Effiziente Signalverarbeitung für PDM-Mikrofonsignale in vielkanaligen Mikrofonarrays

Stephan Herzog[1]

[1] AG Digitale Signalverarbeitung, TU Kaiserslautern, 67663 Kaiserslautern, Email:herzog@eit.uni-kl.de

Einleitung

In den letzten Jahren haben MEMS (Micro Electro Mechanical Systems) Mikrofone sich qualitativ stark weiterentwickelt und auf Grund vieler vorteilhafter Eigenschaften wie hoher Konstanz der Parameter, niedrigen Kosten und digitalen Schnittstellen weite Verbreitung gefunden. Durch ihre guten Eigenschaften werden sie auch für die Anwendung in der Audiomesstechnik interessant, vor allem für Mikrofonarrays mit hohen Kanalzahlen, auch wenn ihre akustischen Eigenschaften noch an die hochwertiger Kondensator-Messmikrofone nicht heranreichen. In der Anwendung in Mikrofonarrays ist neben den geringen Kosten besonders die digitale Schnittstelle als vorteilhaft anzusehen, da aufwändige Spannungsversorgung, Analog-Digital-Wandler und die damit verbundene Technik komplett entfallen können.

Die digitalen Ausgangssignale von MEMS-Mikrofonen liegen meist als PDM (Puls-Dichte-Modulation) Signal vor. In dieser Form ist das Signal mit einer 1-bit Quantisierung und einer sehr hohen Abtastfrequenz (typ. 3,072 MHz) repräsentiert und eine Weiterverarbeitung ist mit herkömmlichen Algorithmen nicht direkt möglich, da diese mit PCM-Signalen arbeiten. Um diese digitalen Signale weiterverarbeiten zu können, werden sie durch Dezimation in PCM-Daten mit typischen Audio-Abtastfrequenzen von z.B. 48 kHz und Wortbreiten von z.B. 16 oder 24 bit konvertiert.

Neben der Möglichkeit, einen Echtzeit-Audiodatenstrom mit voller Bandbreite aus dem PDM-Signal zu berechnen, werden in diesem Artikel weitere Ansätze aufgezeigt. Es sind Anwendungen denkbar, in denen nur die Pegel der einzelnen Mikrofon-Kanäle benötigt werden, z.B. um die Schalldruckverteilung in einem Gebiet zu bestimmen. Durch den Abstand der Mikrofone in einem Mikrofonarray ist Anwendungen in vielen Fällen akustisch eine obere Grenzfrequenz gesetzt. In solchen Fällen werden die Mikrofonsignale nicht in ihrer vollen Bandbreite benötigt und sie kann durch geeignete Vorverarbeitung verringert werden, um die zu übertragende Datenmenge zu reduzieren und die Auswertung zu vereinfachen.

Volldigitale MEMS-Mikrofonarrays für akustische Holographie

Die kleine Bauform und die eingangs erwähnten positiven Eigenschaften von MEMS-Mikrofonen ermöglichen den Aufbau von volldigitalen Mikrofonarrays, die z.B. direkt über eine USB-Schnittstelle mit einem Rechner verbunden werden können. Als Schnittstellenbaustein bieten sich dazu verschiedene Möglichkeiten an, z.B. spezifische Audio-USB-Chips oder Interfaces bzw. Controller-Boards. Die mögliche Anzahl der Kanäle ist dabei meist auf 24 oder 32 begrenzt, was die Möglichkeiten für Mikrofonarrays einschränkt. Eine USB 2.0 - Schnittstelle selbst mit 480 Mbit/s ist dabei nicht der Flaschenhals, die Datenrate bei 96 Kanälen, 48 kHz Abtastfrequenz und 24 bit Wortbreite liegt bei ca. 110 Mbit/s. Über ein Interface mit 192 kHz Abtastfrequenz, 16 Kanälen und 24 bit Wortbreite wäre eine Datenrate von 110 Mbit übertragbar, allerdings müssen die Daten im Falle einer Anzahl von Mikrofonkanälen von $N > 16$ auf jeden Fall umsortiert werden, was ein lokale Signalverarbeitung auf dem Mikrofonarray, z.B. in einem FPGA erforderlich macht. Die Vorverarbeitung der Signale zur Sortierung ist auch nötig, wenn die PDM-Signale von speziellen PDM-ICs in PCM gewandelt werden. So ist es naheliegend, die komplette Umwandlung von PDM zu PCM in einem FPGA zu integrieren und es stellt sich die Frage, welche Signalverarbeitung bei einer möglichst hohen Anzahl von Kanälen der beabsichtigten Anwendung gerecht wird und gleichzeitig möglichst aufwandsgünstig ist.

