Diplomarbeit

Entwicklung eines echtzeitfähigen Systems zur akustischen Detektion der Fahrtrichtung

Christoph Reitbauer

Technische Universität Graz - Institut für Kommunikationsnetze und Satellitenkommunikation
Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Otto Koudelka

Begutachter: Univ.-Prof. Dipl.-Ing. Dr.techn. Otto Koudelka
Betreuer: Dipl.-Ing. Dr. techn. Franz Graf

Graz, Dezember 2011

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, _____ _____

Ort, Datum Unterschrift

Kurzfassung

Zuverlässige Verkehrsüberwachung zur Erhöhung der Sicherheitsstandards auf Autobahnen und Schnellstraßen gewinnt immer mehr an Bedeutung. Dabei ist die Bestimmung der Fahrtrichtung auf Richtungsfahrbahnen von großer Bedeutung, damit Schäden an Gesundheit und Infrastruktur vermieden werden können. Hierbei spielt die automatische Verkehrsüberwachung eine wesentliche Rolle.
Ziel dieser Arbeit ist die Entwicklung eines Systems zur akustischen Fahrtrichtungsdetektion von Fahrzeugen mittels Mikrofonarray. Die Entwicklung des Systems basiert dabei auf einem von Joanneum Research entwickelten mehrspurigen Fahrtrichtungsdetektionsalgorithmus. Die Arbeit umfasst dabei eine echtzeitfähige Implementierung des Fahrtrichtungsdetektionsalgorithmus, die Entwicklung und Realisierung eines kostengünstigen Datenakquisitionssystems, sowie die Implementierung geeigneter Softwaremodule zur Anbindung des Datenakquisitionssystems an die Analyseeinheit.

Abstract

Reliable traffic surveillance is of upcoming interest to meet safety and security standards on roads and highways. The determination of the driving direction on multilane highways is of great importance to avoid severve damage to infrastructure and health. In this context automatic traffic surveillance systems play an essential role. This thesis aims at developing an acoustic driving direction detecion system using a microphone array.
The development of the system is based on a multilane driving direction detection algorithm developed by Joanneum Research. The development task comprises a real-time implementation of the driving direction detection algorithm, the development and implementation of an inexpensive dataacquisition system, as well as the implementation of software modules for connecting the dataacquisition system to the analysis unit.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mir bei der Entstehung dieser Arbeit behilflich waren.

Ein besonderer Dank gilt meinem Betreuer Franz Graf und dem Team von Joanneum Research, insbesonders Michael Stadtschnitzer und Bernhard Rettenbacher für die Unterstützung und die Zusammenarbeit während dieser Diplomarbeit. Bedanken möchte ich mich auch bei Professor Gunter Winkler vom Insitut für Elektronik der TU Graz für seine Unterstützung und Hilfestellung bei der Hardwareentwicklung.

Ich bedanke mich bei Studienkollegen und Weggefährten, allen voran Anna Fuchs und Wolfgang Jäger für die gemeinsame Zeit während des Studiums, sowie Martin Rohrmoser und Klaus Dobbler für offene Ohren und motivierenden Austausch während dieser Arbeit.

Besonders bedanken möchte ich mich bei meinen Eltern Helga und Kurt, die mein technisches Interesse geweckt und mir dieses Studium ermöglicht haben.

Graz, Dezember 2011 Christoph Reitbauer

Inhaltsverzeichnis

Abkürzungen

CPU	Central Processing Unit
FFT	Fast Fourier Transform
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
GCC	Generalised Cross Correlation
GUI	Graphical User Interface
IC	Interated Circuit
IEPE	Integrated Electronics Piezo Electric
LED	Light Emitting Diode
MADI	Multichannel Audio Digital Interface
NI	National Instruments
PHAT	Phase Transform
PLL	Phase Locked Loop
SPDIF	Sony/Philips Digital Interface
SRP	Steered Response Power
TDE	Time Delay Estimation
TDOA	Time Difference Of Arrival
TI	Texas Instruments
TTL	Transistor-Transistor-Logik
UDP	User Datagram Protocol

Kapitel 1

Einleitung

1.1 Motivation und Hintergrund

Verkehrsunfälle mit Geisterfahrern ziehen große Aufmerksamkeit auf sich, weil sie meist tragische Folgen haben. Verglichen mit anderen Verkehrsunfällen ist die Schwere der Verletzungen sehr hoch und es kommen dabei viele unschuldige Verkehrsteilnehmer ums Leben. Laut Daten des Kuratoriums für Verkehrssicherheit sind seit 1987 bereits 105 Menschen bei Geisterfahrerunfällen auf Österreichs Straßen ums Leben gekommen. Die Hauptursachen für Geisterfahrten werden dabei mit Alkoholisierung ($\sim 50\%$), Überforderung ($\sim 50\%$) und bewusstes Wenden auf der Fahrbahn ($\sim 27\%$) angegeben. Trotz zahlreicher Optimierungen bei Auf- und Abfahrten von Autobahnen und weiteren Sicherheitsmaßnahmen auf den Straßen warnte der Radiosender Ö3 im Jahr 2010 402 Mal vor Geisterfahrern. In Deutschland wird die Zahl der Geisterfahrer vom Bundesverkehrsministerium mit rund 1700 pro Jahr angegeben.

Vor diesem Hintergrund gewinnt die zuverlässige Überwachung des Verkehrs auf Autobahnen und Schnellstraßen zunehmend an Bedeutung. Besonders Geisterfahrer müssen zuverlässig erkannt werden damit Schäden an Gesundheit und Infrastruktur vermieden werden können. Die einfachste und gängigste Methode zur Überwachung von Autobahnen und Schnellstraßen erfolgt mittels Videokameras, wobei die Erkennung von Geisterfahrern vom Überwachungspersonal durchgeführt wird. Im Fall eines Geisterfahrers muss das Sicherheitspersonal die Aufmerksamkeit im richtigen Moment auf die richtige Kamera gerichtet haben damit der Geisterfahrer erkannt werden kann. Das Videomaterial kann natürlich gesichert und gegebenenfalls nachträglich analysiert werden. Der Zeitbereich in dem eine Intervention für den Verkehr sinnvoll und notwendig ist, ist aber auf wenige Minuten begrenzt. Eine Analyse von aufgezeichnetem Videomaterial ist daher wenig sinnvoll und von geringem Interesse. Automatische Verkehrsüberwachunssysteme sind daher eine geeignete Methode um diesen Nachteil auszugleichen, da die Information über die falsche Fahrtrichtung mit einer sehr geringen Latenzzeit zur Verfügung steht.

In den letzten Jahren wurden verschiedene Fahrzeugdetektionssysteme entwickelt. Systeme mit Induktionsschleifendetektoren sind ein einfacher und zuverlässiger, allerdings auch sehr kostenintensiver Lösungsansatz. Die Installation und Wartung der Induktionsschleifen sind

dabei immer mit Arbeiten am Fahrbahnbelag und somit auch Straßensperren verbunden. Das fördert die Entwicklung alternativer Lösungsansätze mit geringen Installations- und Wartungskosten.

Systeme, die auf die Analyse des Videobildes setzen erfüllen zwar den Anspruch nach geringen Installations- und Wartungskosten, können derzeit aber speziell bei Regen, Schneefall und Nebel keine zuverlässigen Analysen bieten.
Bei Systemen die auf Mikrowellentechnologie basieren, handelt es sich meist um aktive Systeme mit Sender- und Empfangseinheit. Für eine Fahrzeugdetektion sind immer beide Einheiten notwendig, womit wiederum die Fehleranfälligkeit steigt.
Auf Radartechnologie basierende Systeme erfüllen die oben genannten Anforderungen, sind jedoch in der Anschaffung sehr teuer.

Beim System, dass in dieser Arbeit (weiter)entwickelt wird, wird die Fahrtrichtung von Fahrzeugen auf mehreren Fahrspuren nur mit passiven, akustischen Sensoren (Mikrofonen) ermittelt. Das System kann auf bestehenden Portalen, Brücken oder Lichtmasten montiert werden, sodass keine weiteren Kosten für den Bau notwendiger Infrastruktur aufgewendet werden müssen.

1.2 Ausgangssituation

Diese Arbeit versteht sich als Fortsetzung eines Forschungsprojektes. Ziel dieses Forschungsprojektes war eine Machbarkeitsstudie über eine passive, akustische Fahrtrichtungserkennung mittels Mikrofonarray zu erstellen. Im Rahmen des Forschungsprojektes wurden von Martin Rohrmoser und Christoph Reitbauer verschiedene Mikrofonarraygeometrien entworfen, simuliert und anschließend in Messungen getestet und analysiert. Es wurde mit der räumlichen Position des Mikrofonarrays experimentiert und Messungen im Labor, an Autobahnen und Schnellstraßen durchgeführt um Testdaten zu sammeln. Die aufgenommenen Messdaten (Audio und Video) wurden anschließend ausgewertet und die Fahrzeuge mit Fahrtrichtung per Hand annotiert. Der aus der Projektarbeit resultierende Analysealgorithmus für Fahrzeugdetektion und Fahrtrichtungsdetektion brachte jedoch nicht die gewünschten Detektionsergebnisse.

An die Projektarbeit knüpfte die Diplomarbeit *"Driving direction detection with microphone arrays"* [1] von Peter Dollfuß an, in welcher der aus der Projektarbeit resultierende Analysealgorithmus verbessert werden sollte. Als erstes wurde hier ein Beamformer-Simulationstool in LabView implementiert und die Auswirkungen des Beamformerverhaltens bei Variation von verschiedenen Parametern untersucht. Aus dieser Untersuchung resultierte die Geometrie des 2D-Mikrofonarrays. Der Analysealgorithmus wurde weiter verbessert, sodass die Detektionsrate erhöht werden konnte, die Fehlerwahrscheinlichkeit aber noch relativ hoch war.

Zur Verbesserung der Detektionsraten wurde von DI Michael Stadtschnitzer von JR ein neuer Analysealgorithmus entworfen und in MATLAB implementiert. Mit diesem Algorithmus wurden nun die geforderten Detektionsraten erreicht. Die Ergebnisse des Forschungs-

projektes sind im Endbericht [2] dokumentiert.

Dieser Algorithmus und die aus den vorangegangenen Arbeiten gewonnenen Erkenntnisse über die zu verwendete Arraygeometrie und Montageort dienen als Grundlage für diese Arbeit.

1.3 Ziele

Ziel und Inhalt dieser Arbeit ist, ein Gesamtkonzept eines kostengünstigen und einfach zu installierenden Fahrtrichtungserkennungssystems zu erstellen und einen Prototyp zu entwickeln, der für Demonstrationen und Langzeitmessungen eingesetzt werden kann. Das Einsatzgebiet wurde mit Autobahnen und Schnellstraßen festgelegt. Das System muss also an die dort herrschenden Umwelteinflüsse wie zum Beispiel Witterung und Temperatur angepasst werden.
Das Gesamtsystem soll aus einem Mikrofonarray, der Datenakquisitions- und/oder Übertragungseinheit und einem Analysesystem bestehen mit welchem unter den genannten Umwelteinflüssen möglichst störungsfrei gearbeitet werden kann.
Die Fahrtrichtungsanalyse soll in "Echtzeit" erfolgen, d.h. die Detektionsergebnisse sollen nach einer endlichen Zeit zur Verfügung stehen und in einer Datenbank gespeichert werden.
Der Prototyp soll als Vorlage für eine spätere Produktentwicklung dienen, wobei speziell auf niedrige Gesamtkosten bei der Entwicklung geachtet werden soll. Weiters soll das System möglichst einfach und mit geringem Aufwand zu installieren und einfach in Betrieb zu nehmen sein.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel werden die für die Implementierung des Fahrtrichtungsdetektionsalgorithmus notwendigen theoretischen Grundlagen erläutert. Beginnend mit der Wellenausbreitung werden dann diskrete Sensorarrays, Beamformer und schließlich der -PHAT Algorithmus erläutert.

2.1 Wellenausbreitung

Schallwellen breiten sich in verschiedenen Medien als Longitudinalwellen aus. Unter der Annahme eines idealen Ausbreitungsmediums ohne Viskosität kann die Wellengleichung wie folgt angegeben werden [3]

$$\nabla^2 x\left(t, \mathbf{r}\right) = \frac{1}{c^2} \cdot \frac{\partial^2}{\partial t^2} x\left(t, \mathbf{r}\right) \tag{2.1}$$

$x\left(t, \mathbf{r}\right)$ beschreibt den Schalldruck als Funktion von Zeit und Ort, der Vektor $\mathbf{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ ist ein Vektor relativ zur Schallquellenposition, ∇^2 ist der Laplace-Operator. Die Ausbreitungsgeschwindigkeit c der Welle ist abhängig von Dichte, Druck und Temperatur des Ausbreitungsmediums und liegt für Luft bei ca. $340\,{}^{\mathrm{m}}/_{\mathrm{s}}$ bei einer Temperatur von $\theta = 20\,^{\circ}\mathrm{C}$.

Die allgemeine Wellengleichung 2.1 kann mittels Variablenseparation für eine eben Welle gelöst werden und man erhält [3]

$$x\left(t, \mathbf{r}\right) = A \cdot e^{j(\omega t - \mathbf{kr})} \tag{2.2}$$

A ist die Amplitude der Welle, $\omega = 2\pi f$ ist die Frequenz in ${}^{\mathrm{Radiant}}/_{\mathrm{s}}$.

$$\mathbf{k} = \frac{2\pi}{\lambda} \begin{bmatrix} \sin\Theta\cos\Phi & \sin\Theta\sin\Phi & \cos\Theta \end{bmatrix} \tag{2.3}$$

Die Wellenzahl(vektor) $\mathbf{k}$ beschreibt die Richtung und Geschwindigkeit der sich ausbreitenden Welle. Die Wellenlänge λ ist mit der Ausbreitungsgeschwindigkeit über die Beziehung $\lambda = \frac{c}{f}$ verknüpft. Θ gibt den Winkel in der Vertikalebene (Elevation) und Φ den Winkel in der Horizontalebene (Azimuth) an.

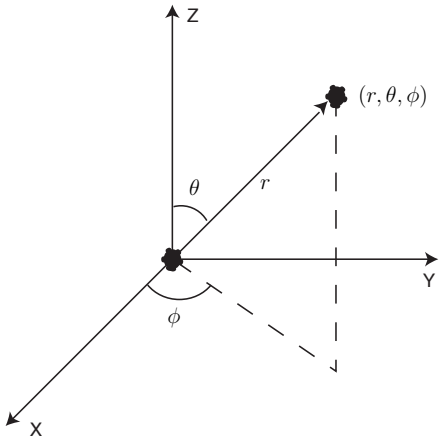


Abbildung 2.1: Parameter Elevation Θ, Azimuth Φ und Radius r [4]

2.2 Kontinuierliche Apertur

Der Begriff Apertur bezeichnet einen räumlich begrenzten Bereich, in den Schallwellen eintreten (passiv) oder austreten (aktiv) können. In der Akustik ist eine Apertur ein elektroakustischer Wandler der akustische Signale in elektrische Signale umwandelt (Mikrofon) und umgekehrt (Lautsprecher) [4].

Der Raum rund um eine Apertur lässt sich in unterschiedliche Bereiche einteilen. In dieser Arbeit sind vor allem das Nah- und Fernfeld von Bedeutung. Befindet sich beispielsweise der Ursprung eines sphärischen Wellenfeldes im Fernfeld der Apertur, so treffen die Wellen nahezu parallel auf die Apertur auf und man nimmt ein ebenes Wellenfeld an. Befindet sich der Ursprung des Wellenfeldes im Nahfeld, so bleibt der sphärische Charakter der Welle beim Auftreffen auf die Apertur erhalten und muss berücksichtigt werden.
Da der Übergang zwischen diesen beiden Bereichen fließend verläuft, existiert für die Bestimmung keine eindeutige Lösung. Als grobe Näherung soll für die weiteren Betrachtungen für den Beginn des Fernfeldes folgende Bedingung gelten:

$$r > \frac{2L^2}{\lambda}. \tag{2.4}$$

r bezeichnet die Entfernung zum betrachteten Raumpunkt und L die maximale Ausdehnung der Apertur.

Die folgenden Betrachtungen haben ausschließlich Gültigkeit im **Fernfeld** der Apertur. Die entsprechenden mathematischen Abhandlungen für die Gültigkeit im Nahfeld sind in [3] und anderen Publikationen detailliert ausgeführt.

2.2.1 Aperturfunktion

Betrachtet man eine allgemeine kontinuierliche Apertur mit Volumen V, auf die eine ebene Welle $x_M(t, \mathbf{r}_P)$ zum Zeitpunkt t auf der räumlichen Position $\mathbf{r_P}$ auftrifft, so nimmt die Apertur nur einen gewissen Teil der einfallenden Welle auf. Dieser Vorgang wird als

räumliche Filterung bezeichnet. Betrachtet man das infintesimale Volumen dV am Ort $\mathbf{r}_P$ als lineares Filter mit der Impulsantwort $a_P(t, \mathbf{r}_P)$, so ergibt sich das empfangene Signal x_R mittels Faltung im Zeitbereich zu [4]

$$x_R(t, \mathbf{r}_P) = \int_{-\infty}^{\infty} x_M(\tau, \mathbf{r}_P)\, a_P(t - \tau, \mathbf{r}_P)\, d\tau \tag{2.5}$$

oder mittels Fourier Transformation im Frequenzbereich zu

$$X_R(f, \mathbf{r}_P) = X_M(f, \mathbf{r}_P)\, A_P(f, \mathbf{r}_P) \tag{2.6}$$

Die verwendeten Indizes dienen nur zur besseren Verständlichkeit und weisen auf Folgendes hin:

$_M$	...	Ausbreitungsmedium
$_P$	...	passive Apertur
$_R$	...	empfangenes Signal

Der Term $A_P(f, \mathbf{r}_P)$ der kontinuierlichen Apertur wird als Aperturfunktion bzw. als Empfindlichkeitsfunktion bezeichnet (engl.: aperture function, sensitivity function) und definiert die Systemantwort als Funktion der räumlichen Position auf der Apertur [4, 3].

