A Differential Microphone Array with Input Level Alignment, Directional Equalization and Fast Notch Adaptation for Handsfree Communication

Bernd Geiser1,2, Hauke Krüger1,2, Peter Vary1, Detlef Wiese3

1Institute of Communication Systems and Data Processing (ind), RWTH Aachen University, 52056 Aachen, Germany
2Java Solutions GmbH, Gallerstraße 33, 52074 Aachen — 3Binauric SE, Am Söldnermoos 17, 85399 Hallbergmoos

Abstract

We revisit the standard differential microphone of [1] and propose three algorithmic enhancements. First, a simple alignment procedure equalizes the power levels of the input signals. Second, we describe an efficient algorithm to equalize the distorted directional response of the original algorithm. This enables a wider operating frequency range and helps to reduce the noise gain of the array. Third, a fast time-domain notch adaptation algorithm is presented to reliably track interfering sound sources. Its tracking performance is not impaired by the activity of desired sound sources. Finally, we describe a fixed-point arithmetic implementation of the modified algorithm for wideband handsfree communication (\(f_s = 16\) kHz) using two miniature digital MEMS microphone capsules.

1 Introduction

For sound acquisition in realistic acoustic environments, microphone arrays proved to be useful. They are designed to attenuate possible noise and interference components while retaining the desired source signal by exploiting different spatial (or directional) characteristics of the different signal sources, see, e.g., [3] for an overview.

A simple, yet efficient approach is the first-order differential array [1] as depicted in Figure 1 that allows to place two symmetrical notches (directions of maximum attenuation) at angles of \(\theta\) and \(360^\circ - \theta\). In this paper, we propose three enhancements to the original algorithm and describe a practical implementation for handsfree communication. The target device is a wireless loudspeaker with two integrated miniature digital microelectromechanical system (MEMS) microphone capsules which facilitate handsfree audio communication.

2 Differential Microphone Array

First, the originally proposed algorithm [1] shall be reviewed. Two closely spaced omnidirectional microphones M1 and M2 are used to capture the acoustic environment. The corresponding digital signals \(x_1(k)\) and \(x_2(k)\) are sampled with a rate of \(f_s\). Due to the small distance \(D\) between M1 and M2, a coherent mutual subtraction[4] of the aligned signals, i.e.,

\[
\begin{align*}
x_1(k) &= x_1(k) - x_2(k) + h_T(k) \quad (1) \\
x_2(k) &= x_2(k) - x_1(k) + h_T(k) \quad (2)
\end{align*}
\]

can be achieved. Thereby, \(h_T(k)\) is a fractional delay filter with a delay time of \(T = D/c\) (c: speed of sound). This value corresponds to the sound propagation time from one microphone to the other. The signals \(x_1(k)\) and \(x_2(k)\) can be interpreted as "forward and backward facing cardiac" signals as the respective directional responses of [4] and [2] form cardiac shapes, see [4] Fig. 3]. For example \(x_1(k)\) does not contain any sound components from the rear direction (\(\theta = 180^\circ\)) while \(x_2(k)\) does not contain components from the front direction (\(\theta = 0^\circ\)). Both signals \(x_1(k)\) and \(x_2(k)\) are finally combined according to

\[
y_{EQ}(k) = h_{EQ}(k) * (x_1(k) - a \cdot x_2(k)) \quad (3)
\]

whereby the low-pass equalizer \(h_{EQ}(k)\) compensates for the high-pass effect of the coherent subtraction operations and the scalar factor \(a\) can be used to control the notch angle(s): \(a = 0\) corresponds to a (double) notch angle of \(180^\circ\) while \(a = 1\) corresponds to two symmetrical notch angles of \(90^\circ\) and \(270^\circ\). Due to the symmetry, all angle specifications are restricted to \(0^\circ...180^\circ\) in the following whereby any statement for a specific angle \(\theta\) also applies to its symmetrical counterpart \(360^\circ - \theta\). In the common operation scenario of the microphone array, the desired sound source lies in the front half plane (\(\theta = 0^\circ...90^\circ\)) while the undesired noise or interference source(s) lie(s) in the rear half plane (\(\theta = 90^\circ...180^\circ\)).