Für die vorgesehene Anwendung eines solchen Arrays, die Berechnung der Abstrahlcharakteristik durch akustische Holographie für Zeilen- oder Flächenlautsprecher werden, wie in [5] und [6] beschrieben, nur die (relativen) Amplituden und Phasen der von den Mikrofonen aufgenommenen Multitonsignale benötigt, wodurch eine erhebliche Reduktion der Datenmenge bzw. -rate möglich ist. Die Reduktion der Datenmenge soll dabei mit möglichst wenig Ressourcen und vergleichsweise einfachen Algorithmen auf dem Mikrofonarray erfolgen, die komplexere Auswertung der Signale wie die Berechnung von Amplituden und Phasen (im Frequenzbereich) und die weitere Verarbeitung auf einem angeschlossenen PC.

Verarbeitung von PDM-Mikrofonsignalen

Wir nehmen für Messungen im Audio-Frequenzbereich eine typische Abtastfrequenz von $f_s = 48$ kHz an. Dies ist die nominelle Abtastfrequenz, auf die das PDM-Signal mit einer Abtastrate von $f_{sH} = 3{,}072$ MHz heruntergetastet und in ein PCM Signal gewandelt wird. Die Abtastratenreduktion von 3,072 MHz auf 48 kHz bedeutet eine Dezimation um den Faktor $D = 64$. Dies erlaubt Messungen in einem nutzbaren Frequenzbereich bis ca. $f_s/2 = 24$ kHz.

Ausgangspunkt der vorgestellten SV-Algorithmen ist das 1 bit PDM Ausgangssignal eines MEMS-Mikrofons. Im Mikrofon wird dieses Signal aus dem analogen Schalldrucksignal mittels Sigma-Delta-Modulation erzeugt [1].

Das PDM-Signal kann wie ein mit hoher Abtastfrequenz abgetastetes und mit einem Bit quantisiertes Analogsignal betrachtet und verarbeitet werden, auch wenn die Daten im PDM-Signal nicht aus einer direkten Abtastung entstanden sind. Die Anforderungen an die SV sind dabei im Allgemeinen geringer Berechnungsaufwand, konstanter Betragsfrequenzgang im Nutzband und eine gute Unterdrückung der Spiegelfrequenzen, die bei einer Unterabtastung entstehen.

In den meisten Anwendungen werden für die Abtastratenreduktion CIC (Cascaded Integrator Comb) Filter in Verbindung mit Filtern zur anschließenden Frequenzgangkorrektur verwendet [7], für die es fertige Bibliotheken für viele Systeme gibt. Sie bieten einen guten Kompromiss der o.g. Eigenschaften. Andere Möglichkeiten zur Abtastratenreduktion sind Polyphasenfilter, Halbbandfilter, Frequenzabtast-Strukturen oder IIR-Dezimatoren [9]. Mit diesen Ansätze wird üblicherweise ein PCM-Signals mit der vollen Bandbreite $f_s/2$ bei möglichst konstantem Frequenzgang und guter Unterdrückung von Spiegelfrequenzen berechnet, für unsere Anwendung sind jedoch nicht alle diese Punkte relevant.

Berechnung von Amplitude und Phase

Aus den auf die nominelle Abtastfrequenz f_s gebrachten Signalen können zur Berechnung der Abstrahlcharakteristik in einem weiteren Verarbeitungsschritt die interessierenden Größen, bei der akustischen Holographie sind dies die Amplituden und Phasen der einzelnen Kanäle, berechnet werden. Dies kann mit Hilfe von Zeitbereichsalgorithmen oder im Frequenzbereich mittels schneller Fouriertransformation, des Goertzel-Algorithmus [5] oder einer 1-Punkt DFT [4] durchgeführt werden.

Der Nachteil dieser Vorgehensweise ist, wie schon erwähnt, dass eine erheblich größere Datenmenge übertragen und ausgewertet werden muss, als es für die Anwendung nötig ist.

f_{sH}	f_s	f_{sL}
6,7 s	0,17 s	0,03 s

Tabelle 1: Für 96 FFTs in Matlab benötigte Zeit bei verschiedenen Abtastfrequenzen. Frequenzauflösung jeweils 1 Hz.