2.2.2 Richtungsfunktion

Wird im Folgenden Text von der Richtungsfunktion gesprochen, so ist mit der Annahme aus 2.2 damit immer die Fernfeld-Richtungsfunktion gemeint.

Der von der Apertur aufgenommene Teil der einfallenden ebenen Welle ist frequenz- und winkelabhängig [4, 3]. Dieses Prinzip ist in Abbildung 2.2 für den Fall, dass ebene Wellen von einer linearen Apertur empfangen werden, dargestellt [4].
Die Richtungsfunktion (engl. directivity function, directivity pattern, beam pattern) für eine allgemeine, kontinuierliche Apertur lässt sich anschreiben als [3]

$$D_P(f, \mathbf{k}) = \mathcal{F}_{\mathbf{r}_P}\{A_P(f, \mathbf{r}_P)\} = \int_{-\infty}^{\infty} A_P(f, \mathbf{r}_P)\, e^{j2\pi \mathbf{k}\mathbf{r}_P}\, d\mathbf{r}_P \tag{2.7}$$

mit dem Positionsvektor auf der Apertur

$$\mathbf{r}_P = \begin{bmatrix} x_P & y_P & z_P \end{bmatrix}^T \tag{2.8}$$

dem Wellenvektor $\mathbf{k}$ der Welle und

$$d\mathbf{r}_P = dx_P dy_P dz_P. \tag{2.9}$$

Der Ausdruck $\mathcal{F}_{\mathbf{r}_P}\{\cdot\}$ bezeichnet hier die Fouriertransformation. Die Frequenzabhängigkeit in obigen Gleichungen ist implizit mit dem Wellenlängenterm $\lambda = \frac{c}{f}$ gegeben.

In Abbildung 2.2 bezeichnet 1 die Wellenfronten der sich ausbreitenden ebenen Schallwellen und 2 den Signalanteil der von der Apertur "gesehen" wird.

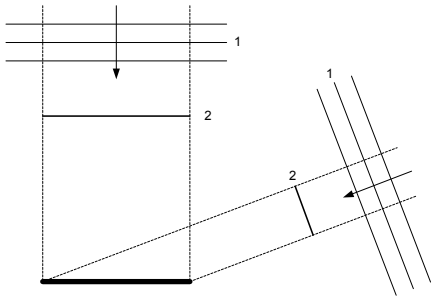


Abbildung 2.2: Ebene Wellen treffen auf eine kontinuierliche, lineare Apertur [4]

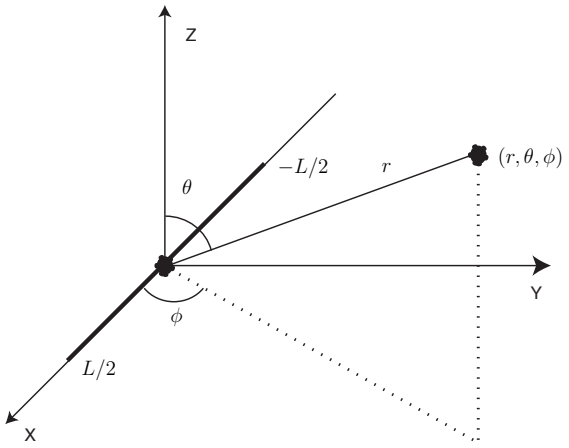


Abbildung 2.3: Kontinuierliche lineare Apertur [4]

2.2.3 Kontinuierliche, lineare Apertur

Betrachtet man eine lineare, kontinuierliche Apertur der Länge L, die, wie in Abbildung 2.3 dargestellt, symmetrisch um den Koordinatenursprung auf der X-Achse liegt, so kann der Positionsvektor mit

$$\mathbf{r}_P = \begin{bmatrix} x_P & 0 & 0 \end{bmatrix}^T \tag{2.10}$$

angegeben werden und die Richtungsfunktion der Apertur ergibt sich zu [4, 3]

$$D_P\left(f, k_x\right) = \mathcal{F}_{x_P}\{A_P\left(f, x_P\right)\} = \int_{-L/2}^{L2} A_P\left(f, x_P\right) e^{j2\pi k_x x_P}\, dx_P \tag{2.11}$$

mit

$$k_x = \frac{\sin\Theta\cos\Phi}{\lambda}. \tag{2.12}$$

Wird die Gleichung als Funktion der Winkel Θ und Φ geschrieben, so erhält man

$$D_P(f,\Theta,\Phi) = \int_{-L/2}^{L2} A_P(f,x_P)\, e^{j\frac{2\pi}{\lambda}\sin\Theta\cos\Phi x_P}\, dx_P. \tag{2.13}$$

Nimmt man eine frequenzunabhängige Aperturfunktion an, so kann diese mit

$$A_P(x_P) = rect\left(\frac{x_P}{L}\right) \tag{2.14}$$

angegeben werden. Mit Hilfe dieser rechteckförmigen Aperturfunktion

$$rect(x/L) \mathrel{\widehat{=}} \begin{cases} 1, & |x| \leq L/2 \\ 0, & |x| > L/2 \end{cases} \tag{2.15}$$

lässt sich mittels Fouriertransformation die Richtungsfunktion wie aus 2.11 folgt berechnen [4, 3]:

$$D_P(f,k_x) = \mathcal{F}\left\{rect\left(\frac{x_P}{L}\right)\right\} = Lsinc(k_xL) \tag{2.16}$$

mit

$$sinc(x) \mathrel{\widehat{=}} \frac{\sin(\pi x)}{\pi x}. \tag{2.17}$$

Betrachtet man nun den Plot der Richtungsfunktion aus Gleichung 2.16 in Abbildung 2.4, so können daraus einige grundlegende Attribute der Richtungsfunktion (auch Beampattern genannt) abgelesen werden. Der Bereich zwischen $-\lambda/L \leq k_x \leq \lambda/L$ wird als Hauptkeule (engl.: main lobe) bezeichnet. Die Breite der Hauptkeule (engl.: beamwidth) ergibt sich für kontinuierliche, lineare Apertur mit $2\lambda/L$ oder auch $2c/fL$. Mit steigender Frequenz wird die Breite der Hauptkeule schmäler [4].

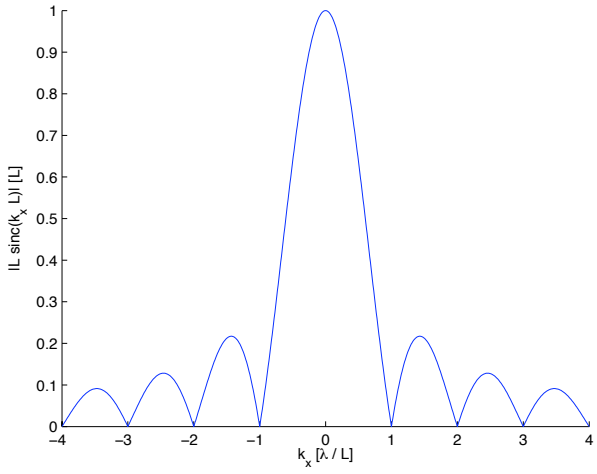
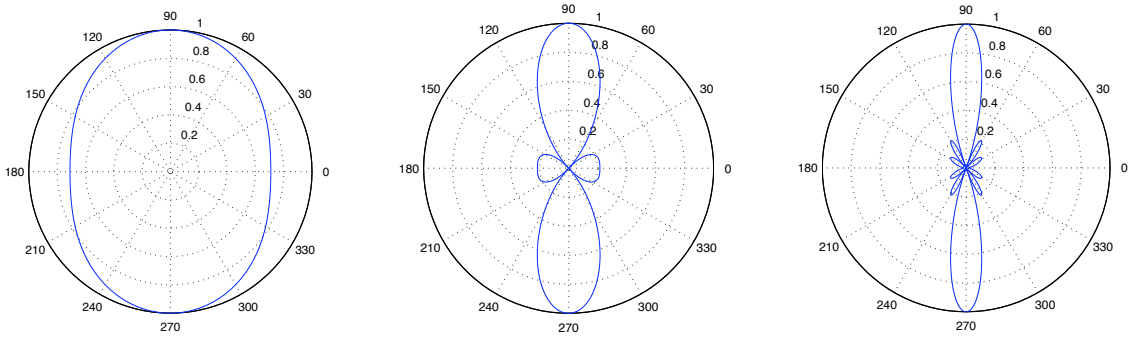


Abbildung 2.4: Normalisierte Richtungsfunktion der Rechteck-Aperturfunktion [4]



(a) $f = 300\,\mathrm{Hz}$ (b) $f = 1000\,\mathrm{Hz}$ (c) $f = 2000\,\mathrm{Hz}$

Abbildung 2.5: Normalisierte, horizontale Richtungsfunktion einer kontinuierlichen, linearen Apertur der Länge $L = 0.5\,\mathrm{m}$ in Polardarstellung

Generell ist es oft üblich, die Richtungsfunktion in einer normalisierten Form [4, 3]

$$D_N\left(f, k_x\right) = \frac{D_P\left(f, k_x\right)}{L} = sinc\left(k_x L\right) \tag{2.18}$$

oder abhängig von den Winkeln Θ und Φ

$$D_N\left(f, \Theta, \Phi\right) = sinc\left(\frac{L}{\lambda}\sin\Theta\cos\Phi\right) \tag{2.19}$$

anzugeben [4]. Als horizontales Beampattern (Elevation $\Theta = 90\,^\circ$) kann die Richtungsfunktion wie folgt angegeben werden

$$D_N\left(f, \frac{\pi}{2}, \Phi\right) = sinc\left(\frac{L}{\lambda}\cos\Phi\right). \tag{2.20}$$

Abbildung 2.5 zeigt die normalisierte, horizontale Richtungsfunktion einer linearen, kontinuierlichen Apertur in Polardarstellung. Die frequenzabhängige Breite der Hauptkeule ist hier deutlich zu erkennen.

2.2.4 Kontinuierliche, planare Apertur

Zweidimensionale Aperturen beliebiger, geometrischer Anordnung werden als planare Aperturen bezeichnet. Die Ausführungen in 2.2.3 lassen sich einfach für zweidimensionale Aperturen weiterentwickeln. Betrachtet man also eine kontinuierliche, planare Apertur beliebiger geometrischer Form, die in der XY-Ebene liegt, so lässt sich Gleichung 2.11 weiterentwickeln zu [3]

$$\begin{aligned} D_P\left(f, k_x, k_y\right) &= \mathcal{F}_{x_P}\mathcal{F}_{y_P}\left\{A_P\left(f, x_P, y_P\right)\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_P\left(f, x_P, y_P\right) e^{j2\pi\left(k_x x_P + k_y y_P\right)}\, dx_P dy_P \end{aligned} \tag{2.21}$$

mit

$$A_P(f, x_P, y_P) = a_P(f, x_P, y_P)\, e^{j\Theta(f, x_P, y_P)} \tag{2.22}$$

und

$$k_x = \frac{\sin\Theta\cos\Phi}{\lambda} \tag{2.23}$$

$$k_y = \frac{\sin\Theta\sin\Phi}{\lambda} \tag{2.24}$$

Die Richtungsfunktion entspricht in diesem Fall einer zweidimensionalen sinc-Funktion.

2.3 Diskrete Sensorarrays - Mikrofonarrays

Ein Sensorarray kann als diskrete Version einer kontinuierlichen Apertur bezeichnet werden, wobei die diskrete Apertur nur an einer endlichen Anzahl an diskreten Punkten angeregt wird. Jedes Sensorelement kann hierbei für sich als kontinuierliche Apertur betrachtet werden.
Die Aperturfunktion des gesamten Arrays wird mittels Superposition der Aperturfunktionen der einzelnen Sensoren bestimmt. Mit einer genügend hohen Anzahl an Sensoren kann mit einem diskreten Sensorarray eine kontinuierliche Apertur angenähert werden [4].

2.3.1 Lineare Arrays

Werden N identische Sensoren entlang einer Raumachse angeordnet, so bilden diese die einfachste Form eines diskreten Arrays, ein lineares Array. Die diskrete Aperturfunktion lässt sich, wie bereits in Abschnitt 2.3 beschrieben mittels Superpositionsprinzip aus den einzelnen Aperturfunktionen der Sensoren bilden und ergibt sich für z.B. ein lineares Array mit ungerader Anzahl an Sensoren zu [4]

$$A_P(f, x_P) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} A_{P_n}(f, x_P) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, e_n(f, x_P - x_n) \tag{2.25}$$

$\omega_n(f)$ stellt die komplexe Gewichtungsfunktion, $e_N(f, x_P)$ die komplexe Systemantwort und x_n die räumliche Position des n-ten Sensorelementes dar. Mittels Fouriertransformation kann aus der Aperturfunktion nun die Fernfeld-Richtungsfunktion ermittelt werden [4]

$$D_P(f, k_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, E_n(f, k_x)\, e^{j2\pi k_x x_n} \tag{2.26}$$

$E_n(f, k_x)$ steht hier für die Aperturfunktion von Sensor n. Sind alle Systemantworten der einzelnen Sensoren identisch (also $E_n(f, k_x) = E(f, k_x), \forall n$), so vereinfachen sich Apertur- und Richtungsfunktion zu

$$A_P(f, x_P) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, \delta(x_P - x_n) \tag{2.27}$$

bzw.

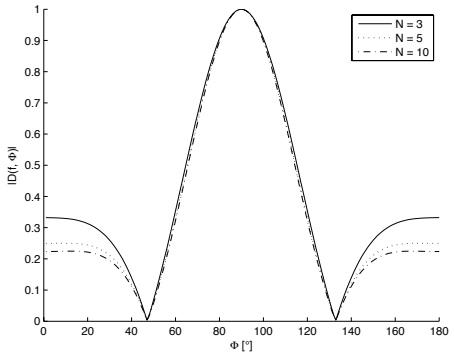
$$D_P(f, k_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, e^{j2\pi k_x x_n}. \tag{2.28}$$

Betrachtet man nun ein lineares, diskretes Sensorarray mit N identischen Sensoren, die im Abstand d äquidistant voneinander positioniert sind, so ergibt sich die horizontal Richtungsfunktion zu

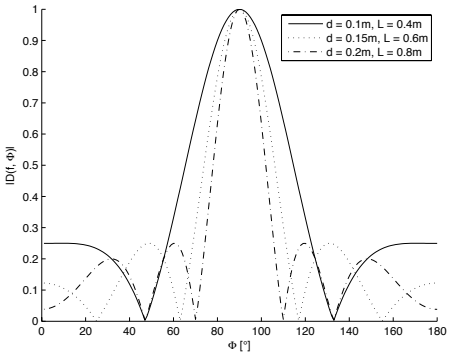
$$D_P(f, \Phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, e^{j\frac{2\pi f}{c} nd\cos\Phi}. \tag{2.29}$$

Wie nun aus Gleichung 2.29 ersichtlich ist, hängt die Richtungsfunktion von der Frequenz f, dem Einfallswinkel Φ, dem Abstand zwischen den Sensorelementen d und der Anzahl der Sensoren N ab.

Um eine möglichst gute Annäherung an eine kontinuierliche Apertur zu erhalten, soll die Anzahl der Sensoren möglichst groß und Abstand zwischen den Sensoren möglichst klein gewählt werden. Die Auswirkungen bei unterschiedlicher Wahl von N bzw. d auf die Richtungsfunktion zeigt Abbildung 2.6.



(a) Variation der Sensoranzahl bei gleich gleichbleibender Frequenz $f = 1\,\text{kHz}$ und Arraylänge $L = 0.5\,\text{m}$



(b) Variation des Sensorabstandes bei gleichbleibender Frequenz $f = 1\,\text{kHz}$ und Sensoranzahl $N = 5$

Abbildung 2.6: Normalisiertes Beampattern eines linearen, diskreten Sensorarrays bei Variation von N bzw. d [4]

In Abbildung 2.6(a) ist gut zu erkennen, dass eine Erhöhung der Sensoranzahl N zu einer Verringerung der Amplitude der Nebenkeulen und somit zu einer Verbesserung des Signal-Rausch-Abstands führt, der als Verhältnis der Amplituden der Hauptkeule zur höchsten Nebenkeule definiert ist. Eine Verkleinerung des Abstands d zwischen den Sensoren und damit der effektiven Arraylänge, bei gleichbleibender Sensoranzahl N führt zu einer Verringerung der Hauptkeulenbreite und damit zu einer Erhöhung der räumlichen Auflösung

[4].

Bei gegebener Arraykonfiguration ist die Breite der Hauptkeule eine Funktion der Frequenz. Abbildung 2.7 zeigt die Richtungsfunktion als Funktion der Frequenz über einen Bereich von 400 Hz $\leq f \leq$ 4000 Hz. Außerdem ist ein weiterer Effekt zu erkennen, der als räumliches Aliasing bezeichnet wird und in Verbindung mit einem großen Sensorabstand d auftritt. Bei dieser Arraykonfiguration tritt ab einer Frequenz von $3000Hz$ räumliches Aliasing auf.

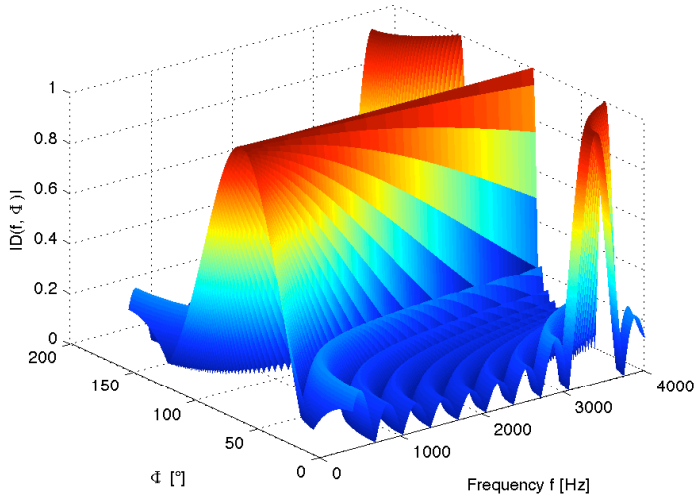


Abbildung 2.7: Normalisierte Richtungsfunktion eines linearen, diskreten Sensorarrays mit $N = 10$, $d = 0.1$ m und 400 Hz $\leq f \leq$ 4000 Hz

Die Theorie des räumlichen Aliasing beruht auf dem Nyquist-Shannon Abtasttheorem, das besagt, dass die Abtastfrequenz f_S mindestens doppelt so groß sein muss wie die maximale Signalfrequenz f_{max} [4]

$$f_S = \frac{1}{T_S} \geq 2f_{max} \tag{2.30}$$

Auf die räumliche Abtastung umgelegt führt dies zu

$$f_{x_S} = \frac{1}{d} \geq 2f_{x_{max}} \tag{2.31}$$

wobei f_{x_S} die räumliche Abtastfrequenz in $[Samples/m]$ und $f_{x_{max}}$ die größte im betrachteten Bereich auftretende Signalfrequenz ist. Mit

$$f_{x_{max}} = \frac{c}{\lambda_{min}} \tag{2.32}$$

kann nun das räumliche Abtasttheorem mit

$$d < \frac{\lambda_{min}}{2} \tag{2.33}$$

angegeben werden [4].