3 Input Level Alignment

A common problem with differential arrays are the tolerances of the employed microphones, leading to a "microphone mismatch" and therefore noise amplification [2]. The digital MEMS microphones in the target device exhibit relatively constant frequency responses, therefore individual microphone equalization is not necessary for the envisaged application. However, their power levels may still vary to a certain extent due to mounting and assembly tolerances. As fully matched input levels are necessary to utilize the full potential for the algorithm, an input level alignment procedure is devised here.

First, the relative gain \(g = \sqrt{\frac{\hat{\sigma}_{x_1}^2}{\hat{\sigma}_{x_2}^2}}\) is computed on the basis of the recursively estimated input channel variances \(\hat{\sigma}_{x_1}^2\) and \(\hat{\sigma}_{x_2}^2\). Then, correction gains are obtained

\[
g_1 = \begin{cases} g_c^{-1} & \text{if } g_c > 1 \\ 1 & \text{else} \end{cases} \quad \text{and} \quad g_2 = \begin{cases} g_c & \text{if } g_c < 1 \\ 1 & \text{else} \end{cases} \quad (4)
\]

which are finally recursively smoothed before they are applied to the respective input channels. With these gains, the "louder" channel’s level is reduced to the "quieter" channel’s level.
This approximation is frequency independent which is sufficient for many applications. However, \( a_{\text{lin}}(\alpha) \) deviates more and more from the optimal \( a(\omega, \alpha) \) as \( \omega \) approaches \( \omega_s \) which is illustrated in Figure 2. The effect already becomes relevant for frequencies well below \( \omega_s \) and also varies with the angle \( \alpha \). For a clear demonstration, \( \alpha = 135^\circ \) will be used in the following examples, because the deviation from the expected behavior is very prominent in this case. The directional response of a differential microphone array using the linear approximation according to \( 7 \) is shown in Figure 4a.

To correct the obvious deformation of the notch characteristic, we propose to integrate the optimal (frequency and angle dependent) steering factor \( a(\omega, \alpha) \) into the time domain realization of the differential array. In principle, this can be achieved by replacing the scalar multiplication operation (factor \( a \) in \( 3 \)) and in Figure 1 with a filtering operation whereby the angle dependent filter transfer functions are given by \( a(\omega, \alpha) \). For a more efficient implementation with low memory and computational requirements we consider the approximation error of the steering factor:

\[
\Delta a(\omega, \alpha) = a(\omega, \alpha) - a_{\text{lin}}(\alpha)
\]

Fortunately, \( \Delta a(\omega, \alpha) \) is separable with good accuracy:

\[
\Delta a(\omega, \alpha) \approx \Delta a(\alpha) \cdot \Delta a(\omega)
\]

The factors \( \Delta a(\omega) \) and \( \Delta a(\alpha) \) can be computed by marginalization of the 2-dim. function \( \Delta a(\omega, \alpha) \) and appropriate normalization. The factor \( \Delta a(\omega) \) is now regarded as the frequency response of a fixed filter. It is transformed to the time domain via periodic extension, inverse DFT, cyclic shifting (to enforce causality) and an appropriate shortening to a desired length. The resulting FIR filter coefficients \( h_{\text{DEQ}}(k) \), e.g., of order 16, are independent of the steering angle \( \alpha \). The angular dependence is then reintroduced with a polynomial approximation (e.g., order 4) of the second factor \( \Delta a(\alpha) \) after a variable transformation from \( \alpha \) to \( a_{\text{lin}}(\alpha) \), i.e., \( P_{\Delta a}(a_{\text{lin}}) \approx \Delta a(a_{\text{lin}}) \).