Tabelle 1 zeigt die Berechnungszeiten für 96 FFTs (für 96 Kanäle) mit 1 Hz Frequenzauflösung für verschiedene Abtastfrequenzen. Niedrige Abtastfrequenzen resultieren wie zu erwarten in einer erheblich kürzeren Berechnungszeit.

Anwendungsbezogene Ansätze zur Abtastratenreduktion

Zur Reduktion der Datenmenge stellen wir mehrere Ansätze, unter anderem Unterabtastung auf eine niedrigere Abtastfrequenzen $f_{sL} < f_s$ vor. Dabei entstehendes Aliasing durch Verletzung des Abtasttheorems kann dabei in Kauf genommen und kontrolliert werden. Unterabtastung wird zum Beispiel bei der Abtastung Bandpasssignalen und Superresolution eingesetzt [2]. Betrachtet man einzelne Töne, so werden durch das Aliasing die im Signal enthaltenen Informationen hinsichtlich Amplitude und Phase nicht beeinträchtigt und mit der Kenntnis der Alias-Frequenzen der Testtöne können die relativen Amplituden für alle Kanäle bestimmt werden. Wollen wir beispielsweise über ein vorhandenes, handelsübliches Interface mit 16 Audiokanälen bei 48 kHz 96 Kanäle übertragen, muss die PDM-Abtastrate von 3,072 MHz um den Faktor $R \cdot D = 384$ auf $f_{sL} = 8$ kHz reduziert werden. Wir betrachten zwei Varianten für die Abtastratenreduktion. Bei der ersten wird ein vorhandenes System mit $f_s = 48$ kHz genutzt und eine weitere Unterabtastung angewendet. Bei der zweiten Variante wird das PDM-Signal direkt auf eine niedrige Abtastrate f_{sL} heruntergetastet. Danach können durch Multiplexing mehr Kanäle als die im Interface zum PC vorhandenen übertragen werden.

Bei der Signalverarbeitung mit Aliasing sind Schwerpunkte das Bestimmen der Alias-Frequenz und die Wahl geeigneter Frequenzen der Testtöne unter der Beachtung der Beschränkungen durch das Aliasing. Zur Berechnung von Amplituden und Phasen können die gleichen Algorithmen wie bei herkömmlicher Verarbeitung angewendet werden.

Wenn wir Aliasing kontrolliert zulassen, würde eine Tiefpassfilterung zur Bandbegrenzung nicht gewünschte Spektralanteile über $f_{sL}/2$ unterdrücken. Eine reine Unterabtastung ohne jegliche Filterung wäre ein sehr rauschempfindlicher Prozess. Ein ressourcenschonender Ansatz zur Rauschreduktion bei gleichzeitig nicht zu starker Unterdrückung hoher Frequenzen ist die Anwendung eines Moving-Average- (MA) Filters. Das MA-Filter berechnet den Mittelwert über M Abtastwerte zu

$$x_m(i) = \frac{1}{M} \sum_{n=i}^{i+M-1} x_P(n) \qquad i = 0, M, 2M, ... \qquad (1)$$

Mit der Mittelwertbildung kann für $M = D$ direkt eine Unterabtastung verbunden werden, in dem nur jeder Mte Abtastwert weiterverarbeitet wird. Man kann in diesem Fall den Vorgang als Polyphasen-FIR-Filter der Länge M betrachten, in dem alle Koeffizienten den Wert $1/M$ haben. Ist M eine Zweierpotenz, so kann durch links- oder rechts-Schieben die Skalierung mit $1/M$ durchgeführt werden, was die Implementierung gegenüber einer Multiplikation weiter vereinfachen kann.

Ein MA-Filter der Länge $M = R \cdot D = 384$ zur Abtastratenreduktion auf $f_{sL} = 8$ kHz hat Nullstellen bei den Frequenzen 8, 16 und 24 kHz. Signale mit diesen Frequenzen werden vollständig unterdrückt und tauchen auch nicht als Alias-Frequenz im Ausgangssignal nach dem Unterabtasten auf. Zusätzlich ist in Bereichen nahe der Nullstellen die Dämpfung des MA-Filters sehr groß, so dass ein solcher Ansatz nicht prktikabel ist.

