Evaluation of various Algorithms to detect acoustic Feedback

Toningenieur Projekt

Sebastian Braun

Betreuung: DI Hannes Pomberger

Graz, February 19, 2012

kunst uni graz

institut für elektronische musik und akustik

Abstract

Annoying howling caused by acoustic feedback is an omnipresent problem in amplified live-sound situations. There are some algorithms that detect feedback howling frequencies automatically and try to eliminate these via notch filters. For the detection of the feedback frequencies, several different approaches and criteria are available, that can be used separately or in combination. The proposed algorithms shall be implemented and evaluated in a suitable simulation environment.
In this work, a new evaluation method is designed, that is more objective than the proposed evaluation in literature. Two evaluation methods are introduced, but the final evaluation focuses on the new developed method.
Further, the improvement of the detection algorithms by using a Constant-Q-Analysis, and the additional use of a second microphone are investigated. This second microphone can gather additional information, such as the estimation of maximum stable gain (MSG) of the amplification system and might improve the performance.
The evaluation results of several detection criteria form a basis for an advanced design of a feedback detector.

Contents

1 Introduction

Almost everyone who visits pop/rock-concerts from time to time, has experienced a situation where the public adress (PA) system started to howl due to acoustic feedback. Whereas at live-music events a qualified sound engingeer is usually present at any time, especially in speech reinforcement situations a technical supervisor is not present or available. But even a properly equalized PA system does not guarantee a howling-free system. If the microphone positions are not static, i.e. a microphone is moved over the stage, the transfer functions change and new situations can occur, that weren't taken into account and cause unstable feedback. To handle this problem, an adaptive feedback canceller is a good solution. The focus of this work lies on the detection of feedback howling frequencies, not on cancelling or suppression methods.

1.1 How occurs feedback howling?

In every real sound reinforcement situation (except in a perfect free-field) a part of the amplified signal is picked up by the microphone and amplified again. Assuming simplified conditions with a dry source signal at the microphone and no additional noise, we state the signal model in Fig. 1. Eq. (1) describes the microphone signal $x(t)$,

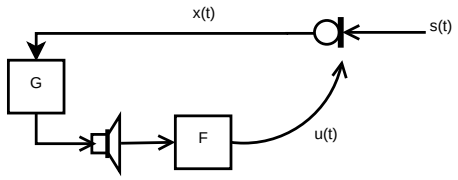

Figure 1: General signal model of feedback loop with source signal $s(t)$, electroacoustical forward path G and feedback path F

consisting of the source signal $s(t)$ and the additional term caused by the feedback via one or more loudspeakers. $g(t)$ is the impulse response of the complete electroacoustical path including microphone, preamp, signal processing (e.g. EQ), amplifier and speaker characteristics. The room impulse response measured between the loudspeaker and microphone positions is called $f(t)$. $G(e^{j\omega})$ and $F(e^{j\omega})$ are the corresponding transfer functions in the frequency domain.

$$x(t) = s(t) + u(t) = s(t) + f(t) * g(t) * x(t) \tag{1}$$

The overall transfer function of the system in Fig. 1 can be calculated to

$$\frac{X(e^{j\omega})}{S(e^{j\omega})} = \frac{1}{1 - G(e^{j\omega})F(e^{j\omega})} \tag{2}$$

A LTI system is BIBO stable, if its impulse response is absolutely summable [7]. In the frequency domain all poles have to be inside the unit circle for BIBO stability. Therefore, for our system follows:

$$\text{If} \quad |G(e^{j\omega})F(e^{j\omega})| \leq 1 \quad \Rightarrow \text{BIBO stable} \tag{3}$$

This means that for certain poles in the transfer function Eq. (2), our system gets unstable, even if a bounded (stable) input signal is given. Depending on the unstable frequency pole, we perceive this unstability as howling, since it occurs most frequent at frequencies between about 200 and 5000 Hz . The howling has a very narrow-band like character similar to a single sine component, because usually only one single frequency is unstable.

1.2 Notch filter based howling suppression

There are several techniques for howling cancellation or suppression available, such as phase modulation, gain reduction, spatial filtering and room modeling methods [1]. Gain reduction methods can be discriminated between automatic gain control (AGC), that reduces the entire frequency range, automatic equalization (AEQ), that reduces critical subbands and notch-filter-based howling suppression (NHS). The latter reduces the gain in narrow frequency bands around critical frequencies and this is the choice for this work.

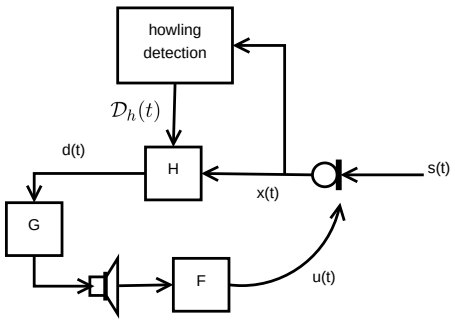

Figure 2: Feedback loop with source signal $s(t)$, Bank of adjustable notchfilters $H(e^{j\omega})$ controlled by the parameter set $\mathcal{D}_h$, electroacoustical forward path $G(e^{j\omega})$ and feedback path $F(e^{j\omega})$

Figure 2 shows again the LTI system from Fig. 1 with an inserted notch filter, controlled by a howling detection algorithm.

$$x(t) = s(t) + f(t) * g(t) * h(t) * x(t) \tag{4}$$

The transfer function is in analogy to Eq. (2)

$$\frac{X(e^{j\omega})}{S(e^{j\omega})} = \frac{1}{1 - H(e^{j\omega})G(e^{j\omega})F(e^{j\omega})} \tag{5}$$

Now the filter $H(e^{j\omega})$ has to be designed, that the condition for BIBO stability (3) is fulfilled.

$$|H(e^{j\omega})G(e^{j\omega})F(e^{j\omega})| \leq 1 \tag{6}$$

The output after the cancelling filter $H(e^{j\omega})$ is the desired signal $d(t)$, which should be howling free. The howling detection algorithm operates on the microphone signal $x(t)$

and outputs a set of howling frequencies for each time frame. These detected frequencies form together with other filter design parameters the parameter set $\mathcal{D}_h(t)$. A notch filter design needs at least the center frequency, the gain reduction value and a bandwidth. An advanced notch-filter design can include a time-variant design for all parameters, not only the center frequency. In simpler implementations some of the parameters can kept static, e.g. the notch-filter bandwidth.

2 Available Criteria

The paper of Toon van Waterschoot and Marc Moonen [2] is used as a starting point for this work. It provides a collection of various criteria for feedback howling detection. In this section, the criteria and the evaluation method proposed in [2] are explained.

The idea behind the criteria is the following: The microphone signal $x(t)$ is buffered and framed with a buffer size of N and hop-size R, windowed and transformed into the frequency domain via FFT.

$$\mathbf{x}(t) = [x(t+R-N) \ \ldots \ x(t+R-1)]^T \tag{7}$$

$$X(\omega_k, t) = \sum_{n=0}^{N-1} w(t_n) x(t_n) e^{-j\omega_k t_n} \tag{8}$$

As not stated different, the values $N = 4096$, $R = \frac{1}{2}N$ and a Blackman window for $w(t)$ are used at a sampling frequency $f_s = 44100$ Hz. The spectra $X(\omega_k, t)$ are processed by a peak picking algorithm, that delivers the angular frequencies ω_i of the detected peaks as a set of "howling candidates" $\mathcal{D}_\omega(t)$. The feedback detection algorithms operate on this set of M howling candidates and calculate a certain criterion value for every howling candidate $\omega_i \in \mathcal{D}_\omega(t)$. If this value exceeds a threshold, the null hypothesis - $\mathcal{H}_0$: howling does not occur - is rejected, otherwise no howling is detected.

The simplest criterion is to take a fixed power threshold value, e.g. $P_0 = 85$ dB SPL to decide whether a frequency bin contains feedback or not. Waterschoot collected 6 such criteria and one, that merges two of these into a new criterion.