2.3.2 Planare Arrays

Bisher wurden nur lineare, diskrete Sensorarrays behandelt. Im Folgenden werden planare Arrays analysiert, da sie die Basis für das zu verwendende Messsystem bilden. Es sollen nun die Gleichungen für die Apertur- und Richtungsfunktion eines linearen, diskreten Sensorarrays für ein zweidimensionales Array weiterentwickelt werden.

Betrachtet wird nun ein planares M x N Grid-Array das in der XY-Ebene liegt, so lässt sich die Aperturfunktion aus Gleichung 2.25 erweitern zu [3]

$$A_P\left(f, x_P, y_P\right) = \sum_{m=-M'}^{M'} \sum_{n=-N'}^{N'} \omega_{mn}\left(f\right) e_{mn}\left(f, x_P - x_m, y_P - y_n\right) \tag{2.34}$$

mit

$$M' = \frac{M-1}{2} \tag{2.35}$$

und

$$N' = \frac{N-1}{2}. \tag{2.36}$$

Die entsprechende Richtungsfunktion lässt sich mittels zweidimensionaler Fouriertransformation wie in Gleichung 2.21 bzw. mit

$$D_P\left(f, k_x, k_y\right) = \sum_{m=-M'}^{M'} \sum_{n=-N'}^{N'} \omega_{mn}\left(f\right) E_{mn}\left(f, k_x, k_y\right) e^{j2\pi(k_x x_m + k_y y_n)} \tag{2.37}$$

berechnen.

Liegt dem Array kein rasterförmiges Muster zugrunde, so lässt sich die Richtungsfunktion als Funktion der Frequenz f und der Winkel Θ und Φ darstellen als

$$D_P\left(f, \Theta, \Phi\right) = \sum_{n=0}^{N-1} \omega_n\left(f\right) e^{j\frac{2\pi f}{c}(p_{n_x} \sin\Theta\cos\Phi + p_{y_n} \sin\Theta\sin\Phi)} \tag{2.38}$$

wobei N für die Anzahl der Sensoren steht und p_{n_x} und p_{y_n} jeweils die räumliche Position des n-ten Sensors beschreiben.

2.4 Beamforming

In den vorigen Unterkapiteln wurde auf die Eigenschaften unterschiedlicher Apertur- und Arraygeometrien eingegangen. Es wurde gezeigt, wie es möglich wird, durch Anordnung von N Sensoren als Array, Signale aus einer bestimmten Raumrichtung bei gleichzeitiger Abschwächung von Signalen aus anderen Raumrichtungen zu verstärken. Mit der Richtungsfunktion (Beampattern) eines diskreten Sensorarrays kann die Sensitivität des Arrays für verschiedene Raumrichtungen abgelesen werden. Durch eine Variation der komplexen Gewichtungsfunktion der Sensoren lässt sich dabei ohne räumliche Neuausrichtung des Arrays dessen räumliche Sensitivität gezielt beeinflussen. Die Theorie hinter der dafür notwendigen systematischen Steuerung der Hauptkeule des Arrays wird als Beamforming bezeichnet. Man unterscheidet grundsätzlich zwischen vom Eingangssignal abhängigen (adaptiven) und vom Eingangssignal unabhängigen Beamforming Algorithmen.

Betrachtet man Gleichung 2.29 für die Richtungsfunktion eines linearen, diskreten Sensorarrays mit N identen Sensoren, die äquidistant auf der x-Achse um den Koordinatenursprung verteilt sind

$$D_P(f, k_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f)\, e^{j2\pi f k_x n d}$$

so wurde bisher eine gleichmäßige Gewichtung aller Sensoren angenommen

$$\omega_n(f) = \frac{1}{N} \tag{2.39}$$

Die berechnete Richtungsfunktion soll nun durch gezielte Variation der im Allgemeinen komplexen Gewichtungsfunktion, beeinflusst werden [4].

$$\omega_n(f) = a_n(f)\, e^{j\phi_n(f)} \tag{2.40}$$

Die Gewichtungsfunktion besteht im wesentlichen aus zwei Teilen, einem frequenzabhängigen Amplitudenterm $a_n(f)$ und einem frequenzabhängigen Phasenterm $\phi_n(f)$. Die Variation von $a_n(f)$ bewirkt eine Änderung der Amplitude der Richtungsfunktion, eine Variation von $\phi_n(f)$ ermöglicht eine Änderung der räumlichen Ausrichtung der Hauptkeule der Richtungsfunktion. Die gezielte Variation der Position der Hauptkeule wird als Beamsteering bezeichnet [3].

Für eine genauere Betrachtung des Beamsteerings soll nun nur der Phasenterm der Gewichtungsfunktion variiert werden. Der Amplitudenterm der Gewichtungsfunktion wird für die folgenden Betrachtungen mit $a_n(f) = 1$ angenommen. Dadurch ergibt sich die Richtungsfunktion zu

$$D_P(f, k_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j(2\pi f k_x n d + \phi_n(f))} \tag{2.41}$$

wählt man nun

$$\phi_n(f) = -2\pi k'_x n d \tag{2.42}$$

mit

$$k'_x = \frac{\sin\Theta' \cos\Phi'}{\lambda} \tag{2.43}$$

so lässt sich die verschobene Richtungsfunktion anschreiben als [4]

$$D'_P(f, k_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j\frac{2\pi}{\lambda}nd(k_x - k'_x)} \tag{2.44}$$

oder

$$D'_P(f, k_x) = D_P\left(f, k_x - k'_x\right). \tag{2.45}$$

In Abbildung 2.8 ist die horizontale Richtungsfunktion für verschiedene Verschiebungswinkel dargestellt. Wird die Verschiebung entlang der k_x Achse (horizontal) betrachtet, so ändert sich die Form des Richtungsfunktion nicht. Wird die Richtungsfunktion als Funktion des Winkels dargestellt (polar), so ändert sich die Form der Richtungsfunktion mit zunehmendem Verschiebungswinkel [4], wie in Abbildung 2.9 dargestellt.

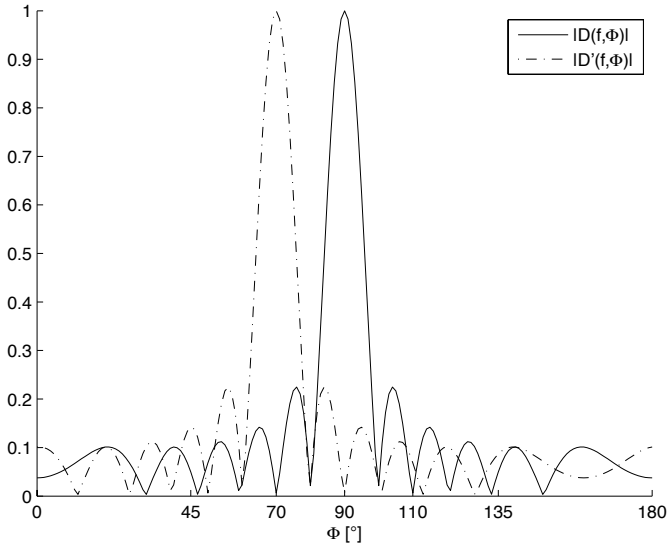
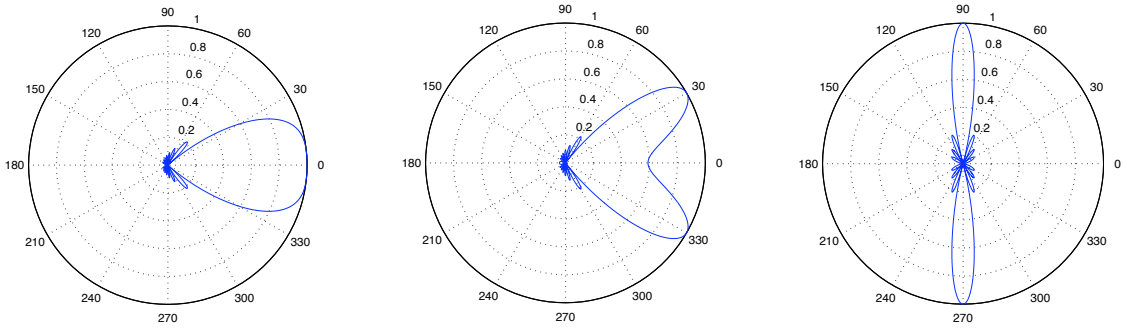


Abbildung 2.8: Normalisierte, horizontale Richtungsfunktion eines linearen, diskreten Sensorarrays mit $N = 10$ und $d = 0.2\,\mathrm{m}$ für $\Phi' = 0\,^\circ$ (durchgezogene Linie) und $\Phi' = 20\,^\circ$ (strichpunktierte Linie)

2.4.1 Delay & Sum Beamforming

Die Beeinflussung der Richtungsfunktion durch Veränderung der komplexen Gewichtungsfunktion wurde im vorigen Kapitel besprochen. Dadurch ist es nun möglich mit der Hauptkeule des Sensorarrays in eine bestimmte Richtung zu "steeren". Trifft nun eine, von einer



(a) $\Phi = 0\,^\circ$ (b) $\Phi = 30\,^\circ$ (c) $\Phi = 90\,^\circ$

Abbildung 2.9: Normalisierte, horizontale Richtungsfunktion einer endlichen, linearen Apertur der Länge L in Polardarstellung für $L/\lambda = 4$ und unterschiedliche Φ

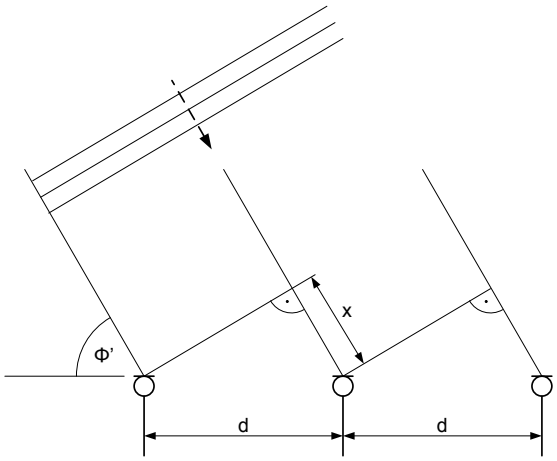


Abbildung 2.10: Ebene Welle trifft auf lineares, diskretes Mikrofonarray

sich im Fernfeld befindlichen Quelle emittierte ebene Welle mit dem Einfallswinkel Φ' auf ein lineares, diskretes Sensorarray (Abbildung 2.10), so sind die an den einzelnen Sensoren gemessenen Signale im Idealfall ident. Sie unterscheiden sich lediglich durch einen zeitlichen Versatz, der sich über den Zusammenhang [4]

$$\tau = \frac{x}{c} = \frac{d \cos \Phi'}{c} \tag{2.46}$$

für zwei benachbarte Mikrofone beschreiben lässt, oder für eine beliebige Mikrofonkombination als

$$\tau_n = \frac{nd \cos \Phi'}{c} \tag{2.47}$$

anschreiben lässt.

Diese zeitliche Verzögerung entspricht einer Phasenverschiebung des Signales im Frequenzbereich. Dieser Zusammenhang wird beim Delay & Sum Beamforming ausgenutzt um mit

dem Sensorarray ein "Steering" des Beams in eine gewünschte Richtung zu erreichen.

Wird nun der Indexbereich der Mikrofone von $-\frac{N-1}{2} \leq n \leq \frac{N-1}{2}$ nach $1 \leq n \leq N$ umgeschrieben, so ergibt sich für den n-ten Sensor folgendes Delay [4]

$$\tau_n = \frac{(n-1)\, d \cos \Phi'}{c} \tag{2.48}$$

Beim Delay & Sum Beamformer werden die Sensorsignale im Zeitbereich um τ_n verzögert anschließend summiert und mit $\frac{1}{N}$ gewichtet. Die komplexe Gewichtungsfunktion ergibt sich im Frequenzbereich zu [4]

$$\omega_n (f) = \frac{1}{N} e^{j \frac{-2\pi f}{c}(n-1) d \cos \Phi'} \tag{2.49}$$

Daraus ergibt sich das Ausgangssignal des Arrays im Frequenzbereich mit

$$y (f) = \frac{1}{N} \sum_{n=1}^{N} x_n (f)\, e^{j \frac{-2\pi f}{c}(n-1) d \cos \Phi'} \tag{2.50}$$

oder im Zeitbereich

$$y (t) = \frac{1}{N} \sum_{n=1}^{N} x_n (t - \tau_n)\,. \tag{2.51}$$

2.4.2 Filter & Sum Beamforming

Der im vorigen Unterkapitel beschriebene Delay & Sum Beamformer kann als weitere Verallgemeinerung der Klasse der Filter & Sum-Beamformer zugeordnet werden. In dieser Klasse sind Amplituden- und Phasenterm der Gewichtungsfunktion (Filterfunktion) frequenzabhängig. Das Ausgangssignal eines Filter & Sum Beamformers kann wie folgt angeschrieben werden [4]

$$y (f) = \sum_{n=1}^{N} \omega_n (f)\, x_n (f)\,. \tag{2.52}$$

2.5 SRP-PHAT Algorithmus

Algorithmen zur Schallquellenlokalisation können grob in drei Kategorien eingeteilt werden: Algorithmen, welche die Energie am Ausgang eines Beamformers auswerten, mit welchem das Schallfeld abgetastet wird (SRP), Konzepte, die für die Quellenlokalisation eine spektrale Schätzung verwenden und Ansätze die die Laufzeitunterschiede zwischen zwei Empfängern auswerten (TDOA - Time Difference Of Arrival) [5].

Im Zuge des *HiMoni*-Projektes hat DI Michael Stadtschnitzer mehrere Algorithmen zur Schallquellenlokalisation miteinander vergleichen, wobei der SRP-PHAT Algorithmus für diese Anwendung die besten Ergebnisse lieferte. Dieser Algorithmus vereint SRP- und TDOA-Konzept zu einem robusten Quelllokalisationsalgorithmus. Im Folgenden werden nun die einzelnen Teile des Algorithmus genauer beschrieben.

2.5.1 Das Impulsantwort-Modell

Ausgehend von ebener Wellenausbreitung wie in Kapitel 2.1 beschrieben, kann der akustische Pfad von der Schallquelle bis zum Mikrofon als lineares System modelliert werden [5]. Das am n-ten Mikrofon empfangene Signal ergibt sich zu

$$x_n(t) = s(t) * h_n(\mathbf{q}_s, t) + v_n(t), \tag{2.53}$$

wobei $s(t)$ das Quellsignal darstellt, $h_n(\mathbf{q}_s, t)$ die Gesamtimpulsantwort zwischen Schallquelle und Mikrofonausgangssignal. Die Gesamtimpulsantwort setzt sich aus zwei kaskadierten Filtern zusammen: der Raumimpulsantwort und der Mikrofonkanal Impulsantwort. Die Raumimpulsantwort charakterisiert den akustischen Pfad zwischen Schallquelle und Mikrofon, inklusive dem Direktschallsignal. Die Mikrofonkanal Impulsantwort beschreibt die elektrischen, mechanischen und akustischen Eigenschaften des Mikrofonsystems, der additive Term $v_n(t)$ beschreibt das zum Signal hinzugefügte Rauschen, dass sich aus dem Rauschen des Mikrofonkanals und etwaigen Umgebungsgeräuschen zusammensetzt. Es wird angenommen, dass dieses Rauschen mit dem Quellsignal $s(t)$ unkorreliert ist. Bei der (Time Delay Estimation) wird der zeitliche Versatz der Direktschallkomponenten in den beiden erhaltenen Mikrofonsignalen ausgewertet. Die Impulsantwort $h_n(\mathbf{q}_s, t)$ kann in Bezug auf das Direktschallkompnenten umgeschrieben werden [5].

$$x_n(t) = \frac{1}{r_n} s(t - \tau_n) * g_n(\mathbf{q}_s, t) + v_n(t) \tag{2.54}$$

r_n beschreibt die Distanz zwischen Schallquelle und Mikrofon, τ_n ist die Verzögerung des Direktschalls, and $g_n(\mathbf{q}_s, t)$ ist die modifizierte Impulsantwort (Gesamtimpulsantwort minus der Direktschallkomponenten) [5].

2.5.2 GCC und die PHAT Gewichtungsfunktion

Für ein Mikrofonpaar $n = 1, 2$ ist die TDOA τ_{12} definiert als $\tau_{12} = \tau_2 - \tau_1$. Wird diese Beziehung auf das Impulsantwortmodell des vorigen Unterkapitels angewandt, so führt dies zu

$$\begin{aligned} x_1(t) &= \frac{1}{r_1} s(t-\tau_1) * g_1(\mathbf{q}_s, t) + v_1(t) \\ x_2(t) &= \frac{1}{r_2} s(t-\tau_1-\tau_{12}) * g_2(\mathbf{q}_s, t) + v_2(t). \end{aligned} \tag{2.55}$$

Sind die Impulsantworten des Mikrofonpaares gleich, so zeigt Gleichung 2.55, dass eine skalierte Version von $s(t-\tau_1)$ im Mikrofonsignal 1, sowie eine zeitverzögerte (und skalierte) Version von $s(t-\tau_1)$ auch in Mikrofonsignal 2 enthalten ist [5].