Die Tiefpass-Filterung vor dem Unterabtastens hat keinen konstanten Betragsfrequenzgang. Für die beabsichtigte Anwendung ist dies nicht relevant, da nur relative Pegel und Phasen der Kanäle benötigt werden. Der

Höhenabfall mit $R = 64$ bei 20 kHz beträgt nur ca. 3 dB, mit $R \cdot D = 128$ ist die Dämpfung 15 dB bei 20 kHz.

Durch das Aliasing werden Testtöne mit $f > f_{sL}/2$ nicht mehr auf ihrer ursprünglichen Frequenz abgebildet und die Alias-Frequenz f_A bzw. der Index des Frequenzbereichswerts im DFT-Spektrum muss berechnet werden.

Die Alias-Frequenz f_A bestimmt sich im Falle tatsächlich vorhandenen Aliasings im Allgemeinen [8] zu

$$f_A = \left| f - f_{sL} \cdot \left\lfloor \frac{f}{f_{SL}} \right\rfloor \right|. \tag{2}$$

Da die verwendeten Testton-Frequenz f immer bekannt ist, kann die Frequenz f_A in allen Fällen berechnet werden. Es kann aber sein, dass das Signal nach dem Aliasing trotzdem nicht mehr die gewünschte Information enthält. Bei Koinzidenz der zwei Spektrallinien eines Einzeltons wird die Phase zu null und somit ist auch keine Auswertung mehr möglich. Dieser Fall tritt bei einer Frequenz von $f = N \cdot f_{sL}/2$ auf. Bei Multiton-Testsignalen ist es wichtig zu beachten, dass die Frequenzen f_A zweier oder mehrerer Töne durch das Aliasing nicht koinzidieren, da eine Trennung in die Ursprungskomponenten dann nicht mehr möglich ist.

Betrachten wir Beispielsweise einen Multiton mit den 28 Norm-Terzmittenfrequenzen. Bis zu einer Frequenz von $f = 4000$ Hz tritt bei $f_{sL} = 8$ kHz kein Aliasing auf. Die Komponente mit 5 kHz fällt durch Aliasing auf 3 kHz, 6,3 kHz auf 1,7 kHz, 8 kHz auf 0 Hz und 12,5 kHz auf 3500 Hz. Da bei diesen Aliasfrequenzen kein anregender Ton liegt, stellt das Aliasing für diese Töne bei Kenntnis der Alias-Frequenzen f_A kein Problem dar. Der 10 kHz Ton fällt auf 2 kHz, 16 kHz auf 0 Hz und 20 kHz auf 4 kHz, also Frequenzen, bei denen jeweils anregende Töne vorhanden sind, bzw. Aliasing-Produkte anderer Töne auftreten. Somit kann die ursprüngliche Information an diesen Frequenzen nicht mehr ausgewertet werden.

Eine Lösung für dieses Problem kann darin bestehen, die Frequenzen der anregenden Töne leicht zu modifizieren. Z.B. entstehen bei den Frequenzen 10001 Hz, 16002 Hz und 19999 Hz Aliasingprodukte bei den Frequenzen 2001 Hz, 2 Hz und 3999 Hz. Bei kohärenter Abtastung können diese Frequenzen problemlos von den 1 Hz benachbarten Alias-Produkten unterschieden werden. In vielen Fällen sind solche hohen Frequenzen zur Messung gar nicht nötig, da die Abstände der Messpunkte bzw. Mikrofone zu groß sind, um eine ausreichende räumliche Abtastung zu gewährleisten.

Ein Echtzeitbetrieb mit beliebigen (Musik-)signalen funktioniert bei einer Abtastfrequenz von $f_{sL} = 8$ kHz natürlich nur bis 4 kHz. Für diese Anwendung ist auch eine angemessene Unterdrückung der Signalanteile über $f_{sL}/2$ nötig.

Kohärentes Abtasten unter Aliasing

Um kohärente Abtastung für Frequenzen mit $f > f_{sL}/2$ zu erreichen, müssen die Alias-Frequenzen f_a der Multiton-Komponenten und die Abtastfrequenz f_{sL} beachtet werden.

$$N^* = \left\lceil \frac{f_{sL}}{ggT(f_{sL}, f_{A1}, ..., f_{Ai})} \right\rceil \tag{3}$$

wobei für die Alias-Frequenzen nach (2) berechnet werden.

