Postfiltering in Adaptive Feedback Cancellation for PA Systems

Authors
Johannes Willem de Vries
Cyril Arnold Weustink

TU Delft

Postfiltering in Adaptive Feedback Cancellation for PA Systems

by

J.W. de Vries & C.A. Weustink

to obtain the degree of Bachelor of Science
at the Delft University of Technology.

Student numbers:	4719972 (J.W. de Vries)	
	4692586 (C.A. Weustink)	
Project duration:	20 April 2020 – 3 July 2020	
Thesis committee:	Prof.dr. J. Schmitz,	TU Delft, proposer
	Dr.ir. R. C. Hendriks	TU Delft, supervisor
	Dr. J. Martinez Castaneda,	TU Delft, supervisor
	Dr.ir. S. Izadkhast,	TU Delft

TU Delft

Abstract

The most promising method for acoustic feedback control in public address systems is adaptive feedback cancellation (AFC). This approach is based on making an estimation of the room impulse response, which is used to subtract an estimate of the feedback from the microphone signal. A postfilter is often used to improve the performance of such an AFC system by reducing residual feedback or increasing stability. The postfilter implemented in this thesis focuses on the latter. It consists of two modules: howling detection and howling suppression. The final design makes use of the peak to harmonic power ratio criterion (PHPR) and second order digital Butterworth notch filter respectively. Simulation measurements show a successful suppression of howling as long as the loudspeaker signal does not get saturated.

Preface

This thesis is written in order to obtain the degree of bachelor of science in electrical engineering at Delft University of Technology. The project treated in this thesis was conceived by prof.dr. John Schmitz, who noted that many practical PA systems still suffer from feedback effects in spite of an abundance of theoretical solutions. We want to thank him for providing us with such an interesting and relevant project topic, as well as for his enthusiasm during the project.

We would also like to express our gratitude to our supervisor dr.ir. Richard Hendriks for introducing us to and arousing interest in the subject of digital signal processing. We would like to express our gratitude to our supervisor dr. Jorge Martinez for guiding us through several projects in preparation of the bachelor graduation project. Of course we also want to thank both for the support they offered during the whole graduation project. A special thanks goes out to Metin Çaliş for his dedication to help us whenever we encountered challenges we could not resolve ourselves.

Early 2020, the coronavirus pandemic hit the Netherlands, causing a partial lockdown nationwide and preventing on-campus education from taking place. Because of this exceptional situation, any inclusion of physical implementations was prohibited for all bachelor graduation projects. Therefore, the scope of this thesis excludes a practical implementation and accompanying validation results and instead discusses possible implementations that could be made based on our theoretical analysis. We regret having to leave out such an interesting part of a project but we understand it is a necessary precaution given the circumstances.

J.W. de Vries and C.A. Weustink
Delft, June 2020

Contents

1 Introduction

A public address system, PA system for short, is an electronic system which is used to amplify a sound signal and present this signal to a large group of people. A PA system consists of a combination of microphones, amplifiers and loudspeakers and possibly other devices like audio mixers or equalisers to perform the desired amplification. These systems are often used at concerts and festivals to make the performing artists audible to everyone in the crowd (for example in Figure 1.1). PA systems are also used as a means to make announcements to a large group of people. This is frequently done in schools and train stations. The composition of these PA system can vary between applications. A PA system used at a concert for instance generally has many loudspeakers and several microphones which all have to be combined into one functional system. A mixer is needed to combine these signals appropriately. The number of microphones and loudspeakers in the system depends greatly on the application of the PA system.

In some applications acoustic feedback can occur. This happens when the audio signal that is transmitted by a loudspeaker is received by the microphone. The loudspeaker signal is then amplified by the PA system and again transmitted by the loudspeaker. A loop exists where the transmitted audio signal is fed back into the PA system. This feedback can cause problems. If the strength of the signal that is received by the microphone from the loudspeaker is large enough, the amplified signal strength might be larger than the signal strength originally transmitted by the loudspeaker. In other words, the loop gain is larger than one. When this is the case the system is unstable and the signal will keep looping through the PA system with increasing signal strength. This often happens first for specific frequencies in the signal that are transferred well between the loudspeaker and the microphone. Which frequencies are transmitted best from the loudspeaker depends on the configuration of the PA system and the acoustics of the environment in which it is used. This type of acoustic feedback results in the amplification of a specific frequency in the audio signal which will eventually drown out the original audio signal. This is often referred to as howling or the Larsen effect. To ensure this effect does not occur, the gain of the PA system must be limited. The maximum gain that can be used without causing howling is called the maximum stable gain, MSG for short. This is the highest possible gain for which the PA system is stable.

If the strength of the signal that is received by the microphone from the loudspeakers is not large enough to cause howling, drowning out all other components of the audio, it can still have an effect on the audio signal. The signal received by the microphone is still sent back to the loudspeaker and will be transmitted again, albeit with a smaller signal stength. This signal will eventually die out but can still be audible. When the system is stable but nearing instability, in other words it has a loop gain approaching 1, this effect can be very disturbing.

The aim of this project is to design a device that can eliminate the aforementioned unwanted effects of acoustic feedback by means of signal processing and by doing so increase the maximum stable gain of the PA system.

Figure 1.1: A PA system used at a large concert.

1.1. State of the Art Analysis

Solving the acoustic feedback problem can be traced back as early as the sixties. Since then, much progress has been made in automatic feedback control theory but practical systems still suffer from feedback effects. Following the classification of Van Waterschoot and Moonen [1], four main categories of automatic feedback control can be identified:

- phase-modulation methods,
- gain reduction methods,
- spacial filtering methods,
- room modelling methods.

.

Phase-modulating feedback control (PFC) is one of the oldest techniques of increasing the MSG [2]. Its advantages are the simplicity of the design, the robustness of the system and the relatively little computational power required. A major drawback is the limited effectivity in terms of MSG increase. Other disadvantages include significant distortion in audio applications [3] and decreasing effectivity for increasing number of microphones and loudspeakers [4].

The simplest gain reduction method reduces the broadband gain between microphone and loudspeaker when howling is detected, thereby increasing the stability of the system. This is a very inelegant solution and is usually reserved as a last-effort fail-safe (e.g. in [5]). Another method is called notch-filter-based howling suppression (NHS). It works similarly, but instead of reducing the total gain only the small frequency band with the howling is filtered out. This stabilises the system and suppresses howling. Disadvantages include a limited increase in MSG (comparable to PFC [2]), a more difficult design process due to the number of algorithm parameters, and a significant signal distortion as howling can usually be perceived before it is suppressed.

Spacial filtering methods make use of directionality of microphone and loudspeaker arrays, which is irrelevant to this project as the envisioned device for acoustic feedback control has no influence on the physical properties of the microphones and loudspeakers in any way.

Room filtering methods consist of modelling the acoustic feedback path. A popular approach is to estimate the acoustic feedback and subtract it from the microphone signal, called adaptive feedback cancellation (AFC). With a good estimate, the loop gain can almost be reduced to zero, leading to a large increase in MSG. A biased estimate is a common problem that arises in AFC, which can be solved by applying decorrelation techniques [6]. Additionally, AFC may be extended with other feedback control methods to improve performance and stability (e.g. in [7–9]). The largest drawback of AFC is its heavy computational complexity, which makes it hard to implement in practical systems. This complexity also limits the ability to apply the technique to systems with multiple microphones and loudspeakers.

In this project, AFC is chosen as the main method of automatic feedback control. As mentioned above, both PFC and NHS have a limited effectivity and there is not much room for improvement while AFC has no theoretical limit to its effectivity [1]. This decision has been made based on Requirement 6 as explained in Chapter 3. The AFC unit can be extended with either a PFC or NHS unit as a postfilter to further increase performance and stability. Currently realised MSG increases can reach 20 dB [10, 11].

1.2. Subdivision of the System and Problem Definition

Based on the analysis of the different available techniques for removing acoustic feedback the decision has been made to design a solution based on adaptive feedback cancellation. Such a system typically consists of three parts [7]:

- adaptive filtering,
- decorrelation,
- postfiltering.

Adaptive filtering and decorrelation are described in more detail in Section 2.2. In this project, the adaptive filtering module is designed by Kos and Bekkering [12] and the decorrelation module is designed by Huijbregts and Jongepier [13]. This thesis describes the design of the postfilter module.

The postfiltering module is responsible for improving the quality and stability of the audio signal after channel estimation is used to remove a large part of the acoustic feedback. Because the channel estimation cannot be done perfectly, some residual feedback components can exist in the compensated audio signal. The postfilter is introduced to eliminate this residual feedback. The postfilter can also be used to implement a backup filter that is activated should the main feedback suppression method provided by the channel estimation and decorrelation parts fail. The postfilter should thus, above all, guarantee stability of the system.

1.3. Thesis Synopsis

This thesis contains the complete design process of the postfilter in context of the AFC filter unit, with the exception of a practical implementation. This was impossible due to external circumstances (see preface) and instead, a brief section on implementation possibilities is included. Chapter 2 highlights some important background knowledge on acoustic feedback and AFC. In Chapter 3, the programme of requirements is outlined. It contains functional and non-functional specifications of the complete feedback control unit as well as requirements of the postfilter specifically. The design of the simulation environment is shown and explained in Chapter 4. The first postfilter approach, filtering residual feedback, is discussed in Chapter 5. In Chapter 6, howling detection is explained in detail. Chapter 7 elaborates on the design and implementation process of howling suppression. An evaluation of the complete postfilter and a discussion of these results can be found in Chapter 8. Chapter 9 contains a brief overview on the possibilities for a practical implementation. Lastly, Chapter 10 contains conclusions on the project, possible recommendations and challenges in future work.

2 Background Knowledge

Before going in depth on postfiltering implementations, some background knowledge about the acoustic feedback problem and adaptive feedback cancellation has to be known. This chapter will elaborate on these subjects.

In this thesis, round brackets represent discrete signals: $x(t)$. Individual samples in time domain are denoted by t and frames (blocks of samples) by k with a frame length of N_f. When the frame is specified, sample $t = 0$ is the first sample of that frame: $x(t;k) = x(kN_f + t)$. Frequency domain signals are denoted with upper case letters as $X(\omega;k)$ for the spectrum of $x(t;k)$. Here, $\omega = \omega_n = 2\pi n/N_f$ with $n = 0, \cdots, N_f - 1$ is the discrete normalised angular frequency. Hat operators $\hat{x}(t)$ denote estimators and bar operators $\bar{x}(t)$ denote averages.

2.1. Acoustic Feedback

Starting with a one microphone and one loudspeaker setup, the acoustic feedback problem can be simplified to the model shown in Figure 2.1. The source signal $s(t)$—the voice or music signal that contains the information—is amplified by a mixing console (or something similar) represented by G. As this gain can be frequency dependent, it is useful to define a broadband gain factor $\bar{G}$ as the average magnitude of $G(\omega)$, i.e.

$$\bar{G} = \frac{1}{2\pi}\int_0^{2\pi} |G(\omega)| d\omega. \tag{2.1}$$

The amplified signal is played through the loudspeaker but unintentionally picked up by the microphone again. This feedback signal, named $y(t)$, is acoustically distorted because of acoustic delays, attenuation and reflections from the walls in the room. The transfer function that represents this is called the room impulse response or RIR, and is denoted by F.

The acoustic feedback creates a loop in the system with loop gain $G(\omega)F(\omega)$. This loop can cause instability in certain conditions, giving rise to the howling effect. The Nyquist stability criterion states that a closed loop system is unstable if

$$\begin{cases} |G(\omega)F(\omega)| \geq 1 \\ \angle G(\omega)F(\omega) = n2\pi, \quad n \in \mathbb{Z} \end{cases} \tag{2.2}$$

holds for any ω [14]. This is equivalent to having a pole of the system with a positive real part. The consequence is that a maximum $\bar{G}$ exists for which the system is on the edge of instability. This value of $\bar{G}$ is known as the maximum stable gain (MSG). As reducing or eliminating feedback is dependent on the gain (which can often be freely chosen by the user of the PA system), the feedback problem can be rephrased to trying to achieve an MSG as high as possible without causing too much disturbance on the source signal.

Another important concept is the gain margin. It is defined as the difference between the MSG and the actual gain of a system. While a system with a gain almost equal to the MSG is still theoretically stable, many practical effects can reduce the MSG. A gain margin of 2–3 dB is recommended [2, 15].

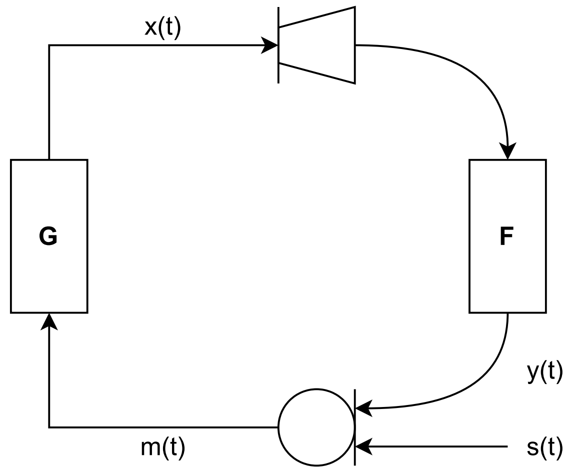


Figure 2.1: A simple model of a one channel PA system with unfiltered feedback.

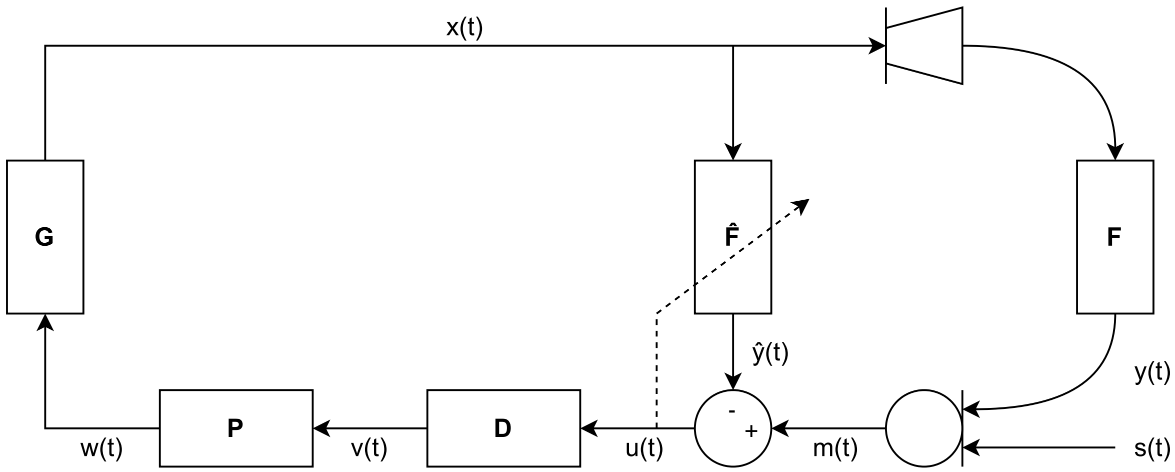


Figure 2.2: A simple model of a one channel PA system with AFC filtering.