2.1 Single-feature criteria

Spectral criteria:

1. *Peak-to-Threshold Power Ratio (PTPR)*: Determines the ratio of the spectral power of the howling candidate ω_i and a fixed absolute power threshold P_0:

$$PTPR(\omega_i, t) \ \text{[dB]} = 10 \log_{10} \frac{|X(\omega_i, t)|^2}{P_0} \tag{9}$$

   If the $PTPR(\omega_i, t)$ value exceeds the threshold T_{PTPR}, howling is detected.

$$PTPR(\omega_i, t) \geq T_{PTPR} \ \text{[dB]} \ \Rightarrow \omega_i \in \mathcal{D}_h(t) \tag{10}$$

2. *Peak-to-Average Power Ratio (PAPR)*: Ratio between the average microphone signal power $\overline{P}_x(t)$ and the howling component power.

$$PAPR(\omega_i, t) \text{ [dB]} = 10 \log_{10} \frac{|X(\omega_i, t)|^2}{\overline{P}_x(t)} \tag{11}$$

$$\overline{P}_x(t) = \frac{1}{N} \sum_{k=0}^{N-1} |X(\omega_i, t)|^2 \tag{12}$$

$$PAPR(\omega_i, t) \geq T_{PAPR} \text{ [dB]} \Rightarrow \omega_i \in \mathcal{D}_h(t) \tag{13}$$

3. *Peak-to-Harmonic Power Ratio (PHPR)*: A spectral feature that determines the ratio of the candidate howling component power $|Y(\omega_i, t)|^2$ and its mth (sub)harmonic component power. Howling has not the same spectral structure as speech or music with its harmonic components. The feature uses this property to discriminate between howling and signal components.

$$PHPR(\omega_i, t) \text{ [dB]} = 10 \log_{10} \frac{|X(\omega_i, t)|^2}{|X(m\omega_i, t)|^2} \tag{14}$$

$$\bigwedge_{m \in \mathcal{M}_{PHPR}} \left[PHPR(\omega_i, t) \geq T_{PHPR} \text{ [dB]} \right] \Rightarrow \omega_i \in \mathcal{D}_h(t) \tag{15}$$

4. *Peak-to-Neighboring Power Ratio (PNPR)*: A spectral feature that uses the fact, that howling is very narrow-band. It determines the ratio between the howling candidate component power and the power of its mth neighboring frequency component.

$$PNPR(\omega_i, t) \text{ [dB]} = 10 \log_{10} \frac{|X(\omega_i, t)|^2}{|X(\omega_i + 2\pi m/N, t)|^2} \tag{16}$$

$$\bigwedge_{m \in \mathcal{M}_{PNPR}} \left[PNPR(\omega_i, t) \geq T_{PNPR} \text{ [dB]} \right] \Rightarrow \omega_i \in \mathcal{D}_h(t) \tag{17}$$

Temporal criteria:

5. *Interframe Peak Magnitude Persistence (IPMP)*: A temporal feature that counts the occurrence of the howling candidate frequencies ω_i in the past Q_M frames. It is based on the idea, that howling typically persists for a longer time than speech or tonal components.

$$IPMP(\omega_i, t) = \frac{\sum_{j=0}^{Q_M - 1} [\omega_i \in \mathcal{D}_\omega(t - jP)]}{Q_M} \tag{18}$$

$$IPMP(\omega_i, t) \geq T_{IPMP} \Rightarrow \omega_i \in \mathcal{D}_h(t) \tag{19}$$

6. *Interframe Magnitude Slope Deviation (IMSD)*: A temporal criterion that determines the deviation over Q_M successive signal frames of the slope. The differetiation is carried out between an old signal frame and more recent signal frames.

$$IMSD(\omega_i,t) = \frac{1}{Q_M-1}\sum_{m=1}^{Q_M-1}\Bigg[\frac{1}{Q_M}\sum_{j=1}^{Q_M-1}\frac{1}{Q_M-j} \cdot \big(20\log_{10}|X(\omega_i,t-jP)| - 20\log_{10}|X(\omega_i,t-Q_MP)|\big) - \frac{1}{m}\sum_{j=0}^{m-1}\frac{1}{m-j}\cdot\big(20\log_{10}|X(\omega_i,t-jP)| - 20\log_{10}|X(\omega_i,t-mP)|\big)\Bigg] \quad (20)$$

$$IMSD(\omega_i,t) \leq T_{IMSD} \;\; \Rightarrow \omega_i \in \mathcal{D}_h(t) \quad (21)$$

2.2 Multi-feature criteria

To gain better performance, multiple criteria can be combined with an logical AND operator. This improves the performance drastically as later can be seen. The AND-combination is demonstrated as an example with the PHPR & IMSD criteria. It can be also used for any other combination.

$$\left(\bigwedge_{m\in\mathcal{M}_{PHPR}}\big[PHPR(\omega_i,t) \geq T_{PHPR}\ \text{[dB]}\big]\right) \wedge IPMP(\omega_i,t) \geq T_{IPMP} \;\; \Rightarrow \omega_i \in \mathcal{D}_h(t) \quad (22)$$

Another feature, that combines basically the PNPR and IMSD criteria, is proposed in [6]. The Feedback Existence Probability (FEP) criterion combines peakness and slopeness features.

$$FEP(\omega_i,t) = 0.7\cdot slopeness(\omega_i,t) + 0.3\cdot peakness(\omega_i,t) \quad (23)$$

$$FEP(\omega_i,t) \geq T_{FEP} \;\; \Rightarrow \omega_i \in \mathcal{D}_h(t) \quad (24)$$

with

$$peakness(\omega_i,t) = \frac{1}{16}\sum_{j=0}^{7}\Bigg\{\Bigg[\frac{1}{6}\sum_{m=2}^{7}PNPR(\omega_i,t-jP,m) \geq 15\ \text{dB}\Bigg] + \Bigg[\frac{1}{6}\sum_{m=-7}^{-2}PNPR(\omega_i,t-jP,m) \geq 15\ \text{dB}\Bigg]\Bigg\} \quad (25)$$

$$slopeness(\omega_i,t) = e^{-|IMSD(\omega_i,t)|} \quad (26)$$

3 Implementation and optimization

This section explains some details about the implementation of the peak picking algorithm and the PHPR criterion. In section 3.4 and 3.5, two approaches for further

perfomance improvement are introduced. 3.5 is not included in the evaluation process, because we could not find a way, that the approach contributes to a performance improvement. But it gives the possibility to estimate the maximum stable gain of a system.

3.1 Peak picking algorithm

The peak picking algorithm [1] - here shown for the spectral magnitude $A(k) = |X(\omega_k)|$ with $k = 0...\frac{N}{2}$ - searches for changes in the its deviation $A'(k)$ from positive to negative values. These are local maxima of $A(k)$,

$$\begin{aligned} &A'(k) = A(k) - A(k+1); \\ &\{A'(k) \geq 0 \ \ \wedge \ \ A'(k+1) \leq 0\} \ \ \Rightarrow \omega_k \in \mathcal{D}_\omega \end{aligned} \tag{27}$$

This algorithm delivers the set of peaks $\mathcal{D}_\omega(t)$. After that we drop peak values for $k = 0$, since a DC peak is not useful and also $k = \frac{N}{2} - 7 \ ... \ \frac{N}{2}$, because it is easier for the implementation of some algorithms (PNPR, FEP), which operate on the picked peaks. At frequencies near 20 kHz the damping through loudspeakers and the air is usually that high, that acoustic feedback won't occur there in a practical scenario.

3.2 Improvement of the PHPR algorithm

As stated in Eq. (14), there could occur two problems:
– What if a peak with a continuous frequency $|Y(\omega_i, t)|$ lies between two DFT bins $|Y(\omega_k, t)|$ and $|Y(\omega_{k+1}, t)|$?
– What if the multiplication by the factors m does not correspond exactly to the harmonic component and misses it by some neighbor bins?

The solution to the first problem is to interpolate for every detected howling frequency. For the second problem, we define a tolerance bandwidth for harmonics. If a peak inside this tolerance bandwidth around a peak $|Y(m\omega_i, t)|$ is found, then this peak replaces the calculated bin. A quadratic interpolation is used, which follows the equation $y(x) = a(x - p)^2 + b$, to calculate the peak location

$$p = \frac{y_{-1} - y_{+1}}{2(y_{-1} - 2y_0 + y_{+1})} \tag{28}$$

and peak height

$$y = y_0 - 0.25(y_{-1} - y_{+1})p \tag{29}$$

for three adjacent samples $[y_{-1} \ y_0 \ y_{+1}]$. This is shown graphically in Fig. 3. If the interpolated frequency $m \cdot \omega_{i,int}$ lies inside the bandwidth $B = \frac{1}{30}$ with another frequency in the set $\mathcal{D}_\omega(t)$, the value from $\mathcal{D}_\omega(t)$ is taken instead of $|Y(m \cdot \omega_{i,int}, t)|$. This is only relavant at higher frequencies, where the DFT resolution has a finer grid than $\frac{1}{30}$ octave.

1. copyright 1994, by C.S. Burrus, J.H. McClellan, A.V. Oppenheim, T.W. Parks, R.W. Schafer, & H.W. Schussler. For use with the book "Computer-Based Exercises for Signal Processing Using MATLAB" (Prentice-Hall, 1994).