Vergleich verschiedener Ansätze in Matlab

Zum Vergleich wurden zwei Einzeltöne mit der gleichen Frequenz, aber unterschiedlichen Amplituden und Phasen erzeugt und zusätzlich weißes Rauschen auf beide Signale addiert. Mit einem Sigma-Delta-Modulator 2. Ordnung wurden daraus PDM-Signale erzeugt und mit Hilfe der verschiedenen Ansätze die Differenzen in Phase und Amplitude zwischen beiden Tönen bestimmt. Die Berechnung der interessierenden Größen Amplitude und Phase wird mittels einer FFT durchgeführt.

1. Unterabtastung auf f_s mit einem MA-Filter mit $M = D = 64$

Bei diesem Ansatz wird das PDM-Signal nach der Filterung mit einem Polyphasen-MA-FIR-Filter entsprechend (1) mit auf 48 kHz heruntergetastet. Die Signalverarbeitung erfordert weniger Aufwand als bei herkömmlichen Ansätzen wie CIC-Filtern. Der Nachteil dieses Ansatzes ist die schlechtere Sperrbanddämpfung des Moving-Average-Filters im Vergleich zu anderen Ansätzen und eine dadurch bedingte höhere Rauschempfindlichkeit.

2. Unterabtastung auf f_{sL} mit MA-Filter der Länge $M = 128$

Hier ist die Länge $M = 128$ des MA-Filters nicht gleich dem Faktor der Unterabtastung $D \cdot R = 384$. Der Moving-Average-Filter mit der Länge $M = 128$ ist das längste MA-Filter, das bei den betrachteten Abtastfrequenzen keine Nullstellen im Audioband bis $f_s/2$ aufweist. Es stellt also einen Kompromiss zwischen guter Rauschunterdrückung und Amplitudenfehler / Nullstellen im Durchlassband dar.

3. Unterabtastung auf f_{sL} aus einem Signal mit f_s ohne weitere Filterung

Falls das Signal schon auf f_s unterabgetastet vorliegt, kann durch einfache Dezimation, d.h. einfacher Übernahme jedes Rten Werts eine Unterabtastung erfolgen. Dies kann der Fall sein, wenn z.B. MEMS-Mikrofone verwendet werden, die direkt ein PCM-Ausgangssignal mit f_s liefern, oder die PDM- zu PCM-Wandlung in dezidierten Abtastratenwandler-Chips erfolgt. Durch die fehlende Tiefpassfilterung wird das Rauschen innerhalb der Bandbreite $f_s/2$ kaum reduziert.

4. Direkte Unterabtastung auf f_{sL} ohne Filterung

In dieser Variante wird das PDM-Signal ohne jegliche Filterung dezimiert, wozu keinerlei Arithmetik benötigt wird. Bei geringen Rauschpegeln funktioniert auch dieser Ansatz zur Berechnung der Amplituden- und Phasen-

unterschiede. Unter praktischen Bedingungen ist er auf Grund des sehr schlechten SNR kaum verwendbar.

Ergebnisse und Diskussion

Die beiden Einzeltöne haben ein Amplitudenverhältnis von $\Delta a = a_1/a_2 = 2$ und eine Phasendifferenz von $\Delta\varphi = (\varphi_1 - \varphi_2)/\pi = -0.2$. Der Signal-Störabstand (SNR) wurde für die jeweilige Bandbreite der unterabgetasteten Signale berechnet. Als Frequenzen werden 28 Terz-Mittenfrequenzen verwendet. Die vier vorgestellten Ansätze wurden dann hinsichtlich der erzielbaren Genauigkeit bei der Berechnung von Amplituden- und Phasendifferenz untersucht. Ausgewertet werden der Mittelwert und die Standardabweichung σ aller berechneten Amplituden- und Phasenunterschiede.

/	1	2	3	4
$\sigma_{\Delta a}$	0,004	0,008	0,012	0,35
$\overline{\Delta a}$	2	2	2,002	1,99
$\sigma_{\Delta\varphi}$	8,6e-4	0,001	0,002	0,036
$\overline{\Delta\varphi}$	-0,2	-0,2	-0,2	-0,21
SNR	12,5	15,09	13,97	-16,87

Tabelle 2: Ergebnisse mit $a_1 = 0.2$, $a_2 = 0.1$, $a_n = 0,8$. Mittel über alle Terzfrequenzen.

In allen vier Ansätzen funktioniert die Bestimmung der Amplituden- und Phasenunterschiede trotz teilweise sehr schlechter SNR-Werte, aufgrund der selektiven Auswertung im Frequenzbereich.