2.2. Feedback Filtering with AFC

The most promising method of feedback reduction is adaptive feedback cancellation (AFC) as mentioned in Section 1.1. This approach is based on estimating the RIR and subtracting it from the microphone signal. Figure 2.2 shows an overview of the system.

The adaptive filtering module $\hat{F}$ is responsible for adaptively modelling the acoustic feedback path of the sound signal from loudspeaker to microphone. The loudspeaker signal $x(t)$ is filtered with $\hat{F}$, yielding a feedback signal estimate $\hat{y}(t)$. The module linked to this project [12] is based on a multidelay Block Frequency Domain Adaptive Filter (MDF).

Because of a potentially substantial correlation between the source and loudspeaker signals, the adaptive filter can converge to a biased estimate. This will lead to partial filtering of the source signal as well as the feedback. To prevent this from happening, a decorrelation module D is inserted in the system, decorrelating the feedback $y(t)$ from the source signal $s(t)$ and thereby increasing the performance of the adaptive filter. The module linked to this project [13] is based on frequency shifting. The empirical mean correlation between the input and output of the module is under 0.05.

The estimate of the adaptive filter is not a perfect copy of the feedback; a residue signal $r(t) = y(t) - \hat{y}(t)$ will persist. A postfilter P can be used before or after decorrelation to further minimise the residue signal or to prevent instability. The following chapters will discuss postfilter implementations in full detail.

3
Programme of Requirements

In this chapter the requirements that need to be met by the product are listed. The program of requirements is divided into two parts:

- requirements for the complete system;
- requirements for the postfiltering submodule.

The requirements can be divided into two types:

- mandatory requirements;
- trade-off requirements.

The mandatory requirements (denoted by 'must') describe properties the final design must have for it to be considered successful. The trade-off requirements (denoted by 'should') describe properties of the final design that can be implemented to different degrees of satisfaction. Each of these types of requirements can be further divided into functional and non-functional requirements. Functional requirements describe what the the product should do and in what way it should do this. Non-functional requirements describe the aspects of the product that do not relate to its operation directly.

3.1. Requirements for the Complete System

In this section all requirements concerning the complete system that is to be created between the subgroups are listed. The mandatory and trade-off functional requirements are as follows.

1. The system must be realisable as a module that can be inserted into the electrical part of an existing PA system using the incoming microphone signal and the outgoing loudspeaker signal.
2. The system must be suitable for both speech and music.
3. The system must have a bandwidth that at least covers the audio frequencies between 20 Hz and 20 kHz, which is the frequency band that can be heard by a human. The sample frequency f_s should therefore be at least 40 kHz following the Nyquist—Shannon sampling theorem [16].
4. The system must operate automatically without any human interaction. Initialisation of the system could still require human interaction.
5. The system must operate in real time applications: sample based calculations must not take longer than the sample period $T_s = 1/f_s$ and frame based calculations must not take longer than the frame period $T_f = N_f/f_s$.
6. The system should increase the MSG as much as possible, preferably by at least 10 dB. This is the maximum achievable increase of the MSG when using the commonly used PFC or NHS feedback suppression methods [1].

7. The system should output a signal with highest possible audio quality. The quality is measured using the PEAQ algorithm [17]. An ODG of above -0.5 (as good as imperceptible) is preferred.

8. The system should introduce as little delay and jitter as possible to the audio signal.

The mandatory and trade-off non-functional requirements are as follows.

9. The system must be designed in the given time span.

10. The system must be suitable for indoor and outdoor environments and must be operational in typical indoor and outdoor temperature ranges.

11. The system must be suitable for Europe mains electricity consumption with a nominal voltage of 230 V and a frequency of 50 Hz.

12. The system should be able to be produced for as cheap a price as possible. A price in the order of below €100 is deemed to be acceptable.

3.2. Requirements for the Postfiltering Submodule

Apart from the previously mentioned requirements, these specific requirements also apply to the postfilter module. The extra mandatory and trade-off functional requirements are as follows.

13. The postfilter must prevent all instability in the system passed on (or created) by the other modules.

14. The postfilter should output audio signals without any quality loss when no howling is present.

15. The postfilter should not attenuate frequencies where no howling is detected more than 3 dB.

The postfilter module does not introduce any new non-functional requirements.

4 Simulation Environment

In order to test implementations of the postfilter virtually, a simulation environment is needed that describes the PA system. The MATLAB-based graphical programming environment Simulink is chosen for this task as it allows for easy and efficient modelling, simulating and analysing of dynamical systems. The MATLAB workspace is used in combination to provide signals and filter coefficients.

On the one hand, the complete model of the PA system for the postfilter should ideally include models of the adaptive filter and decorrelation modules. However, making accurate models of these modules depends on the progress of the other subgroups, which is available only near the end of the project. On the other hand, Chapter 5 concludes that residual feedback filtering will not be applied. Howling detection and suppression should not work any different with or without AFC—it will only be applied more often. For these reasons, the PA model does not contain an AFC model. Instead, the final postfilter will be tested directly together with the final adaptive filtering and final decorrelation.

4.1. Model of the PA System

An overview of the created simulation environment is shown in Figure 4.1. The input and output of the system are linked to the MATLAB workspace by means of the *From Workspace* and *To Workspace* blocks. The *From Workspace* block imports a sound signal array from the workspace, sampled at 44.1 kHz. This is the sampling frequency in the CD format and satisfies Requirement 3. Three different types of input signals are used, all fifteen seconds and monophonic:

- English speech: a fragment of the script of the movie ”Bee Movie” read by VoiceoverPete;
- instrumental music: a fragment of the intro of ”Smooth” by Santana featuring Rob Thomas;
- music with vocals: a fragment of the first verse of ”Take On Me” by A-ha.

These signals are chosen to test the performance of the PA system for speech and musical applications as it should be able to handle both. The *To Workspace* block saves the received audio signal to the workspace as a similar array.

The gain of the PA system is implemented by a *Gain* block, multiplying the signal that enters with a scalar. This is a simplified version of the gain described in Chapter 2, as it does not include any frequency dependence that might be present in the gain of a PA system. This choice has been made because this frequency dependence differs between PA systems and a gain that does not depend on the frequency is generally seen as ideal. A *Saturation* block is included to mirror the maximum volume that can be produced by a loudspeaker. It also prevents the amplitude of the output signal from getting too large when the system becomes unstable, which can cause the simulation to crash. The saturation limit is set sufficiently high to ensure howling can still occur.

The *RIR Filter* block implements the room impulse response as well as the loudspeaker and microphone transfers. This is used to imitate the effect of the audio signal travelling from the loudspeaker, through the room, into the microphone. A discrete finite impulse response filter is used to do so. This filter contains the room impulse response as coefficients. The impulse response of a physical room is

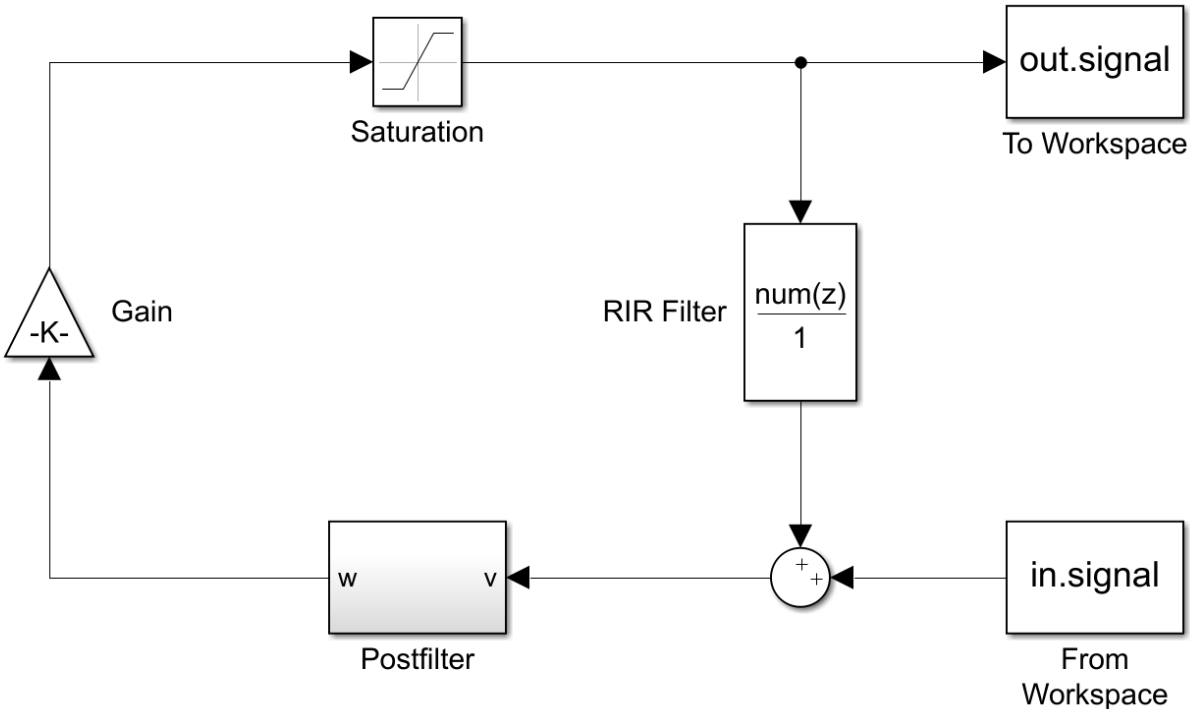


Figure 4.1: Overview of the simulation environment.

used in the simulation. It is obtained by making impulse response measurements. How this is done will be explained in the next section.

The *Postfilter* block is used to implement the functionality of the postfilter. It initially only contains a connection from input to output. Depending on what feature of the postfilter is being implemented, the *Postfilter* block can be expanded.

4.2. Room Impulse Response Measurements

To ensure that the implementation of the room impulse response is realistic, several room impulse responses are measured in a hall often used for musical performances. Different distances between microphone and louspeaker are used. A MATLAB script has been created to perform these measurements. This MATLAB script can be found in Appendix B.1. The method of measuring the impulse response relies on the use of a logarithmic sine sweep signal. This method is described by Stan, Embrechts and Archambeau [18]. The method has been chosen as it performs well in an unoccupied and quiet room, which is applicable to the used test room.

The MATLAB script generates a signal that increases in frequency over time according to

$$x(t) = \sin\left(\frac{2\pi f_1 T}{\ln\left(\frac{f_2}{f_1}\right)}\left(e^{\frac{t}{T}\ln\left(\frac{f_2}{f_1}\right)} - 1\right)\right). \tag{4.1}$$

The starting and ending frequencies are chosen to be $f_1 = 10$ Hz and $f_2 = 22$ kHz respectively. These values are chosen to ensure that the frequency band of interest, namely the band between 20 Hz and 20 kHz as stated in Requirement 3, is represented in the impulse response. The frequency of the signal increases logarithmically over time from the starting to the ending frequency, in a $T = 1.5$ s period of time. This signal is played through the loudspeaker and subsequently recorded by the microphone. To obtain the room impulse response from the recorded signal, it is linearly convolved with the flipped version of the sine sweep signal $x(t)$. $x(t)$ is flipped by reversing the signal and delaying it so that it exists in the positive part of the time axis. The magnitude spectrum of the result is divided by the squared magnitude spectrum of the original sine sweep signal $x(t)$ to obtain the room impulse response. The acquired room impulse responses will be used for simulations during the design of the postfilter.

5

Residual Feedback Filtering

One way to improve the signal that results from adaptive filtering is to identify the feedback that still resides in the feedback compensated signal and then attempt to remove this residual feedback. Several methods are opted for identifying and removing residual feedback from a signal. Two of these methods are looked into in this chapter.

5.1. Residual Feedback Filtering Based on Magnitude Spectra

One method of removing residual feedback from the signal is proposed by Janse and Belt [8]. This method is implemented in the frequency domain and uses the magnitude responses of several signals to determine the magnitude response of the postfilter that removes the residual feedback from the signal. The method is based on two equations

$$\left|P(\omega;k)\right| = \delta\left|P(\omega;k-1)\right| + (1-\delta)\left|\tilde{P}(\omega;k)\right| \tag{5.1}$$

and

$$\left|\tilde{P}(\omega;k)\right| = \max\left(\frac{\left|M(\omega;k)\right| - \gamma\left(\left|\hat{Y}(\omega;k)\right| + \left|\hat{R}(\omega;k)\right|\right)}{\left|V(\omega;k)\right|}, 0\right) \tag{5.2}$$

where $|M(\omega;k)|$, $|\hat{Y}(\omega;k)|$ and $|V(\omega;k)|$ are the magnitude spectra of the corresponding signals shown in Figure 2.2. $|\hat{R}(\omega;k)|$ is the estimated residual feedback. The γ in (5.2) is the subtraction factor. The value of γ depends on the maximum loop gain. The δ in (5.1) indicates the dependence of $|P(\omega;k)|$ on its former value an the value of $|\tilde{P}(\omega;k)|$ with $0 < \delta < 1$. A high value for δ limits the rate at which $|P(\omega;k)|$ can change while a low value for δ enables $|P(\omega;k)|$ to change rapidly, thus δ can be used for smoothing of $|P(\omega;k)|$.

This method of residual feedback removal depends on the magnitude spectrum of the estimate of the residual feedback signal $|\hat{R}(\omega;k)|$. This residual feedback signal is equal to the difference between the source signal entering the microphone $s(t)$ and the feedback compensated signal $v(t)$ as shown in Figure 2.2. The estimation of the residual feedback is paramount to the accuracy of the postfilter, as a perfect estimate completely eliminates the residue but an inaccurate estimate can make the audio quality even worse. Here the problem arises that the signal $s(t)$ is not readily available, because the signal that is received by the microphone also contains the feedback signal $y(t)$. Determining the residual feedback is thus not a straightforward task. Janse and Belt [8] do not provide a method of finding an accurate estimate of the residual feedback signal $r(t)$. This implementation is therefore incomplete and needs a practical method of estimating the residual feedback for it to work.