Die Kreuzkorrelation c_{12} zweier Signale ist im Zeitbereich definiert als

$$c_{12}(\tau) = \int_{-\infty}^{+\infty} x_1(t)\, x_2(t+\tau)\, dt \tag{2.56}$$

und ist ein Maß für die Ähnlichkeit der Signale. Sie wird zur Ermittlung von Laufzeitunterschieden zwischen zwei Signalen verwendet und liefert ein Maximum beim Zeitversatz zwischen beiden Signalen, welcher der TDOA, also τ_{12} entspricht.
Eine allgemeine Form der Kreuzkorrelation stellt die GCC (Generalised Cross Correlation) Funktion $R_{12}(\tau)$ dar und ist definiert als Kreuzkorrelation von zwei gefilterten Versionen von $x_1(t)$ und $x_2(t)$. Sie kann im Zeitbereich angeschrieben werden als

$$R_{12}(\tau) = \int_{-\infty}^{+\infty} (g_1(t) * x_1(t))\,(g_2(t) * x_2(t+\tau))\, dt, \tag{2.57}$$

oder mit Hilfe der Fouriertransformation im Frequenzbereich als [5]

$$R_{12}(\tau) = \int_{-\infty}^{+\infty} (G_1(f)\, X_1(f))\,(G_2(f)\, X_2(f))^*\, e^{j2\pi f\tau} df. \tag{2.58}$$

Die Filterfunktionen können als $\Psi_{12}(f) = G_1(f)\, G_2(f)^*$ zusammengefasst werden und somit ergibt sich die GCC Funktion zu

$$R_{12}(\tau) = \int_{-\infty}^{+\infty} \Psi_{12}(f)\, X_1(f)\, X_2(f)^*\, e^{j2\pi f\tau} df \tag{2.59}$$

Idealerweise ergibt sich damit ein explizites, globales Maximum an der Stelle des Zeitversatzes τ_{12} (der TDOA) zwischen den beiden Signalen.

Im Allgemeinen hat $R_{12}(\tau)$ mehrere lokale Maxima, die durch Reflexionen entstehen können. Das Ziel der Gewichtungsfunktion Ψ_{12} ist es, die GCC bei der wahren TDOA hervorzuheben und alle anderen Maxima zu dämpfen. Die PHAT (Phase Transform) Gewichtungsfunktion ist definiert als [5]

$$\Psi_{12}(f) = \frac{1}{|X_1(f)\, X_2^*(f)|}. \tag{2.60}$$

Bei Verwendung dieser Gewichtungsfunktion werden alle Signalfrequenzen in gleichem Maß betont und das Signal erhält dadurch eine weiße Färbung. Die PHAT Gewichtungsfunktion bildet zusammen mit dem "Steered"-Beamformer die Basis des SRP-PHAT Algorithmus.

Rechentechnisch wird die Kreuzkorrelation bzw. die GCC in der Regel über die inverse Fouriertransformation des Kreuzleistungsdichtespektrum $S_{X_1X_2}(f)$ ermittelt [6]. Das normierte Kreuzleistungsdichtespektrum berechnet sich aus

$$S_{X_1X_2}(f) = \frac{X_1^*(f) \cdot X_2(f)}{|X_1(f)\, X_2^*(f)|}. \tag{2.61}$$

Die Kreuzkorrelationsfunktion kann auch angeschrieben werden als

$$R_{12}(\tau) = \int_{-\infty}^{+\infty} S_{X_1X_2}(f)\, e^{j2\pi f\tau} df \tag{2.62}$$

2.5.3 SRP basierte Lokalisation

Das Impulsantwort-Modell aus 2.5.1 zeigt, dass für ein Array aus N Mikrofonen im Aufnahmebereich der Schallquelle jedes Mikrofonsignal eine verzögerte, gefilterte und mit Rauschen beaufschlagte Version des Quellsignals $s(t)$ enthält. Das Ausgangssignal eines Delay & Sum Beamformers kann, wie bereits in Abschnitt 2.4.1 erläutert, angeschrieben werden mit

$$y(t, \mathbf{q}) = \sum_{n=1}^{N} x_n(t + \Delta_n) \tag{2.63}$$

Δ_n beschreibt das Steering Delay, für eine Fokussierung des Arrays auf eine bestimmte räumliche Position $\mathbf{q}$. Im Prinzip reicht die Kenntnis der TDOA's (und somit der Steering Delays) aus, um den Beamformer auf eine bestimmte räumliche Position zu fokussieren [5]. Damit nun alle auf das Mikrofonsignal auf dem Weg von der Schallquelle zum Mikrofon wirkenden Einflüsse besser abgebildet werden können, kann statt des Delay & Sum Beamformers der allgemeine Ansatz des Filter & Sum Beamformers verwendet werden.
Das Ausgangssignal eines N-elementigen Filter & Sum Beamformer kann im Frequenzbereich angeschrieben werden als

$$Y(f, \mathbf{q}) = \sum_{n=1}^{N} G_n(f)\, X_n(f)\, e^{j2\pi f\Delta_n} \tag{2.64}$$

wobei $X_n(f, \mathbf{q})$ und $G_n(f, \mathbf{q})$ die Fourier Transformierte des n-ten Mikrofonsignales und des dazugehörigen Filters darstellen. Die Mikrofonsignale werden dabei mit den Steering Delays entsprechend phasenverzögert womit sich eine Fokussierung auf die räumliche Position $\mathbf{q}$ ergibt.

Die Wahl der PHAT Gewichtungs- bzw. Filterfunktion, die alle Frequenzkomponenten gleich gewichtet hat für eine praktische Anwendung einige Vorteile wie zum Beispiel verbesserte Robustheit bei moderaten Raumreflexionen.

Der Beamformer kann zur Lokalisation von Schallquellen eingesetzt werden, indem das Array zu bestimmten räumlichen Positionen "gesteered" wird und das Ausgangssignal, üblicherweise die Energie, am betrachteten Punkt ausgewertet wird. Steered Beamformer ermöglichen kurze Analyseintervalle. Die SRP, also die Energie am Ausgang eines Filter & Sum Beamformers kann für eine bestimmte räumliche Position $\mathbf{q}$ mit

$$P(\mathbf{q}) = \int_{-\infty}^{+\infty} |Y(f)|^2 df \tag{2.65}$$

angegeben werden, der Schätzer für die Position kann dann mit

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} P(\mathbf{q}) \tag{2.66}$$

gefunden werden [5].

2.5.4 SRP-PHAT Algorithmus

Mit den Ausführungen in den vorangegangenen Unterkapiteln kann nun der SRP-PHAT Algorithmus definiert werden. Das Ziel des SRP-PHAT Algorithmus ist es, die Vorteile des "Steered"-Beamformers für Quelllokalisation (kurze Analyseintervalle) mit der Robustheit der PHAT Gewichtungsfunktion zu kombinieren [5].

Die SRP eines Filter & Sum Beamformers kann als

$$P(\mathbf{q}) = \sum_{l=1}^{N} \sum_{k=1}^{N} \int_{-\infty}^{+\infty} \Psi_{lk}(f) X_l(f) X_k^*(f) e^{j2\pi f(\Delta_k - \Delta_l)} df \tag{2.67}$$

angeschrieben werden, wobei $\Psi_{lk}(f) = G_l(f) G_k^*(f)$ dem 2-kanaligen GCC Gewichtungsterm in Gleichung 2.60 entspricht. Die mehrkanalige Version der PHAT-Gewichtungsfunktion kann mit

$$\Psi_{lk}(f) = \frac{1}{|X_l(f) X_k^*(f)|} \tag{2.68}$$

angegeben werden, was im Kontext des Filter & Sum Beamformers aus Gleichung 2.64 dem Äquivalent des Filters

$$G_n(f) = \frac{1}{|X_n(f)|} \tag{2.69}$$

entspricht [5].

Die SRP für den betrachteten Punkt $\mathbf{q}$ ergibt sich somit zu [5]

$$P(\mathbf{q}) = \sum_{l=1}^{N} \sum_{k=1}^{N} \int_{-\infty}^{+\infty} \frac{X_l(f) X_k^*(f)}{|X_l(f) X_k^*(f)|} e^{j2\pi(\Delta_k - \Delta_l)} df \tag{2.70}$$

oder anders angeschrieben zu

$$P(\mathbf{q}) = \sum_{l=1}^{N} \sum_{k=1}^{N} R_{lk} (\Delta_k - \Delta_l) . \tag{2.71}$$

Die SRP entspricht der Summe aller paarweisen GCC Permutationen, welche um die zugehörigen Steering Delays verschoben werden. In dieser Summe ist auch die Summe von N Autokorrelationen enthalten, welche allerdings nur einen Gleichanteil zur Gesamtenergie liefern, da sie unabhängig von den Steering Delays sind.

Die Schallquellenlokalisation wird nun wie bei Standard SRP-basierten Ansätzen durchgeführt, indem im betrachteten räumlichen Bereich nach Maxima gesucht wird [5].

Kapitel 3

Systemstruktur

Die in den vorangegangenen Arbeiten und Projekten verwendete Systemstruktur kann in zwei Bereiche unterteilt werden:

- Messsystem
- Analysesystem

Mit dem Messsystem wurden in mehreren Messreihen Messdaten in verschiedenen Umgebungen und mit verschiedenen Mikrofonarraykonfigurationen aufgezeichnet. Diese Messdaten bildeten die Grundlage für die Entwicklung des Fahrtrichtungsdetektionsalgorithmus, der auf dem Analysesystem (PC) ausgeführt wurde.

In diesem Kapitel werden das bisher verwendete Messsystem und das Analysesystem bzw. der Analysealgorithmus genauer beschrieben und darauf eingegangen, welche Änderungen an der Systemstruktur für ein "echtzeitfähiges" System notwendig sind.

3.1 Messsystem

Das Messsystem besteht, wie in Abbildung 3.1 dargestellt, aus dem Mikrofonarray mit eingebauter Videokamera und einer Datenakquisitionseinheit, welche in diesem Fall aus National Instruments (NI) Hardware und einem Messnotebook mit entsprechender Software zur Aufzeichnung der Daten besteht.

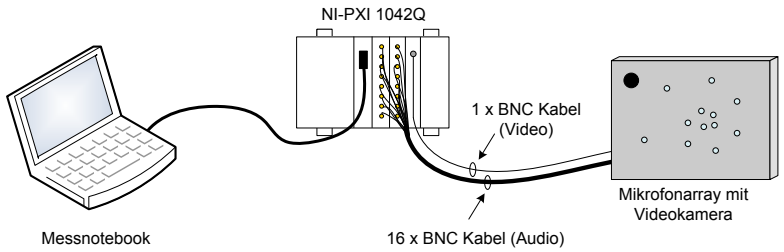


Abbildung 3.1: Schematische Darstellung des Messsystems

3.1.1 Mikrofonarray

Das Mikrofonarray besteht aus einer Box mit einer austauschbaren Frontplatte, in welcher die Messmikrofone eingebaut werden. Als Messmikrofone sind für diese Anwendung 1/4" Konstantstrommessmikrofone vorgesehen.

Die Arraygeometrie, also die räumliche Verteilung der Mikrofone auf der Frontplatte des Beamformers wurde in den vorangegangenen Projekten und Arbeiten entwickelt. Insbesonders sei hier auf die Diplomarbeit von Peter Dollfuß [1] verwiesen, der in seiner Arbeit eine Arraysimulation und ein Arraydesign speziell für diese Anwendung entwickelt hat.

Durch die austauschbare Frontplatte kann die Arraygeometrie verändert bzw. an andere Umgebungsbedingungen angepasst werden. Als Tradeoff zwischen Genauigkeit der Auflösung und der zu verarbeitenden Datenmenge, wurde die Anzahl der zu verwendenden Mikrofone mit 16 festgelegt.

Zur visuellen Kontrolle des Analysebereiches wurde eine Kamera in das Mikrofonarray eingebaut. Die Konstantstrommessmikrofone werden über BNC-Kabel an die Datenakquisitionseinheit angeschlossen und von dieser mit Energie versorgt.

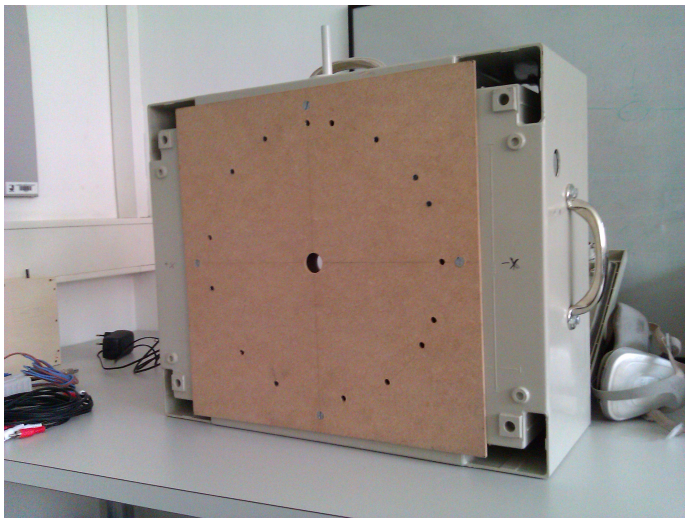


Abbildung 3.2: Verwendetes Mikrofonarray mit irregulärer, kreisförmiger Geometrie

3.1.2 Datenakquisitionseinheit

Die Datenakquisitionseinheit bestand bei den vorangegangenen Messungen aus einem NI PXI-System, einem Messnotebook und entsprechender Software zu Aufzeichnung der Daten.

Das PXI-System besteht aus:

- PXI-Controller
- 2 PXI-4472 Modulen für A/D-Wandlung, IEPE-Versorgung, großer Dynamikbereich
- PXI-1411 Module für Videoaufzeichnung

Die Verbindung zwischen Messnotebook und PXI-System wurde mittels PCMCIA Cardbus hergestellt, welcher eine PCI-zu-PCI-Brücke mit einer Bandbreite von 50 $^{\mathrm{MBit}}/_{\mathrm{s}}$ ermöglicht. Die Software für die Datenaufzeichnung wurde in LabView entwickelt. Als Datenformat wurde das NI TDMS-Format verwendet. Über Parameter kann die Abtastrate und die Wortbreite der Audiodaten bei der Aufzeichnung eingestellt werden. Prinzipiell wurde bei allen Messungen 24 Bit Wortbreite bei 48 kHz Abtastrate für die Audiodaten festgelegt. Die Framerate des Videostreams beträgt 25 Bilder pro Sekunde.

3.2 Analysesystem

Das Analysesystem bildet ein normler PC, auf dem die gemessenen Daten nach abgeschlossener Aufzeichnung (also offline) mit dem in MATLAB implementierten Analysealgorithmus analysiert werden. Die Detektionsergebnisse (Fahrzeug, Fahrtrichtung und Geschwindikeit) werden anschließend in einem GUI (Graphical User Interface), zusammen mit den aufgezeichneten Videodaten ausgegeben.

3.2.1 Analysealgorithmus

Der Analysealgorithmus besteht aus zwei Stufen:

- Generierung des akustischen Energiebildes der Analysefläche
- Detektion von Fahrzeugen und Fahrtrichtung

Generierung der akustischen Bilder

Damit der Algorithmus angewandt werden kann, müssen zuerst noch einige benötigte Parameter genauer bestimmt werden. Die hier beschriebenen Abmessungen der gesamten Konfiguration stammen von einer der letzten Testmessungen an der S31, der Burgenland Schnellstraße im Bereich der Autobahnauffahrt Sieggraben. Abbildung 3.3 zeigt eine schematische Darstellung des Messaufbaus bei dieser Messung.

Die Analysefläche auf der Fahrbahn umfasst vier Fahrspuren, je zwei in dieselbe Richtung. Dieser Bereich wurde anschließend in eine Analysepunktmatrix unterteilt.

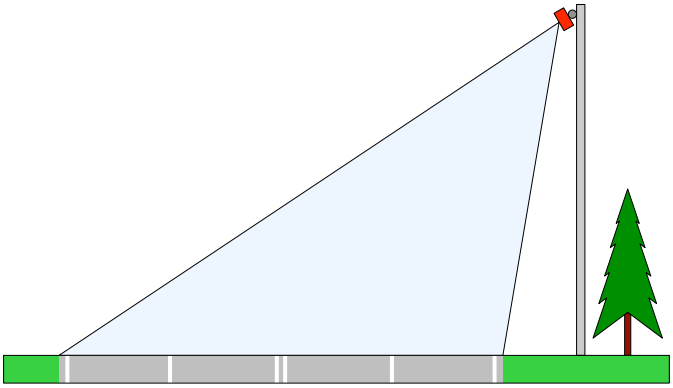


Abbildung 3.3: Schematische Darstellung des Messaufbaus bei der Messung an der S31

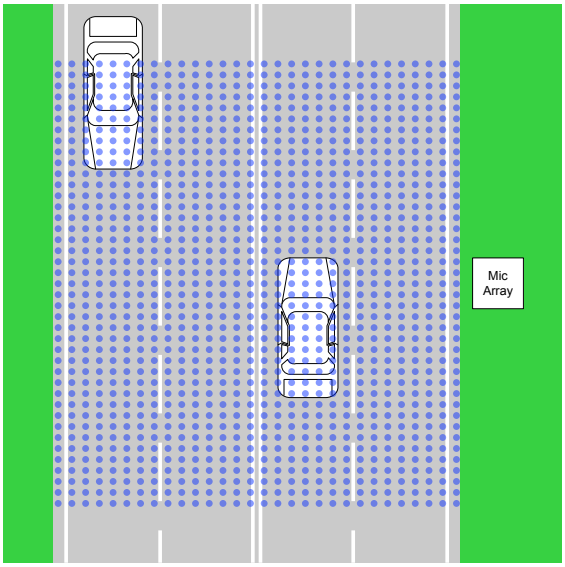


Abbildung 3.4: Schematische Darstellung der Analysefläche bei der Messung an der S31

Als Mikrofonarraykonfiguration wurde ein Array verwendet, das in 8 m Höhe am Rand der Fahrbahn montiert war und in einem Winkel von ca. 48 ° auf die Fahrbahnmitte ausgerichtet war. Am Mikrofonarray war zusätzlich zu den Mikrofonen noch eine Kamera montiert, mit der der Analysebereich aufgezeichnet wurde.