Ansatz 1 liefert bei allen drei Simulationen gute Ergebnisse hinsichtlich der Amplituden und Phasenunterschiede. Noch etwas besser schneidet Ansatz 3 auf Grund der geringeren Bandbreite beim SNR ab, allerdings werden Δa und $\Delta\varphi$ durch das zurückgefaltete Rauschen stärker beeinträchtigt. Durch die Tiefpassfilterung mit kleiner Grenzfrequenz schneidet Ansatz 2 beim SNR am besten ab, dadurch, dass $M! = D \cdot R$ ist (?), sind die Ergebnisse für Amplituden- und Phasenunterschiede etwas schlechter als bei 1. Ansatz 4, ist trotz des sehr schlechten SNR auf Grund fehlender Tiefpassfilterung bei geringen Rauschpegeln anwendbar.

/	1	2	3	4
$\sigma_{\Delta a}$	0,02	0,03	0,04	1,13
$\overline{\Delta a}$	2	2	2	1,99
$\sigma_{\Delta\varphi}$	0,003	0,005	0,01	0,37
$\overline{\Delta\varphi}$	-0,2	-0,2	-0,2	-0,22
SNR	0,43	3	1,94	-28,9

Tabelle 3: Ergebnisse mit $a_1 = 0.05$, $a_2 = 0.025$, $a_n = 0,8$. Mittel über alle Terzfrequenzen.

Zusammenfassung

In diesem Beitrag wurden verschiedene Ansätze zur effizienten Signalverarbeitung mit dem Ziel der anwendungsspezifischen Reduktion der Datenmenge von digitalen Mikrofonarrays gezeigt. Die verschiedenen Ansätze

/	1	2	3	4
$\sigma_{\Delta a}$	0,04	0,136	0,085	0,95
$\overline{\Delta a}$	2	2,04	1,99	1,5
$\sigma_{\Delta\varphi}$	0,005	0,013	0,01	0,47
$\overline{\Delta\varphi}$	-0,2	-0,2	-0,2	-0,23
SNR	-5,6	-3	-4,1	-33,9

Tabelle 4: Ergebnisse mit $a_1 = 0.025$, $a_2 = 0.0125$, $a_n = 0,8$. Mittel über alle Terzfrequenzen.

zeigen, dass eine vereinfachte Signalverarbeitung möglich ist und die Datenmenge über die Verwendung von niedrigeren Abtastfrequenzen deutlich reduziert werden kann. Dies eröffnet Möglichkeiten zur Erhöhung der Anzahl der Kanäle eines Mikrofonarrays und zu einer aufandsgünstigeren eingebetteten Signalverarbeitung.

Literatur

[1] Pervez M. Aziz, Henrik V. Sorensen and Jan Van der Spiegel "An Overview of Sigm a-Delta Converters: How a 1-bit ADC achieves more than 16-bit resolution," in *IEEE Signal Processing Magazine, Volume 13, Issue 1, September 1996, page s 61-84.*, 1996.

[2] Patrick Vandewalle, Luciano Sbaiz, Joos Vandewalle and Martin Vetterli "How to take advantage of aliasing in bandlimited signals," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[3] Evert Start, "Analysis of dds-controlled loudspeaker arrays by near field acoustic holography," in *Deutsche Jahrestagung für Akustik, DEGA 2002*, 2002.

[4] Carlos Martin Orallo and Ignacio Carugati, "Single Bin Sliding Discrete Fourier Transform," in *Fourier Transforms High-tech Application and Current Trends, Intech open*, 2017.

[5] Alexander Potchinkov and Stephan Herzog "Messverfahren für Zeilenlautsprecher mit optimalen Multitonsignalen und Schmalbandanalysen," in *DAGA 2020 Hannover*, 2020.

[6] Stephan Herzog "Fourierbasierte eindimensionale akustische Holographie zur Berechnung des Schallfelds von Lautsprechern," in *DAGA 2020 Hannover*, 2020.

[7] Bruno da Silva, An Braeken, Kris Steenhaut, and Abdellah Touhafi "Design Considerations When Accelerating an FPGA-Based Digital Microphone Array for Sound-Source Localization", in *Hindawi Journal of Sensors, Article ID 6782176, Vol 2017*, 2017.

[8] Dilip Sarwate, *DSP Stackexchange. https://dsp.stackexchange.com/questions/2456/alias-frequency-formula*, 2012.

[9] Uwe Meyer-Baese, "Digital Signal Processing With Field Prgrammable Gate Arrays", Third edition, Springer Verlag., 2007.