5.2. Residual Feedback Filtering Based on Estimating Power Spectral Densities

Another method of removing residual feedback from the signal is proposed by Ortega, Lleida and Masgrau [7] and Gallego, Lleida, Masgrau, and Ortega [9]. This method is based on the fact that the ideal frequency response of the postfilter would be equal to

$$P(\omega;k) = \frac{1}{1+G(\omega;k)R(\omega;k)}, \tag{5.3}$$

exactly cancelling the effect that the room and channel estimation have on the source signal. This equation can be transformed into [7]

$$P(\omega;k) = 1-\sqrt{\frac{S_r(\omega;k)}{S_v(\omega;k)}}. \tag{5.4}$$

Here $S_r(\omega;k)$ is the power spectral density of the residual feedback signal $r(t)$ and $S_v(\omega;k)$ is the power spectral density of the feedback compensated signal $v(t)$. (5.4) shows that the frequency response of the postfilter can be determined when the power spectral densities of the residual feedback signal and feedback compensated signal are found. The power spectral density of the feedback compensated signal can be estimated from $v(t)$ directly. The power spectral density of the residual feedback signal poses a bigger challenge. As mentioned before, the residual feedback signal $r(t)$ is not readily available due to the fact that the source signal $s(t)$ is combined with the feedback signal $y(t)$ as shown in Figure 2.2. The residual feedback can therefore not be determined by directly comparing the feedback compensated signal $v(t)$ with the source signal $s(t)$. Because of this, another method is needed to find the power spectral density of the residual feedback signal. Estimating $S_r(\omega;k)$ can be done recursively [7]. This method uses the relations

$$\hat{S}_r(\omega;k) = \delta\cdot\hat{S}_r(\omega;k-1)+(1-\delta)\cdot\tilde{S}_r(\omega;k) \tag{5.5}$$

and

$$\tilde{S}_r(\omega;k) = \left(\lambda+(1-\lambda)\frac{\hat{S}_r(\omega;k-1)}{\hat{S}_v(\omega;k-1)}\right)^2\hat{S}_v(\omega;k-1). \tag{5.6}$$

(5.6) provides an instantaneous estimation of the power spectral density of the residual feedback signal. Here λ is the bias term, ensuring the estimate of the residual feedback spectral density does not end up converging to zero. In (5.5) time-averaging is achieved, similar to (5.1). Using the estimate of $S_r(\omega;k)$ provided by (5.5) and (5.6) and the power spectral density of the feedback compensated signal $s_v(\omega;k)$, (5.4) can be used to determine the desired frequency response of the postfilter.

The method of estimating the residual feedback spectral density explained above has some drawbacks. Because the estimation method relies on recursion, the initial residual feedback spectral density estimate is very important. Determining a proper residual feedback spectral density estimate for initialisation of the system provides a similar problem to the method discussed in Section 5.1, where the needed information is not available in the system. An inaccurate estimation of the initial residual feedback spectral density can be corrected over time through recursion, but the performance of the postfilter will initially be poor. This could even result in a reduction of the sound quality of the PA system, as the audio signal is incorrectly compensated by the postfilter which can result in alteration of components of the source signal that are present in $v(t)$ which should be left untouched.

Another drawback of Ortega's method is that it is mostly aimed at speech applications. Mel scale based frequency smoothing [19] can be used on the power spectral density of the feedback compensated signal to improve the performance of the postfilter by more effectively filtering out frequencies that are sustained for a longer period of time. This is useful in speech applications because speech generally does not contain frequencies that are sustained for a long period of time. The adaptive feedback cancellation system should be usable for both speech and musical applications as stated in Requirement 2. Music generally does contain frequencies that are sustained for longer periods of time. The Mel scale based frequency smoothing can therefore not be used, as this would affect the quality of musical audio signals presented by the PA system.

5.3. Consideration of the Residual Feedback Filtering Methods

The use of residual feedback filtering is based on the assumption that adaptive filtering yields suboptimal results. While this is indeed the case in practical applications of adaptive filtering, residual feedback filtering might not be the most effective way of reducing the loss of audio quality in the AFC filter. As mentioned above, the available methods for residual feedback filtering suffer from incomplete knowledge of the residue signal that needs to be removed. There is little literature available on the estimation of this residue signal and improper estimation could even cause a deterioration of the audio quality. Solving this problem is outside the scope of this bachelor graduation project, because the bachelor did not provide sufficient knowledge and there is too little time allocated for the project to research the subject adequately, meaning Requirement 9 would not be met. Improving the performance of the AFC filter is therefore better done by improving the adaptive filtering and decorrelation stages, where the problem of incomplete information is less present. It has been decided that residual feedback filtering will not be part of the final product. Instead the postfilter will focus on increasing stability, which is conform with Requirement 13, as this would add more value to the AFC filter than residual feedback filtering does.

6 Howling Detection

A second way to improve the signal that results from the channel estimation is to specifically detect howling frequencies and subsequently taking measures to remove this howling from the signal, as shown in Figure 6.1. The howling detection module takes the input signal $v(t)$ of the postfilter, and outputs one or more frequencies that correspond to howling, represented by the set $Q(k)$. This set of frequencies is calculated per frame k and passed on to the howling suppression module (described in Chapter 7).

The detection of howling frequencies can be done in many different ways. The reliability of the howling detection method can have a large impact on the performance of the postfilter. In this chapter, two different groups of howling frequency detection techniques are implemented and evaluated: feature comparison after peak detection and feature comparison before peak detection. Of all possibilities, the best option is chosen to be integrated in the final postfilter design.

6.1. Peak Detection

Howling detection makes use of spectral analysis. Here the power spectrum of the audio signal is used to identify howling frequencies. At these frequencies peaks occur in the power spectrum, as a large part of the signal power is concentrated at the howling frequency. The peaks in the spectrum corresponding to howling need to be distinguished from other peaks that correspond to components of the actual audio signal.

The first step in this process is calculating the power spectrum of the input signal $\left|V(\omega,k)\right|^2$. This is done with a discrete Fourier transform on the input signal per frame. A large frame length of $N_f = 4096$ samples is chosen to provide a high frequency resolution (as in [20]). Section 7.2 shows why a higher resolution produces a higher quality. Frames can overlap, with a higher overlap generally meaning a faster suppression response but more computational complexity. An overlap between 25 % and 50 % is found to be a good compromise [21]. In the following simulations, an overlap of 50 % is chosen leading to a hop size of $N_h = 2048$ samples, but this value can be changed when the computational complexity needs to be lowered.

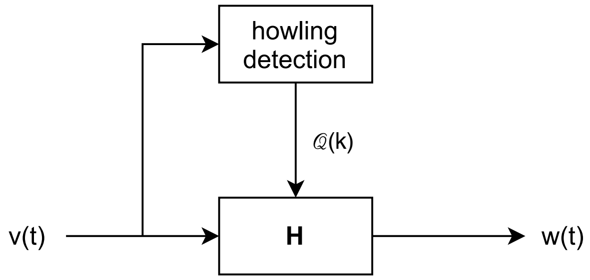


Figure 6.1: Model of the postfilter based on filtering howling frequencies, consisting of a detection module and a suppression module H. $Q(k)$ is a set of design parameters determined per frame.

Taking a single frame from the whole input signal is equivalent to multiplying the whole signal with a rectangular window function (with values 1 at samples inside the frame and 0 outside the frame). Computing the Fourier transform of the framed signal will therefore contain unwanted effects of the frame selection (a convolution in frequency domain with a sinc function). This effect is called spectral leakage. The solution to this problem is to multiply the whole input signal with a better window function $b(t)$ before taking the Fourier transform, resulting in

$$V_b(\omega;k) = \frac{1}{N_f}\sum_{t=0}^{N_f-1} b(t)v(t;k)e^{-j\omega t} = B(\omega) * V(\omega;k). \tag{6.1}$$

To get the framed signal as closely matching to the original input signal as possible, $B(\omega)$ should approximate the Dirac delta function as closely as possible. Harris [22] lists a multitude of possible window functions, concluding with a recommendation of either the Kaiser–Bessel window [23] or the Blackman–Harris windows [24]. The Blackman–Harris windows particularly perform best for detecting nearby tones of significantly different amplitudes. For the following simulations, the (nonexact) Blackman window

$$b(t) = \begin{cases} 0.42 - 0.50\cos\left(\frac{2\pi t}{N_f}\right) + 0.08\cos\left(\frac{4\pi t}{N_f}\right), & 0 \le t < N_f \\ 0, & t < 0 \lor t \ge N_f \end{cases} \tag{6.2}$$

is used. Time and frequency domain plots can be found in Figure A.1.

The magnitude of the resulting framed spectrum is then squared to obtain the power spectrum. This spectrum is used for feature comparisons in the following sections. Before detecting the highest peak in the spectrum, a mask is applied that hides frequencies below 20 Hz. This frequency range falls below human hearing and can be filtered out completely without loss of quality, adhering to Requirement 3. The resulting peak frequency $\breve{\omega}$ is passed on to the feature comparison as well as the peak magnitude.

The Simulink implementation of the peak detection is fairly straightforward. The input signal enters a buffer to divide the signal into frames. A window function block applies the Blackman window to the framed input. The fast Fourier transform (FFT) algorithm is used for the discrete Fourier transform. The magnitude squared FFT block combined with a gain block that divides by the frame length squared provide this operation. The resulting signal can be used for different feature comparison techniques. Before using a block that finds the highest peak and outputs the index and magnitude, the power spectrum is multiplied elementwise with a mask array that is defined as a constant to hide frequencies below 20 Hz for the reason mentioned before.

6.2. Post-Peak Feature Comparison

Following Van Waterschoot and Moonen [21], six different signal features are commonly used to determine whether a detected peak corresponds to howling. In this section, these methods—as well as a modification on and a combination of these features—are explained and implemented. An evaluation of the performance of these methods can be found in Section 6.4.

Peak to Threshold Power Ratio

The most obvious method of howling identification is to compare the signal power at the detected peak to a set threshold, known as the peak to threshold power ratio method, PTPR for short. This is expressed in the inequality

$$\frac{\left|V(\breve{\omega};k)\right|^2}{P_0} \ge T_{PTPR}. \tag{6.3}$$

Here P_0 is the absolute power threshold. This parameter depends on properties of the physical setup like the distance between loudspeaker and microphone and is used to tune the implementation to the environment. T_{PTPR} is used as the howling threshold. Any value that surpasses this threshold is identified as howling. By multiplying both sides of (6.3) by P_0, the absolute power threshold can be combined with T_{PTPR} into a single threshold. This simplifies implementation. An implementation of PTPR using Simulink is shown in Figure A.2.

Peak to Average Power Ratio
Peak to average power ratio, PAPR for short, is a method used for identification of howling that extends upon the PTPR method. Instead of comparing the detected peak to an absolute power threshold P_0, the average power $\bar{P}_v(k)$ of a frame is used as shown in

$$\frac{\left|V(\breve{\omega};k)\right|^2}{\bar{P}_v(k)} \geq T_{PAPR}, \qquad \bar{P}_v(k) = \frac{1}{N_f}\sum_{n=0}^{N_f-1}\left|V(\omega_n;k)\right|^2 \tag{6.4}$$

removing the need to manually tune the implementation to the physical setup by finding a proper value for P_0. After each frame the average power $\bar{P}_v(k)$ is updated to the average power of the most recent frame. A Simulink implementation of PAPR is shown in Figure A.3.

Peak to Local Average Power Ratio
Instead of comparing the signal power at the peak to the average of the whole spectrum, the average of a frequency subband around $\breve{\omega}$ can also be used. This approach will be referenced to as peak to local average power ratio (PLAPR). The spectrum is divided into a number of bins of sample length N_b, and the average power of bin b containing $\breve{\omega}$ is used in place of the total average power resulting in

$$\frac{\left|V(\breve{\omega};k)\right|^2}{\bar{P}_v^b(k)} \geq T_{PLAPR} \tag{6.5}$$

where

$$\bar{P}_v^b(k) = \frac{1}{N_b}\sum_{n=bN_b}^{(b+1)N_b-1}\left|V(\omega_n;k)\right|^2, \quad \breve{\omega} \in \{\omega_n \middle| n = bN_b, \cdots, (b+1)N_b - 1\}. \tag{6.6}$$

This method theoretically improves determining whether a peak is considered howling, especially for signals with highly varying local averages. The bin size parameter has great influence on the performance, where $N_b = N_f$ should act identical to PAPR and $N_b = 1$ would either always or never return howling (depending on T_{PLAPR}). The PLAPR method implementation can be found in Figure A.4.

Peak to Harmonic Power Ratio
Another property that can be used to identify howling in an audio signal is the presence of harmonics in the spectrum. This is done in the peak to harmonic power ratio method, PHPR for short. Harmonics are audio components at an integer multiple of the frequency of the original audio component. While tones in speech and music typically have harmonics, howling does not. This property can be used to distinguish between the audio signal and howling. In

$$\frac{\left|V(\breve{\omega};k)\right|^2}{\left|V(m_h\breve{\omega};k)\right|^2} \geq T_{PHPR} \tag{6.7}$$

the power of the detected peak is compared to that of its m_hth harmonic. When the peak corresponds to a part of the audio signal this comparison should yield a small value because the harmonics should be present in the audio signal and therefore also create peaks in the power spectrum. When this comparison is done for a peak corresponding to howling a large value is expected as the audio signal should not contain harmonics of the howling frequency. The result of the comparison of the detected peak with its harmonics is compared with the threshold value T_{PHPR} to decide whether the peak is considered to be howling. This method can be extended by comparing the detected peak to several of its harmonics at the same time. This is done by determining the peak to harmonic power ratio for each of the harmonics separately and only identifying the peak as howling when all of these peak to harmonic power ratio surpass the set threshold. The detected peak can also be compared to its subharmonics. These are located at the frequency of the peak divided by an integer value.

It should be noted that the harmonics of a detected peak are not always part of the spectrum, as a multiple of the frequency at the detected peak might be outside of the spectrum bandwidth. When this is the case it is not possible to use the peak to harmonic power ratio as a means of identifying howling. To ensure the implementation also works for detected peaks without harmonics within the spectrum bandwidth, the spectrum is taken and extended by concatenating an array containing the average signal power multiplied by the threshold T_{PHPR} and divided by the threshold T_{PAPR}. By doing this the detected peaks that have harmonics outside of the spectrum bandwidth will be compared to the average signal power using the peak to average power ratio method. A Simulink model that implements the PHPR howling identification method along with the mentioned extensions can be found in Figure A.5.

