation converges quicker, but once converged, it seems that there is no clear benefit of choosing one algorithm over the other with regards to cost and array effort.

5.3 Memory comparison of adaptive filters with closed form filters

In chapter 1, an example of creating dynamic sound zones with closed form sets of filters was made. Here it was told that closed form sets of filters could be calculated for a number of spatially sampled positions in the environment, and either cross-fading techniques between the filters or just using the filters for the closest position of the users could be used. It was argued that this would require much memory to store all the filters. This section aims to give an estimate of how much memory that can be saved by using adaptive filters compared to closed form filters.

For time domain filtering, if 256-tap FIR filters are used in a 20 loudspeaker system, delivering playback signal A to zone A while maintaining silence in zone B means storing $256 \cdot 20 = 5{,}120$ filter coefficients in memory. The same applies for delivering playback signal B to zone B while maintaining silence in zone A. For two specific static zones 10,240 filter coefficients must be stored. In the case of the adaptive filter algorithms, the filter coefficients are stored in the frequency domain, and due to the overlap-save method, the filters must be twice as long. Therefore it is necessary to store 20,480 filter coefficients in memory, but these are updated as the location of the sound zones changes, meaning no further memory is needed for filter coefficients.

As the closed form filters are calculated for specific locations of sound zone A and B, the closed form filters must be calculated for every combination of locations of zone A and zone B in the environment. In practice it can be chosen to only maintain signal in zone A and maintaining silence in zone B, and then calculate the closed form filters for all location combinations. Delivering two playback signals can then be done by choosing the two sets of filters where the location of zone A and B are switched. One playback signal must then be filtered with one set of filters, while the other playback signal is filtered with the other set. Therefore a single location configuration requires 5,120 filter coefficients.

If a rectangular room that is 10 meters long and 5 meters wide is spatially sampled every 0.5 meters in both directions, the room will contain $(10/0.5 + 1) \cdot (5/0.5 + 1) = 231$ different locations for a single sound zone. If the two sound zones can be located in all these locations independently of each other, there exists $231 \cdot 231 = 53{,}361$ different combinations of the location of the two sound zones. As it is necessary to store 5,120 filter coefficients for one combination of sound zone locations, it will be necessary to store $53{,}361 \cdot 5{,}120 = 273{,}208{,}320$ filter coefficients for all combinations of the two sound zones in the 10 by 5 meters room. If every filter coefficient is represented in 16 bits, the filter coefficients will require

$$\text{Memory}_{tot} = \frac{273{,}208{,}320 \cdot 16}{1024^3} = 4.07 \text{ Gib} = 0.51 \text{ GiB} \tag{5.9}$$

of memory. It is not known whether the spatial sampling is adequate, and therefore (5.9) can only serve as an estimate of the necessary memory for the given room. It can however be seen that it is not a trivial amount of memory. Especially when compared to the adaptive filters, which only would require 40 KiB for a setup of the same configuration. The necessary memory for the adaptive filters will not change as long as the filter length, number of sound zones and the number of loudspeakers is kept constant. The necessary memory for the closed form filters will furthermore depend heavily on the size and spatial sampling of the room.

5.4 Comparison of computational complexity

This section aims to estimate the computational complexity of calculating the closed form filters together with the complexity of the adaptive filter algorithm. For the comparisons only the adaptive filter algorithm without signal compensation will be used. The algorithm is found in algorithm 1.

Recall the variable N denoting the frequency domain filter length, L denoting the number of loudspeakers and M denoting the number of measuring points where $M \geq L$. Also recall that $\mathbf{H} \in \mathbb{C}^{M \cdot N \times L \cdot N}$, $\mathbf{U} \in \mathbb{C}^{L \cdot N \times L \cdot N}$ and $\mathbf{P}_r \in \mathbb{C}^{M \cdot N \times 1}$. In the comparison of the computational complexity, $\mathbf{U}_B(b)$ from Algorithm 1 is set to zero, such that zone B is silent. The algorithm then reduces to finding a single set of filters, which is also the case when using the closed form solution. When comparing the computational complexity the big O notation will be used. This gives an upper bound of how the complexity of the different operations grows as the problem size is increased [32]. It is here chosen to show how the different variables, N, L and M influences the upper bound of the complexity. The calculation of the closed form filters and the adaptive filters is partitioned into intermediate calculations to give better insight in the complexity of the two methods. In Table 5.2 the computational complexity for the intermediate operations involved when calculating the closed form filters are listed [33][34].

Operation	**Complexity**	**Note**
$var_1 = \mathbf{HU}$	$\mathcal{O}(ML^2N^3)$	$\mathbf{U}$ is sparse
$var_2 = var_1^H var_1$	$\mathcal{O}(ML^2N^3)$	
$var_3 = \mathbf{U}^H\mathbf{U}$	$\mathcal{O}((LN)^3)$	$\mathbf{U}$ is sparse
$var_4 = (var_2 + \beta var_3)^{-1}$	$\mathcal{O}((LN)^3)$	
$var_5 = var_1^H \mathbf{P}_r$	$\mathcal{O}(LMN^2)$	
$\mathbf{Q_{cf}} = var_4 var_5$	$\mathcal{O}((LN)^2)$	

Table 5.2: Table of computational complexity for calculating the closed form filters.

If $M > L$, the operation with the greatest complexity is the computation of var_1 and var_2 from Table 5.2. It is also seen that four of the six intermediate operations grows with N^3 and two grows with L^3.