Mikrofonsignale aus DB auslesen: Hier werden die 16 Mikrofonkanäle aus der Recording Datenbank (DB) frameweise ausgelesen und für die weitere Verarbeitung in Vektoren gespeichert. Das Videosignal wird mit einer Framerate von 25 $^{\text{Frames}}/_{\text{s}}$ aufgezeichnet, was einer Zeitdauer von 40 ms für einen Frame entspricht. Damit die Audiodaten mit den Videodaten bei der Auswertung einfach zusammenzuführen sind, werden die Audiodaten in 80 ms Frames, mit einer Überlappung der Audioframes von 40 ms verarbeitet. Bei einer Abtastrate von 48 kHz beinhaltet ein 80 ms Audioframe 3840 Samples, die Überlappung der Frames beträgt 1920 Samples.

Filtern und Fenstern der Mikrofonsignale: Der in den Mikrofonsignalen enthaltene Gleichanteil wird mit einem Nullphasigen Hochpassfilter gefiltert und anschließend mit einem Hamming Fenster multipliziert.

FFT der Mikrofondaten erzeugen: Alle Mikrofonsignale müssen zuerst mittels Fast Fourire Transformation (FFT) in den Frequenzbereich transformiert werden. Das Zero-Padding für Signallängen ungleich einer 2^n Potenz übernimmt die MATLAB-Funktion *fft()*.

Abbildung 3.5 zeigt das für einen vorbeifahrenden PKW generierte akustische Bild der Analysefläche, wobei die x-Achse in Fahrtrichtung ausgerichtet ist. Der berechnete Energiewert entspricht der Färbung. Je heller die Färbung, desto höher ist der berechnete Energiewert für den entsprechenden Analysepunkt, je dunkler, desto niedriger ist dieser Wert.

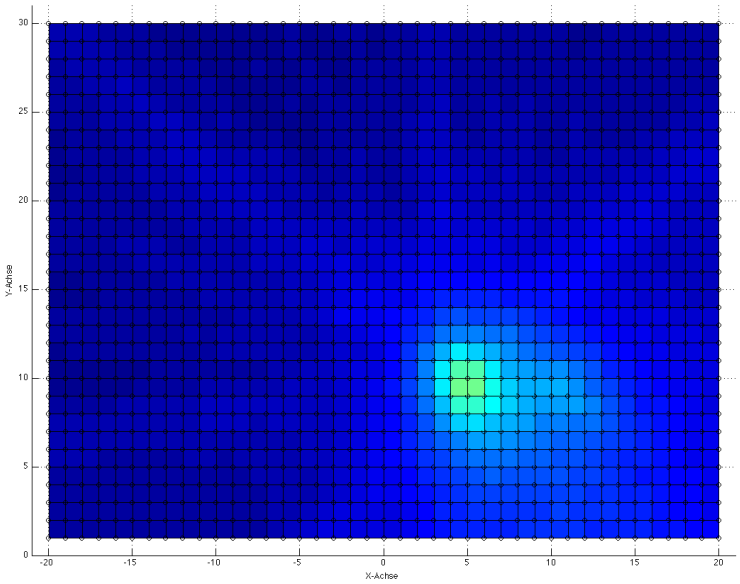


Abbildung 3.5: Berechnetes akustisches Bild für einen PKW

In dreidimensionaler Darstellung sieht das akustische Bild des PKW's wie in Abbildung 3.6 dargestellt aus.

Detektion von Fahrzeugen und Fahrtrichtung

Das akustische Bild (Energiebild) der Analysefläche dient als Grundlage für die Detektion von Fahrzeugen und in weiterer Folge deren Fahrtrichtung und Fahrtgeschwindigkeit. Fahrzeuge erscheinen hier als lokale Energiemaxima. Die Form der Energiemaxima kann nicht genau festgelegt werden, da sie von der Art des Fahrzeugs (einspurig, zweispurig), Anzahl der Achsen, Aufbau und noch vielen weiteren Faktoren abhängt. An dieser Stelle möchte ich wieder auf die Diplomarbeit von Peter Dollfuß verweisen, der dieses Thema in seiner Diplomarbeit [1] behandelt hat. Es kann auf jeden Fall davon ausgegangen werden,

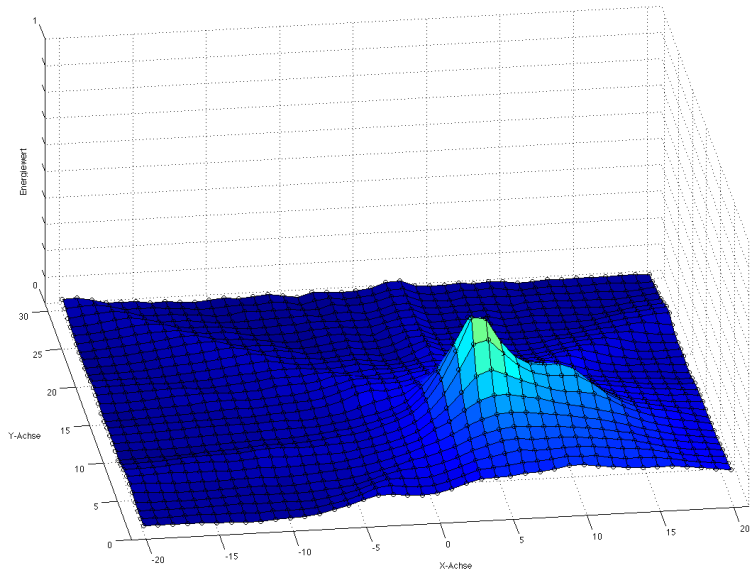


Abbildung 3.6: Berechnetes akustisches Bild für einen PKW in dreidimensionaler Darstellung

dass ein die Analysefläche passierendes Fahrzeug mindestens ein lokales Energiemaximum im akustischen Bild verursacht.

Fahrzeuge sollen bei einer Durchfahrt durch den Analysebereich detektiert werden. Da die Fahrtrichtung der Fahrzeuge prinzipiell nicht bekannt ist bzw. detektiert werden soll, wird im Analysebereich ein Detektionsbereich definiert. Abbildung 3.7 zeigt die Analysefläche mit eingezeichnetem Detektionsbereich.

Die Fahrzeugdetektion kann grob in zwei zusammenhängende Stufen eingeteilt werden:

1. Ermittlung eines lokalen Energiemaximums (Fahrzeug) im Detektionsbereich
2. Bestimmung der Fahrtrichtung, Fahrtgeschwingkeit und Fahrspur

Der Detektionsalgorithmus verwendet eine Raster-Suche zur Bestimmung der lokalen Energiemaxima im Detektionsbereich.

Wird in der 1. Stufe ein Fahrzeug detektiert, so wird die 2. Stufe aufgerufen und versucht Fahrtrichtung und Fahrtgeschwindigkeit des Fahrzeugs zu ermitteln. Konnte die 2. Stufe erfolgreich ausgeführt werden, so werden die Koordinaten der Detektion im Detektionsbuffer gespeichert. Nach einer erfolgreichen Detektion wird die Suche nicht abgebrochen, sondern mit den Folgekoordinaten fortgesetzt.
Wurden alle Koordinaten der Detektionsfläche abgearbeitet, so wird das Detektionsergebnis gespeichert und der Vorgang beginnt mit dem nächsten Frame von vorne.

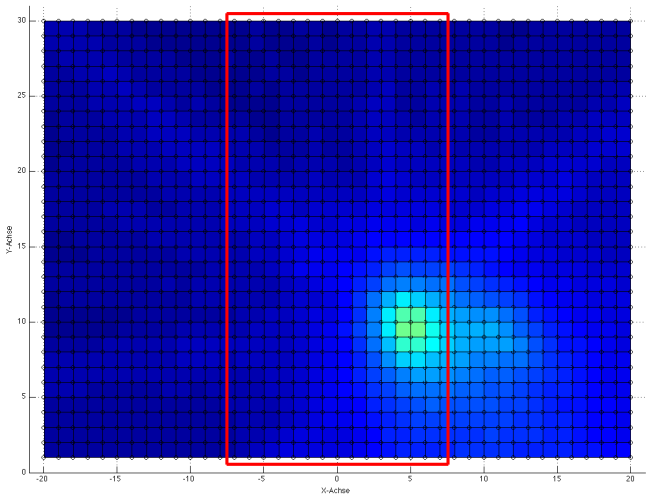


Abbildung 3.7: Energiebild der Analysefläche mit markiertem Detektionsbereich

3.2.2 Ausgabe der Detektionsergebnisse

Nach erfolgter Berechnung für den gewählten Zeitbereich wird das aufgezeichnete Video zusammen mit den dazugehörigen berechneten akustischen Bildern und den Detektionsergebnissen in einem MATLAB GUI ausgegeben.

3.3 Notwendige Änderungen

Bei der bisherigen MATLAB Implementierung des Analysealgorithmus stand nur die Funktionalität und nicht die Effizienz im Vordergrund. Für die Verarbeitung von 5 Minuten Audiomaterial sind in dieser Implementierung des Algorithmus 20 Minuten Rechenzeit notwendig. Die technischen Daten des verwendeten PC sind in Anhang A.1 zu finden. Das maximale Zeitfenster, in welchem die Verarbeitung der Daten abgeschlossen sein muss, hängt von der Framerate der generierten akustischen Bilder ab und ist im vorliegenden Fall wegen der Synchronizität mit dem aufgezeichneten Video mit 40 ms angenommen worden. Für eine Online Verarbeitung, also einer Echtzeit-Verarbeitung der Daten muss die Berechnung des akustischen Bildes innerhalb dieses Zeitfensters erfolgen. Dafür sind einige Änderungen im Aufbau und der Struktur des Analysealgorithmus notwendig. Des weiteren soll im Rahmen dieser Arbeit die verwendete NI Datenakquisitionshardware durch kostengünstigere Hardware ersetzt werden.

3.3.1 Spezifikationen

Die hier angeführten Spezifikationen geben den Rahmen für die, im nächsten Kapitel folgende Marktrecherche zu Datenakquisitionssystemen vor.

Mikrofone

Die in den bisherigen Messungen verwendeten Konstantstrommessmikrofone sollen auch weiterhin verwendet werden. Eine entsprechende Konstantstromversorgung ist also notwendig. Der Übertragungsbereich der Mikrofone wird mit $20\,\mathrm{Hz} - 20\,\mathrm{kHz}$ festgelegt.

Mikrofonarray

Das aus den vorangegangenen Arbeiten übernommene Konzept des Mikrofonarrays mit 16 Mikrofonen die in einer austauschbaren Frontplatte montiert sind, soll beibehalten werden. Die Arraygeometrie kann somit nachträglich verändert werden. Durch die hohe Montage des Arrays soll das Gewicht möglichst gering gehalten werden. Im Mikrofonarray darf kein zusätzlicher Schall durch Lüfter oder andere Schallquellen emittiert werden. Des weiteren soll der Verkabelungsaufwand zwischen Mikrofonarray und der Analyseeinheit möglichst gering und einfach gehalten werden um mögliche Fehlerquellen von vornherein auszuschließen.

Analyserechner und digitale Audiodaten

Als Analyserechner ist eine normaler PC vorgesehen, der räumlich getrennt vom Mikrofonarray aufgebaut ist und digitale Audiodaten von 16 Kanälen bei einer Abtastrate von 48 kHz mit einer Wortbreite von 24 Bit verarbeiten kann. Die örtliche Lage der Datenakquisitionseinheit, bzw. eine eventuelle notwendige Datenübertragung zum Analysesystem wurde nicht genauer spezifiziert, um keine weiteren Einschränkungen für die Marktrecherche zu haben. Auf jeden Fall sollten Treiber für die Anbindung des Datenakquisitionsystems an den Analyserechner vorhanden sein.

Umwelt

Das System soll vorwiegend auf Autobahnen eingesetzt werden, d.h. es muss von einer störungsreichen Umgebung ausgegangen werden. Das Mikrofonarray bzw. die Datenakquisitionseinheit muss für einen erweiterten Temperaturbereich ausgelegt sein, da diese Teile des Systems samt Einhausung der direkten Sonneneinstrahlung ausgesetzt sind. Es können Temperaturen von $-20\,^{\circ}\mathrm{C} - +70\,^{\circ}\mathrm{C}$ angenommen werden.

Kapitel 4

Kommerzielle Systeme

In diesem Kaptiel werden am Markt erhältliche Systeme für die Akquisition und Übertragung von 16 Mikrofonsignalen des Mikrofonarrays zum Analyserechner zusammengestellt. Recherchiert wurde im Internet in den Bereichen Audiotechnik und elektrische Messtechnik nach den in Abschnitt 3.3.1 angeführten Spezifikationen.

Eine Vorgabe war, ein System zusammenzustellen, welches in Zukunft eventuell auch zu einem Produkt weiterentwickelt werden kann. Ein gutes Preis-Leistungs-Verhältnis ist dafür gefordert und notwendig.
Da die Verwendung von Konstantstrommessmikrofonen in der Audiotechnik nicht üblich ist, muss für A/D-Wandlerkarten aus diesem Bereich noch eine geeignete Konstantstromversorgung vorgeschalten werden. Die Konstantstrom-Versorgungsmodule mit zugehöriger Signalaufbereitung werden entweder zugekauft oder selbst entwickelt.

Die folgenden Setups wurden anhand der vorgegebenen Spezifikationen zusammengestellt und auf die Tauglichkeit für den Einsatz in diesem Projekt untersucht.

4.1 Setup 1

Dieses System besteht aus zwei BEHRINGER ADA8000 PRO-8 8 Kanal A/D-Wandlerkarten und einer RME HSP 9652 PCI Audiokarte. Die A/D-Wandlerkarten werden über ADAT-Kabel an die RME Audiokarte angeschlossen. Die RME-Audiokarte liefert das Masterclock-Signal für die A/D-Wandlerkarten.

Fazit

Sehr günstiges System mit einfacher Verkabelung. Die Kabellängen für ADAT-Verkabelung sind allerdings nur für wenige Meter (bis 20 m) spezifiziert. Der Temperaturbereich der A/D-Wandlerkarten ist im Datenblatt mit “Raumtemperatur” angegeben und nicht näher festgelegt. Für eine Verwendung im Mikrofonarray müsste hierfür wahrscheinlich eine Klimatisierung (Heizung/Kühlung) vorgesehen werden. Die fehlende Konstantstromversorgung müsste zugekauft oder selbst entwickelt werden und ist wie die benötigte Klimatisierung im Preis nicht enthalten.

Abtastrate	bis 48 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	Nicht vorhanden
Temperaturbereich	0 °C – +40 °C
Host-Anbindung	RME HSP 9652 PCI Audiokarte mit optischem ADAT
Treiber	Audiotreiber für Windows, OSX und Linux
Verkabelungsaufwand	2 x ADAT Lichtwellenleiter, 1 x Wordclock
Preis (Stand: 12/2010)	ca. 900 €

Tabelle 4.1: Daten Setup 1

4.2 Setup 2

Dieses System besteht aus einer FERROFISH A16 MK-II A/D-Wandlerkarte die über eine optische -Verbindung mit einem RME HDSP MADI- PCI-Karte verbunden ist.

Abtastrate	32 kHz bis 192 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	Nicht vorhanden
Temperaturbereich	0 °C – +40 °C
Host-Anbindung	PCI-MADI Karte
Treiber	Windows, OSX, Linux
Verkabelungsaufwand	MADI Lichtwellenleiter
Preis (Stand: 12/2010)	2500 €

Tabelle 4.2: Daten Setup 2

Fazit

Qualitativ hochwertiges Audiosystem mit einfacher Verkabelung. Die Kabellängen für optisches MADI sind mit 2000 m festgelegt, was für die Aufgabenstellung mehr als ausreichend ist. Der Temperaturbereich der A/D-Wandlerkarte ist wieder mit ´´Raumtemperatur" angegeben, für eine Montage im Mikrofonarray müsste also wieder eine Klimatisierung vorgesehen werden. Die Klimatisierung und die fehlende Konstantstromversorgung wurden im angegebenen Preis nicht berücksichtigt.

4.3 Setup 3

Dieses System besteht aus einer DEWE-ORION-1624-20x 16-kanaligen PCI A/D-Wandlerkarte des österreichischen Messtechnikherstellers Dewetron.

Abtastrate	bis 204.8 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	Nicht vorhanden
Temperaturbereich	0 °C – +50 °C
Host-Anbindung	PCI-Karte
Treiber	Windows
Verkabelungsaufwand	68-Pin SCSI Stecker
Preis (Stand 12/2010)	4000 €

Tabelle 4.3: Daten Setup 3

Fazit

Diese A/D-Wandlerkarte besitzt sehr gute elektrische Signaleigenschaften, ist jedoch nur als PCI-Steckkarte erhältlich. Das Analysesystem müsste demnach in der Nähe bzw. im Mikrofonarray verbaut werden, da die Anbindung an die Steckkarte mittels 68-Pin SCSI Kabel erfolgt und die Kabellänge auf wenige Meter begrenzt ist. Die Klimatisierung und die fehlende Konstantstromversorgung und Signalaufbereitung ist im Preis nicht enthalten.

4.4 Setup 4

Bei diesem Setup werden kaskadierbare, 4-kanalige ETH-8824 Data Translation Ethernet Messmodule eingesetzt. Die einzelnen Module besitzen vier differentielle Analogeingänge mit wählbaren Eingangsspannungsbereichen und können miteinander synchronisiert werden.

Abtastrate	bis 48 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	Nicht vorhanden
Temperaturbereich	0 °C – +55 °C
Host-Anbindung	Gigabit-Ethernet
Treiber	Windows
Verkabelungsaufwand	1 x Ethernetkabel
Preis (Stand: 12/2010)	10000 €für 4 Module zu je 2500 €

Tabelle 4.4: Daten Setup 4

Fazit

Bei diesem System wird eine Ethernetverbindung zur Datenübertragung verwendet, wodurch sich der Verkabelungsaufwand auf ein einzelnes Kabel reduziert. Des weiteren kann ein Glasfaserkabel für die Datenübertragung verwendet werden, wodurch sich die Länge des Übertragungskanales deutlich erhöht. Durch den hohen Preis pro Modul kann dieses Setup für ein späteres Produkt allerdings nicht herangezogen werden.