Peak to Neighbouring Power Ratio

Another signal feature that can be used to identify howling is the ratio between the signal power at detected peak and the signal power at its neighbouring frequencies. The peak to neighbouring power ratio method, PNPR for short, makes use of the assumption that howling is a sinusoidal signal, creating a very sharp peak in the power spectrum. Because tones in music and speech typically are not purely sinusoidal, the peaks created in the power spectrum are generally wider than those caused by howling. This difference in peak width can be evaluated using

$$\frac{\left|V(\breve{\omega};k)\right|^2}{\left|V\left(\breve{\omega}+\frac{2\pi m_n}{N_f};k\right)\right|^2} \geq T_{PNPR} \tag{6.8}$$

with m_n the number of the neighbouring frequency and N_f the frame length. For detected peaks that correspond to speech or music the peak to neighbouring power ratio should yield a relatively small value as the neighbouring frequencies would also be part of the detected peak. When a peak is detected that corresponds to howling, the peak to neighbouring power ratio should be high, as the neighbouring frequencies are not part of the peak and should thus have a significantly lower power than the peak itself. This difference is utilised to differentiate between howling and the actual audio signal. Similar to the peak to harmonic power ratio method, the peak to neighbouring power ratio method can be extended by comparing the peak to several of its neighbours simultaneously. It is possible that the detected peak is located at the edge of the power spectrum. If this is the case its neighbours might not exist. To ensure the implementation will still work when this is the case, the power spectrum is extended in the same way as the peak to harmonic ratio method, using the peak to average ratio method to evaluate these detected peaks. An implementation of the peak to neighbouring power ratio method can be found in Figure A.6.

Temporal Feature Comparisons

Instead of comparing the detected peak to other frequencies, different temporal features can also be used. Interframe peak magnitude persistence (IPMP) is based on the assumption that howling persists longer than a single frame in speech and most music. A specific howling frequency should be detected multiple times in the past K frames. The number of occurrences divided by K can be compared to a threshold between 0 and 1 to test whether howling occurs or not.

Interframe magnitude slope deviation (IMSD) is based on the assumption that howling increases exponentially. The deviation of the slope of the detected peaks (in logarithmic scale) is calculated over the past K frames and compared to a threshold. Low values indicate an exponential increase meaning that howling is present.

A problems arises with these comparison methods: the detection gets more accurate for a larger K. This means that accuracy comes at the direct expense of response speed. Van Waterschoot and Moonen [21] propose $K = 20$ for IPMP and $K = 32$ for IMSD which both yield mediocre results but reduce the response time to $N_f K/f_s \approx 2$ s and 3 s respectively. $K = 16$ is suggested for the more successful combination of PHPR, PNPR and IMSD with a response time of 1.5 s. Notable howling can however already occur in this timeframe. The only solution would be to decrease the frame length (as in [25]), but that would mean a lower frequency resolution and consequently a lower audio quality (as explained in Chapter 7). The conclusion is that temporal feature techniques are not appropriate for this project following Requirement 13.

Peak to Harmonic and Neighbouring Power Ratios

An effective way to possibly improve correct detection is to combine different criteria. The two most convoluted criteria are PHPR and PNPR, which are both based on different principles. When combined, a criterion is created based on the union of conditions (6.7) and (6.8). An implementation of this principle is shown in Figure A.7. While a significant improvement of correct detection is possible, it should be noted that such a combination does increase computational complexity and also requires more design parameters to be chosen.

6.3. Pre-Peak Feature Scaling

In all previous methods, the feature comparison is done after detecting the peak. This means howling can only be suppressed after it exceeds all other frequency components. On the one hand, this should always be the turning point where howling becomes dominant in a signal. On the other hand, if howling is to be detected before exceeding all other frequency components, detection (and subsequently suppression) can be done earlier. This would in theory reduce audible howling even further.

Because the previously mentioned principles behind detection still hold, the same techniques can be implemented before peak detection. Instead of scaling only the peak value, the whole frequency range has to be scaled. The peak detection now finds the frequency for which the chosen criterion holds most. As the threshold and average criteria are independent of the peak frequency, these methods would not be different from their post-peak counterparts. For this reason only PLAPR, PHPR and PNPR will be transformed to pre-peak scaling methods.

Spectrum to Local Average Power Ratio

Similar to PLAPR, spectrum to local average power ratio (SLAPR) calculates the average power per bin of frequencies. But now the whole spectrum is divided by the average power of the respective bins. This makes it possible to detect the highest peak relative to it's bin. An implementation of SPLAPR using Simulink is found in Figure A.8.

Spectrum to Harmonic Power Ratio

The spectrum to harmonic power ratio (SHPR) is found by dividing the spectrum by a scaled-down version of itself, such that every frequency component is divided by the frequency component of double the frequency. This is implemented in Simulink with the *Downsample* block, taking every odd entry in the spectrum array. After this peak is detected, normal comparisons with harmonics are made as in PHPR, to test if the peak is indeed howling. The Simulink implementation is shown in Figure A.9. As with PHPR, the shortage in range of the scaled spectrum is supplemented with values of the average power.

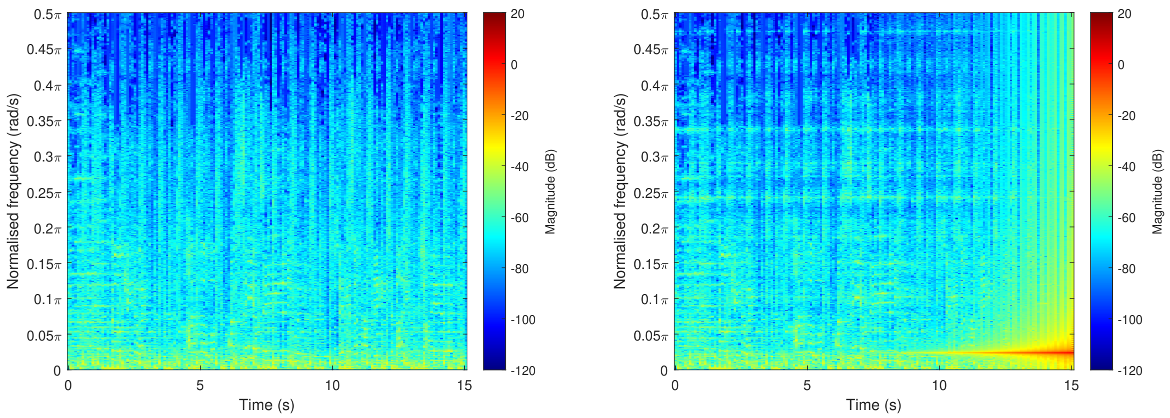
Spectrum to Neighbouring Power Ratio

The last pre-peak method is spectrum to neighbouring power ratio (SNPR), a similar method to PNPR. Before detecting the highest frequency component, the spectrum is scaled by neighbouring values. As multiple neighbours should be included in this scaling, a geometric mean of the neighbouring frequency components is taken (which has the same result as taking a arithmetic mean in decibel scale). The peak detection now finds the highest frequency component in contrast to its neighbours. Figure A.10 shows an implementation of SNPR for $m_n \in \{-2, -1, 1, 2\}$. This implementation is also supplemented with values of the average power.

6.4. Evaluation of Howling Detection Methods

To evaluate the different detection modules and make a comparison, a practical situation has to be tested for each module. The PA system model is used in combination with the fifteen second input signal containing both sung speech and music as described in Section 4.1. The spectrogram of the audio signal is shown in Figure 6.2a. The chosen room impulse response and broadband gain produce howling at 530 Hz that starts to become clear around eight seconds. The spectrogram of the audio signal with generated howling is shown in Figure 6.2b.

The detection problem can be seen as a binary classifier system with varying discrimination threshold. Frames at which howling is present are positive conditions (with N_P the number of positive conditions) and frames without howling are negative conditions (N_N). Frames at which howling is successfully detected are true positives (N_{TP}) and frames at which howling is detected where there is none are false positives (N_{FP}). In the test setup, the boundary for when howling starts is chosen as starting from frame 175. While this boundary is somewhat arbitrary and only a single test setup is used, comparative (relative) evaluation still holds.



(a) Spectrogram of the audio fragment used as input.

(b) Spectrogram of the audio fragment with generated howling.

Figure 6.2: The spectrogram of the input audio signal and the signal with generated howling

With this data, the probability of detection can be calculated as $P_D = N_{TP}/N_P$ and the probability of false alarm as $P_{FA} = N_{FP}/N_N$. Varying the thresholds T will produce different values for P_D and P_{FA}. A plot containing both these probabilities is called a receiver operating characteristic curve (ROC curve) and is used to measure the performance of a feature comparison method. Figure A.11 shows the ROC curves for all six post-peak feature comparisons. As the postfilter should be used only as backup, keeping the probability of false alarm as small as possible is prioritised as to comply to Requirement 14. After prioritising a small probability of false alarm, the probability of detection should be as close to 1 as possible.

PTPR and PAPR show very similar results, both relatively poor, although it should be noted that PAPR can adapt better to dynamic signal and should therefore always be preferred. PLAPR appears also very similar to PAPR or even worse for too small bin sizes. While this could partly be due to the specific circumstances of the test setup, taking local averages does not seem to be worth the added complexity. PHPR without subharmonics greatly improves results. PNPR starts detecting howling from a significant probability of false alarm, rendering the criterion on its own less useful. The combination of PHPR and PNPR is the most promising, yielding the best results while also being completely dynamic. With thresholds $T_{PHPR} = 25$ dB and $T_{PNPR} = 2$ dB, a P_D is achieved of over 95 % while keeping P_{FA} under 1 %.

The pre-peak feature scaling implementations show varying results. Figure A.12 shows the ROC curves of these implementations. The ROC curves of the SLAPR method look quite promising, as they are similar to those of its post-peak detection counterpart PLAPR. It however only works well for fairly large bin sizes, reducing the added value of pre-peak howling detection: a large bin size (or low amount of bins) results in an implementation that approaches that of PAPR. Because the detection performance shown in the ROC curves of SLAPR, even for large bin sizes, is significantly worse than that of some of the post-peak howling detection methods, it is concluded, based on Requirement 13, that the minimal added value it offers by providing earlier detection of howling does not justify the poorer performance of the detection. When looking at the ROC curves of SHPR it can be seen that the probability of detection saturates far below the ideal value of 1, indicating that this method sometimes fails to identify howling in the audio signal irrespective of the used thresholds. It seems that, although performing the feature comparison before the peak detection theoretically allows for more timely identification of howling, this also decreases the effectively of the feature comparison methods used by SHPR by increasing the probability that howling is not identified as such. Because accurate detection of howling is a priority, as formulated in Requirement 13, this pre-peak howling detection method is discarded for the final postfilter design. SNPR looks more promising, as it approaches the ideal probability of detection of 1. It does this however at a relatively high probability of false alarm, meaning that this high probability of detection can not be achieved without falsely identifying non howling frequencies as howling. This again is not conform with Requirement 13, making SNPR not suitable for the final postfilter design.

7

Howling Suppression

After detecting frequencies where howling is present, the corresponding howling has to be filtered out of the input signal of the postfilter. This function is performed by the howling suppression module, represented by H in Figure 6.1. The aimed filter could be either linear or nonlinear, but has to be time-variant. This is because the detected howling frequency can change over time, and the filter must be disabled when no howling is detected (following Requirement 14). This chapter discusses the different filter options, highlights the design of the chosen filter type in detail and concludes with the Simulink implementation and evaluation.

7.1. Filter Choice

The most straightforward howling suppression methods are based on gain reduction. A simple but inelegant approach is automatic gain control (AGC): reducing the gain equally in the entire frequency range whenever any howling is detected [26]. The main strength is the reliability, as instability is guaranteed to be resolved with enough broadband gain reduction. The drawback is that it does not increase the MSG of the system as it also reduces frequency components without howling. This approach is therefore mostly reserved as backup (as in [5, 27]). Automatic equalisation (AEQ) is similar to AGC, but instead of reducing the broadband gain only the frequency subband containing the howling is reduced (as in [27], where the detection is also done per subband). When howling occurs, most subbands are still left intact and the audio does not completely disappear. This results in an MSG increase. The last gain reduction method, notch filter based howling suppression (NHS), takes AEQ further by applying a bandstop filter to the signal with centre frequency set to (or in the vicinity of) the howling frequency. Combined with an accurate howling detection, frequencies without howling are preserved and howling itself is attenuated. For this reason, NHS is chosen as howling suppression method.

There are two options available for implementing a bandstop filter. The first option is to design the bandstop filter using a finite impulse response (FIR). This kind of filter makes use of a finite number of delayed input samples for its implementation. FIR filters are always stable. The transfer function of an Nth order FIR filter is

$$H(z) = \sum_{n=0}^{N} b_n z^{-n} = b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}, \quad z = e^{j\omega}. \tag{7.1}$$

An infinite impulse response (IIR) filter directly depends on delayed samples of both input and output, making the filter recursive and therefore indirectly dependent on all previous input samples. The transfer function of an IIR filter with feedforward order N and feedback order M is

$$H(z) = \frac{\sum_{n=0}^{N} b_n z^{-n}}{1 + \sum_{m=1}^{M} a_m z^{-m}} = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}}. \tag{7.2}$$

The advantage of using an IIR filter over an FIR filter is that it allows for more efficient implementation: the same filtering performance can be achieved with a much lower order filter when compared to an FIR

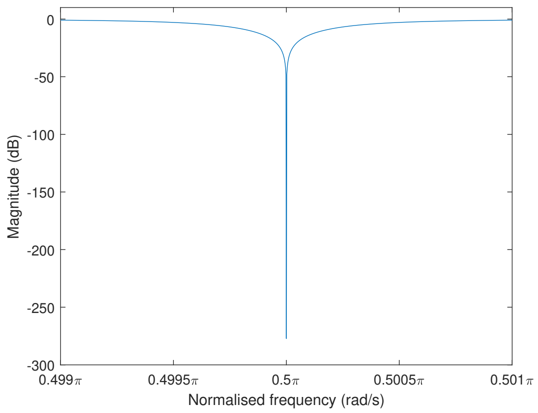


Figure 7.1: Magnitude transfer of a second order IIR bandstop filter with $\omega_0 = 0,5\pi$ rad s^{-1}.