Similarly, the computational complexity of the adaptive filter algorithm for a single update of the filters is estimated. These are seen in Table 5.3.

Operation	**Complexity**	**Note**
$var_1 = \mathbf{U}(\text{b})\mathbf{Q}(\text{b})$	$\mathcal{O}((LN)^2)$	$\mathbf{U}$(b) is sparse
$\mathbf{Y}(\text{b}) = \mathbf{H}(\text{b})var_1$	$\mathcal{O}(MLN^2)$	
$var_2 = \beta var_1$	$\mathcal{O}(LN)$	
$\mathbf{E}(\text{b}) = \mathbf{Y}(\text{b}) - \mathbf{P}_r(\text{b})$	$\mathcal{O}(MN)$	
$var_3 = \mathbf{H}^H(\text{b})\mathbf{E}(\text{b}) + var_2$	$\mathcal{O}(LMN^2)$	
$var_4 = \mathbf{U}^H(\text{b})var_3$	$\mathcal{O}((LN)^2)$	$\mathbf{U}$(b) is sparse
$var_5 = \mathbf{D}(\text{b})var_4$	$\mathcal{O}((LN)^2)$	$\mathbf{D}$(b) is sparse
$\mathbf{\Phi}(\text{b}) = \text{IFFT}[var_5]$	$\mathcal{O}(N\log(N))$	Must be done for each loudspeaker.
$var_5 = \text{FFT}\begin{bmatrix}\mathbf{\Phi}(\text{b})\\ \mathbf{0}\end{bmatrix}$	$\mathcal{O}(N\log(N))$	Must be done for each loudspeaker.
$\mathbf{Q}(\text{b}+1) = \mathbf{Q}(\text{b}) - \alpha var_5$	$\mathcal{O}(LN)$	

Table 5.3: Table of computational complexity for calculating a single iteration of the filters with the adaptive filter algorithm.

In Table 5.3 the greatest complexity is when calculating $\mathbf{Y}(\text{b})$ and var_3 if $M > L$. It is however also seen that for all the intermediate computations, the complexity only grows squared instead of cubed. It is however necessary to calculate multiple iterations of the adaptive filter algorithm as showed in the results from section 5.1.

5.5 Anechoic assumption in a reflective environment

As it was decided in section 3.1 that the pressure model should only include acoustical transfer functions from assumptions of anechoic conditions, it is of interest to investigate whether this assumption is a fair assumption. Furthermore the results from the investigation will indicate how alike the simulation are compared to practices. A test is therefore conducted which is found in Appendix A.

The idea of the test is to compare simulated and recorded reproduction playback signals in both sound zones. The simulated signal will be modeled as being in an anechoic environment, while the recorded signal will consist of the playback signal with reflections and noise from the environment. As the conversion factor between the strength of the simulated signal and the recorded signal is not know, and since both signals are hard to synchronize, the comparison of the two environments will be conducted by calculating the cross-talk (CT), which indicated the levels of acoustical separation. By using CT the relative pressure difference between the bright sound zone and the dark sound zone is used, such that no synchronization or conversion factors are needed.

The CT for a measuring point in each of the sound zones are defined as [5]

$$CT = \frac{1}{N}\sum_{n=1}^{N}\left(20 \cdot log_{10}\frac{|P_B(n)|}{|P_D(n)|}\right), \tag{5.10}$$

where $P_B(n)$ and $P_D(n)$ is the pressure in the bright and dark zone respectively at time n and N being the total number of time domain samples.

For the test, four playback signals are used, where three of the playback signals are songs and one is noise. By using different playback signals, and thereby different spectrums, it is believed that a more general representation of what influence a reflective environment have on the CT is found.

Each of the playback signals are run through the computed closed form filters from (4.37), with $\mathbf{U}$ being a $L \cdot N$ by $L \cdot N$ identity matrix. Here $Pos_A = [2.4,\ 3.3]^T$ defines sound zone A which is bright, and $Pos_B = [-2.4,\ 3.3]^T$ defines sound zone B, which is dark.

The contribution from the filtered playback signal for each of the loudspeakers, is both simulated and recorded in Pos_A and Pos_B. For each of the choices of the playback signal, the average CT for samples equivalent to 60 seconds of recording are computed, and the results can be seen in Table 5.4.

Playback signal	**Sim. CT [dB]**	**Rec. CT [dB]**	**Sim. CT - Rec. CT [dB]**
Michael Jackson - Smooth Criminal	15.33	10.33	-5.00
Nirvana - Serve the Servants	12.97	10.39	-2.58
Aerosmith - Livin' On The Edge	12.32	8.93	-3.39
Noise - $\sim \mathcal{N}(\mu = 0, \sigma = 0.2)$	10.40	8.76	-1.64

Table 5.4: Results from the CT computed via simulations with the assumption of anechoic conditions, and recordings from a reflective and noisy environment with the closed form filters.

From the results in Table 5.4 it can be seen that the average CT depends on the playback signal in both the measured and simulated test. It can also be seen that the average recorded CT is constantly lower than the simulated average CT. However the difference between the simulated and recorded CT is not constant and is therefore also dependant on the playback signal.