The resulting differential array with directional equalization is shown in Figure 4b whereby the “directionally equalized” backward cardioid signal \( x_{\text{DEQ}}(k) \) is given as

\[
x_{\text{DEQ}}(k) = \mathcal{F}^{-1} \{ \Delta a(\omega, \alpha) \} \ast x_b(k)
\]

\[
\approx \mathcal{F}^{-1} \{ a_{\text{lin}}(\alpha) \ast \Delta a(\omega) \cdot \Delta a(\alpha) \} \ast x_b(k)
\]

\[
= a_{\text{lin}}(\alpha) \cdot x_b(k) + \Delta a(\alpha) \cdot \mathcal{F}^{-1} \{ \Delta a(\omega) \} \ast x_b(k)
\]

\[
= a_{\text{lin}}(\alpha) \cdot x_b(k) + P_{\Delta a}(a_{\text{lin}}) \cdot h_{\text{DEQ}}(k) \ast x_b(k)
\]

where \( \mathcal{F}^{-1}(\cdot) \) denotes the inverse Fourier transform. The effect of directional equalizing can be observed in Figure 4b which displays an almost frequency-invariant behavior over the entire wideband frequency range.

4 Directional Equalization

In the following, it will become clear that the directional response of the original algorithm \( 1 \) is distorted, i.e., the actual notch angle clearly deviates from the desired angle, in particular for high frequencies. To correct this behavior, we propose an efficient “directional equalization” approach.

It can be shown (proof omitted) that the angle-dependent transfer function of the differential array (without the output equalizer) is

\[
H(\omega, \phi) = 2j \cdot e^{-j \frac{\omega T}{2} \sin(\phi)} \left[ \sin \left( \frac{\omega D}{2c} \left( 1 + \cos \phi \right) \right) - a \cdot \sin \left( \frac{\omega D}{2c} \left( 1 - \cos \phi \right) \right) \right]
\]

(5)

with the (angular) frequency \( \omega = \Omega \cdot f_s = 2\pi f \) and the respective angle of observation \( \phi \). Now this transfer function should become zero for a specific angle, i.e., the so called steering angle \( \alpha \). Hence, by requiring \( H(\omega, \phi = \alpha) = 0 \), the optimal steering factor \( a \) can be deduced from \( 5 \):

\[
a(\omega, \alpha) = \frac{\sin \left( \frac{\omega D}{2c} \right)}{\sin \left( \frac{\omega D}{2c} \right)} \left( 1 + \cos \alpha \right)
\]

(6)

which obviously depends on \( \alpha \) and on the frequency \( \omega \). The steering angle \( \alpha \) should, ideally, be adapted to match the interference incidence angle \( \theta \). This is discussed in Section 5.

We note that if we approach the spatial alias frequency, i.e., \( \omega \rightarrow \omega_s = \frac{f_s}{2} \), we have \( a(\omega, \alpha) \rightarrow 1 \) regardless of the angle \( \alpha \), i.e., the steering angle is not controllable at \( \omega_s \) which renders it a natural cutoff frequency for the entire array. The frequency dependency of \( a \) is, however, a contradiction to the scalar multiplication operation in \( 3 \). Therefore, usually, small values of \( \omega D \) are assumed and a linear approximation of \( 6 \) is used \( 4 \):

\[
a_{\text{lin}}(\alpha) = \frac{1 + \cos \alpha}{1 - \cos \alpha}.
\]

(7)

Figure 3: Directional frequency responses (rear half plane) of a wideband \( (f_s = 16 \text{ kHz}) \) array with \( D = 1.8 \text{ cm} \) steered towards \( \alpha = 135^\circ \) using the linear approximation of \( 7 \) and the directional equalizer of Figure 4.

Figure 2: Opt. steering factor vs. linear approximation.

(a) linear approximation \( a_{\text{lin}}(135^\circ) \)

(b) with directional equalization

Figure 4: Differential array with directional equalizer.
The practical operation of the modified version of the microphone array (Figure 5) does not significantly differ from the conventional version (Figure 1): The desired notch angle \( \alpha \) is still easily controlled by adapting the scalar factor \( a_{\text{lin}} \). As an additional step, the polynomial \( P_{\Delta}(a_{\text{lin}}) \) must be evaluated.