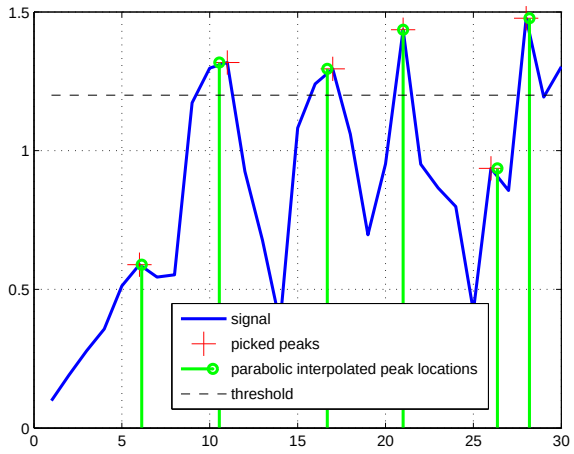

Figure 3: Peak picking and interpolation for PHPR

3.3 Cascade temporal criterion IPMP with other criteria as post-processor

The combination of the temporal IPMP feature does not improve the performance, if it is AND connected with other criteria as a multi-feature (see section 2.2). Toon van Waterschoot's and our simulations showed this result likewise, so we introduce a little tweak: We don't use a logical AND conjunction for IPMP, but we cascade the feature after all the others. The IPMP feature operates now as a post-processor on the output howling bins $\omega_h \in \mathcal{D}_h$ of one or more AND-joint features. So the input to the IPMP processor is a selection, not the whole set of peaks $\mathcal{D}_\omega(t)$. Fig. 4 illustrates the signal flow. This results in a kind of temporal smoothing, because the best working threshold turned out to be $T_{IPMP} = Q_M - 1$. If not stated otherwise, $Q_M = 5$ and $T_{IPMP} = 4$ is used.

$$IPMP(\omega_h, t) = \frac{\sum_{j=0}^{Q_M-1} [\omega_h \in \mathcal{D}_h(t - jP)]}{Q_M} \tag{30}$$

$$IPMP(\omega_h, t) \geq T_{IPMP} \quad \Rightarrow \omega_i \in \mathcal{D}_{h'}(t) \tag{31}$$

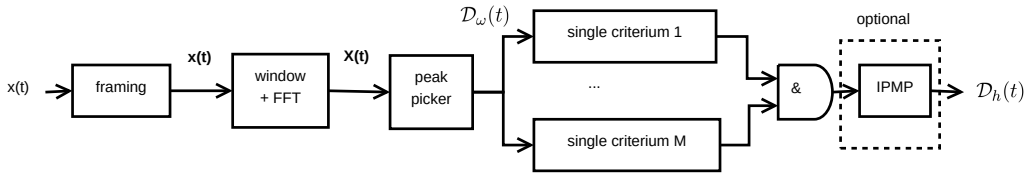

Figure 4: Calculation of the notch filter parameters $\mathcal{D}_h$ from microphone signal $x(t)$

3.4 Constant-Q Analysis

The spectral criteria PHRP and PNPR show a strongly varying performance The given linear frequency resolution of the DFT is quite low at lower octaves and high at the upper octaves, A logarithmic frequency resolution would fit a perceptive grid for musical content as howling components better. The Constant-Q-Transformation (CQT) [3] is a suited time-to-frequency domain transformation for this problem. It allows a spectral analysis holding logarithmic spaced spectral bins with equal bandwidth.
The CQT of a discrete-time signal $x(n)$ is defined by

$$X^{CQ}(k,n) = \sum_{j=n-\lfloor N_k/2 \rfloor}^{n+\lfloor N_k/2 \rfloor} x(j) a_k^*(j-n+N_k/2) \tag{32}$$

where the basis functions $a_k(n)$, also called time-frequency atoms are

$$a_k(n) = \frac{1}{N_k} w\left(\frac{n}{N_k}\right) \exp\left[-i2\pi n \frac{f_k}{f_s}\right]. \tag{33}$$

w denotes a window function and the center frequencies obey

$$f_k = f_1 2^{\frac{k-1}{B}}. \tag{34}$$

The frequency dependent window length N_k computes as follows, with $0 < q \leq 1$ being a scaling factor and B the number of bins per octave.

$$N_k = \frac{q f_s}{f_k(2^{\frac{1}{B}} - 1)} \tag{35}$$

For the implementation in Matlab, the CQT-Toolbox published in [5] was used, that gives a quick usable and efficient implementation of a constant-Q transform.

3.5 Two Microphone Method

An additional idea to get better performance of howling detection algorithms is to use a second microphone capsule, coincident or spatially separated, to get more information and calculate the features from this second microphone signal. But since the information, the second microphone picks up, is not really different to the first microphone, the feature calculation cannot be improved further.
At least we can use the second capsule to calculate the actual gain factor a, that is placed in the electroacoustical forward path $G(e^{j\omega}) = a \cdot G_n(e^{j\omega})$ with $\max(G_n(e^{j\omega})) = 1$.

$$y_1(t) = V_1 s(t) + a H_1 y_1(t); \;\Rightarrow\; y_1(t) = \frac{V_1 x(t)}{1 - a H_1} \tag{36}$$

$$y_2(t) = H_2 x(t) + y_1(t) \cdot a V_2 = H_2 x(t) + \frac{V_1 x(t)}{1 - a H_1} \cdot a V_2 \tag{37}$$

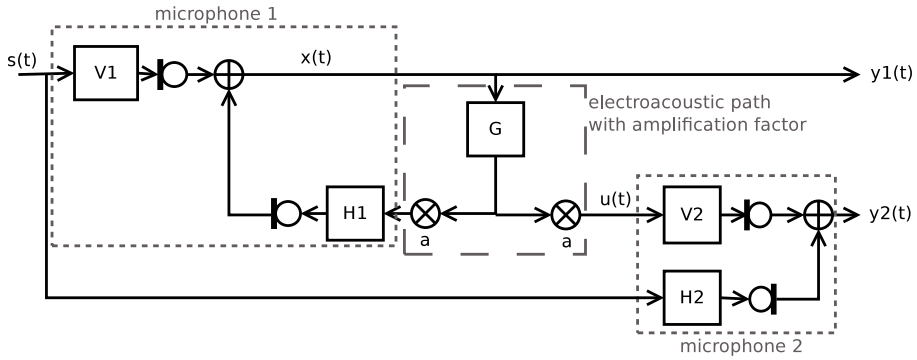

Figure 5: Source leakage using two microphone capsules, the first pointing to the signal $s(t)$, the second aiming at the loudspeaker signal $u(t)$

$$\frac{y_2(t)}{y_1(t)} = \frac{H_2(1 - aH_1)}{V_1} + aV_2 = \frac{H_2}{V_1} + a \cdot \left(V_2 - \frac{H_1 H_2}{V_1} \right) \tag{38}$$

Assuming that the front-transfer functions of the microphones are ideal, $V_1 = V_2 = 1$, we can estimate the absolute value of the gain factor in Eq. (39) by squaring each side of Eq. (38) for energy and resolving the equation to a.

$$|a| \approx \frac{H_1 H_2 - 1}{2\left(H_2 \pm |\frac{y_2(t)}{y_1(t)}|\right)} \tag{39}$$

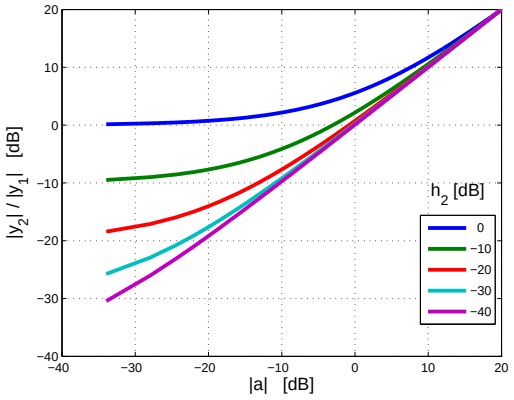

Figure 6: Relationship between feedback loop gain factor a and the signal ratio of two microphones for different backwards attentuations h_2 of microphone 2

The relationship depicted in Fig. 6 becomes more linear with a decreasing backwards damping factor $|h_2|$. This lets assume that the optimal directivity pattern for the second microphone would be a cardiod, pointing backwards to the speaker.

4 Evaluation of the criteria

The evaluation as Toon van Waterschoot et al. did, seemed to have some weak points. After discussing the realization according to his method, another new developed method to evaluate the algorithms is presented.

4.1 First Environment

The method evaluates an audio file, that contains a signal and a building up howling component. For this reason, a feedback loop was built up in Pure Data and several audio files were generated with some combinations of signals and impulse responses. They are listed in Table 1. A signal - impulse response pair is named after the signal label (Tab. 1 left column) and the impulse response identifier letter (Tab. 1 3^{rd} column) as appendix or as subscript. The audiofiles are truncated when they reach a certain absolute signal

signals		impulse responses	
label	description	label	description
Mendel	Classical Orchestra: Mendelssohn wedding march	F	livingroom, measured with Fostex nearfield monitors
speech	female speech	K	conference room
Tenor	singing tenor		
Violine	Violine (original used by [2])		impulse response used by [2]

Table 1: Labeling and description of audiofiles and impulse responses

power threshold. This results in a varying length of about 30 seconds, depending on how fast the howling builds up. The following analysis is implemented in Matlab. The whole audiofile is analyzed and the detected howling bins of each criterium are recorded.