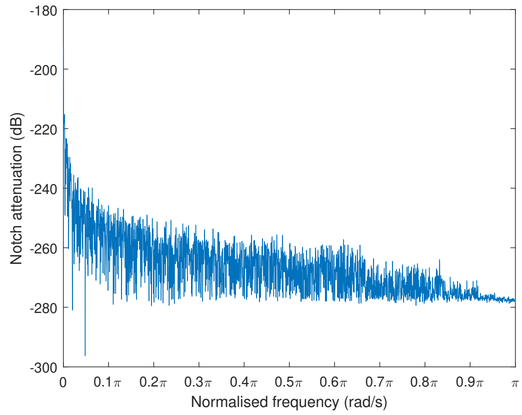


Figure 7.2: Maximum attenuation a second order IIR bandstop filter for different centre frequencies.

filter [28]. A drawback of using an IIR filter is that it is not guaranteed to be stable due to the location of the poles that appear in the transfer. An IIR filter is chosen as bandstop filter as the lower order required demands less computation time, complying to Requirement 5.

7.2. Filter Design

Digital IIR filters are often implemented by designing an analog filter and transforming it into a digital filter by means of the bilinear transform [29]. Basing the design of an IIR filter on an existing analog filter design has the benefit of ensuring that the designed filter is stable [30]. Several different analog filter implementations are available that are suitable for the design of an IIR filter, each with different characteristics. The four most well known filter implementations are the Butterworth, Chebyshev type I and II and elliptic filters.

Before a filter can be designed, some design specifications need to be established. The first specification is the 3 dB bandwidth of the filter. This is the width between the points in the magnitude response 3 dB below the passband at 0 dB. The normalised width of the stopband should thus be larger than $2\pi/N_f$. According to Requirement 15, frequencies where no howling is detected should not be attenuated by more than 3 dB. The maximum normalised 3 dB bandwidth therefore is equal to twice the detection resolution. The resolution of the howling detection is determined by the frame length N_f. The normalised resolution of the detection is equal to $2\pi/N_f$. From this it follows that the maximum normalised 3 dB bandwidth is equal to $4\pi/N_f$. The second specification is the stop band attenuation. This attenuation needs to be sufficient to counteract the build up of howling at the detected frequency.

Based on these specifications an IIR filter can be designed. The lowest order IIR band stop filter that can be designed is of order 2. In this case all of the four different analog filter implementations yield similar IIR bandstop filters. The design equation of such a filter is [31]

$$H(z) = \beta \frac{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}{1 - 2\beta\cos(\omega_0)z^{-1} + (2\beta - 1)z^{-2}}, \quad \beta = \frac{1}{1 + \tan(\Delta\omega/2)} \tag{7.3}$$

where ω_0 is the normalised centre frequency and $\Delta\omega$ is the 3 dB bandwidth. Figure 7.1 shows the magnitude response of a second order IIR filter based on the design equation with an ω_0 of $\pi/2$ and a 3 dB bandwidth of $4\pi/N_f$.

The figure shows that the filter has a maximum attenuation of more than 250 dB, which is more than enough to suppress howling. An example can be observed in Figure 6.2b, where the magnitude difference between howling power and the average audio signal power never exceeds 50 dB even well after howling appears. For this implementation to work however, the attenuation should be sufficient for all possible centre frequencies. Figure 7.2 shows the maximum attenuation for all possible centre frequencies. From this figure it can be seen that the smallest amount of attenuation achieved by the filter more than 180 dB, which is still more than enough for effective suppression of howling. From this it is concluded that the second order IIR bandstop filter meets the desired specifications.

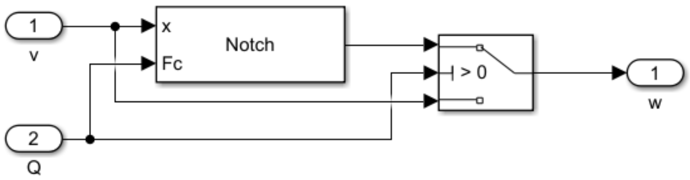
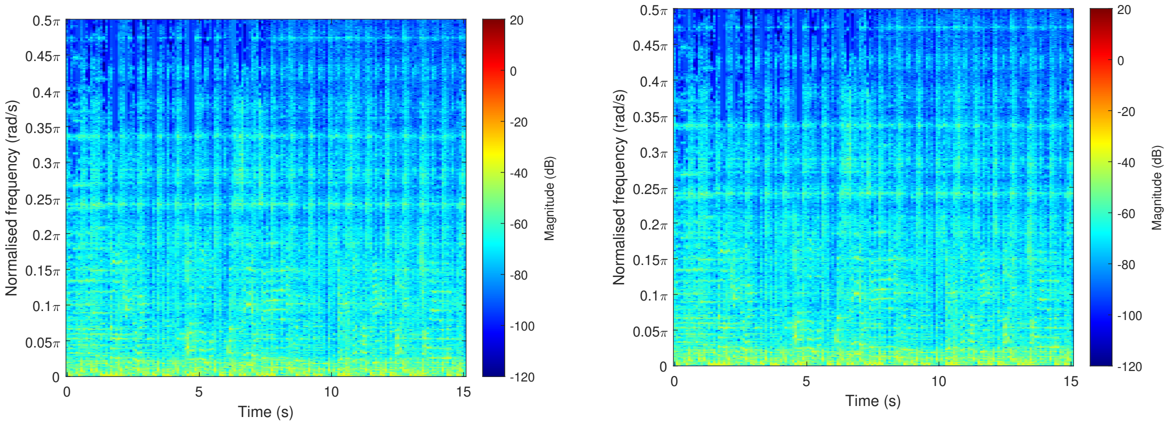


Figure 7.3: Overview of the suppression module in Simulink.



(a) Spectrogram of the audio fragment with the filter activated.

(b) Spectrogram of the audio fragment with feedback but without howling.

Figure 7.4: Comparison between the filtered audio fragments with howling generated at 530 Hz and the audio fragment without generated howling.

7.3. Filter Implementation and Evaluation

Implementing the suppression module in Simulink is quite trivial. The *Notch–Peak Filter* block takes an input signal and applies the second order IIR filter as described in (7.3). The centre frequency can be taken as variable input and the filtered signal is outputted. The *Switch* block is used to bypass the filter when no howling is present (when Q is zero). The full implementation is shown in Figure 7.3.

To evaluate the performance of the implemented filter, the same audio fragment is used as for the evaluation of the detection methods. In this fragment howling is generated at a frequency of 530 Hz. A spectrogram of the audio fragment with howling shown in Figure 6.2b. A spectrogram with the filter activated is created, shown in Figure 7.4a. From these figures it can be seen that the howling that occurs in the unfiltered spectrogram is not present in the spectrogram with the filter enabled. The filter therefore seems to successfully cancel the howling.

Besides effectively filtering howling, the filter should not alter the rest of the audio signal significantly. To see if this is the case the spectrogram of the filtered signal can be compared to the audio signal without howling (but with feedback, achieved with a broadband gain slightly below MSG). The spectrogram of this signal is shown in Figure 7.4b. When comparing this spectrogram to that of the filtered audio fragment shown in Figure 7.4a, it is concluded that both are very similar, indicating that the filter does not distort the rest of the audio signal. From the fact that the filter can effectively remove audio from the audio fragment, without distorting the rest of the signal, it is concluded that the filter operates as desired.

8

Final Design and Discussion of Results

After the detection and suppression modules have been designed, they can be combined to create the complete postfilter. In this chapter the postfilter is tested and the results of these tests are discussed. Additionally, the postfilter is discussed in the context of the complete AFC filter designed in this project.

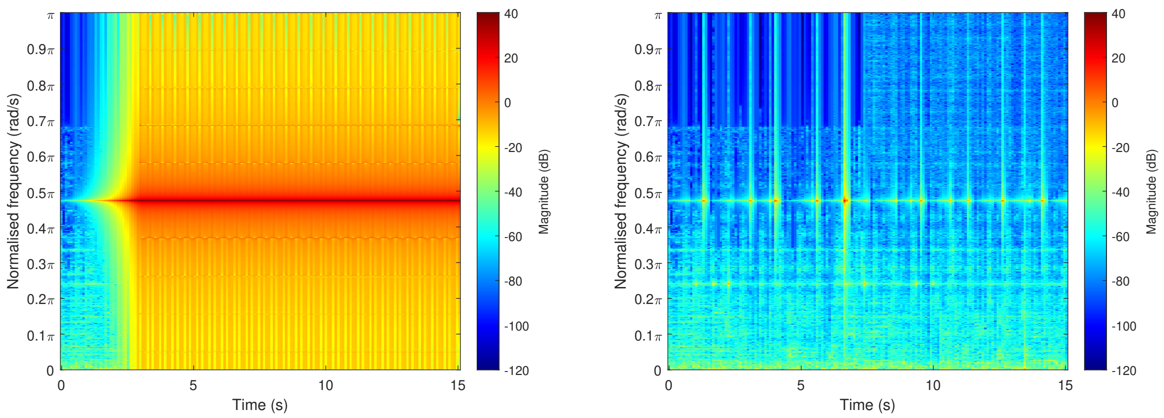
8.1. Postfilter Evaluation

From the comparison of howling detection methods conducted in Chapter 6, it has been concluded that the method combining PHPR and PNPR looks the most promising. This module is combined with the suppression module designed in Chapter 7 to create a functioning postfilter. To test this postfilter, it is inserted in the simulation environment as shown in Figure 4.1. Several different audio fragments are used as input signals to test the performance of the postfilter, as described in Section 4.1. By increasing the gain in the simulation environment howling is generated in the audio signals. This howling should be suppressed by the postfilter module. Spectrograms of the situation are used to visualise the results of these simulations.

When running this simulation while using the combination of PHPR and PNPR as the detection method it becomes clear that no howling is detected even though the audio signal does contain howling. Further investigation points out that the PNPR part of the detection never successfully identifies the howling, while the PHPR part identifies the howling as expected. Because the combined detection implementation only activates the suppression when both the PNPR and PHPR detect howling, the suppression is never activated due to the failing of the PNPR. The failure of the PNPR detection method seems to be caused by a wide howling peak in the power spectrum, contradicting the assumption made in Section 6.2 that the howling peak would be narrow. This means that some of the 'neighbours' are also part of this peak, preventing the PNPR from satisfying its detection condition. It seems that, while PNPR can perform very well in some situations, it performs rather badly in others, making it an unreliable method of howling detection. Based on this it has been decided not to use PNPR. PHPR is chosen as the detection method for the postfilter as this method provided the best results after the combination of PHPR and PNPR when tested in Chapter 6 and does not show the same unreliability that PNPR shows.

The new postfilter implementation contains PHPR detection alongside the same suppression. This version of the postfilter is simulated and tested with the same input audio signals. Focusing on the A-ha fragment (whose spectrogram is shown in Figure 6.2a), Figure 8.1a shows the output signal with feedback but without postfilter. As the broadband gain is set higher than the MSG, substantial howling takes place and starting at around 3 s, the whole spectrum is distorted due to the saturation that is included in the model. When including the postfilter, howling is successfully suppressed such that the saturation point is never reached. This can be observed in Figure 8.1b. While audible howling is still present, the systems stays stable and the signal itself can still clearly be heard. This is in accordance with Requirements 13 & 15 respectively. The same conclusions can be drawn from the other input signals, as can be seen from Figure A.13.

A simulation with a broadband gain substantially lower than the MSG shows that the postfilter input remains largely unfiltered; only at some frames (around 5 % for the A-ha signal) howling is incorrectly



(a) Spectrogram of the audio fragment without postfilter.

(b) Spectrogram of the audio fragment with postfilter.

Figure 8.1: Simulated PA system with 15 second audio fragment of 'Take on me' by A-ha, without and with postfilter.

detected and consequently the filter is incorrectly activated. While this is performance is not ideal, it still largely follows Requirement 14.

A drawback of the postfilter implementation is in situations where howling saturates. Saturated howling consists of a high amplitude sine function that gets cut off, yielding a signal that resembles a square wave. Similar to the square wave, the cut-off sine also contains odd harmonics. The PHPR method implemented checks the third harmonic and will therefore not detect howling. This drawback can be circumvented by suppressing howling before it reaches saturation. Another shortcoming is that a substantial range of harmonic frequencies are outside of the range of the signal, especially as the fourth harmonic is included. This problem is solved by extending PHPR with PAPR at those frequencies, but this method has a worse performance and can therefore have a small negative influence at high howling frequencies.

8.2. Complete AFC Filter Evaluation

The ultimate goal of the postfilter module is to work together with the adaptive filtering and decorrelation modules [12, 13] to form a complete and functioning AFC filter. Although the designed postfilter is tested on its own, even in very nonideal situations, new and unexpected problem can always arise when integrating multiple different designs to one system. In order to thoroughly validate the postfilter, it should be extensively tested in the complete AFC system.

Combining the adaptive filtering and decorrelation modules together in a working setup has proven unsuccessful in the given time span for the thesis. This regrettably means that the postfilter module could not be tested in a complete setup. However, due to how independent the postfilter operates from the other modules, expectations are that adding the postfilter should not have posed a problem. This can largely be attributed to the fact that the postfilter leaves the input signal unaltered when no howling is present. One factor that could influence the postfilter's functionality is how the adaptive filtering module creates instability. If this instability is fundamentally different from ordinary howling (without AFC), the postfilter might prove less useful. This is currently unknown and should be tested in further research.

9

Possible realisations of the prototype

Although no prototype will be made of the designed postfilter as a result of the ongoing pandemic, it is still valuable to look at the different possibilities available for doing so. This chapter discusses the options and the benefits each of these options could bring.

9.1. Prototype based on a single-board computer

As the postfilter has been designed as a model in Simulink, it is possible to directly generate C code that implements the designed functionality as a standalone script. An obvious way to make a prototype would then be to use a single-board computer and run the generated C code on it. A benefit of creating the prototype this way is that the designed model can directly be used on the prototype, without requiring manual redesign of the individual components of the postfilter. Another benefit of this method is that the postfilter module can directly be combined with the other modules of the adaptive feedback cancellation system. Because both the adaptive filtering and decorrelation modules are also designed in Simulink, all can be combined and one C script can be generated that implements all of the three modules. It is important to note that the adaptive feedback cancellation system uses a digitised audio signal. Because the incoming and outgoing audio signals are analog, analog-to-digital and digital-to-analog converters are needed for the inputs and outputs, with a sampling frequency of 44.1 kHz. This is the sampling frequency the system has been designed to use based on Requirement 3 Some single-board computers already include these converters, but it is always possible to install these separately. A weather-proof enclosure can be made for the prototype to satisfy Requirement 10. Most single board computers can be powered from mains power using an adapter, meaning that Requirement 11 is also satisfied. The price of the prototype depends mostly on the single-board computer that is used. A sufficiently powerful model can be purchased for around 60 euros. Combined with the cost of other parts of the prototype the total costs will not exceed 100 euros, which is conform with Requirement 12.