To illustrate the cause of the lower CT in the recordings compared to the simulations, a selected part of the test with "Micheal Jackson - Smooth Criminal" is illustrated in Figure 5.9. It is here seen that in the simulation, the relative amplitude difference between the bright and dark zone are large compared to the recorded signals. Another interesting observation is the signal in the dark zone of the recording after 600 samples. The signal in the dark zone looks similar to a delayed and attenuated version of the signal in the bright zone. This delay and attenuation could be caused by a reflection of the sound from a wall. It can also be seen that this reflection is not present in the simulated case.

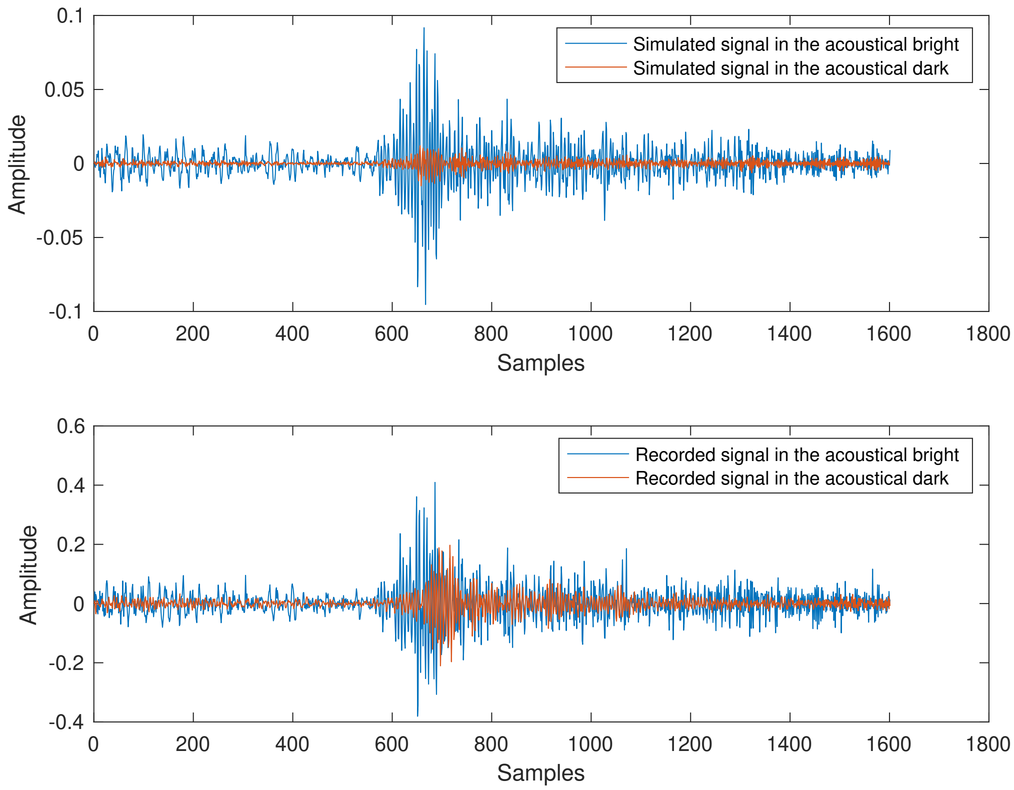


Figure 5.9: Simulated and recorded reproduction signal of ”Michael Jackson - Smooth Criminal” in both the acoustical bright and dark zone for a chosen number of samples at approximately 18 sec. of the playback signal.

Based on the presented results in Table 5.4 and Figure 5.9, the model used when calculating the filters could be expanded such that it included reflection, if a better separation between the dark zone and the bright zone is desired. To include reflections in the model will however require knowledge of where the reflections are coming from, and to which degree the reflected signal is attenuated. On the other hand if a general solutions is desired, that is independent on the environment, and only consider the distance from the speaker to the measuring points that forms the sound zones, then the assumption of anechoic conditions can be used on the compromise of lower separation.

6 Discussion

The limitations of the frequency range of the playback signal when using the adaptive filters, unfortunately also limits the applicable scenarios, since it is assumed that few are willing to listen to music not having the full audible range etc. It will therefore be necessary to incorporate other methods of creating sound zones for the frequencies not covered by the adaptable filters. In the high frequency range, super directive turn able loudspeakers could for example be used. It is however still argued that if a full audible frequency range is not a requirement, such as in the intercommunication system example where the playback signals are speech, the adaptable filters could be used as a stand alone solution.

The test from section 5.1 showed that dynamic sound zones require many additional iterations per block if the cost should not be influenced by the speed of movement compared to static sound zones. The test showed that 50 iterations per block is necessary when moving with 1.4 meters per second. This could make the adaptive filter algorithm difficult to execute in real time, due to the need of multiplying the rather large matrices and vectors. In a relative naive implementation of the adaptive filter in MATLAB R2020a, it was experienced that 2 iterations per block can run in real time on a Intel Core i5-4460 processor. This processor has 4 cores with 1 thread per core and a clock frequency of 3.2 GHz. The algorithm can however be heavily parallelized if enough hardware is available. Calculating the pressure contribution in all the measuring points can be done parallel for both the individual frequency bins, but also for the individual measuring points. Calculating the gradient of the cost function can be done in parallel for all frequency bins and all loudspeakers. Constraining the gradient can be done in parallel for all the loudspeakers, and updating the filters can be done parallel for all frequency bins and for all loudspeakers. If a GPU is available for the adaptive filter algorithm, 50 iterations per block does not seem far-fetched for a real time implementation.