4.5 Setup 5

Dieses Setup besteht aus einem ROGA VibDAQ16+ Datenlogger. Alle Eingänge bieten eine Konstantstromversorgung und entsprechende Signalkonditionierung. Die Anbindung an das Analysesystem erfolgt per USB. Für eine verlustfreie Datenübertragung besitzt das Modul einen $512MB$ großen Pufferspeicher.

Abtastrate	bis 97 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	vorhanden
Temperaturbereich	0 °C – +70 °C
Host-Anbindung	USB
Treiber	Windows
Verkabelungsaufwand	1 x USB-Kabel
Preis (Stand: 12/2010)	4500 €

Tabelle 4.5: Daten Setup 5

Fazit

Dieses System ist zwar als Datenlogger konzeptioniert, ist aber für die Anwendung sehr gut geeignet, da es bereits eine integrierte Konstantstromversorgung besitzt, in einem höheren Temperaturbereich einsetzbar ist und auch Treiber für die Softwareanbindung vorhanden sind. Lediglich die Anbindung an das Analysesystem mittels USB-Schnittstelle lässt es für die Anwendung im Mikrofonarray ausfallen, da die Kabellänge bei USB auf wenige Meter begrenzt ist und der Einsatz in störungsreicher Umgebung nur bedingt möglich ist.

4.6 Setup 6

Dieses Setup besteht aus einem 8-kanaligem IOtech ZonicBook/618E Interface plus einen Kanalerweiterungsboard auf 16 Kanäle. Die Module besitzen Spannungseingänge und eine integrierte Konstantstromversorgung. Die Anbindung an das Hostsystem erfolgt mittels Gigabit-Ethernet.

Abtastrate	bis 100 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	vorhanden
Temperaturbereich	+5 °C – +40 °C
Host-Anbindung	Ethernet
Treiber	Windows
Verkabelungsaufwand	1 x Ethernetkabel
Preis (Stand: 12/2010)	15000 €

Tabelle 4.6: Daten Setup 6

Fazit

Damit alle technischen Anforderungen erfüllt sind, müsste der Temperaturbereich noch mit einer entsprechenden Klimatisierung angepasst werden. Durch den hohen Preis ist an einen späteren Einsatz in einem Produkt allerdings nicht zu denken.

4.7 Setup 7

Setup 7 wurde aus der NI Compact DAQ Serie zusammengestellt und besteht aus vier NI-9234 Modulen mit jeweils vier Konstantstromversorgten Spannungseingängen, die auf einem NI DAQ-9188 Modul montiert sind. Die Anbindung an das Hostsystem erfolgt über Gigabit-Ethernet.

Abtastrate	bis 51.2 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	vorhanden
Temperaturbereich	$-40\,^{\circ}\mathrm{C} - +70\,^{\circ}\mathrm{C}$
Host-Anbindung	Ethernet
Treiber	Windows
Verkabelungsaufwand	1 x Ethernetkabel
Preis (Stand: 12/2010)	7500 €

Tabelle 4.7: Daten Setup 7

Fazit

Dieses System entspricht genau den technischen Anforderungen, der Preis ist für ein späteres Produkt allerdings sehr hoch.

4.8 Setup 8

Setup 8 besteht dem 16-kanaligen Hummingbird II Akquisition System des deutschen Herstellers HGL Dynamics. Das Akquisitionssystem bietet Konstantstromversorgung und eine Abtastrate von 200 kHz pro Kanal. Es wurde als Datenlogger konzipiert und beinhaltet eine SSD, auf welcher die Daten aufgezeichnet werden. Die Daten können anschließend über Gigabit-Ethernet an ein Hostsystem weitergeleitet werden.

Fazit

Mit einem Gewicht von ca. 4 kg ist das System wohl das leichteste am Markt und könnte auch in das Mikrofonarray eingebaut werden. Für die Anbindung ans Analysesystem müsste eine eigene Schnittstelle mit dem Protokoll des Herstellers implementiert werden, alle dafür notwendigen Informationen würden vom Hersteller zur Verfügung gestellt werden. Zusätzlich müsste eine, nicht im Preis enthaltene Klimatisierung vorgesehen werden.

Abtastrate	bis 200 kHz
Auflösung / Wortbreite	24 Bit
Konstantstromversorgung	vorhanden
Temperaturbereich	0 °C – +55 °C
Host-Anbindung	Ethernet
Treiber	keiner
Verkabelungsaufwand	1 x Ethernetkabel
Preis (Stand: 12/2010)	7000 €

Tabelle 4.8: Daten Setup 8

4.9 Zusammenfassung der Marktrecherche

Es sind einige Geräte bzw. Systeme am Markt erhältlich, die den technischen Anforderungen dieses Projektes genügen, jedoch ist in diesen Fällen der Preis für eine Verwendung in einem zukünftigen Produkt viel zu hoch. Ist der Preis entsprechend niedrig, müssen bei den technischen Anforderungen Abstriche gemacht werden.

Aus den Erfahrungen der Marktrecherche und den, bei den zusammengestellten Systemen der verschiedenen Hersteller verwendeten Hardwarekonzepten wurde nun ein eigenes Konzept für ein Datenakquisitionssystem entwickelt, das sowohl den technischen, als auch den wirtschaftlichen Anforderungen genügt, entwickelt.

Kapitel 5

Realisierung des eigenen Systems

Bei allen im vorigen Kapitel zusammengestellten kommerziell erhältlichen Datenakquisitionssystemen erfüllte kein System alle Anforderungen. Aus diesem Grund wurde in Zusammenarbeit mit dem StartUp Unternehmen xFace ein Hardwarekonzept erarbeitet und entwickelt. xFace entwickelt digitale Hardware für eingebettete Systeme im Bereich Kommunikationstechnik.

5.1 Datenakquisitionssystem

Das in Abbildung 5.1 dargestellte Konzept für das Datenakquisitionssystem wurde gemeinsam mit der Firma xFace entwickelt und besteht aus den Komponenten, die im Folgenden beschriebenen werden.

Mikrofonboard

Die Mikrofonboards bestehen aus vier konstantstromversorgten IEPE-Mikrofoneingängen (Integrated Electronics Piezo Electric), Signalverstärker und anschließender A/D-Wandlung samt dafür notwendiger Signalaufbereitung. Jeweils zwei Mikrofonkanäle werden an einem

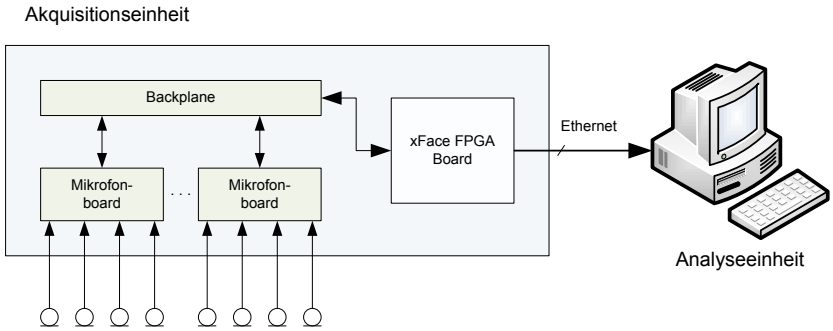


Abbildung 5.1: Schematische Darstellung des Hardware Aufbaus

Stereo A/D-Wandler zusammengefasst. Die Mikrofonboards werden über eine Steckverbindung mit der Backplane verbunden.

Backplane

Die Backplane stellt die Verbindung zwischen dem xFace FPGA Board und den Mikrofonboards her. Alle Signale vom und an das FPGA Board laufen über die Backplane. Des weiteren ist sie für die Spannungsversorgung bzw. -Verteilung an die Mikrofonboards verantwortlich. An die Backplane können bis zu 8 Mikrofonboards angeschlossen werden.

xFace FPGA Board

Beim FPGA Board handelt es sich um ein SP601 Evaluation Board von Xilinx. Es beinhaltet das FPGA, Speicher, sämtliche benötigte Schnittstellen und die digitale Hardware von xFace. Die digitale Hardware liefert die Taktsignale für die A/D-Wandler auf den Mikrofonboards und nimmt deren digitale Datenströme entgegen, wandelt die empfangenen Daten in UDP-Datenpakete um und sendet diese via Gigabit Ethernet an die Analyseeinheit.

Analyseeinheit

Die Analyseeinheit besteht aus einem normalen PC mit einer Gigabit-Ethernetschnittstelle, welche den UDP-Datenstrom vom FPGA empfängt und anschließend die Daten für die weitere Verarbeitung zur Verfügung stellt.

5.2 Spezifikationen

In diesem Abschnitt werden alle notwendigen Spezifikationen für das zu entwickelnde Akquisitionssystem festgelegt. Beginnend von den Umgebungsbedingungen, den eingesetzten Mikrofonen über die zu erwartenden Schallpegel, Schnittstellen zwischen den einzelnen Teilen des Akquisitionssystems bis hin zu den Übertragungsprotokollen bilden diese Spezifikationen die Grundlage für die weitere Entwicklung.

5.2.1 Umgebungsbedingungen

Das System soll, wie bereits in Kapitel 3.3.1 für den Einsatz auf Autobahnen konzipiert werden. Die Datenaqkuisitionseinheit muss für einen erweiterten Temperaturbereich von $-20\,°\mathrm{C} - +70\,°\mathrm{C}$ ausgelegt und gut gegen elektromagnetische Störungen geschützt werden.

5.2.2 Mikrofone und Mikrofonsignale

Bei den bisherigen Messungen mit dem Mikrofonarray wurden Messmikrofone verwendet. Diese oder ähnliche konstantstromversorgte Mikrofone sollen als Vorgabe auch in Zukunft im Mikrofonarray eingesetzt werden können.

Für den Einsatz auf Autobahnen und Schnellstraßen kann der Schalldruckpegel in einem Bereich von $50\,\mathrm{dB(SPL)} - 130\,\mathrm{dB(SPL)}$ angenommen werden. Dieser liegt innerhalb des

vom Mikrofon abbildbaren Schalldruckpegels, wodurch für die Ermittlung des Ausgangsspannungsbereiches des Mikrofons der Störspannungsabstand und der SPL Spitzenwert verwendet werden können.

$$L_{p\ dB(SPL)} = 20 \cdot \log \frac{p}{p_0}. \tag{5.1}$$

Als Bezugsschalldruck werden $p_0 = 20\,\mu\text{Pa}$ angenommen. Daraus ergibt sich nach dem Umformen der Schalldruck p zu

$$p = p_0 \cdot 10^{\frac{L_{p\ dB(SPL)}}{20}} \tag{5.2}$$

Für den Schalldruckpegelbereich von $30\,\text{dB(SPL)} - 130\,\text{dB(SPL)}$ ergeben sich die zugehörigen Schalldrücke mit:

$$p_{min} = p_0 \cdot 10^{\frac{30}{20}} = 0.63245\,\text{mPa}$$
$$p_{max} = p_0 \cdot 10^{\frac{130}{20}} = 63.2455532\,\text{Pa}$$

Da es sich beim Schalldruck (p_{min}, p_{max}) um Effektivwerte handelt, berechnen sich die Effektivwerte der Mikrofonausgangsspannung zu

$$U_{min\ \text{eff}} = p_{min} \cdot 100\frac{mV}{Pa} = 63.24555\,\mu\text{V}$$
$$U_{max\ \text{eff}} = p_{max} \cdot 100\frac{mV}{Pa} = 6.324555\,\text{V}$$

und die dazugehörigen Scheitelwerte der Mikrofonausgangsspannungen zu

$$\hat{U}_{min} = \sqrt{2} \cdot U_{min\ \text{eff}} = 89.4427\,\mu\text{V}$$
$$\hat{U}_{max} = \sqrt{2} \cdot U_{max\ \text{eff}} = 8.94427\,\text{V}$$

Es sind also Mikrofonausgangsspannungen im Bereich von $\sim 90\,\mu\text{V} - 9\,\text{V}$ zu erwarten.

5.2.3 Mikrofonkanal

Der Mikrofonkanal ist im Aufbau an den Mikrofonkanal des in Abschnitt 3.1.2 beschriebenen NI Akquisitionssystem angelehnt und besteht aus folgenden Teilen: Konstantstromversorgung, Anzeige zur Sensorkontrolle, Verstärker und A/D-Wandler mit entsprechender Signalaufbereitung. Abbildung 5.2 zeigt den schematischen Aufbau eines Mikrofonkanals.

Konstantstromversorgung

Die Verwendung von Konstantstrommikrofonen bedingt natürlich eine entsprechende Konstantstromversorgung.

IEPE steht für "Integrated Electronics Piezo Electric" und ist ein verbreiteter Standard für das Ausgangssignal von Mikrofonen. Die im Mikrofon integrierte Schaltung wandelt das hochimpedante und störungsempflindliche Ladungssignal des Piezoelements in ein Spannungssignal mit niedriger Impedanz um. Um die im Sensor integrierte Elektronik mit Energie zu versorgen, wird ein Konstantstrom auf der Signalleitung eingeprägt und gleichzeitig von der nachfolgenden Signalverarbeitung entkoppelt. Für die Energieversorgung und

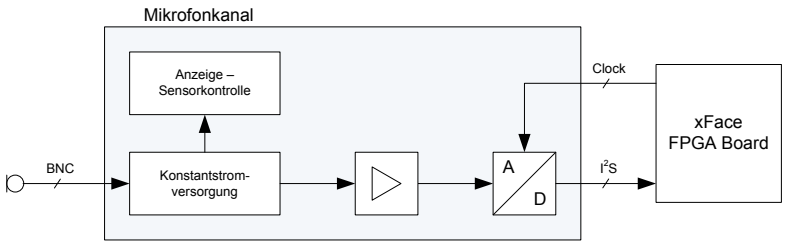


Abbildung 5.2: Schematische Darstellung eines Mikrofonkanals

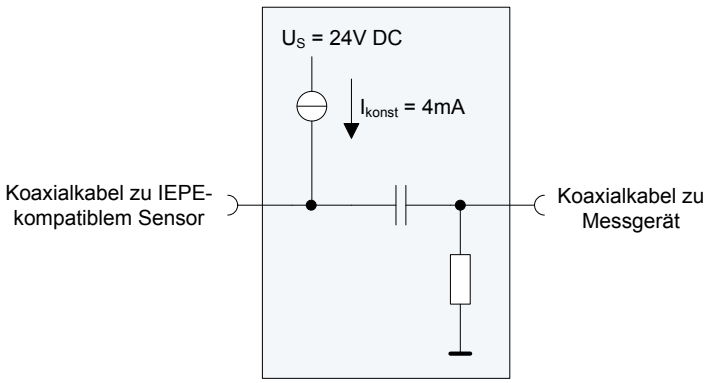


Abbildung 5.3: Prinzipschaltbild des Konstantstromversorgungsmoduls

Messsignalübertragung wird also dieselbe Signalleitung verwendet. Abbildung 5.3 zeigt ein Prinzipschaltbild des Konstantstromversorgungsmoduls. Durch Einprägen eines Stromes entsteht am Sensor eine positive Gleichspannung, die je nach Hersteller zwischen 5 V und 14 V liegt. Um diese Arbeitspunktspannung schwingt das Sensorsignal [7].

Für die selbst zu entwickelnde Konstantstromversorgung wird die Versorgungsspannung mit $U_S = 24\,\mathrm{V}$ und der einzuprägende Konstantstrom mit $I_{konstant} = 4\,\mathrm{mA}$ festgelegt. Da das Messmikrofon einen Übertragungsbereich von $20\,\mathrm{Hz} - 20\,\mathrm{kHz}$ hat, wird die untere Grenzfrequenz mit 20 Hz festgelegt.

Sensorkontrolle

Für die Überwachung des Konstantstromkreises soll eine dreistufige LED-Kontrolle wie beim M28 Modul mit folgenden Zuständen implementiert:

- LED aus: Kein Sensor angeschlossen oder der Stromkreis ist unterbrochen
- LED gelb: Sensorspannung liegt im Bereich zwischen 1 V und 24 V
- LED rot: Sensorspannung liegt unter 1 V oder der Sensor ist kurzgeschlossen

Signalverstärker

Da die errechnete minimale Ausgangsspannung der verwendeten Mikrofone bei $\sim 90\,\mu V$ liegt und nicht immer der gesamte Dynamikbereich von $\sim 80\,dB$ genutzt wird, soll ein Signalverstärker mit den Verstärkungsfaktoren 1, 10 und 100 implementiert werden. Die Verstärkungsfaktoren sollen in der Schaltung nachträglich noch geändert werden können. Es soll ein möglichst rauscharmer Operationsverstärker für die Implementierung des Verstärkers verwendet werden.

A/D-Wandler

Mit dem A/D-Wandler sollen Abtastraten von $48\,kHz - 192\,kHz$ bei einer Wortbreite von 24 Bit möglich sein. Der A/D-Wandler muss im Slave-Modus vom FPGA gesteuert arbeiten können und die digitalisierten sollen im I^2S-Format am Ausgang ausgegeben werden. Der THD+N Wert des Wandlers sollte als Vorgabe $> 100\,dB$ sein.

5.2.4 Spezifikation der xFace FPGA Board Datenübertragung

Das FPGA Board soll die digitalen Audiodaten von 32 Kanälen simulatan empfangen und diese Daten anschließend via Gigabit Ethernet an das Analysesystem übertragen.

Datenübertragung A/D-Wandler - FPGA Board

Die Datenübertragung zwischen den A/D-Wandlern und dem FPGA Board erfolgt über eine serielle I^2S Schnittstelle, wobei das FPGA hier als Master fungiert. Das FPGA Board stellt die für die A/D-Wandlung und für die Datenübertragung notwendigen Taktsignale zur Verfügung. Eine entsprechende Pufferung der Takt- bzw. Bussignale muss implementiert werden.

Datenübertragung FPGA Board - Analyse PC

Die xFaceStream Schnittstelle im FPGA wandelt die empfangenen digitalen Audiodaten in UDP Pakete um und überträgt diese dann via Gigabit Ethernet an dem Analyse PC. Hierfür ist das gemeinsam von JR und xFace entwickelte xFaceStream Protokoll vorgesehen. Das Protokoll verwendet eine fixe Größe der Nutzdaten von 1024 Byte. Dies führt zu einer maximalen Anzahl von 256 Samples bei 32 Bit Wortbreite in einem Datenpaket.
Der strukturelle Aufbau eines xFaceStream Pakets ist in Tabelle 5.1 dargestellt.