4.1.1 Evaluation method

To calclulate a probability of detection P_D and a probability of false alarm P_{FA} we need to know the number of bins, that correspond to true howling, the number of true positives N_{TP}, and the bins that are delivered by the algorithm as howling bins, but don't correspond to howling: the false positives N_{FP}. By dividing the true positives by all true howling bins N_P, we calculate the hitrate to

$$P_D = \frac{N_{TP}}{N_P} \tag{40}$$

The negative realizations of the data set N_N allows us to calculate the false alarm rate. The number of negative realizations is obtained from the number of the peak picker outputs without the number of true positives.

$$P_{FA} = \frac{N_{FP}}{N_N} \qquad \text{with} \quad N_N = N_{\mathcal{D}_\omega} - N_{TP} \tag{41}$$

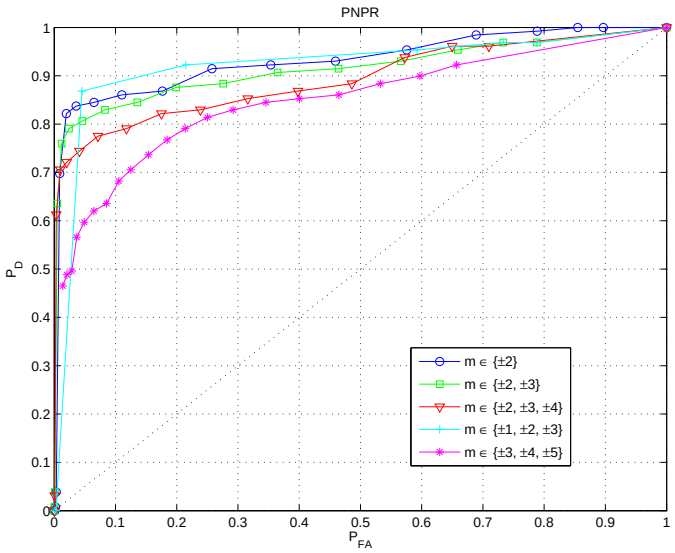

Figure 7: ROC of PNPR algorithm for the violine signal

Plotting these two rates against each other results in the so-called Receiver-Operating-Curves (ROC). Fig. 7. is shown as an example for the $PNPR$ algorithm. The used signal and room impulse response were the same as used in [2]; these are available on a website [2].

4.1.2 Defining the true howling frequencies

A difficulty here is, how to define the "true" howling frequencies and their quantity N_P? The howling frequency is determined by DFT analysis of buffered signal frames with high howling power and its pitch is controlled by ear and a sinus generator. Then the strongest

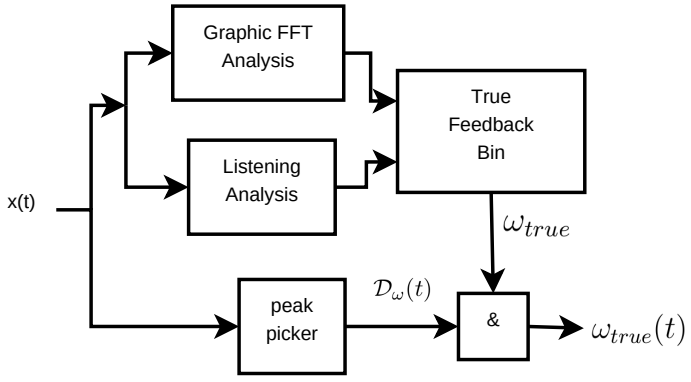

Figure 8: Defining the true howling bin per frame $\omega_{true}(t)$

bin in this region is assumed to be the howling bin. By analyzing the signal with the peak picker (see 3.1), all peaks $\mathcal{D}_\omega(t)$ corresponding to this strongest howling bin ω_{true}

2. `ftp://ftp.esat.kuleuven.be/pub/SISTA/vanwaterschoot/abstracts/09-207.html`

are assumed to be true howling and not so the rest. This process is schematically shown in Fig. 8.

4.1.3 Problems of this simulation method

As in this method a precomputed audio file that already contains feedback is used to calculate the feedback criteria from it, several problems arise:
– With increasing file length, the howling gets more prominent.
– In the sequence where the feedback level has already built up higher over the signal, it is easy for the algorithms to detect the howling frequency.
– The false alarm rate is therefore dependent on the file length, respectively on how fast the feedback builds up and is powerful in the signal.

With this method the criteria are only comparable, if one single audio file is used. As audio material can differ drastically, the method is not objective for all sound sources, only for the one specific tested case. For that reason another test scenario is developed as follows.

4.2 New Simulation Environment

A feedback loop is built in the real-time software Pure Data with a defined impulse response. As impulse resonse, a bandpass filter is used as a bell-peak filter at 1 kHz. The filter $h[n]$ is realized as FIR because of the ability, that it can be a linear-phase filter. As the gain in the feedback loop gets close to 1, the gain at 1 kHz is greater than 1 and howling occurs. As we know the shape of the feedback path, we can insert a inverse filter $g[n]$ to stop howling. The frequency responses of both filters are depicted in Fig. 9. So everytime the feedback criterion detects the frequency bin of $f_i = 1$ kHz, the filter $g[n]$ is inserted for 50 ms (slightly more than one hop-size duration), which prevents the system from getting unstable. This way we can run the system as long as we want, getting better statistical values.

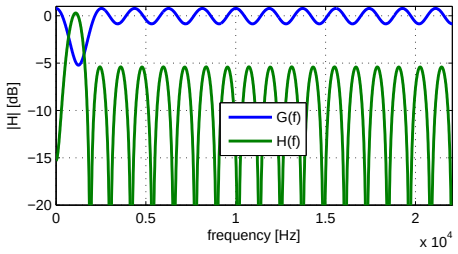

Figure 9: Artificial room impulse response $h[n]$ and compensation filter $g[n]$

We now can't go for a hit rate, as the howling disappers everytime it is detected. Instead we compute two other values to get a performance measure of the algorithms:
– The time between two hits is the detection time t_D. We compute the mean value as the mean detection time $\bar{t}_D$.

– The signal power of the whole feedback loop. The longer the algorithm needs to detect and cancel the right howling frequency, the higher the signal power rises.

The signal can influence the mean detection time t_D since it can take longer for howling to build up, if the signal power is very weak or zero right after the filter $g[n]$ is deactivated. The other way round, t_D can be very short if the signal power is great. So if we take the average signal power, we have a quite good measure for how much howling occurs in the system, until it gets detected and cancelled.

To get a relative value, a second reference feedback loop is built with the filters $h[n]$ and $g[n]$ always activated. We now measure the signal power of the actual feedback loop $E_S(t)$ and the reference system $E_{ref}(t)$ and can get a relative power value in Eq. (42) by dividing the mean values. $\overline{E}$ describes how much additional signal power is added by howling because of the system being unstable.

$$\overline{E} = \frac{\overline{E}_S(t)}{\overline{E}_{ref}(t)} \tag{42}$$

The system is depicted in Fig. 10 with the actual feedback loop on top and the reference loop below. Since the filter $g[n]$ introduces a delay of d samples to the forward path, a delay expressed as Dirac-Delta function $\delta[n]$ compensates the path without $g[n]$. The only purpose of the reference loop is to calculate E_{ref}. The gain factor a controls the amount of feedback. If $|ah[n]| > 1$ for one frequency bin (1 kHz in this case), the system gets unstable and starts howling at this frequency.

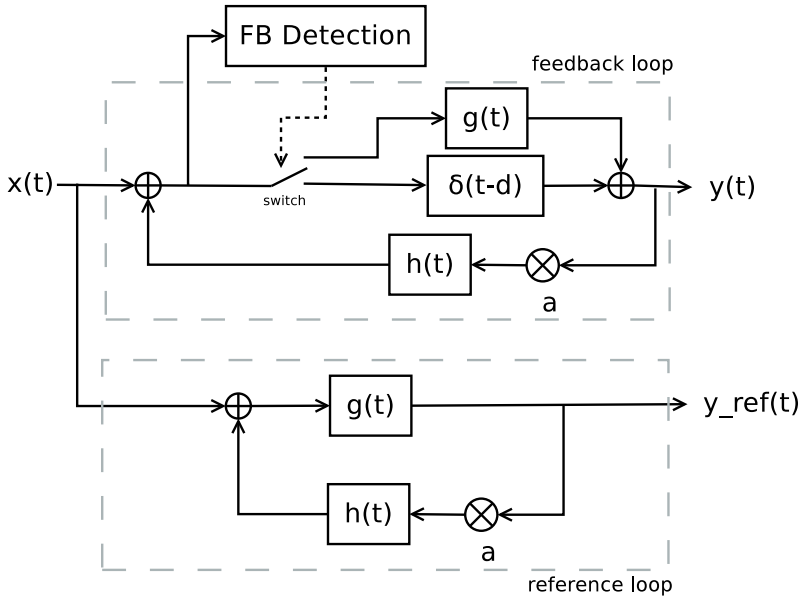

Figure 10: Feedback loop and reference loop. The detection algorithm switches the filter $g[n]$ in for 50 ms to keep the loop stable.

Testsignals To reduce the effort of the simulation, three test signals are chosen. They have already been presented in section 4.1. We chose only single instrument sources, since in a given situation it is more likely that we have a close-miking setup. The three test samples are the *Violine*, a singing *Tenor* and female *Speech*, that are labeled likely.

4.2.1 Additional criterion: HBPF

A new criterion is introduced because of the signal-interactive behaviour of this new developed evaluation method. In contrast to the method proposed in 4.1, it matters here especially in terms of the detection speed, that can't be measured with the method of section 4.1.
The detected howling frequencies $\mathcal{D}_h(t)$ are straight used to steer the set of notch-filters. A further idea, how to lower the detection of false howling frequencies, is based on this assumption: Only one frequency per time frame can be a true howling feedback, all others are assumed to be false detected ones. To prevent this, only the *Highest Bin Per Frame (HBPF)* from the set $\mathcal{D}_h(t)$ is taken, that means the howling component with the greatest power. Using this additional criterion, only one notchfilter can be changed or set per frame.

$$HBPF(\omega_h, t) = \max\big(|X(\omega_h, t)|\big) \tag{43}$$

Fig. 11 shows the signal flow graph including now all implemented processing blocks. One or more of the 6 single criteria (section 2.1) can be combined with a a AND conjunction. The following post-processors for IPMP and HBPF can be switched in optional.