Another thing to consider regarding the number of iterations per block, is how this parameter is chosen in an online algorithm. It was discussed in section 5.1 that the number of iterations per block needed, would depend on the movement speed of the user, and also the step size used in the algorithm, which in turn depends on the non constant hessian of the cost function. Furthermore, it was experienced in simulations that different movement patterns of two dynamic sound zones influenced the cost differently. It could therefore be that moving the sound zones in some pattern would require additional iterations per block. How to chose a upper limit for the number of iterations per block in an online algorithm is therefore a rather difficult question to answer.

In section 5.3 an estimate of the necessary memory for storing filter coefficients was made for closed form filters in a spatially sampled room and for the adaptive filters. Through this thesis the sound zones has only been defined for the horizontal plane. If the sound zones are created in three dimensions, instead of two, the number of filter coefficients for the adaptive filters will only scale linearly with the number of loudspeakers for such a setup. On the other hand for the closed form filters, the combinations of sound zones locations now scales cubed, instead of squared, with the resolution of the spatial sampling, and so will the necessary memory. Similarly if an additional sound zone is defined, the adaptive filter coefficients will still scale linearly. For the closed form filters, the combinations of sound zone placements is now approximately the number of spatially sampled points cubed, instead of squared when only two sound zones were used. This further emphasizing the advantage of the adaptive filters from a memory perspective.

For this thesis, two adaptive filter algorithms had been created. It was expected that the algorithm that incorporated both signals when updating the filters could obtain the same level of reproduction error, as the algorithm which only used one, but with a lower array effort. The test from section 5.2 however showed that there was no clear benefit of choosing one algorithm over the other with respect to reproduction error and array effort when using music as playback signals. It was manually checked to some degree that the paired playback signals had a somewhat similar frequency spectrum, and it was therefore surprising that the two methods produced such similar results.

When modelling the pressure contribution from the loudspeakers in the measuring points, the model included only the direct path. The test in section 5.5 showed that there is a clear difference in the acoustical separation of the sound zones between simulations an real measurements. This indicates that the model is somewhat lacking if the highest separation is desired. However as discussed before, it is simple to create and it is independent of the location of reflective objects. Creating a model with reflection, while still modeling the pressure instead of measuring it, could be done by incorporating mirror sources in the room model. As argued previously in this chapter, the algorithm is however difficult to execute in real time without parallel hardware, and making the model more complex to create, might therefore not be a desired choice.

Conclusion 7

Through this thesis, there has been worked with the following research question:

> ***"How can adaptive filters be constructed and applied for creating dynamic sound zones?"***

Two dynamic sound zones consisting of multiple dynamic measuring points were defined, and their spatial locations was determined by user location data. It was defined that a playback signal should be available in only a single sound zone. Creating the two sound zones, each with their own playback signal, was done by filtering the playback signals with multiple adaptable FIR filters before being reproduced by multiple loudspeakers in a known constellation.

The dynamic measuring points made it possible to create two dynamic models, that describes the pressure contribution of the filtered playback signals in the measuring points when reproduced by the loudspeakers. One model included both playback signals, and the other only included a single playback signal. The two models were created to be independent of the environment in which the sound zones are located.

By using a block implementation, that created overlapping blocks of the playback signals, transforming these to the frequency domain, and defining and updating the dynamic model in the frequency domain, it was possible to efficiently calculate the pressure contribution in the dynamic measuring points. The block implementation furthermore allowed for efficient filtering of the playback signals.

By creating a reference pressure in the dynamic measuring points based on their spatial location and the current block of playback signals, it was possible to define a convex optimization problem that was used to minimize the difference between the modelled- and reference pressure, with the frequency domain FIR filters as optimization variable.

The optimization problem was solved iteratively, which allowed for continuous correction of the FIR filters, making them adaptable to any change in the dynamic measuring points. Here a fixed number of iterations was calculate every block.

The two models showed to have equal reproduction error, and almost equal array effort in simulations.

The iterative optimization allowed for an adaptive filter algorithm with low memory requirement.

It was found that the movement speed of the dynamic sound zones greatly influenced the necessary number of iterations per block, in order for the adaptable filters to only being influenced by the current location of measurement points. For static sound zones, but dynamic playback signals, it was found that five iterations per block was adequate to be nearly as good as filters calculated as a closed form solution.

8 Future works

In order to make an online algorithm, it will be necessary to know how many iterations per block to calculate in order to determine an upper limit for the algorithms computational complexity. It will therefore be advantageous to further study the influence of different movement patterns on the cost. Furthermore, a scheme where the step size is varied dependent on the hessian of the cost function could be implemented to increase the convergence rate, and thereby decrease the required number of iterations per block.

Updating the transfer functions from all loudspeakers to all measuring points based on the latest user location data is a rather computational expensive operation, and since it was done for every block in this thesis, the operation was time consuming. It could therefore be advantageous to study how often in practice it will be necessary to update the room model, and to what degree a user will be influenced by the choice of update frequency. If a user can not hear the difference in the sound field when only updating the room model every third block, then updating the room model every block seems wasteful.

When one sound zone is in front of the other relative to the loudspeaker array, the acoustical separation between the sound zones becomes very poor. To address this problem, a different loudspeaker setup could be used. For example, if two loudspeaker arrays are used in a rectangular room, where the arrays are placed on two perpendicular walls, the array for which the distance between the sound zones are greatest could be used to produce the sound zones. This would consequently mean that the source location of produced sound changes, which could confuse some users.