### 5 Fast Notch Adaptation

The goal of a notch adaptation algorithm is to automatically align the notch angle \( \alpha \) of the differential array with the incidence angle \( \theta \) of the (main) interferer. In the common application scenario, the undesired noise or interference sources are assumed to lie in the rear half plane, i.e., \( \theta = 90^\circ \ldots 180^\circ \). The standard approach to adapt the factor \( a \) (or \( a_{\text{lin}} \), if directional equalization is used) and therefore the notch angle \( \alpha \) is the \textit{normalized} least mean square (NLMS) algorithm. The goal here is to minimize the power of the output signal \( y(k) \), i.e.

\[
E \{ [x_k(k) - a \cdot x_0(k)]^2 \} \rightarrow \text{min},
\]

where, usually, \( 0 < a < 1 \), i.e., \( 180^\circ \geq \alpha \geq 90^\circ \) is enforced. The stepwise update rule of this algorithm is (e.g., [4])

\[
a(k) = a(k - 1) + \frac{\mu}{\sigma^2_0} \cdot x_0(k) \cdot y(k)
\]

with the short term power estimate \( \sigma^2_0 \) of the backward cardioid signal and the constant stepsize \( \mu \).

For our proposed notch adaptation algorithm, which is inspired by the “optimum stepsize NLMS” of [6], we separate the coherent and the incoherent (ambient) components of the acoustic environment. The array output (before the lowpass equalizer) therefore becomes

\[
y(k) = x_c(k) - a \cdot x_0(k) - x_a(k) - a \cdot x_{ha}(k).
\]

This equation represents the error signal of a single-tap adaptive filter with a noisy input. In the context of the differential array, the noise signal \( n(k) \) is due to the incoherent (ambient) noise that cannot be suppressed. The coherent contribution to \( y(k) \) should ideally be zero. However, due to the instantaneous misadaptation of the factor \( a \), an error signal \( e(k) \) appears at the output. According to adaptive filter theory, the optimal (adaptive) stepsize parameter \( \mu \) (13, 56]) is

\[
\mu_{\text{opt}} = \frac{E \{ e^2(k) \}}{E \{ n^2(k) \} + E \{ e^2(k) \}}.
\]

From the signals available, the best approximation of \( E \{ n^2(k) \} \) is the level \( \sigma^2_0 \) of the array’s output \( y(k) \) while for \( E \{ e^2(k) \} \), the assumption of a fixed attenuation factor for the backward cardioid signal is made, i.e. \( E \{ e^2(k) \} \approx \kappa \cdot \sigma^2_0 \). We set \( \kappa = 0.01 \) (assumed attenuation of 20 dB) and the adaptive stepsize parameter is hence

\[
\mu_{\text{opt}} \approx \frac{\kappa \sigma^2_0}{\sigma^2_0 + \kappa \sigma^2_0}
\]

with the recursively estimated short term powers \( \sigma^2_0 \) and \( \sigma^2_0 \) which leads to the new NLMS update rule

\[
a(k) = a(k - 1) + \frac{\sigma^2_0}{\sigma^2_0 + \kappa \sigma^2_0} \cdot x_0(k) \cdot y(k).
\]

The adaptation can be deliberately slowed down by the factor \( 0 < \xi \leq 1 \) to avoid artifacts that stem from the single-tap prediction which does not apply any smoothing. The combination of the proposed NLMS notch adaptation with the directional equalizer of Section 4 is straight forward. The equalizer can indirectly influence and enhance the notch adaptation via the array output signal \( y(k) \).