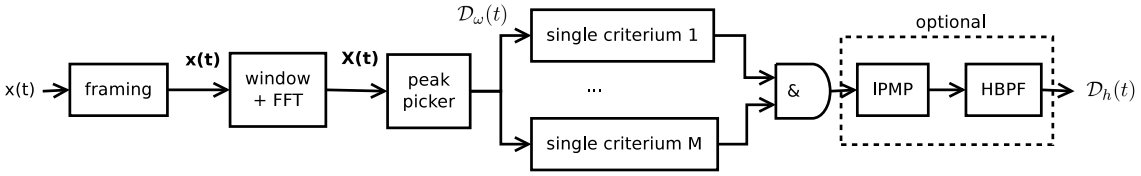

Figure 11: Calculation of the notch filter parameters $\mathcal{D}_h$ from microphone signal $x(t)$

5 Simulation results

5.1 First simulation method

The diagrams obtained with the first simulation method are similar to the reference publication [2], but they can't be reproduced exactly. Even when using the same signal and impulse response in the feedback loop, they come close but some differences remain. To compare the algorithms with different signals and impulse responses, we compute the "Partial Area Under the Curve" (PAUC) for every realization.
We can observe, that the signal spectrum dependent criteria $PHPR$ and $PNPR$ are

influenced by some characteristics of the signals or impulse responses, for example if the howling frequency bin is shared with a harmonic part of the signal, or the howling frequency lies exactly between two bins. For the other features, the tendencies remain more constant in relative manner. The simplest features $PTPR$ and $PAPR$ seem to be the best though.

5.1.1 Temporal development of Hitrate and False Alarm Rate

To get a little more insight into the temporal (frame-wise) behaviour of the algorithm, we analyze the development of $P_D(t)$ and $P_{FA}(t)$. The hitrate can only reach a value of 1 or 0 per frame, since we have only one true howling frequency. We calculate a cumulative hitrate, which takes all hits until the actual point of time into account. For the false alarm rate we do the same.

$$P_{D,cum}(t) = \sum_{k=1}^{k=t} \frac{N_{TP}(k)}{N_T(k)} \tag{44}$$

$$P_{FA,cum}(t) = \sum_{k=1}^{k=t} \frac{N_{FP}(k)}{N_N(k)} \tag{45}$$

$P_D(t)$ is a solid line, $P_{FA}(t)$ is dashed. For the female speech signal the temporal development is shown in Fig. 12. A little algorithm was used to search for the "best" threshold and parameter values of each criterion. It searches the highest possible P_D at a chosen $max(P_{FA}) \leq 0.1$.

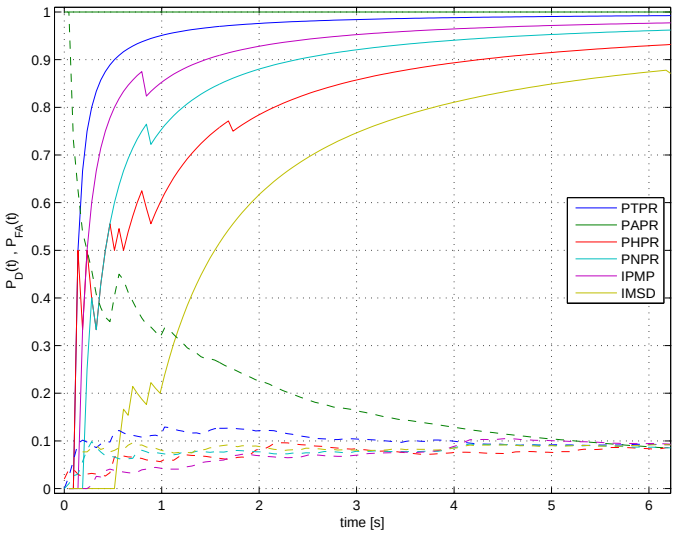

Figure 12: Temporal development of hitrate and false alarm rate

We can observe that an (exponential) growth of the curves against its final values. While a P_D-curve is growing, it detects the right howling bin; if it decreases, it looses

the howling. $IPMP$ has the smallest problem with losing the howling, because it is just kind of a temporal smoothing filter. Fig. 12 also points out very good that the results of the ROC-Curves differ, depending on the audiofile length. One can read the temporal probability values by just cutting vertically through the curves at one specific point of time.
The $PAPR$ algorithm has found the true howling frequency from the beginning. But the problem of this evaluation method is clarified most with the $PAPR$-curve: The longer the audio segment is, the more P_{FA} grows against zero. So this evaluation method diminishes the evaluation emphasis of the critical phase, where the howling begins to build up and is not louder than the signal. But this is the critical and essential phase of such an algorithm. If the howling power lies already several dB over the signal power, it is easy to detect the howling. The early phase where howling builds up, is where the algorithms should later operate at, because if they detect howling, it will be immediately notched out. That's the reason why the second evaluation method in 4.2 is developed.

5.2 Comparison of CQT vs. DFT

By comparing three audiofiles using the same signal but three different impulse responses with different howling frequencies (280, 506 and 6008 Hz), Fig. 18 shows a improvement for the low frequency compared to Fig. 17. The middle (Fig. 15 & 16) and high frequency howling ROCs (Fig. 13 & 14) show no significant change in performance. The plots for the three situations are opposed with the usual DFT analysis on the left and with CQT analysis on the right side.

If the computing power is available for a CQT analysis, the use in a detection system might improve the performance. But if one can assume, that howling is not a problem at lower frequencies < 500 Hz, the usage of a CQT analysis has no improving effect. Also the needed computational power gets quite high, if a reasonable hop-size has to be held. Fig. 13 - 18 were generated using a CQT analysis with 96 bins per octave. By dropping the CQT resolution to 48 or 24 bins per octave, the advantage decreases.