To increase the acoustical separation between sound zones, reflective paths could be included in the room model. Here mirror sources could be used to model the reflections. It will here be necessary to study how many reflections should be included to give a meaningful increase of the acoustical separation. If a user is not able to perceive the increase in separation, additional reflections will only increase the computational complexity.

Bibliography

[1] F. B. Dutton, “Dalton’s law of partial pressures,” *Journal of Chemical Education*, vol. 38, no. 8, p. A545, 1961. [Online]. Available: https://doi.org/10.1021/ed038pA545.1

[2] D. Bies, *Engineering noise control : theory and practice.* London New York: E & FN Spon, 1996.

[3] T. Betlehem, W. Zhang, M. Poletti, and T. Abhayapala, “Personal sound zones: Delivering interface-free audio to multiple listeners,” *IEEE Signal Processing Magazine*, vol. 32, pp. 81–91, 03 2015.

[4] W. Druyvesteyn and J. Garas, “Personal sound,” *Journal of the Audio Engineering Society*, vol. 45, no. 9, pp. 685–701, 1997.

[5] M. F. S. Gálvez, D. Menzies, and F. M. Fazi, “Dynamic audio reproduction with linear loudspeaker arrays,” *J. Audio Eng. Soc*, vol. 67, no. 4, pp. 190–200, 2019. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=20451

[6] F. Olivieri, F. Fazi, M. Shin, and P. Nelson, “Pressure-matching beamforming method for loudspeaker arrays with frequency dependent selection of control points,” *138th Audio Engineering Society Convention 2015*, vol. 2, 05 2015.

[7] F. Olivieri, F. Fazi, P. Nelson, M. Shin, S. Fontana, and Y. Lang, “Theoretical and experimental comparative analysis of beamforming methods for loudspeaker arrays under given performance constraints,” *Journal of Sound and Vibration*, vol. 373, 03 2016.

[8] M. B. Møller and J. Østergaard, “A moving horizon framework for sound zones,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 256–265, 2020.

[9] F. Wefers and M. Vorlaender, “Efficient time-varying fir filtering using crossfading implemented in the dft domain,” *European Acoustic Association*, 09 2014.

[10] J.-W. Choi and Y.-H. Kim, “Generation of an acoustically bright zone with an illuminated region using multiple sources,” *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1695–1700, Apr. 2002. [Online]. Available: https://doi.org/10.1121/1.1456926

[11] M. Shin, S. Q. Lee, F. M. Fazi, P. A. Nelson, D. Kim, S. Wang, K. H. Park, and J. Seo, “Maximization of acoustic energy difference between two spaces,” *The Journal of the Acoustical Society of America*, vol. 128, no. 1, pp. 121–131, Jul. 2010. [Online]. Available: https://doi.org/10.1121/1.3438479

[12] P. Coleman, P. J. B. Jackson, M. Olik, M. Møller, M. Olsen, and J. A. Pedersen, “Acoustic contrast, planarity and robustness of sound zone methods using a circular loudspeaker array,” *The Journal of the Acoustical Society of America*, vol. 135, no. 4, pp. 1929–1940, Apr. 2014. [Online]. Available: https://doi.org/10.1121/1.4866442

[13] O. Kirkeby and P. A. Nelson, "Reproduction of plane wave sound fields," *The Journal of the Acoustical Society of America*, vol. 94, no. 5, pp. 2992–3000, Nov. 1993. [Online]. Available: https://doi.org/10.1121/1.407330

[14] M. Shin, F. M. Fazi, P. A. Nelson, and F. C. Hirono, "Controlled sound field with a dual layer loudspeaker array," *Journal of Sound and Vibration*, vol. 333, no. 16, pp. 3794–3817, Aug. 2014. [Online]. Available: https://doi.org/10.1016/j.jsv.2014.03.025

[15] S. Haykin, *Adaptive Filter Theory.* USA: Prentice-Hall, Inc., 1986.

[16] T. ElAli, *Discrete systems and digital signal processing with MATLAB.* Boca Raton, FL: CRC Press, 2012.

[17] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. USA: Prentice Hall Press, 2009.

[18] L. Kinsler, *Fundamentals of acoustics.* New York: Wiley, 2000.

[19] C. H. Hansen, "1 fundamentals of acoustics." [Online]. Available: https://pdfs.semanticscholar.org/b99c/98062ebee8d54aa44de4ad496f5f22f04b06.pdf

[20] D. Self, *Audio engineering explained : professional audio recording.* Oxford: Focal, 2009.

[21] H.-E. de Bree, "The microflown e-book," Available at https://www.microflown.com/resources/e-books/e-book-the-microflown-e-book/ (2020/04/02), 2009.

[22] R. Rabenstein and S. Spors, "Spatial aliasing artifacts produced by linear and circular loudspeaker arrays used for wave field synthesis," in *Audio Engineering Society Convention 120*, May 2006. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=13515

[23] S. Spors, H. Buchner, and R. Rabenstien, "A novel approach to active listening room compensation for wave field synthesis using wave-domain adaptive filtering," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2004, pp. iv–iv.

[24] R. Bader, *Springer handbook of systematic musicology.* Germany: Springer Verlag, 2018.

[25] D. Davis, *Sound system engineering.* New York: Focal Press, 2013.