Das Darstellungsformat der Nutzdaten ist in Little Endian Format, was eine direkte Verarbeitung der Daten in x86-basierten Systemen erlaubt. Das 24 Bit Audiosample ist im 32 Bit Wort links ausgerichtet, was bedeutet, dass das LS-Byte keine Audiodaten beinhaltet.

Byte	Header	Datenfelder
$0-13$ (14 Byte)	Ethernet Header	-
$14-33$ (20 Byte)	IP Header	-
$34-41$ (8 Byte)	UDP Header	2 Byte Quellport (0xFACE) 2 Byte Zielport (0xFACE) 2 Byte Länge (1042) 2 Byte Checksumme
$42-51$ (10 Byte)	xFaceStream Header	1 Byte Port (0x00) 1 Byte Typ (0x5B: 32 Kanäle, 32 Bit, VZ) 4 Byte time_h: Sekunden, Little Endian 4 Byte time_l: Sample, Little Endian
$52-1075$ (1024 Byte)	256 Audio Samples	$256 \cdot 4$ Byte signed PCM, Little Endian, links ausgerichtetes 24 Bit Audiosample

Tabelle 5.1: xFaceStream Protokoll für Audiodatenübertragung

Wenn ein xFaceStream Paket generiert wird, wird jeweils ein 64 Bit Zeitstempel generiert. Der 64 Bit Zeitstempel wird in jeweils zwei 32 Bit Worte aufgespalten. Im Datenfeld *time_h* wird die am FPGA eingestellte Zeit in Sekunden, im Datenfeld *time_l* wird die Samplenummer an das Hostsystem übertragen. Bei einer Abtastrate von 48 kHz wird die Samplenummer also von $0-47999$ gezählt. Wird das FPGA nun mit z.Bsp.: 32 Kanälen betrieben, so finden in einem Datenpaket $256/32 = 8$ Audiosamples Platz, d.h. die Samplenummer wird bei jedem versendeten Paket um 8 erhöht. Soll nun jedes Sample mit einem genauen Zeitstempel versehen werden, so muss dies am Analysesystem erfolgen.

Durch die Verwendung eines Zeitstempels können Paketverlust am Analysesystem sehr einfach detektiert und entsprechend darauf reagiert werden.
Mit folgenden Datenraten muss auf der Netzwerkkarte gerechnet werden: Mit einer Nutzdatenlänge von 256 Audiosamples und bei Übertragung von 32 Audiokanälen ergibt sich eine UDP-Datenpaketrate von

$$DP = \frac{\text{Anzahl Audiosamples pro Datenpaket}}{\text{Anzahl Audiokanäle}} = \frac{256}{32} = 8\,\text{Datenpakete/s} \tag{5.3}$$

Bei einer Gesamtdatenpaketlänge von 1075 Byte und einer Abtastrate von $f_S = 48\,\text{kHz}$ ergibt sich die benötigte Bandbreite zu:

$$B_{Ethernet} = \frac{f_S}{DP} \cdot 1075 \cdot 8 = \frac{48000}{8} \cdot 1075 \cdot 8 = 51.6\,\text{MBit/s} \tag{5.4}$$

In Tabelle 5.2 sind Bandbreiten für verschiedene mögliche Übertragungskonfigurationen aufgelistet. Mit diesem Übertragungssystem sollten also theoretisch auch 64 Kanäle bei einer Abtastrate von 192 kHz möglich sein.

Abtastrate $[kHz]$	Anzahl Audiokanäle	Bandbreite $\left[\frac{MBit}{s}\right]$
48	32	51.6
96	32	103.2
192	32	206.4
48	64	103.2
96	64	206.4
192	64	412.8

Tabelle 5.2: Bandbreitenberechnung für die Datenübertragung per Ethernet

5.3 Mikrofonboard

Wie bereits in der Systembeschreibung in Kapitel 5.1 erwähnt, besteht ein Mikrofonboard aus vier konstantstromversorgten IEPE-Mikrofoneingängen. Jeder Mikrofoneingang hat eine eigene Spannungsversorgung, Konstantstromversorgung und Signalverstärker mit Signalaufbereitung für den A/D-Wandler. Es werden jeweils zwei Kanäle auf einem Stereo A/D-Wandler zusammengefasst, welcher mittels I^2S-Bus über die Backplane mit dem xFace FPGA Board verbunden ist. Abbildung 5.4 zeigt den schematischen Aufbau des 4-kanaligen Mikrofonboards.

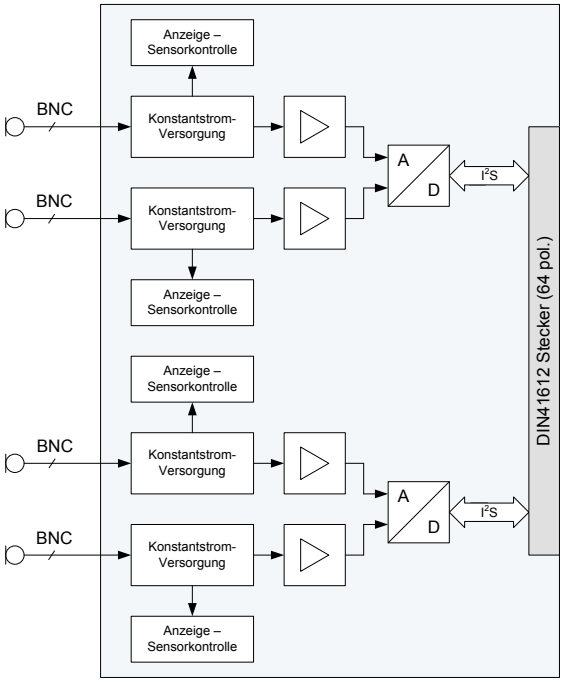


Abbildung 5.4: Schematische Darstellung des Mikrofonboards

In diesem Kapitel werden die einzelnen Teile des entwickelten Mikrofonboards genauer betrachtet, elektronische Schaltungen entworfen und dimensioniert, den Vorgaben entsprechende Bauteile ausgesucht bis letztendlich die Gesamtschaltung, das Platinen-Layout und die Bauteilliste für das Mikrofonboard entstehen.

5.3.1 Konstantstromversorgung

Wie bereits in Abschnitt 5.2.3 festgelegt, soll die Konstantstromversorgung für einen Konstantstrom von 4 mA bei einer Versorgungsspannung von 24 V dimensioniert werden.

Dimensionierung der Konstantstromversorgung

Nach eingehender Recherche wurde LM234 IC von STMicroelectonics als Konstantstromquelle ausgesucht. Dieser IC entspricht den Anforderungen an Versorgungsspannung, einzustellendem Konstantstrom und Temperaturbereich. Genauere Daten können dem Datenblatt [8] entnommen werden. Die folgende Dimensionierung stammt aus dem Datenblatt des IC's und wurde den eigenen Anforderungen entsprechend angepasst. Die genaue Herleitung der einzelnen Teile kann dort nachgelesen werden.

Da die Konstantstromquelle in einem relativ großen Temperaturbereich eingesetzt werden soll, wird die im Datenblatt angegebene Schaltung zur Kompensation der Temperaturdrift verwendet, die in Abbildung 5.5 dargestellt ist.

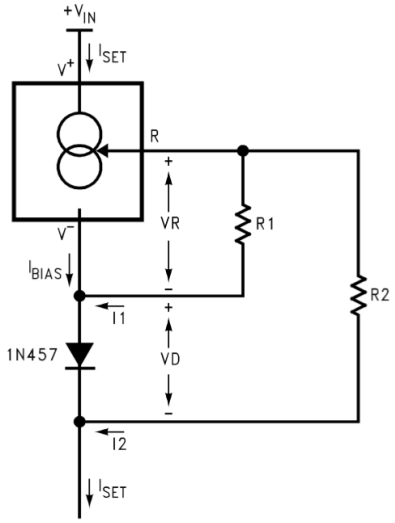


Abbildung 5.5: Blockschaltbild der temperaturkompensierten Konstantstromquelle [8]

Bei einem einzustellendem Konstantstrom von $I_{SET} = 4\,\text{mA}$ werden die Widerstände laut Datenblatt wie folgt berechnet:

$$I_{SET} = \frac{0.134V}{R_1} \tag{5.5}$$

$$R_1 = \frac{0.134}{I_{SET}} = \frac{0.134}{0.004} = 33.5\,\Omega \tag{5.6}$$

Da der Widerstandswert von $33.5\,\Omega$ nicht im Handel erhältlich ist, wird der nächstgelegendste Wert von $R_1 = 33.2\,\Omega$ gewählt. R_2 berechnet sich mit

$$R_2 \approx 10 \cdot R_1 = 10 \cdot 33.2 = 332\,\Omega \tag{5.7}$$

Die vom Hersteller empfohlene Diode 1N457 ist mittlerweile nicht mehr im Handel erhältlich und wird durch die BAS16 von Fairchild Semiconductors ersetzt, die den Spezifikationen der empfohlenen Diode entspricht.

Dimensionierung der Spannungsversorgung für die Konstantstromversorgung

Durch die Regelung des Konstantstromes mit dem LM234 IC kann es durch den Regelvorgang zu Belastungen bzw. Schwankungen der Versorgungsspannung kommen. Damit sich diese Schwankungen nicht direkt auf die Versorgung der Konstantstromquellen aller anderen Kanäle auswirkt, wird für jeden einzelnen Kanal eine eigene Versorgungsspannungsregelung implementiert. Hierbei ist eine gute "Ripple Rejection" also eine gute Unterdrückung von Versorgungsspannungsschwankungen gefordert.

Die Wahl fiel auf den LM317LM Spannungsregler von National Semiconductor. Dieser Spannungsregler weist, bei entsprechender Beschaltung, eine Ripple Rejection Ratio von typischerweise 80 dB auf, was für die Anwendung als ausreichend erscheint und entspricht auch sonst den Anforderungen an den Temperatur- und Versorgungsspannungsbereich.

Die Dimensionierung der in Abbildung 5.6 dargestellten Schaltung ist dem Datenblatt [9] entnommen und wurde für diese Anwendung entsprechend angepasst.

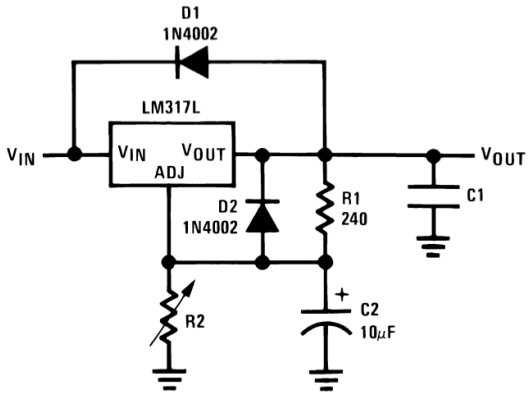


Abbildung 5.6: Blockschaltbild der Versorgungsspannungsregelung der Konstantstromquelle [9]

Die Ausgangsspannung U_{OUT} der Schaltung berechnet sich mit

$$U_{OUT} = U_{REF} \cdot \left(1 + \frac{R_2}{R_1}\right) + I_{ADJ} \cdot R_2. \tag{5.8}$$

I_{ADJ} wird auf kleiner 100 μA geregelt, deshalb kann dieser Term für die Anwendung vernachlässigt werden bzw. mit 100 μA angenommen werden. Die Referenzspannung U_{REF} wird vom IC selbst erzeugt und beträgt $U_{REF} = 1.25\,\text{V}$. Die Ausgangsspannung wird, wie bereits in Abschnitt 5.2.3 beschrieben mit $U_{OUT} = 24\,\text{V}$ festgelegt, der Widerstand mit $R_1 = 240\,\Omega$ wie im Datenblatt angegeben angenommen.

Der Widerstand R_2 berechnet sich nach Umformung von Gleichung 5.8 zu:

$$R_2 = \frac{V_{OUT} - V_{REF}}{\frac{V_{REF}}{R_1} + I_{ADJ}} = \frac{24 - 1.25}{\frac{1.25}{240} + 100\ \cdot 10^{-6}} = 4286.714\,\Omega \tag{5.9}$$

$$R_2 = 4300\,\Omega \text{ ... gewählt}$$

Die in der Schaltung abgebildeten Schutzdioden müssten bei Ausgangsspannungen von $U_{OUT} < 25\,\text{V}$ eigentlich nicht verwendet werden, da hierfür die eingebauten Schutzbeschaltung im IC ausreichen würde. Weil aber der durch die Dioden entstehende Schaltungsaufwand minimal ist, werden sie trotzdem verwendet. Diode D_1 verhindert, dass sich bei kurzgeschlossenem Eingang des Spannungsreglers der Kondensator C_1 nicht über den IC entlädt, die Diode D_2 verhindert eine Entladung des Kondensators C_2 über den IC bei kurzgeschlossenem Ausgang des Spannungsreglers.

Der Hersteller garantiert einen Ausgangsstrom von 100 mA bei den angegebenen Spezifikationen, was zur Versorgung der Konstantstromquelle bei einem eingestellten Konstantstrom von $I_{SET} = 4\,\text{mA}$ und für die Versorgung der Sensorkontrolle auf jeden Fall ausreichend sein sollte.

5.3.2 Anzeige - Sensorkontrolle

Die in Abschnitt 5.2.3 spezifizierte Sensorkontrolle wird mit einer Komparatorschaltung realisiert, welche eine 2-farbige LED ansteuert. Die Spannungsversorgung der Komparatorschaltung und der LED erfolgt über die Spannungsversorgung der Konstantstromquelle. Als Komparator IC wird der LM224DG von Texas Instruments (TI) eingesetzt, bei dem vier Komparatoren platzsparend in einem Gehäuse zusammengefasst sind.

5.3.3 Signalverstärker

Beim Signalverstärker sollen, wie in 5.2.3 erläutert, die Verstärkungsfaktoren 1, 10 und 100 implementiert werden. Dies wird mit einem als Nichtinvertierenden Verstärker geschalteten Operationsverstärker realisiert.

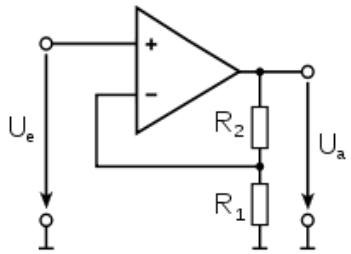


Abbildung 5.7: Schaltung eines Nichtinvertierenden Verstärkers

Der Verstärkungsfaktor errechnet sich mit

$$v = 1 + \frac{R_2}{R_1} \tag{5.10}$$

Für die drei Verstärkungsfaktoren wird der Widerstand R_1 mit Hilfe eines steckbaren Jumpers variiert. Da sich das thermische Widerstandsrauschen im Rückkopplungszweig sehr stark auf das Gesamtrauschverhalten der Schaltung auswirkt, sollten die Widerstandswerte des Rückkopplungswiderstandsnetzwerkes möglichst klein gehalten werden. Um den

virtuellen Nullpunkt im Rückkopplungsnetzwerk möglichst störungsfrei zu halten wird die Jumperleiste zwischen R_1 und Ground platziert. Tabelle 5.3 zeigt die letztendlich verwendeten Widerstandswerte für die verschiedenen Verstärkungsfaktoren.

Verstärkungsfaktor	R_2	R_1
1	1 kΩ	offen
10	1 kΩ	110 Ω
100	1 kΩ	10 Ω

Tabelle 5.3: Berechnete und gerundete Widerstandswerte für Nichtinvertierenden Verstärker

Für die Implementierung der Schaltung wurde der Operationsverstärker OPA1611 von TI gewählt. Dieser Operationsverstärker hat laut Datenblatt einen THD+N Wert kleiner −136 dB, also ein ausgezeichnetes Rauschverhalten und sehr geringe Verzerrungen. Genaue Spezifikationen können dem Datenblatt entnommen werden [10].

Simulation

Die Schaltung wurde in TINA-TI Spice für die drei verschiedenen Verstärkungsfaktoren mit den entsprechenden Widerstandswerten simuliert. R_S wurde für alle Verstärkungsfaktoren mit 100 Ω angenommen. Die für die Simulation verwendete Schaltung ist in Abbildung 5.8 dargestellt. Als Eingangsspannung wurde eine sinusförmige Spannung von 1 V gewählt, das Simulationsmodell für den Operationsverstärker wird vom Hersteller zur Verfügung gestellt.

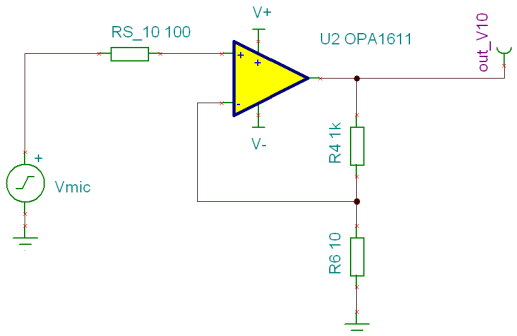
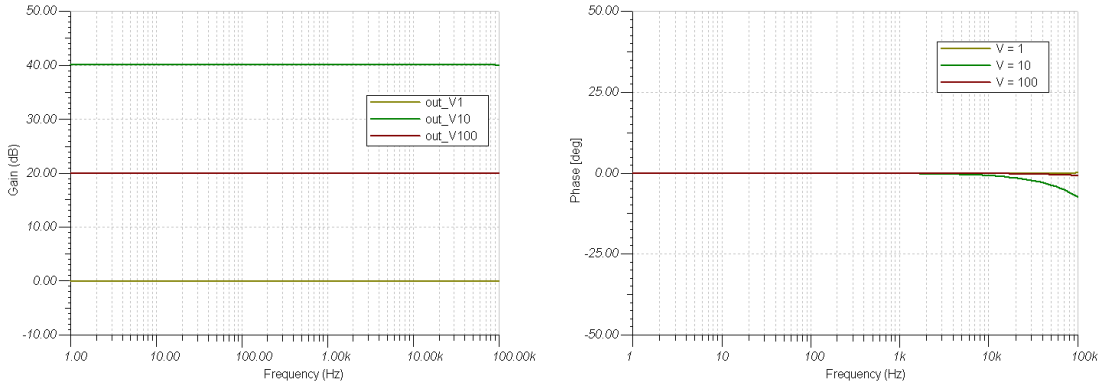


Abbildung 5.8: Simulationsschaltung des Signalverstärkers

Das Übertragungsverhalten kann mit der Simulation relativ gut abgebildet werden, die Simulation des Rauschverhaltens dagegen ist nicht wirklich aussagekräftig, da dieses sehr stark vom Platinenlayout und der Bauteilauswahl abhängig ist und diese Parameter in der Simulation nicht berücksichtigt wurden.
In Abbildung 5.9 ist das simulierte Übertragungsverhalten der Schaltung für verschiedene Verstärkungsfaktoren dargestellt.