9.2. Prototype as a piece of software

As the previously described prototype realisation is based on one C script that implements the complete adaptive feedback cancellation system, the possibility of realising the prototype as a piece of software can also be considered. This way no physical prototype needs to be made. Instead any computer can be used to run the script. Doing this comes with the benefit that there is no need for dedicated hardware. This version of a prototype will however become less plug-and-play as the process of connecting the inputs and outputs of the prototype can differ between devices. This might require peripheral hardware. This is not the case for the single-board computer version, as it has its own dedicated input and output ports. The prototype is also not guaranteed to be weather-proof, as this depends on the device which is used to run the script. It can therefore not be stated that Requirement 10 is satisfied as this is device dependent. Most consumer devices can be powered using the mains electricity from which it can be concluded that Requirement 11 can be assumed to be satisfied. The price of the prototype is a strong point of a software implementation, as no hardware needs to be purchased for the prototype Requirement 12 is satisfied.

9.3. Prototype created in hardware

Another option for the prototype would be to mirror the designed Simulink model in hardware. Because the Simulink model consists of separate blocks, each performing a specific operation, it might be possible to create hardware components that perform these operations and to combine them to recreate the complete system. This could be done by using a hardware description language like VHDL. An FPGA can be used to test the design and the prototype can be realised as an integrated circuit. A benefit of doing this is that a hardware implementation generally speaking is more efficient than a software implementation, as the hardware is specifically designed for the task at hand. Creating an integrated circuit of the prototype is rather expensive, as only a few prototypes are needed while making integrated circuits really only becomes economically viable when a large amount of the same integrated circuit is created. Making a hardware implementation as a prototype is therefore hard to justify as it violates Requirement 12. Besides this a hardware implementation can not easily be tweaked. This means that when a change is made to the design, a completely new integrated circuit needs to be created. When using a single-board computer or software implementation any changes can be made by simply updating the C script, which is far more practical. A prototype made in hardware can be made weather-proof and compatible with mains electricity, which means that Requirements 10 and 11 can be satisfied.

10
Conclusions, Recommendations and Future Work

This thesis describes the complete design process of a postfilter to be used as backup after AFC filtering. After evaluating literature on state of the art technologies in this field of research, one of several functionalities for the postfilter is selected: ensuring stability by removing howling. The design is split in two modules: a howling detection and a howling suppression module.

For the detection module, multiple different existing techniques are described and simulated, and some new techniques are proposed and simulated. All these techniques are evaluated with the use of ROC curves and one technique is selected for the postfilter based on the set of requirements, being the peak to harmonic power ratio criterion. For the suppression module, a notch filter is designed that strikes a balance between the requirements, specifically regarding the 3 dB bandwidth, notch attenuation and filter order. A second order Butterworth infinite impulse response filter with a very narrow 3 dB bandwidth is chosen.

The detection and suppression are combined, resulting in a complete postfilter. This complete postfilter is simulated and the results are positive regarding the requirements. However, an insufficient amount of simulations have been run regarding the postfilter when no howling is present (related mostly to Requirement 14). While howling itself is correctly detected and suppressed in multiple setups with considerable howling, the postfilter might perform less ideal when no howling is present. Some requirements regarding practical realisations (e.g. real time operation, Requirement 5) could not strictly be adhered to as no practical realisation could be made. However these requirements have been kept in mind during the design process.

It can be concluded that the postfilter works as intended, especially in situations where a backup filter is often required (such as for a mediocre AFC filter).

If given more design time in the project, a number of improvements and additions could have been included. These improvements and additions can be taken as recommendation for future work improving on the postfilter designed in this thesis.

- A first possible improvement could be to change how the spectrum to harmonics power ratio criterion works. At present, only the second harmonic is used pre-peak. Similar to the spectrum to neighbouring power ratio criterion, multiple harmonics could be taken and used in the form a geometric mean. Dividing the spectrum by this mean could potentially result in more accurate howling detection, such that the ROC curve in Figure A.12b more closely approaches $P_D = 1$. The extra implementation would not be very hard to design as taking other harmonics simply means copying and changing the downsample rate in Figure A.9. The geometric mean calculation can be borrowed from the spectrum to neighbouring power ratio implementation in Figure A.10.
- An addition to the detection module could be to detect multiple different peaks simultaneously. The set of howling frequencies ($\mathcal{Q}(k)$ as in Figure 6.1) would then potentially contain multiple frequencies per frame. The suppression module should be adjusted to contain multiple filters such that multiple frequencies can be suppressed at the same time. The advantages would

be threefold: the detection module can increase the detection probability by considering more possible howling frequencies, howling can be detected earlier as not only the largest peak is evaluated, and howling can be better suppressed if it occurs at multiple frequencies (including howling at saturation as discussed in Section 8.1). Disadvantages that have to be kept in mind are the higher potential to falsely detect howling which filters the signal undesirably (going against Requirement 14), and introducing more computational complexity (going against Requirement 5). The principle of multiple peak detection has already been proposed, for example by Van Waterschoot and Moonen [21].

- If more time could be spent on the suppression module, a higher order filter could possibly be designed for the postfilter. With a higher order, the transition between passband and stopband could be made sharper with the advantage that the passband will be less affected by the filter while keeping the 3 dB bandwidth the same. The maximum attenuation in the stopband can also be increased. Designing a higher order filter generally takes more time to guarantee system stability, especially with dynamic design parameters. As the second order filter was adequate already, no extra time was spent here in this project. As with the previous addition, the main disadvantage is the increase in computational complexity.

- A last possible improvement to the suppression module is to change the way the filter is activated and deactivated. Currently, the filter is applied only at frames where howling is detected. Howling usually persists for multiple frames and can return after filtering. A solution (more elegant than simple letting the filter switch on and off repeatedly) is to let the filter hold for a certain amount of time. Few research results are available on the timing of disabling the filter after enabling [1], making this a valuable subject to research further. A good starting point would be the deactivation criterion by Terada and Murase [32], which links the activation duration inversely to the time period between two successive occurrences of howling. A related topic is about how the suppression is activated. Instead of applying the full filter immediately, the stopband attenuation could be increased from 0 dB to the full attenuation over the course of several samples or frames. This could reduce the introduced error when falsely detecting howling, making the probability of false alarm a less important parameter.

- Another way of improving the postfilter performance might be to find a way of implementing residual feedback filtering. As described in Chapter 5 the main challenge in implementing residual feedback filtering is finding an accurate approximation of the residual feedback signal. At the moment there are no reliable methods available to make this approximation. When a reliable method can be found of approximating the residual feedback, residual feedback filtering can be a valuable addition to the postfilter, as it can improve the signal quality at the output of the postfilter. Development of such a residual feedback approximation method can therefore be an interesting subject for further research.

A
Figures

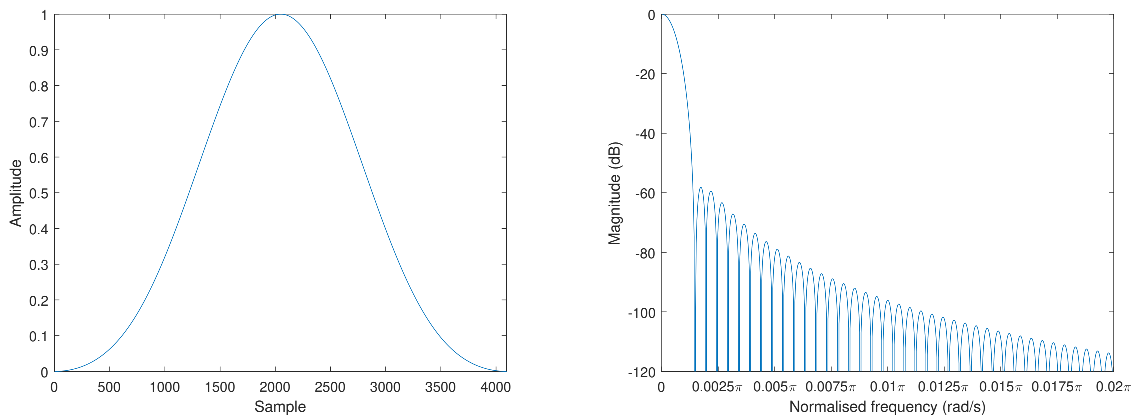


Figure A.1: Time and frequency domain plots of the Blackman window.

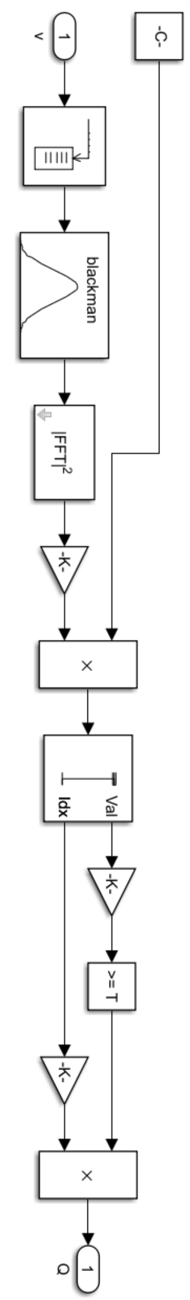


Figure A.2: Peak to threshold power ratio (PTPR) howling detection model.

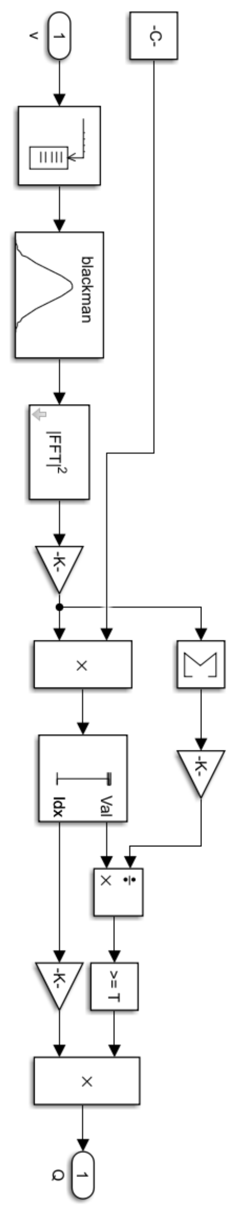


Figure A.3: Peak to average power ratio (PAPR) howling detection model.

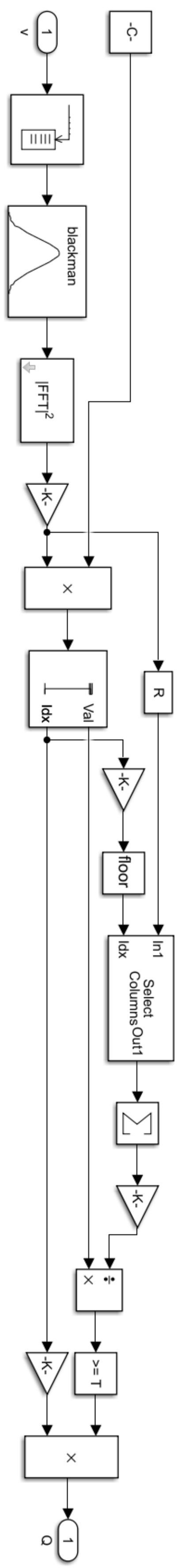


Figure A.4: Peak to local average power ratio (PLAPR) howling detection model.

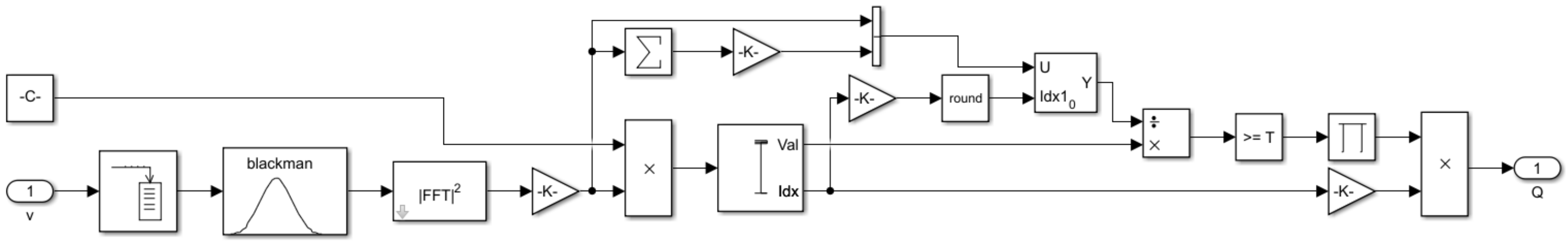


Figure A.5: Peak to harmonic power ratio (PHPR) howling detection model.

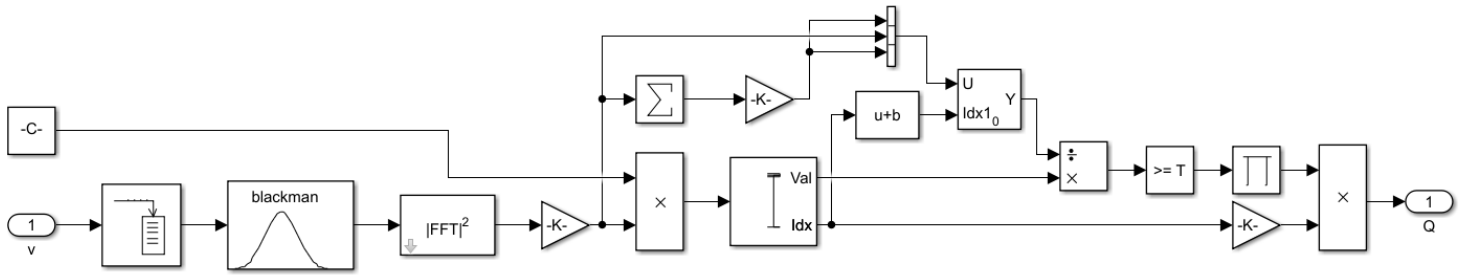


Figure A.6: Peak to neighbouring power ratio (PNPR) howling detection model.