[26] K. Stevens, *Acoustic phonetics.* Cambridge, Mass: MIT Press, 1998.

[27] A. Antoniou, *Practical optimization : algorithms and engineering applications.* New York: Springer, 2007.

[28] L. Trefethen, *Numerical linear algebra.* Philadelphia: Society for Industrial and Applied Mathematics, 1997.

[29] A. Cariow and G. Cariowa, "An algorithm for complex-valued vector-matrix multiplication," *Przeglad Elektrotechniczny*, vol. R88, pp. 213–216, 09 2013.

[30] S. Arunachalam, S. M. Khairnar, and B. S. Desale, "The fast fourier transform algorithm and its application in digital image processing," *Mathematical theory and modeling*, vol. 3, pp. 267–273, 2013.

[31] M. Schimpl, C. Moore, C. Lederer, A. Neuhaus, J. Sambrook, J. Danesh, W. Ouwehand, and M. Daumer, "Association between walking speed and age in healthy, free-living individuals using mobile accelerometry—a cross-sectional study," *PLoS ONE*, vol. 6, no. 8, p. e23299, Aug. 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0023299

[32] "Big o notation," http://web.mit.edu/16.070/www/lecture/big_o.pdf, accessed: 2020-06-31.

[33] M. V. Wickerhauser, "Appendix b - basics, technicalities, and digressions," in *Mathematics for Multimedia*, M. V. Wickerhauser, Ed. San Diego: Academic Press, 2004, pp. 273 – 295. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780127484518500082

[34] R. Oshana, "4 - overview of digital signal processing algorithms," in *DSP Software Development Techniques for Embedded and Real-Time Systems*, ser. Embedded Technology, R. Oshana, Ed. Burlington: Newnes, 2006, pp. 59 – 121. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780750677592500065

A Test of the influence of a reflective environment

The purpose of this test is to investigate whether the assumption of anechoic conditions used in the transfer functions is good in practice, when creating sound zones in reflective environments.

The idea of the test is to compare simulated and recorded reproduction playback signals. The simulated signal will be modeled as being in an anechoic environment, while the recorded signal will consist of reflections and noise from the environment. As the conversion factor between the strength of the simulated signal and the recorded signal is not know and that both signals is hard to synchronize, the comparison of the two environments will be conducted by using cross-talk (CT). By using CT the relative difference between the pressure in the bright sound zone and the dark sound zone is used, such that no synchronization or conversion factors are needed.

The CT for one measuring point in each of the sound zones are defined as [5]

$$CT = \frac{1}{N}\sum_{n=1}^{N}\left(20 \cdot log_{10}\frac{|P_B(n)|}{|P_D(n)|}\right), \tag{A.1}$$

where $P_B(n)$ and $P_D(n)$ is the pressure in the bright and dark zone respectively at time n.

For the test four playback signals are used as the difference in spectrum might show a trend in the CT. The playback signals are monophonic, shorted down to a duration of 60 sec picked in the middle of the songs and bandpass filtered from 350 Hz to 3500 Hz. The playback signals used are

- Michael Jackson - Smooth Criminal
- Nirvana - Serve The Servants
- Aerosmith - Livin' On The Edge
- Noise $\sim \mathcal{N}(\mu = 0, \sigma = 0.2)$

Each of the playback signals are run through both the computed closed form filters, with a assumed flat spectrum of magnitude one, and the filters obtained with the adaptive filter algorithm without signal compensation. The measuring points used for the algorithms are as illustrated in Figure 3.5, where the center point for sound zone A and B are chosen to be

$$Pos_A = [2.4, 3.3]^T$$
$$Pos_B = [-2.4, 3.3]^T.$$

In the algorithm the acoustically bright and dark sound zone is Pos_A and Pos_B respectively.

The parameters, if used in the algorithm, and measuring points for both algorithms are identical. The choice of parameter values are picked as these seemed to work relatively well and is chosen to be

- $\beta = 0.001$ - array effort regularization
- $\gamma = 0.5$ - forgetting factor for input power
- $\alpha = 3$ - gradient update step size
- one gradient update per input block

With the filtered playback signals, two things have to be done. First a simulated reproduced playback signal in the center of both the bright and dark sound zone for all filtered playback signals are computed. These individual simulated reproduction playback signals are used to compute the CT for the assumption of anechoic conditions. Second, the filtered playback signals are played through a loudspeaker array and the pressure in the two defined spatial positions for the two sound zones are recorded. The measurements are recorded in a listening room of 8.09-by-7.35-by-2.87 meters (L-W-H), where the center of the loudspeaker array are located against a wall in the center, which is defined as the Cartesian coordinate (0,0). The test setup for the recordings can be seen in Figure A.1 and Figure A.2 and the equipment used are listed in Table A.1.

Figure A.1: Picture of the test setup for the recordings of pressure in the acoustical bright and dark sound zone for CT computations.

Equipment	**AAU/Serial No.**	**Type**
RME M-16 DA	86844	DA Converter
RME M-16 DA	86844	DA Converter
RME M-16 DA	108243	DA Converter
RME Fireface UCX	-	AD Converter
RME Fireface UFX+	108249	Soundcard
7x150 W ICEPOWER amplifier	0001	Power amplifier
7x150 W ICEPOWER amplifier	0002	Power amplifier
7x150 W ICEPOWER amplifier	0003	Power amplifier
GRAS 40AZ, w. 26CC preamp	78161	Measuring microphone
GRAS 40AZ, w. 26CC preamp	78162	Measuring microphone
Microphone calibrator 94 dB ref. 20 μPa	78301	Brüel & Kjær - TYPE 4231
20 x 2 inch loudspeakers	-	Loudspeakers
MATLAB R2020a	-	Data processing software

Table A.1: Equipment for the room reflection impact on CT.