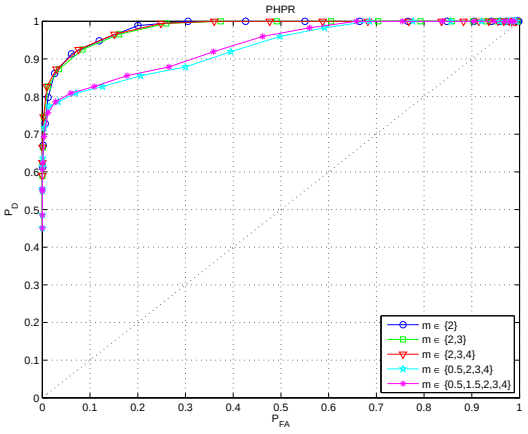

Figure 13: $f_{howl} = 6008$ Hz

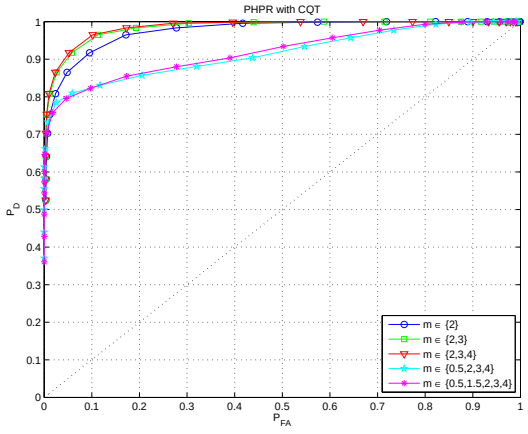

Figure 14: $f_{howl} = 6008$ Hz

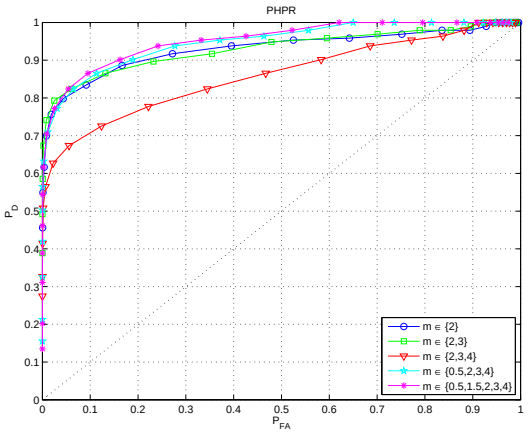

Figure 15: $f_{howl} = 506$ Hz

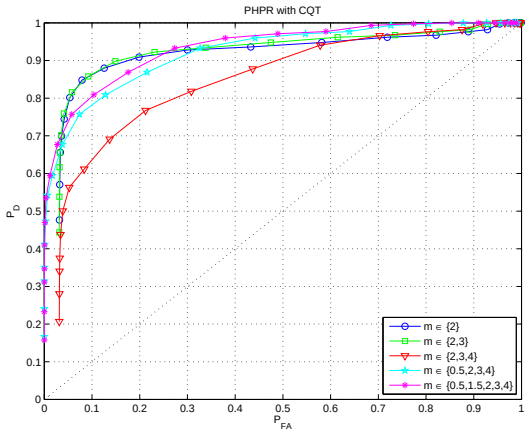

Figure 16: $f_{howl} = 506$ Hz

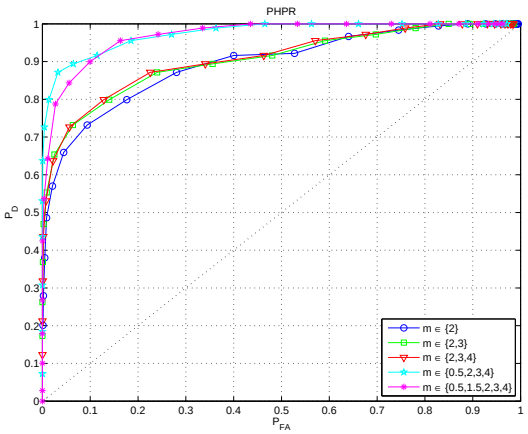

Figure 17: $f_{howl} = 280$ Hz

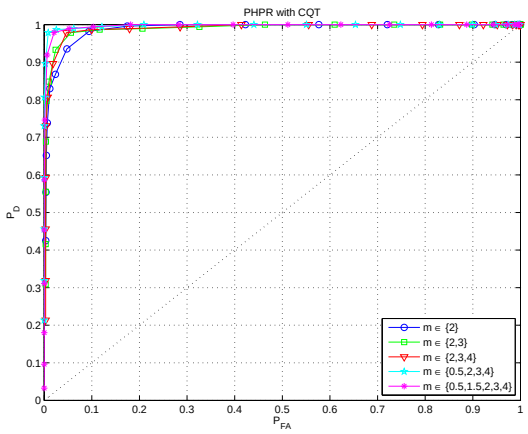

Figure 18: $f_{howl} = 280$ Hz

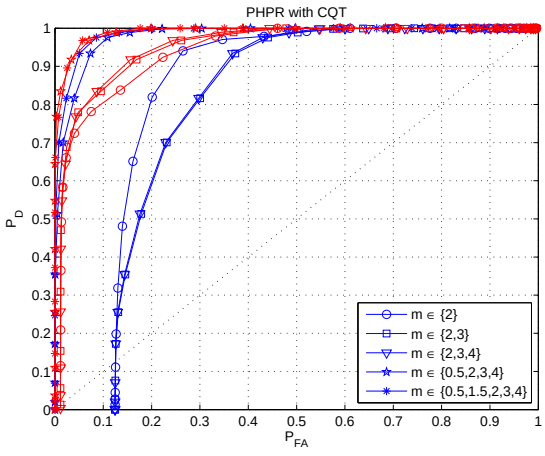

Figure 19: PHPR with CQT transformation 24 bins per octave (blue) and 48 (red), $f_{howl} = 280$ Hz

Fig. 19 shows the low-frequency howling file with 24 (blue) and 48 (red) bins/octave. With 48 bins/octave there is still a small improvement to Fig. 17, using just 24 bins might be a quality decrease. Another advantage of a CQT analysis is the variable window length, decreasing with the frequency. This might be useful for a howling detection as well, since lower frequencies build up slower than high frequencies. This corresponds to the low hop-size of the CQT at the bottom and a high hop-size on the top.

5.3 New Real-time simulation method

Labeling of the evaluation plots in this section For reasons of saving space, the labeling in the plots is kept short. Since most features have more parameters than just the threshold T, they are abbreviated as listed in Table 2. The index i stands in m_i for the set of factors/adders and in Q_i for the number of frames, the criterion operates on.

PHPR		PNPR		IMSD		FEP	
m_1	$m \in \{2\}$	m_1	$m \in \{\pm 2\}$	Q_i	$Q_M = i$	Q_i	$Q_M = i$
m_2	$m \in \{2, 3\}$	m_2	$m \in \{\pm 2, \pm 3\}$				
m_3	$m \in \{2, 3, 4\}$	m_3	$m \in \{\pm 2, \pm 3, \pm 4\}$				
m_4	$m \in \{0.5, 2, 3, 4\}$						
m_4	$m \in \{0.5, 1.5, 2, 3, 4\}$						

Table 2: Labeling of the following graphics

5.3.1 Evaluation of single criteria

In this section we discuss the results of the second simulation method, described in 4.2. As performance measures there are always two basic parameters to evaluate. Between these one has to find a trade-off:

(+) How fast and secure the algorithm recognizes the howling, in the optimal case even before it's noticeable to a common audience. You can also consider the hit-rate as in the first simulation.

(−) How many wrong howling frequencies are detected, which can destroy the sound signal, if too many notches are set. This describes the false alarm rate.

To measure the detection speed, the mean time $\bar{t}_D$ between correct detected howling bins is plotted in Fig. 20. To avoid the problem described in 4.2 that $\bar{t}_D$ is signal dependent, the mean signal power expresses the same tendencies without correlating with the signal pauses (see Fig. 20 and 21). Unfortunately the units of $\overline{E}$ in dB are not as intuitive to read as $\bar{t}_D$, but the tendencies are the same and therefore it should be considered a better and more appropriate performance measure.

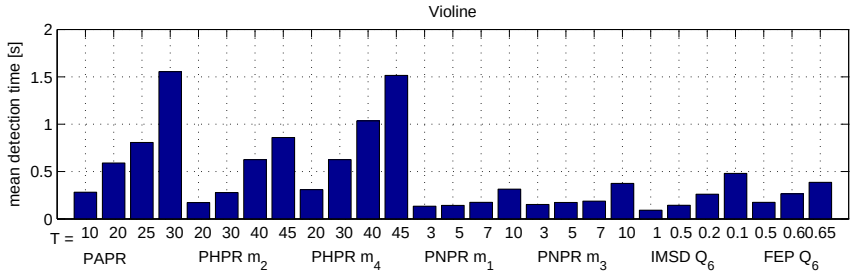

Figure 20: $\bar{t}_D$ for the single criteria with different Thresholds T

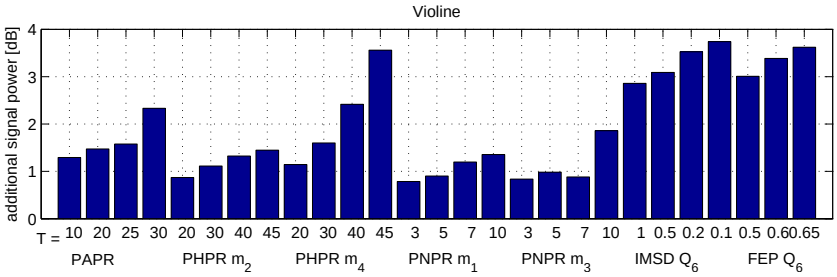

Figure 21: $\overline{E}$ for the single criteria with different Thresholds T

First the false alarm rate is calculated for every frame after

$$P_{FA}(t) = \frac{N_{FP}(t)}{N_N(t)}. \tag{46}$$

Now we can analyze the statistical spread of $P_{FA}(t)$, shown in Fig. 22 as boxplots. It implies that with increasing thresholds within each criterion, the spread diminshes. Since the same tendencies are represented by just taking the average over time of $P_{FA}(t)$, we found a value to represent the false alarm properly. To take care of the outliers or the

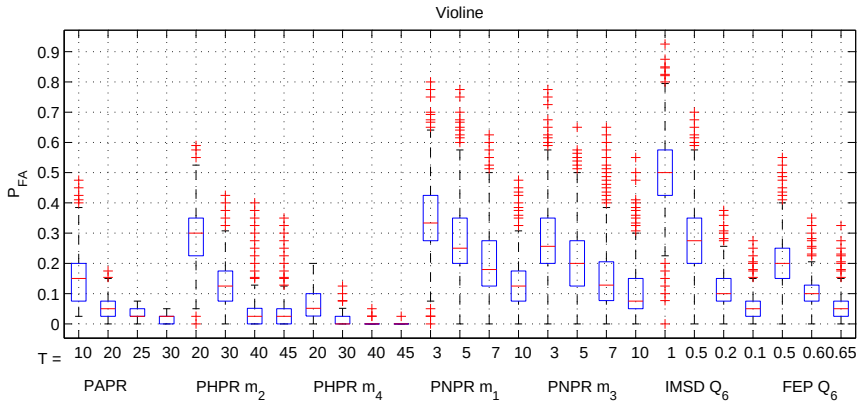

Figure 22: statistical spread as boxplot for $P_{FA}(t)$

maximum $P_{FA}(t)$, that occurred in one frame, we combine average and maximum value with a weighting after Eq. (47). Fig. 23 shows how the new built representative false

alarm rate $P_{FA}^{\sim}$ consists of the blue mean value and the red maximum.

$$P_{FA}^{\sim} = 0.9 \cdot \overline{P}_{FA} + 0.1 \cdot max\big(P_{FA}(t)\big) \tag{47}$$

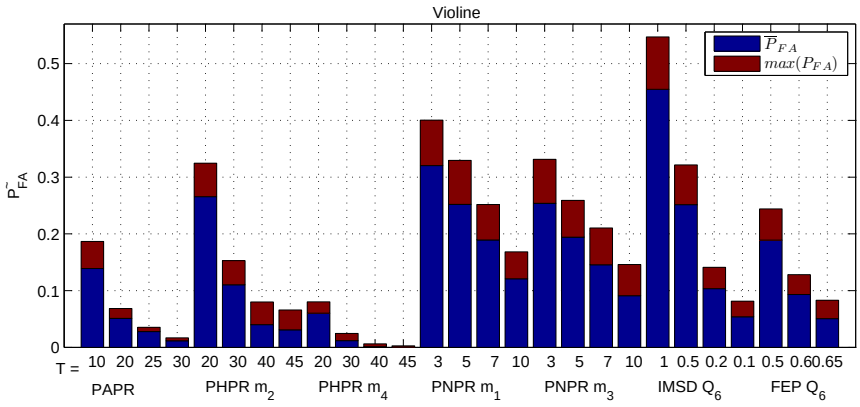

Figure 23: Combination of average and maximum value to $P_{FA}^{\sim}$

5.3.2 Evaluation of additional post-processing criteria IPMP and HBPF

Fig. 24 shows how the addition of the optional cascaded features improves the performance. The diagram uses already the new developed evaluation method in advance, which is explained in chapter 4.2 in detail. The false alarm rate $P_{FA}^{\sim}$ is plotted against the average additional signal power caused by instable feedback $\overline{E}$. The optimal point is therefore in the lower left corner.