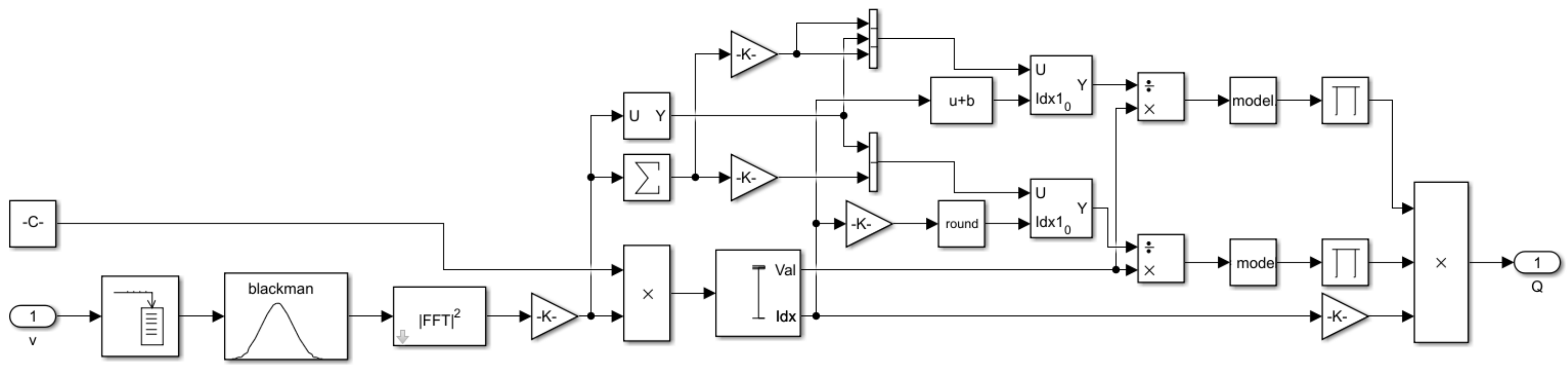


Figure A.7: Peak to harmonic and neighbouring power ratios howling detection model.

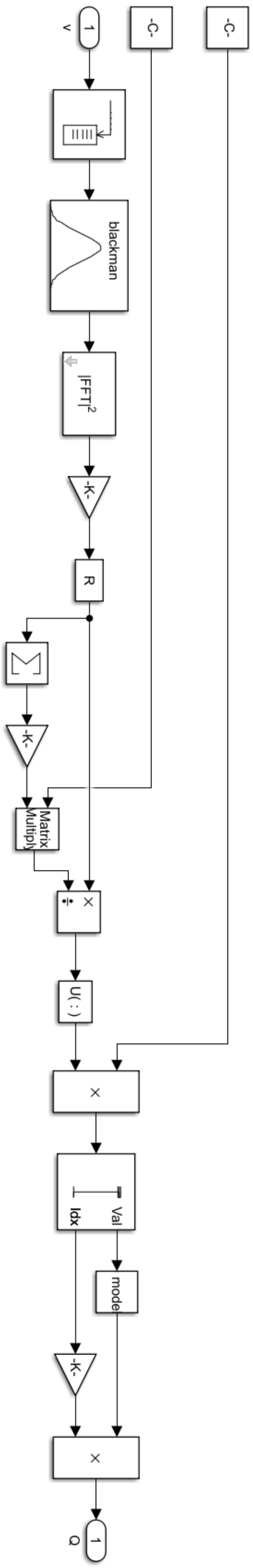


Figure A.8: Spectrum to local average power ratio (SLAPR) howling detection model.

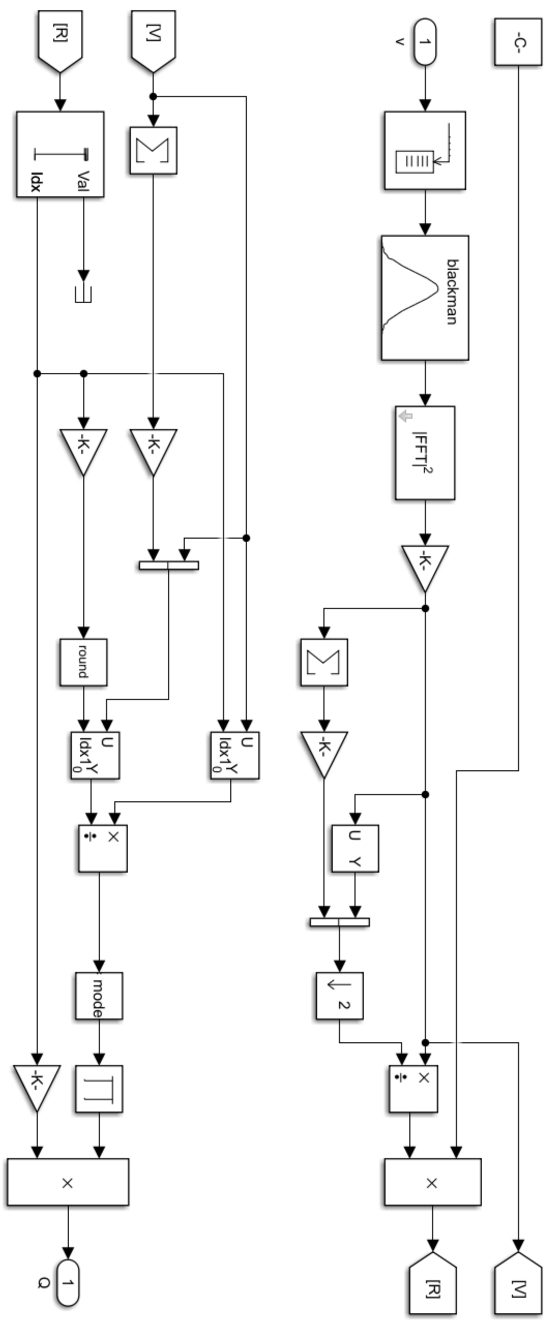


Figure A.9: Spectrum to harmonic power ratio (SHPR) howling detection model.

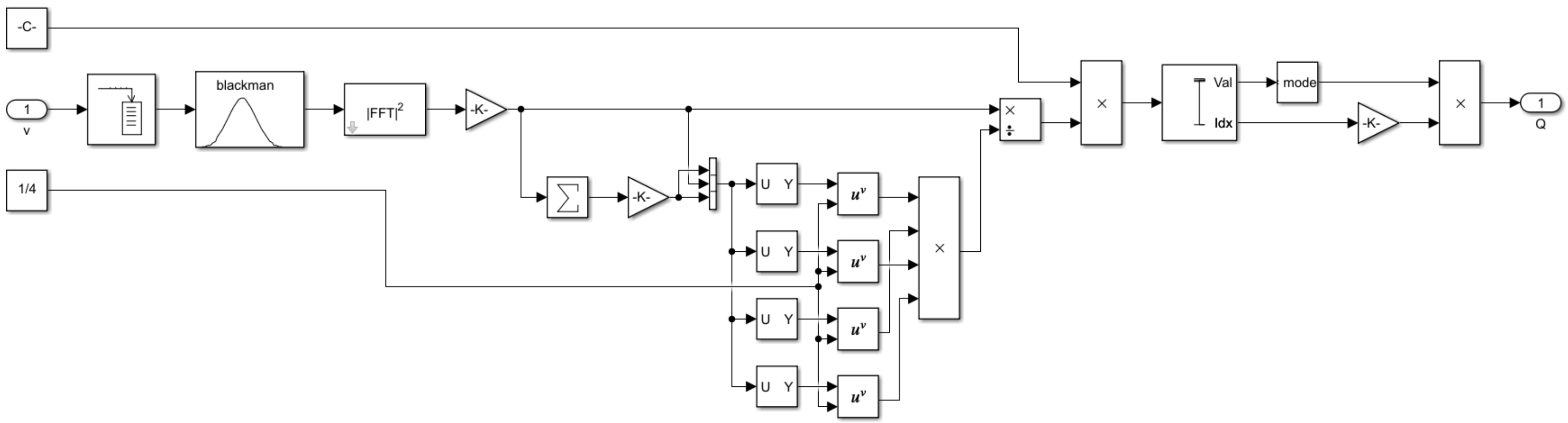
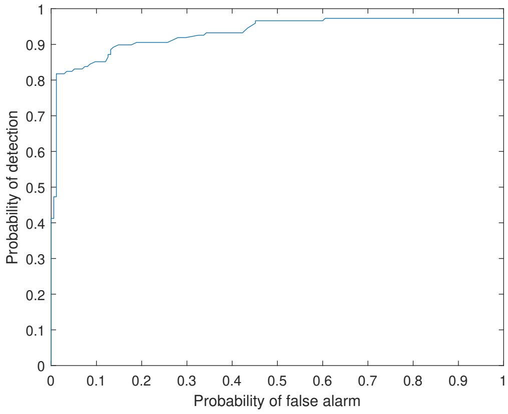
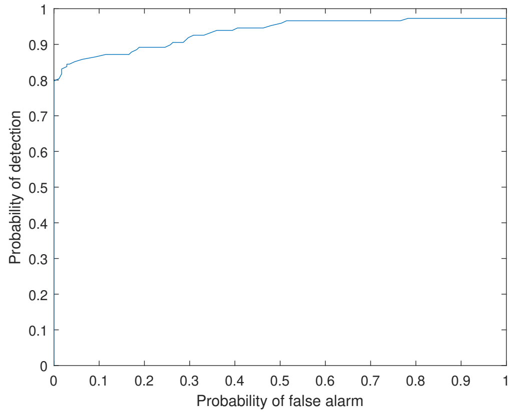


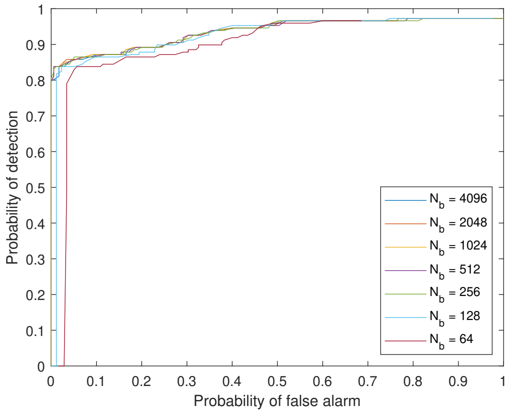
Figure A.10: Spectrum to neighbouring power ratio (SNPR) howling detection model for $m_n \in \{-2, -1, 1, 2\}$.



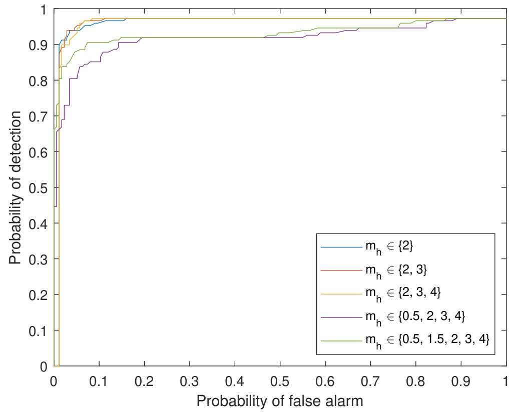
(a) Peak to threshold power ratio criterion with $T_{PTPR} \in (-\infty, +\infty)$ dB.



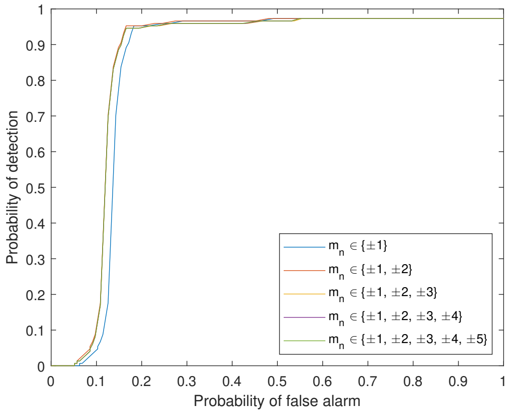
(b) Peak to average power ratio criterion with $T_{PAPR} \in (-\infty, +\infty)$ dB.



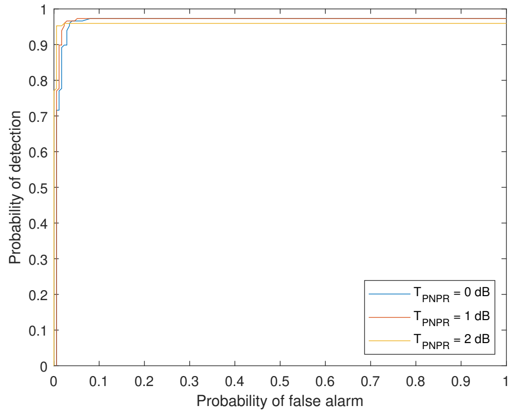
(c) Peak to local average power ratio criterion with $T_{PLAPR} \in (-\infty, +\infty)$ dB for different bin sizes N_b.



(d) Peak to harmonic power ratio criterion with $T_{PHPR} \in (-\infty, +\infty)$ dB for different harmonics sets.

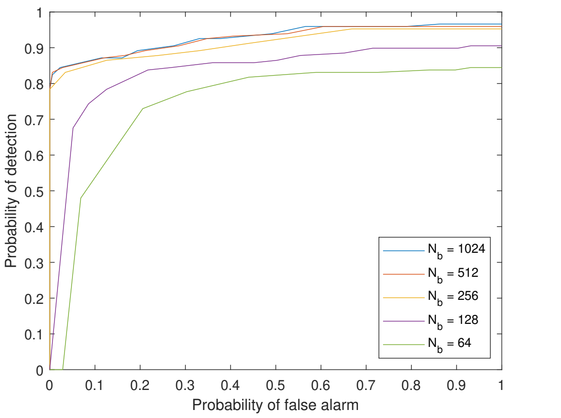


(e) Peak to neighbouring power ratio criterion with $T_{PNPR} \in (-\infty, +\infty)$ dB for different neighbour sets.

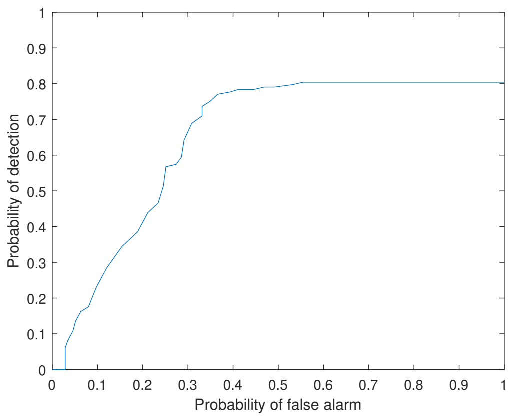


(f) Peak to harmonic and neighbouring power ratio criteria with $T_{PHPR} \in (-\infty, +\infty)$ dB for different PNPR thresholds T_{PNPR}.

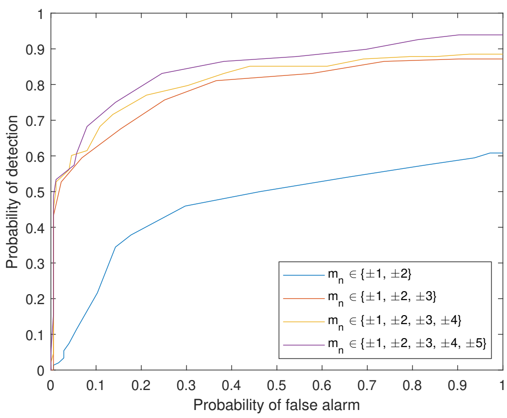
Figure A.11: ROC curves for different post-peak detection criteria.