The individual recordings of the reproduction playback signals are used to compute the CT for

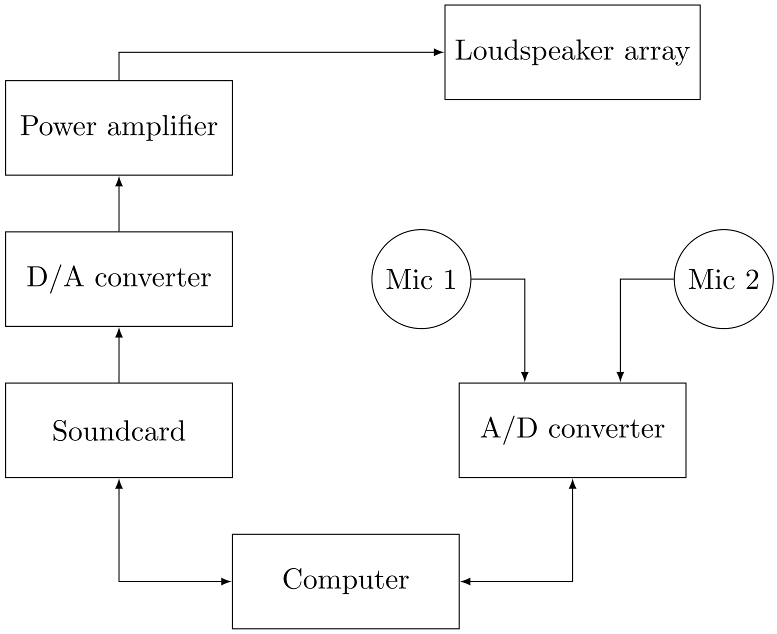


Figure A.2: The test setup for the recordings of pressure in the acoustical bright and dark sound zone for CT computations.

a reflective environment. Remark that the recordings have been highpass filters with a 2. order butterworth filter that have a cutoff frequency of 50 Hz before computing the CT. Furthermore the CT for the adaptive filters is computed with the transient to the converged filters. The test results can be seen in Table A.2.

Playback signal	Filters	Sim. CT [dB]	Rec. CT [dB]	Delta [dB]
Michael Jackson	Closed form	15.33	10.33	−5.00
	Adaptive	15.65	10.53	−5.12
Nirvana	Closed form	12.97	10.39	−2.58
	Adaptive	13.13	10.31	−2.82
Aerosmith	Closed form	12.32	8.93	−3.39
	Adaptive	12.50	8.93	−3.57
Noise	Closed form	10.40	8.76	−1.64
	Adaptive	10.50	8.82	−1.68

Table A.2: Results from the CT computed via simulations with the assumption of anechoic conditions and recordings from a reflective and noisy environment.

Looking at the results from Table A.2 the CT varies relative much from playback signal to playback signal, which indicate that the reflections have an impact on the CT. Furthermore the closed form filters and the adaptive filters for the same playback signal tends to behave the same. It is also seen that "Michael Jackson - Smooth Criminal" differs from the other in that the difference between the CT of the simulated and recorded are relative high compared to the other playback signals. To investigate what the cause may be, the simulated and recorded reproduction signal, generated from the closed form filters, for both the acoustical bright and dark zone are plotted, and is shown in Figure A.3.