The single $PHPR$ criterion with harmonic factors $m \in \{2, 3\}$ is drawn in blue for four different thresholds. The addition of the $HBPF$-feature (green) shows a significant improvement: the false alarm rate is and even the additional signal power is reduced. The cascade of $IPMP$ with $PHPR$ (red) reduces for three data points also the false alarm rate. $\overline{E}$ is slightly higher. The addition of both features at the same time (cyan) reduces the false alarm rate dramatically on the cost for a higher $\overline{E}$.

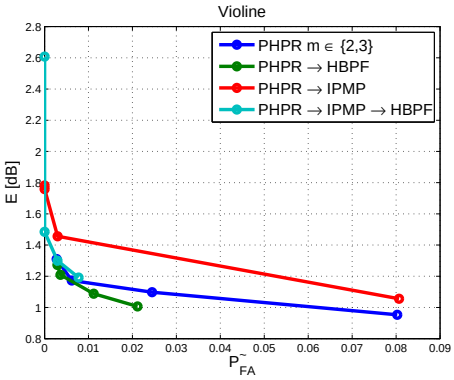

Figure 24: Improvement through cascade with HBFP and IPMP features

5.3.3 Comparison and evaluation of the best performing features

These two representative tradeoff-measures - $\overline{E}$ and $P_{FA}^{\sim}$ - can be plotted against each other to obtain similar diagrams as in the first simulation method in chapter 5.1, but mirrored horizontally. For three different signals, this depicted in Fig. 25, 27 and 29 comparing the single criteria.

In a practical case we want a criterion that delivers the notchfilter-parameter set $\mathcal{D}_h$ with a quite low false alarm rate, otherwise we still have to add another post-processing and election of the delivered howling bins. Therefore most of the single-feature criteria are not good enough anyway, only the PAPR and PHPR with m_4 are settled in a resonable ratio. So we have a look mainly at the combined features. Since the maximum number of outputs of the peak picker is

$$|\mathcal{D}_\omega(t)|_{max} = 40, \tag{48}$$

we aim at features, that hold a false alarm rate

$$P_{FA}^{\sim} < \frac{1}{40} = 0.025. \tag{49}$$

This means that the feature detects in average not more than one false howling frequency per frame and the number of maximum false frequencies doesn't exceed to excessively.

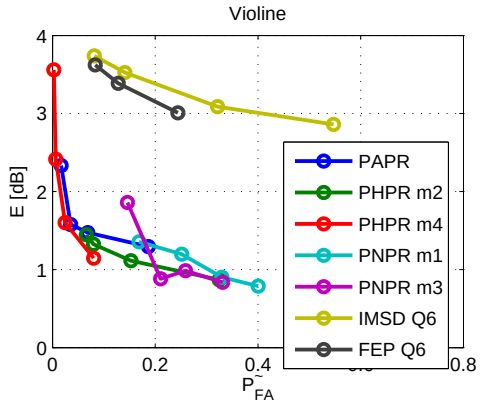

Figure 25: performance of single feature criteria with different thresholds

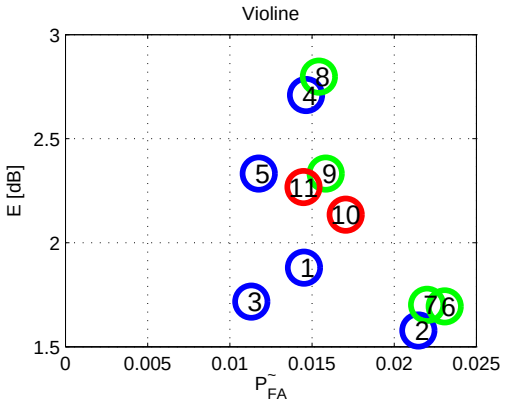

Figure 26: combined features

Figures 26, 28 and 30 show some combined features, that hold the in Eq. (49) postulated criterion. The numbers inside the circles label different criteria combinations and threshold settings. They are listed in Tables 3, 4 and 5 with the threshold values as subscripts. The additional power E cannot be taken into account as absolute measure to compare between the different used signals. But most of the features stay more or less constant regarding their relative position for the each audio file. The combinations of PHPR, PNPR and HBPF (label 3&4) and the combinations of the three best single features (label 10) seem to be stable and deliver reasonable results.

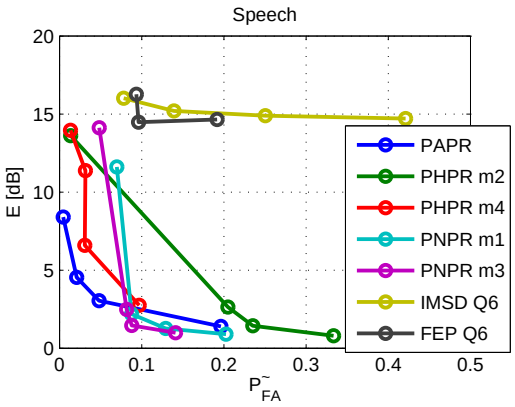

Figure 27: performance of single feature criteria with different thresholds

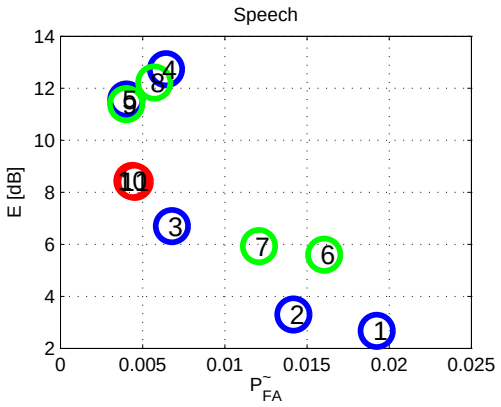

Figure 28: combined features

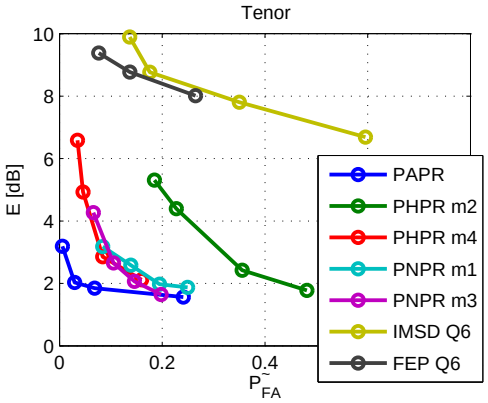

Figure 29: performance of single feature criteria with different thresholds

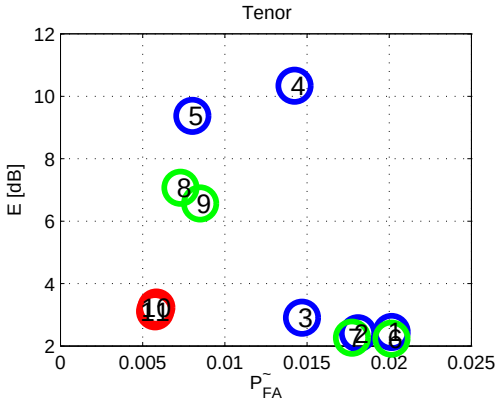

Figure 30: combined features

plot label	feature	$P_{FA}^{\sim}$ [%]	$\overline{E}$ [dB]	$\bar{t}_D$ [ms]
1	$PHPR_{10}$ $PNPR_{10}$ HBPF	1.45	1.88	867
2	$PHPR_{20}$ $PNPR_{10}$ HBPF	2.15	1.58	485
3	$PHPR_{30}$ $PNPR_{10}$ HBPF	1.13	1.72	624
4	$PHPR_{20}$ $PNPR_{10}$ IPMP	1.46	2.71	1216
5	$PHPR_{15}$ $PNPR_{5}$ IPMP HBPF	1.18	2.33	1049
6	$PAPR_{20}$ $PHPR_{15}$ HBPF	2.31	1.69	910
7	$PAPR_{25}$ $PHPR_{25}$ HBPF	2.20	1.70	910
8	$PAPR_{25}$ $PHPR_{25}$ IPMP	1.54	2.80	1597
9	$PAPR_{20}$ $PHPR_{20}$ IPMP HBPF	1.58	2.33	1207
10	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$	1.71	2.13	1374
11	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$ HBPF	1.45	2.27	1445