(a) Spectrum to local average power ratio criterion with $T_{SLAPR} \in (-\infty, +\infty)$ dB for different bin sizes N_b.

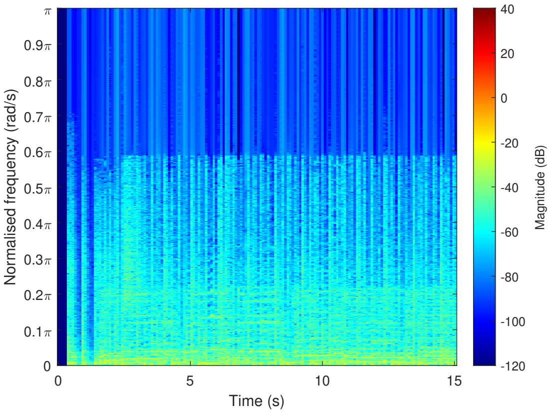


(b) Spectrum to harmonic power ratio criterion with $T_{SHPR} \in (-\infty, +\infty)$ dB for harmonics $m_h \in \{2, 3, 4\}$.

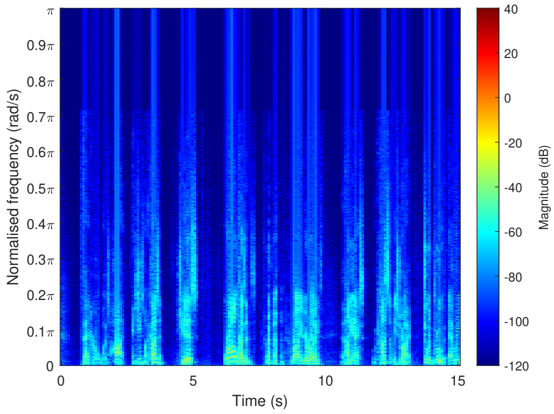


(c) Spectrum to neigbouring power ratio criterion with $T_{SNPR} \in (-\infty, +\infty)$ dB for different neighbour sets.

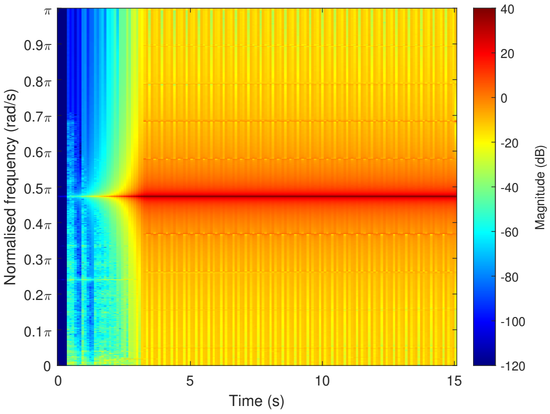
Figure A.12: ROC curves for different pre-peak detection criteria.



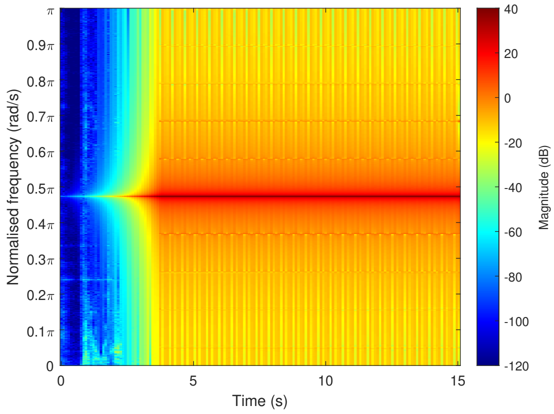
(a) Spectrogram of the instrumental audio fragment.



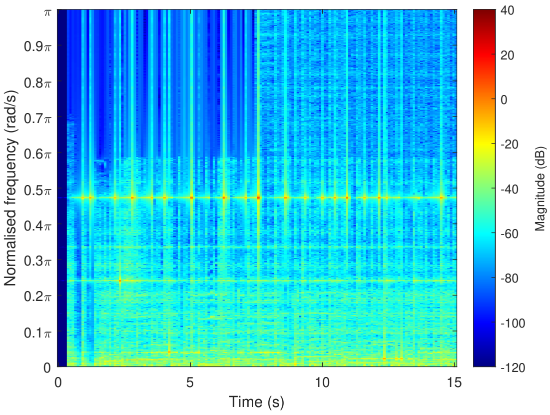
(b) Spectrogram of the speech audio fragment.



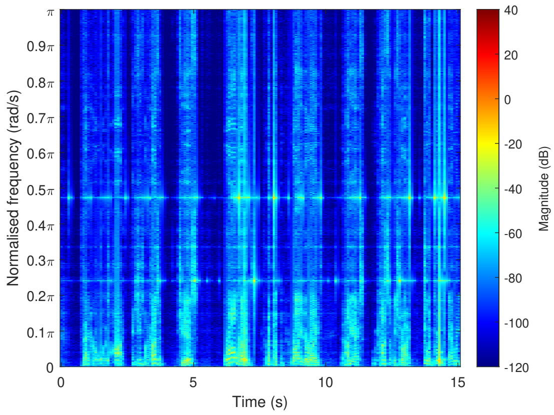
(c) Spectrogram of the instrumental audio fragment with feedback, without postfilter.



(d) Spectrogram of the speech audio fragment with feedback, without postfilter.



(e) Spectrogram of the instrumental audio fragment with postfilter.



(f) Spectrogram of the speech audio fragment with postfilter.

Figure A.13: Simulated PA system with a 15 second instrumental audio fragment of 'Smooth' by Santana featuring Rob Thomas (left) and a 15 second English speech fragment (right).

B

Matlab Code

B.1. tp_roomimpulse

```matlab
%==========================================================================
% ** tp_roomimpulse
%--------------------------------------------------------------------------
% This function plays and records a logarithmic sine sweep function and
% calculates the room impulse response using convolution.
% Authors: C.A. Weustink & J.W. de Vries, 05.05.20
%==========================================================================

function [RIR] = tp_roomimpulse(dist)
%--------------------------------------------------------------------------
% * Define parameters and initialise recorder
%--------------------------------------------------------------------------
Fs = 44100;
T = 1.5;
w1 = 2*pi*10;
w2 = 2*pi*22000;

r = audiorecorder(Fs,16,1);
%--------------------------------------------------------------------------
% * Create Logarithmic SineSweep
%--------------------------------------------------------------------------
t = 0 : 1/Fs : T;
x = sin(T*w1/log(w2/w1)*(exp(t/T*log(w2/w1))-1));
f = flip(x);
%--------------------------------------------------------------------------
% * Record Logarithmic SineSweep and calculate room impulse response
%--------------------------------------------------------------------------
sound(x,Fs)
recordblocking(r,T+1);
RIR.y = getaudiodata(r);

H = fft(conv(RIR.y,f));
RIR.h_long = ifft(H./abs(fft([x';zeros((length(H)-length(x)),1)])).^2);
delay = round(dist / 343 * Fs);
index = find(RIR.h_long == max(RIR.h_long),1,'first');
RIR.h = RIR.h_long(index-delay:end);

end
```

B.2. tp_compare

```
%==========================================================================
% ** tp_compare
%--------------------------------------------------------------------------
% This script takes model parameters, simulates the detection model and
% calculates ROC curve data.
% Authors: C.A. Weustink & J.W. de Vries, 26.05.20
%==========================================================================

%--------------------------------------------------------------------------
% * Define simulation parameters and initialise results
%--------------------------------------------------------------------------
steps = 200;
domain = [0 9];
clear results;
%--------------------------------------------------------------------------
% * Define model parameters
%--------------------------------------------------------------------------
model.frame = 4096;
model.hop = 2048;
model.harmonics = [2 3 4];
model.neighbours = [-5 -4 -3 -2 -1 1 2 3 4 5];
model.Ta = 380;
model.Tn = 1.5;
%--------------------------------------------------------------------------
% * Simulate system and calculate results
%--------------------------------------------------------------------------
for i = (1 : steps + 1)
model.T = 10^((i-1)/(steps/(domain(2)-domain(1))) + domain(1));
simOut = sim('Model_Postfilter_Sim.slx');
boundary = 176;

N_FP = nnz(simOut.det_frequencies(1:(boundary-1)));
N_N = boundary - 1;
P_FA = N_FP/N_N;

N_TP = nnz(simOut.det_frequencies(boundary:end) < 539 & simOut.
    det_frequencies(boundary:end) > 516);
N_P = length(simOut.det_frequencies(boundary:end));
P_D = N_TP/N_P;

results(i,:) = [model.T P_D P_FA];
disp(i/(steps+1)*100)
end
%--------------------------------------------------------------------------
% * Clear variables
%--------------------------------------------------------------------------
clearvars steps domain boundary i;
clearvars N_FP N_N P_FA;
clearvars N_TP N_P P_D;
```

Bibliography

[1] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," *Proc. IEEE*, vol. 99, pp. 288–327, Feb. 2011.

[2] M. R. Schroeder, "Improvement of acoustic-feedback stability by frequency shifting," *J. Aucoust. Soc. Amer.*, vol. 36, pp. 1718–1724, Sep. 1964.

[3] P. U. Svensson, *On reverberation enhancement in auditoria*. PhD thesis, Dept. Appl. Acoust., Chalmers Univ. Technol., 1996.

[4] M. A. Poletti, "The stability of multichannel sound systems with frequency shifting," *J. Aucoust. Soc. Amer.*, vol. 116, pp. 853–871, Aug. 2004.

[5] D. Somasundaram, "Feedback cancellation in a sound system," Mar. 2008. Eur. Patent App. EP1 903 833 A1.

[6] T. van Waterschoot, G. Rombouts, and M. Moonen, "On the performance of decorrelation by prefiltering for adaptive feedback cancellation in public address systems," in *Proc. 4th IEEE Benelux Signal Process. Symp.*, (Hilvarenbeek, The Netherlands), pp. 167–170, Apr. 2004.

[7] A. Ortega, E. Lleida, and E. Masgrau, "Speech reinforcement system for car cabin communications," *IEEE Trans. Speech Audio Process.*, vol. 13, pp. 917–929, Sep. 2005.

[8] C. P. Janse and H. J. W. Belt, "Sound reinforcement system having an echo suppressor and loudspeaker beamformer," Feb. 2003. WIPO Patent Appl. WO/2003/010 996.

[9] F. Gallego, E. Lleida, E. Masgrau, and A. Ortega, "Method and system for suppressing echoes and noises in environments under variable acoustic and highly feedback conditions," Dec. 2002. WIPO Patent Appl. WO/2002/101 728.

[10] S. Kamerling, K. Janse, and F. van der Meulen, "A new way of acoustic feedback suppression," in *Preprints AES 104th Conv.*, (Amsterdam, The Netherlands), May 1998.

[11] G. Rombouts, T. van Waterschoot, and M. Moonen, "Robust and efficient implementation of the pem—afrow algorithm for acousic feedback cancellation," *J. Audio Eng. Soc.*, vol. 55, pp. 955–966, Nov. 2007.

[12] C. H. Kos and M. C. Bekkering, "Adaptive filtering in adaptive feedback cancellation for pa systems." B.S. Thesis, Delft Univ. Technol., Delft, 2020.

[13] L. C. A. Huijbregts and M. A. Jongepier, "Decorrelation in adaptive feedback cancellation for pa systems." B.S. Thesis, Delft Univ. Technol., Delft, 2020.

[14] H. Nyquist, "Regeneration theory," *Bell Syst. Tech. J.*, vol. 11, pp. 126–147, 1932.

[15] R. W. Guelke and A. D. Broadhurst, "Reverberation time control by direct feedback," *Acustica*, vol. 24, no. 1, pp. 33–41, 1971.

[16] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, pp. 10–21, Jan. 1949.

[17] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, and C. Colomes, "Peaq-the itu standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, vol. 48, no. 1/2, pp. 3–29, 2000.

[18] G.-B. Stan, J.-J. Embrechts, and D. Archambeau, "Comparison of different impulse response measurement techniques," *J. Audio Eng. Soc.*, vol. 50, pp. 249–262, Apr. 2002.

[19] S. S. Stevens, J. Volkmann, and E. B. Newman, “A scale for the measurement of the psychological magnitude pitch,” *J. Aucoust. Soc. Amer.*, vol. 8, no. 3, pp. 185–190, 1937.

[20] M. P. Lewis, T. J. Tucker, and D. M. Oster, “Method and apparatus for adaptive audio resonant frequency filtering,” Sep. 1993. U.S. Patent 5,245,665.

[21] T. van Waterschoot and M. Moonen, “Comparative evaluation of howling detection criteria in notch-filter-based howling suppression,” *J. Audio Eng. Soc.*, vol. 58, pp. 923–940, Nov. 2010.

[22] F. J. Harris, “On the use of windows for harmonic analysis with the discrete fourier transform,” *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[23] F. F. Kuo and J. F. Kaiser, *System analysis by digital computer*. New York, NY, USA: Wiley, 1966.

[24] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra*. New York, NY, USA: Dover, 1958.

[25] N. Osmanovic and V. Clarke, “Acoustic feedback cancellation system,” Feb. 2010. U.S. Patent 7,664,275.

[26] E. T. Patronis, Jr., “Electronic detection of acoustic feedback and automatic sound system gain control,” *J. Audio Eng. Soc.*, vol. 26, pp. 323–326, May 1978.

[27] S. Ando, “Howling detection and prevention circuit and a loudspeaker system employing the same,” Jun. 2001. U.S. Patent 6 252 969.

[28] L. R. Rabiner, J. F. Kaiser, O. Herrmann, and M. T. Dolan, “Some comparisons between fir and iir digital filters,” *Bell System Technical Journal*, vol. 53, no. 2, pp. 305–331, 1974.

[29] A. Tustin, “A method of analysing the behaviour of linear systems in terms of time series,” *J. Inst. Electr. Eng. 2A*, vol. 94, no. 1, pp. 130–142, 1947.

[30] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*. Harlow, Essex, UK: Pearson Education, 4th ed., 2014.

[31] S. J. Orfanidis, *Introduction to signal processing*. Upper Saddle River, NJ, USA: Prentice–Hall, 1995.

[32] Y. Terada and A. Murase, “Howling control device and howling control method,” Mar. 2007. U.S. Patent 7,190,800.