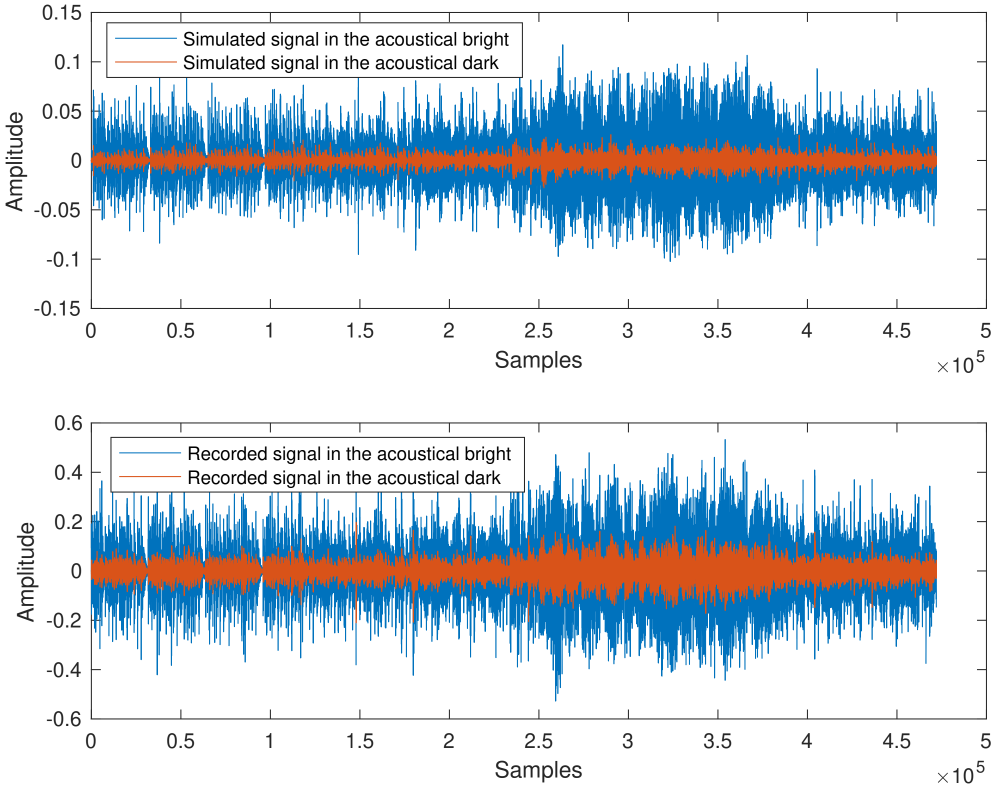


Figure A.3: Simulated and recorded reproduction signal of "Michael Jackson - Smooth Criminal" in both the acoustical bright and dark zone.

By inspection of the simulated reproduction signal plot, it is seen that the relative amplitude difference from the acoustical bright and dark zone is high. The signal in the acoustical dark zone is caused by that it is impossible to create perfect separable sound zones.

For the recorded reproduction signal it is seen that the relative difference is smaller. This is due to that both the signals suffers from noise in the environment, that the recordings are conducted in a reflective environment, that the speakers/speaker system may not produce the correct pressure and that it is impossible to create perfect separable sound zones. The reflections in the environment are most likely to cause the CT to fall the most, as the reflected signals will contribute to the measurements. An example of this can be seen in Figure A.4. Here it is believed that a reflection from the bright sound zone contributes to the pressure in the dark zone. As a reflection is a delayed and attenuated version of the signal, which is also discussed in section 3.1. Furthermore it is assumed that the playback signal "Michael Jackson - Smooth Criminal" suffers the most in terms of CT, as the beat have abrupt transitions to silence in a repeating manner. In these silent moments, contributions from the reflections are present in the recordings. This can also explain why the reflective environment do not impact the CT for noise as much, as there is almost no silent samples.

The test indicates that if the greatest separation, in terms of CT, between the sound zones is desired then other assumptions than anechoic conditions should be considered. This could be to include the transfer functions from the environment in the calculations of the filters.

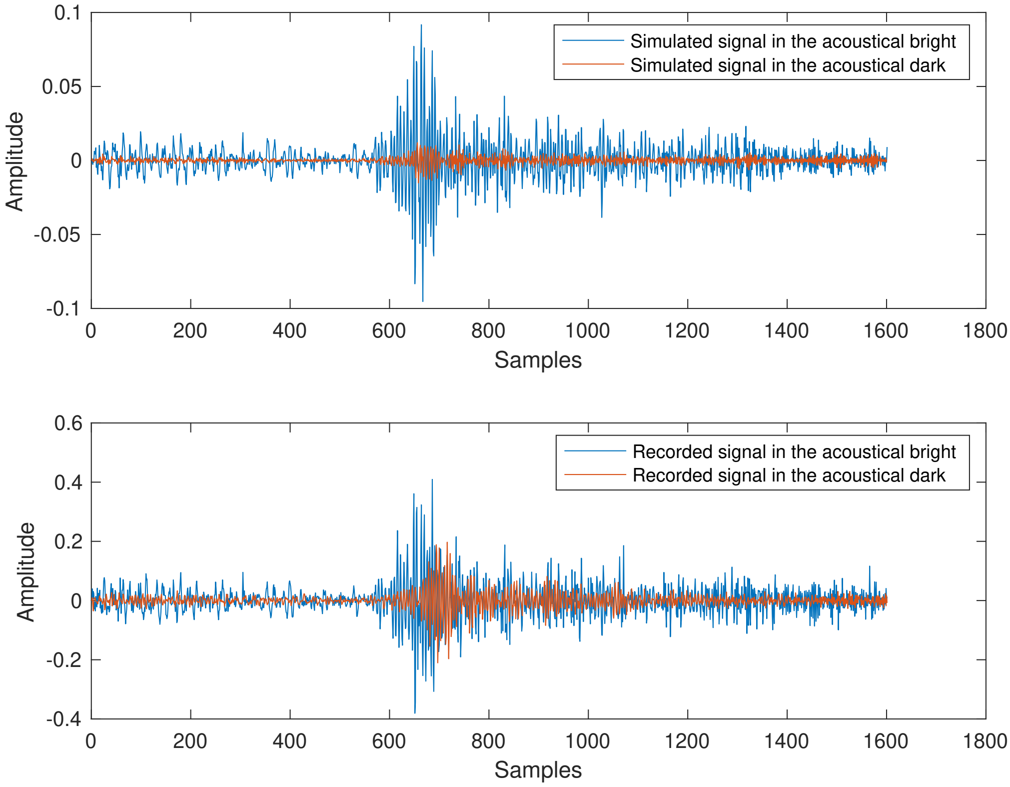


Figure A.4: Simulated and recorded reproduction signal of ”Michael Jackson - Smooth Criminal” in both the acoustical bright and dark zone zoomed in.

On the other hand if a general solutions is desired, that is independent on the environment and only consider the distance from the speaker to the measuring points that forms the sound zone, then the assumption of anechoic conditions can be used on the compromise of lower separation.