Table 3: Violine: Results of best features

plot label	feature	$P_{FA}^{\sim}$ [%]	$\overline{E}$ [dB]	$\bar{t}_D$ [ms]
1	$PHPR_{10}$ $PNPR_{10}$ HBPF	1,92	2,67	665
2	$PHPR_{20}$ $PNPR_{10}$ HBPF	1,42	3,30	307
3	$PHPR_{30}$ $PNPR_{10}$ HBPF	0,68	6,70	1037
4	$PHPR_{20}$ $PNPR_{10}$ IPMP	0,64	12,74	908
5	$PHPR_{15}$ $PNPR_{5}$ IPMP HBPF	0,40	11,57	756
6	$PAPR_{20}$ $PHPR_{15}$ HBPF	1,60	5,60	374
7	$PAPR_{25}$ $PHPR_{25}$ HBPF	1,21	5,93	462
8	$PAPR_{25}$ $PHPR_{25}$ IPMP	0,57	12,22	1067
9	$PAPR_{20}$ $PHPR_{20}$ IPMP HBPF	0,40	11,39	736
10	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$	0,44	8,44	701
11	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$ HBPF	0,45	8,40	694

Table 4: Speech: Results of best features

plot label	feature	$P_{FA}^{\sim}$ [%]	$\overline{E}$ [dB]	$\bar{t}_D$ [ms]
1	$PHPR_{10}$ $PNPR_{10}$ HBPF	2.02	2.45	299
2	$PHPR_{20}$ $PNPR_{10}$ HBPF	1.81	2.40	309
3	$PHPR_{30}$ $PNPR_{10}$ HBPF	1.47	2.91	452
4	$PHPR_{20}$ $PNPR_{10}$ IPMP	1.42	10.34	723
5	$PHPR_{15}$ $PNPR_{5}$ IPMP HBPF	0.80	9.37	711
6	$PAPR_{20}$ $PHPR_{15}$ HBPF	2.01	2.24	426
7	$PAPR_{25}$ $PHPR_{25}$ HBPF	1.77	2.27	438
8	$PAPR_{25}$ $PHPR_{25}$ IPMP	0.73	7.07	847
9	$PAPR_{20}$ $PHPR_{20}$ IPMP HBPF	0.85	6.57	647
10	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$	0.58	3.23	611
11	$PAPR_{30}$ $PHPR_{30}$ $PNPR_{10}$ HBPF	0.58	3.11	593

Table 5: Tenor: Results of best features

5.4 Subjective evaluation with PD patch

A patch is developed in *Pure Data* for a real-time subjective evaluation. Two options corresponding to the evaluation methods are available:

- `Feedback_Detection_only.pd`: A "static" implementation, where a audiofile containing a growing howling component is played back. The algorithm detects the howling and attenuates it down via notchfilters.
- `fb_sim_and_cancel.pd`: A "dynamic" implementation containing a feedback loop with a measured room impulse response. After having chosen the algorithm and setting it up, a signal can be played back through a feedback loop. By turning up the feedback gain, the system starts to get unstable and to howl. If the algorithm is set up properly it will detect the howling and set a notchfilter, so that the system is stable again.

Fig. 31 shows the graphical user interface to set up the threshold values and parameters of the different algorithms. It is possible to use three of the "usual" criteria that are AND-conjuncted. The $IPMP$ and $HBPF$ features can be switched in separately. The 20 notch-filter gains and center frequencies are displayed on the bottom. The number of detected frequencies can be observed in the orange field on the right. These displays make it easier to keep track of the momentary state and support the listening impression. The observation of false detections is quite interesting.
If the $HBPF$-feature is switched off, it is useful to use more than one detection features with higher threshold values. Otherwise it can happen, that up to all 20 notch-filter parameters are changed quickly. This degrades the audio signal extremely and the audio processing of *PD* could get stuck by too high CPU load.

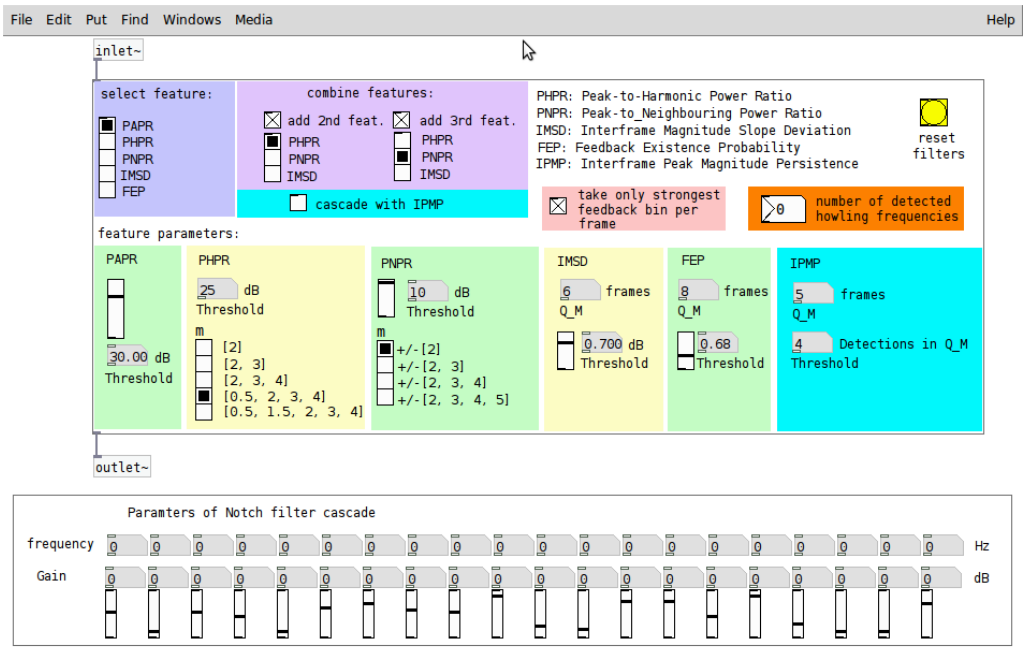

Figure 31: GUI of the PD-patch to set up the detection algorithms

Notch-filter design The control and design algorithm for the bank of 20 notch-filters is kept as simple as possible. Nevertheless, it this algorithm is essential for a subjective good performance of the howling canceller, so some tweaks are listed here.
- Blocksize: $N = 2048$ samples, $N/2$ overlapping, $f_s = 44100\frac{1}{s}$.
- Fixed notch-filter bandwidth: $\frac{1}{30}$ octave.
- If a frequency is within the bandwidth, where already a notchfilter was or is set, the same existing filter with its center frequency unchanged is used.
- Gain-reduction value: -3 dB. Everytime a new or before existing howling frequency is detected, the corresponding notch-filter gain is reduced by -3 dB until a maximum value of -30 dB.
- Gain-make-up value: +2 dB. If a notch-filter gain was not reduced for some consecutive frames, the filter gain is made up by +2 dB again until it reaches 0 dB or is reduced again.
- Time constant: 2 sec. Make-up gain by +2 dB after 10 frames, if no corresponding howling frequency is detected.

This design ensures a smooth operation of not too fast gain reductions, which can annoy the listener. After some time, notch-filters are taken out again, if they are not needed anymore. This can be due to a change of the room situation. If a feedback frequency would be always present, it levels to a stable reduction value.

6 Conclusion and outlook

By implementing and examinating the most common feedback howling detection criteria, a good overview of the state of the art is gathered. A new, more stable method for evaluation is developed and applied. Moreover, a method to estimate the feedback loop gain by using a second microhpone is proposed, but could not be tested sufficiently within the framework of this work.
The most effective algorithms focus on the short-time spectrum. The incorporation of temporal features gives a little more stability (i.e. reduction of false alarm), but at the cost of a much slower reaction time.

To get a perceptual motivated evaluation of the algorithms, a subjective listening test should be considered for future work. But that includes also strongly the notch-filter controlling and design. The effects of spectral distortion by setting unneeded notch-filters is an important information to design a good algorithm. Since the algorithms are coded in Matlab and Pure Data, they can form the basis of a following research and develop a more advanced method to control and design the notch-filter parameters.

References

[1] Waterschoot, Toon van and Moonen, Marc: "50 Years of Acoustic Feedback Control: State of the Art and Future Challenges", Proc. IEEE, vol. 99, no. 2, Feb. 2011, pp. 288-327.

[2] Waterschoot, Toon van & Moonen, Marc: "Comparative evaluation of howling detection criteria in notch-filter-based howling suppression", AES Journal vol. 58, no. 11, Leuven, 2010

[3] J. C. Brown, "Calculation of a constant Q spectral transform", J. Acoust. Soc. Am., vol. 89, no. 1, pp. 425-434, 1991.

[4] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform", J. Acoust. Soc. Am., vol. 92, no. 5, pp. 2698-2701, 1992.

[5] Schörkhuber, Christian and Klapuri, Anssi: "Constant-Q Transform Toolbox for Music Processing", 7th Sound and Music Computing Conference, Barcelona, Spain, 2010.

[6] N. Osmanovic, V. E. Clarke, and E. Velandia, "An In-Flight Low Latency Acoustic Feedback Cancellation Algorithm",123rd AES Convention, J. Audio Eng. Soc., 2007 Oct., convention paper 7266.

[7] Horn, Martin and Dourdoumas, Nicolaos, "Regelungstechnik", Pearson Studium, München 2004.