Although the frequency dependency of \( a(k) \) is disregarded in the NLMS update, this slight error is immediately compensated for at the next sample instant when the DEQ is in turn adapted to the new steering factor \( a(k + 1) \) (or \( a_{\text{lin}}(k + 1) \) in this case).
Evaluation & Discussion

The performance of the proposed fast notch adaptation algorithm is contrasted with the conventional NLMS using a fixed stepsize in Figure 5. The graph illustrates the adaptation process for a synthetic sound field with a single sound source that arrives from changing angles $\theta$. In the example, only the angle of $\theta = 0^\circ$ is associated with a desired source. In this case, the adaptation should not drift towards the 90$^\circ$ boundary but rather maintain the previously identified steering factor $\alpha$. The underlying assumption is that an interferer does not move while being inactive. The fast version of the constant stepsize NLMS (13) (gray curve) for example drifts towards 90$^\circ$ easily in case of activity of the desired sound source, but even the slower version (blue curve) is not able to maintain a once identified steering factor in all situations. The proposed adaptive NLMS (16) (green curve) mostly solves this problem while converging almost instantaneously towards new interfering sound sources.

However, all three approaches exhibit an unstable behavior for $\theta = 135^\circ$. These surprisingly strong fluctuations, which are clearly audible and also disturbing, are explained by intermittent high frequency sounds in the interferer source (such as fricatives in speech). If a high frequency sound event occurs from $\theta = 135^\circ$, the adaptation algorithm (which minimizes the output power) shifts the distorted notch characteristic of Figure 3-a to the left, i.e., toward $\alpha = 90^\circ$ or $a_{\text{lin}} = 1$. Low frequency sounds yield the expected result ($\alpha = 135^\circ$ or $a_{\text{lin}} \approx 0.17$) instead. If the DEQ algorithm from Section 4 is activated (red curve in Figure 5), the unstable adaptation behavior vanishes almost completely.

6 Implemented System

Using fixed point arithmetic, the described differential microphone array (including the proposed enhancements) has been implemented on a signal processor of a wireless loudspeaker (Binauric Boom Boom) which is, at the same time, a handsfree communication device. Two miniature digital MEMS microphone capsules are placed on the top of the device with $D = 1.8$ cm. The microphones offer SNRs of more than 60 dB which open up the possibility of a differential microphone array with a sufficiently low noise level. An example application scenario is a handsfree call in an office where another colleague is working on the opposite side of the desk. The colleague’s noise (typing, voice, etc.) can then be canceled out when placing a call with Boom Boom.

The signal processing software has been developed with the help of the RTProc rapid real-time prototyping framework [8] — the developer interface for algorithm parametrization is shown in Figure 6. Beginning with a Matlab prototype (based on frame-wise processing), several other versions have been subsequently developed: A parametrizable C version, a C version with generated parameter tables, a C version based on fixed point arithmetic with an emulated instruction set and generated parameter tables, and finally optimized assembler code for the signal processor with generated parameter tables. All versions can be verified against each other and there is the possibility to step back to Matlab and add or modify features. The measured complexity of the assembler code is approximately 7 MIPS for the wideband sampling rate ($f_s = 16$ kHz).

7 Conclusions

A number of algorithmic enhancements for the standard differential microphone of [1] have been proposed in this paper:

- The described input level alignment procedure enables the array to work reliably despite certain assembly and sensor tolerances.

- With the proposed directional equalizer, a frequency invariant notch characteristic can be obtained even for larger microphone distances. A larger distance also helps to confine the noise gain of the array. Therefore, the array becomes practically usable even for higher sampling rates (e.g., 16 kHz).

- With the fast notch adaptation algorithm, the noise (or interferer) suppression works more reliably in a broader range of acoustic scenarios. Also, the desired source does not compromise the notch adaptation anymore. Moreover, the combination with the directional equalizer leads to a more stable direction of arrival tracking.

The implementation of the proposed techniques in a new commercial handsfree communication device was described.

Finally, the subjective listening impression confirms that, with the new algorithms, interfering sound sources are suppressed much more reliably than with the conventional microphone array. The now frequency-invariant notch characteristic is not the only reason for this. Rather, only the combined application of both DEQ and fast notch adaptation facilitates the clearly improved direction of arrival tracking and interference suppression.

References