(a) Amplitudengang (b) Phasengang

Abbildung 5.9: Simulierte Übertragungsfunktion des OPA1612 als Nichtinvertierender Verstärker für verschiedene Verstärkungsfaktoren

5.3.4 A/D-Wandler

Die Wahl des geeigneten A/D-Wandler fiel wiederum auf ein Produkt von TI, nämlich den PCM4202 IC. Dabei handelt es sich um einen 24 Bit Stereo A/D-Wandler mit differentiellen analogen Spannungseingängen. Als Wandlungsprinzip kommt ein $\Delta\Sigma$-Modulator zum Einsatz. Die digitalisierten Daten können über eine serielle I^2S-Schnittstelle zum Master-System übertragen werden. Des weiteren bietet der IC einen integrierten digitalen Hochpassfilter, mit welchem ein möglicherweise verbliebener DC-Offset aus dem Signal gefiltert werden kann. In Tabelle 5.4 ist ein kurzer Auszug aus dem Datenblatt aufgelistet, die vollständigen technischen Daten können unter [11] nachgelesen werden.

Abtastrate		$32\,\text{kHz} - 216\,\text{kHz}$
THD+N	typ.	$< -103\,\text{dB}$
Dynamik (A-gewichtet)	typ.	$118\,\text{dB}$
Kanaltrennung	typ.	$120\,\text{dB}$
Analoge Eingangsspannung	differentiell	$6\,\text{V}_{\text{PP}}$
Eingangsimpedanz	unsymmetrisch	$10\,\text{k}\Omega$
Max. Stromaufnahme	Analogteil	$55\,\text{mA}$
	Digitalteil	$10\,\text{mA}$
Verlustleistung	typ.	$308\,\text{mW}$
Betriebstemperatur	typ.	$-10\,°\text{C} - +70\,°\text{C}$

Tabelle 5.4: Datenblattauszug des PCM4202 A/D-Wandler von TI

Ein wichtiger Grund für die Wahl dieses A/D-Wandlers war das Vorhandensein einer Referenzimplementierung in Form des Evaluierungsmoduls PCM4202EVM und der dazugehörigen Referenzschaltung bzw. dem Referenzlayout, an welches meine eigene Implementierung angelehnt ist. Für die A/D-Wandler von TI ist des weiteren unterstützende Literatur in Form von "Application Notes" verfügbar.

Die in Kapitel 5.2.2 berechnete maximale Mikrofonausgangsspannung $\hat{U}_{max} = 8.9442\,\mathrm{V}$ muss nun auf die maximale Eingangsspannung bzw. das entsprechende Eingangsspannungsformat (differentiell) des A/D-Wandlers angepasst werden. Zur Vereinfachung der Berechnung des Skalierungsfaktors wird die maximale Mikrofonausgangsspannung mit $\hat{U}_{max} \sim 9\,\mathrm{V}$ angenommen.

Skalierungsschaltung

Damit die im Datenblatt des A/D-Wandlers angegebenen Spezifikationen erreicht werden können, wird für die Anpassung der Eingangsspannung in Pegel und Format eine geeignete Skalierungsschaltung verwendet. Die Grundlage für diese Schaltung bilden die Ausführungen im Designdokument [12] und die im PCM4202EVM Evaluierungsmodul [13] verwendete Skalierungsschaltung.

Für die Umwandlung und Skalierung des unsymmetrischen Mikrofonsignales wird der OPA1632 IC [14] von TI verwendet. Dabei handelt es sich um einen differentiellen Audio Operationsverstärker, der als Treiber für High-Performance A/D-Wandler verwendet wird. Abbildung 5.10 zeigt die im Datenblatt des PM4202 vorgeschlagene und im PCM4202EVM Evaluierungsmodul verwendete Skalierungs- und Treiberschaltung für den A/D-Wandler.

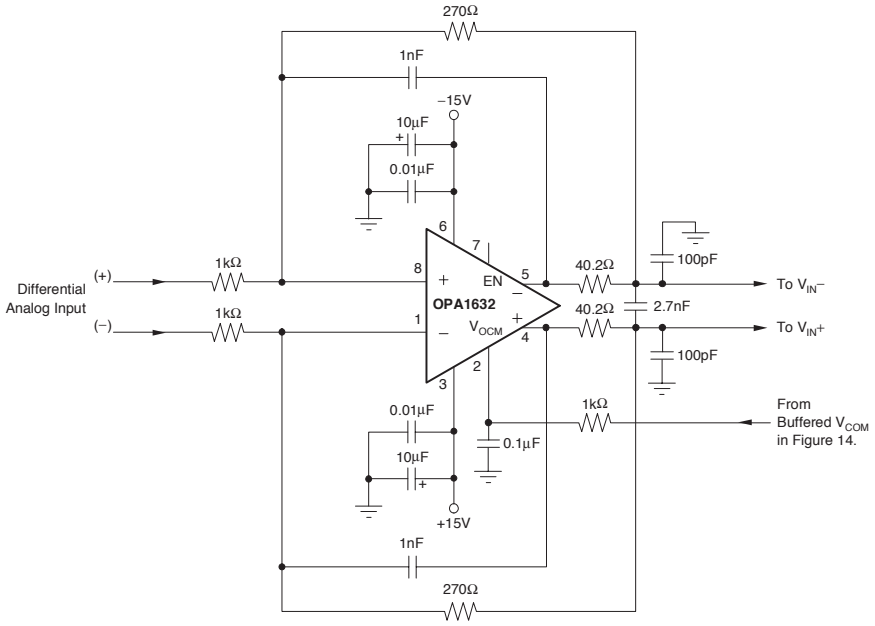


Abbildung 5.10: Skalierungs- und Treiberschaltung mit OPA1632 [11]

Der Skalierungsfaktor wird über das Widerstandsverhältnis von $\frac{R_1}{R_3}$ bzw. $\frac{R_2}{R_4}$ eingestellt. Die Widerstände in den Rückkopplungszweigen (R_1, R_2) sollten für ein gutes Rauschverhalten der Schaltung möglichst klein gehalten werden, da das durch sie verursachte Rauschen direkt an den Eingang zurückgekoppelt wird und somit das Rauschverhalten der

Gesamtschaltung wesentlich beeinflusst. Werden die im Datenblatt abgebildeten Werte von $R_3 = R_6 = 270\,\Omega$ und $R_1 = R_2 = 1\,\mathrm{k}\Omega$ verwendet, ergibt sich daraus eine Verstärkung von $v = 0.27$, was bei der Annahme von $\hat{U}_{max} = 9\,\mathrm{V}$ am Ausgang der Skalierungsschaltung eine Spannung von

$$\hat{U}_{out} = v \cdot \hat{U}_{max} = 0.27 \cdot 9 = 2.43\,\mathrm{V}$$

ergibt. Um eine gute Balance des erzeugten differentiellen Ausgangssignal zu erreichen, sollten die Widerstände R_1 - R_2 und R_3 - R_4 möglichst gleich sein. Eine Widerstandswerttoleranz von 1% wird im Datenblatt als ausreichend angegeben. Damit der Spannungseingang am A/D-Wandler voll ausgesteuert wäre, könnte der Skalierungsfaktor bis 0.33 erhöht werden, damit jedoch später noch ein AB-Vergleich der pinkompatilben A/D-Wandler PCM4202 und PCM1804 gemacht werden kann, wird der Skalierungsfaktor bei 0.27 belassen. Der AB-Vergleich sollte wegen der deutlich geringeren Stromaufnahme und somit auch einer geringeren Verlustleistung durchgeführt werden. Durch Änderung von R_3 und R_4 auf $330\,\Omega$ kann der Faktor angehoben werden.

Die in der Schaltung in Abbildung 5.10 verwendeten Widerstände R_5, R_6 und C_3 bilden ein Tiefpassfilter und einen Ladungsspeicher zum Abfangen von Spannungsspitzen am Eingang des A/D-Wandlers. Die Kondensatoren C_1 und C_2 bilden zusammen mit den Rückkopplungswiderständen ein Tiefpassfilter und werden zur Bandbreitenbegrenzung verwendet. Das unsymmetrische Eingangssignal wird an den differentiellen + Eingang des OPA1632 angeschlossen, der − Eingang wird auf Ground gelegt.

Als digitale serielle Schnittstelle wird in diesem Projekt die I^2S-Schnittstelle des A/D-Wandlers verwendet. Die Schnittstellenspezifikationen können unter [15] nachgelesen werden. Das xFace FPGA fungiert als Master für die Steuerung und die Kommunikation der A/D-Wandler am Mikrofonboard, stellt die benötigten Taktsignale für die Wandlung (SCK) und die serielle Kommunikation (BCK, WS) zur Verfügung und nimmt den digitalen Datenstrom (SD) vom Ausgang der A/D-Wandler entgegen. Damit die Taktsignale vom FPGA einer möglichst geringen kapazitiven Belastung ausgesetzt sind bzw. der digitale Datenstrom vom A/D-Wandler zum FPGA einer genügend hohen kapazitiven Belastung standhält werden diese Signale mit einem SN74ALVC Treiber IC gepuffert. Jeder Ausgang des Treibers ist in der Lage eine kapazitive Last von 50 pF zu treiben, was für diese Anwendung ausreichen sollte.

5.3.5 Gesamtschaltung

In den vorangegangen Unterkapiteln wurden die einzelnen Schaltungsteile des Mikrofonboards beschrieben und dimensioniert. In diesem Kapitel werden nun die einzelnen Schaltungsteile zu einer Gesamtschaltung zusammengefügt.

Vor der aktuellen Schaltung wurde bereits ein 2-kanaliger Mikrofonboard Prototyp gefertigt, aufgebaut und anschließend mit dem AP 2700 Messgerät der TU Graz vermessen. Aus fertigungstechnischen Gründen wurde dabei ein 2-lagiges Platinenlayout verwendet. Dabei wurde die gesamte Signalkette von Mikrofoneingang bis zum digitalen Ausgang am xFace FPGA Board hinsichtlich des Frequenzgangs und des Rauschverhaltens bei verschiedenen Samplingfrequenzen vermessen. Die Taktsignale für das Mikrofonboard bzw. der

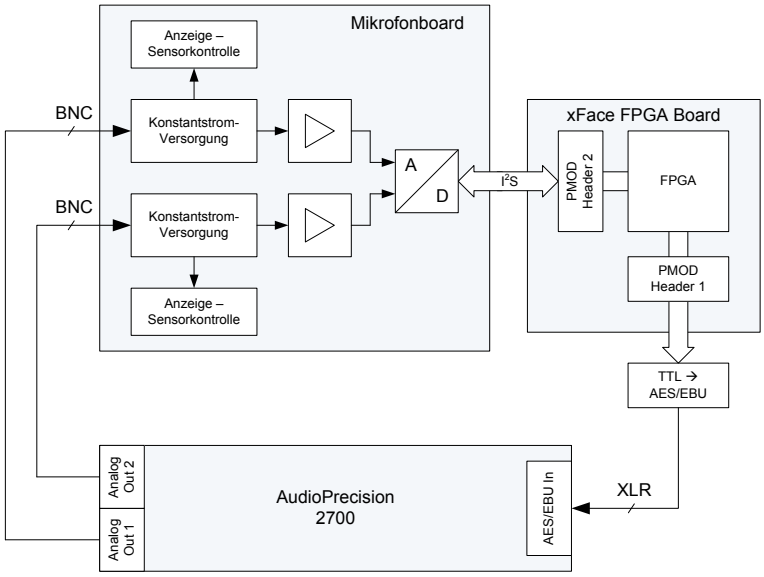


Abbildung 5.11: Schematische Darstellung des Messaufbaus bei Prototypmessungen

serielle Datenstrom an das FPGA wurden über den PMOD Header 1 am FPGA ausgegeben bzw. entgegengenommen. Die Ausgabe der digitalisierten Daten erfolgte über die SPDIF Schnittstelle, die in Kapitel 5.6.1 genauer beschrieben ist. Da das AudioPrecision Messsystem jedoch nur einen AES/EBU Eingang hat, musste für diese Tests ein zusätzlicher Pegel- und Formatwandler gebaut werden, der in Anhang B.1 zu finden ist. Abbildung 5.11 zeigt die schematische Darstellung des verwendeten Messaufbaus für die Messungen am Prototypen.

Mit dem 2-kanaligen Prototypenaufbau des Mikrofonboards konnte in der Messung bei einer Abtastrate von 48 kHz ein maximaler THD+N Wert von 95 dB bei Vollaussteuerung des A/D-Wandlers ermittelt werden. Für steigende Abtastraten verschlechterte sich dieser Wert deutlich. Für diese Messungen muss angemerkt werden, dass die Spannungsversorgung durch Labornetzteile erfolgte.

Die aktuelle Schaltung der Mikrofonboards (siehe **??**) ist eine Weiterentwicklung des Prototypenaufbaus und wurde diesmal als 4-lagiges Platinendesign ausgeführt (siehe **??**). Damit soll nun ein THD+N Wert erreicht werden, der ungefähr dem des A/D-Wandlers entspricht. Bei der Entwicklung des aktuellen Mikrofonboards wurde die Breite des Boards mit 10 cm festgelegt, damit können im Handel erhältliche Eurocard-Systeme für die Einhausung verwendet werden. Die Verbindung zur Backplane wird über einen 64-poligen DIN41612 Stecker hergestellt, über den die Platine mit den benötigten Spannungen (+27 V, ±15 V, +5 V und +3.3 V) versorgt und die digitalen Signale übertragen werden.

5.4 Backplane

Die Verbindung zwischen Mikrofonboards und FPGA Board wird mit dem Backplaneboard realisiert. Abbildung 5.12 zeigt den schematischen Aufbau des Adapterboards.

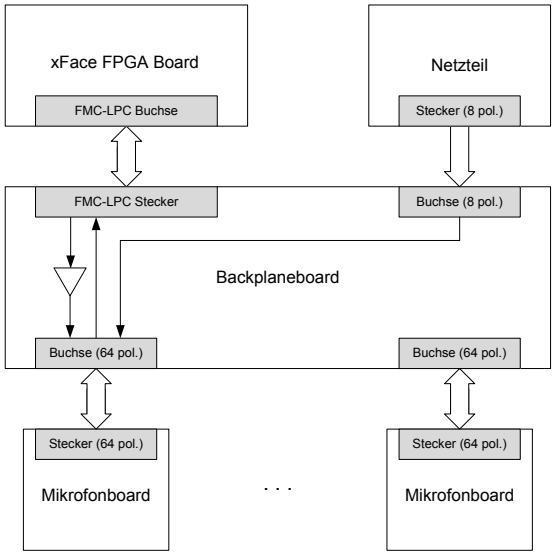


Abbildung 5.12: Schematische Darstellung des Backplaneboards

Als Steckverbindung zwischen FPGA Board und dem Backplaneboard wird ein VITA57.1 FMC-LPC Stecker verwendet. LPC steht dabei für *Low Pin Count* und bezeichnet die Anzahl an Pins (160) die für die Datenübertragung zur Verfügung stehen. Die gewählte Steckerbelegung ist in Anhang **??** zu finden. Genauere Informationen über FMC und die VITA51.1 Spezifikationen können unter [16] nachgelesen werden. Die Ankopplung der Mikrofonboards wird, wie bereits in 5.3.5 ausgeführt, eine 64-polige DIN41612 Steckverbindung verwendet. Das Netzteil wird in der ersten Prototypvariante des Backplaneboards mit Bananenbuchsen angeschlossen. Die in Abbildung 5.12 dargestellte Versorgung des Mikrofonboards mit Taktsignalen, Datenrückleitungen und Spannungen ist für alle weiteren Mikrofonboardsteckplätze simultan implementiert.

Der Schaltplan, das Layout und die dazugehörige Bauteilliste des prototypischen Backplaneboards für den Anschluss von 8 Mikrofonboards sind in Anhang **??** zu finden.

Die vom FPGA zur Verfügung gestellten Taktsignale müssen an alle Mikrofonboards weitergeleitet werden. Damit die Originaltaktsignale nicht durch die kapazitive Belastung, die durch Steckverbindungen, Leitungslängen auf der Platine und Eingangskapazitäten der mit dem Taktsignal zu versorgenden IC's entstehen verfälscht wird, werden 1:10 Clock Driver verwendet, die diese Signale puffern.

Bei dem in der Schaltung verwendeten PI49FCT32807 Clock Driver IC von Pericom kann jeder Ausgang mit bis zu 15 pF belastet werden, was für die Verteilung an die Mikrofonboards als ausreichend abgeschätzt wurde. Diese Abschätzung muss jedoch erst durch

Messungen am fertig aufgebauten Prototypen bestätigt werden. Genaue Spezifikationen des Clock Drivers können unter [17] nachgelesen werden.

5.5 Netzteil

Die Entwicklung eines entsprechenden Netzteiles ist nicht Teil dieser Arbeit. Bei allen Messungen mit aufgebauten Prototypen wurden Labornetzteile verwendet. In Tabelle 5.5 sind alle für den Betrieb des Systems notwendigen Versorgungsspannungen aufgelistet.

Spannung	Platine	Verwendung
+27 V	Mikrofonboard	Mit dem Spannungsregler werden daraus +24 V für Konstantstromquelle und Sensorkontrolle erzeugt
±15 V	Mikrofonboard	Versorgung der Signalaufbereitungs IC's
+5 V	Mikrofonboard	Versorgung Analogteil A/D-Wandler
+3.3 V	Mikrofonboard	Versorgung Digitalteil A/D-Wandler und digitaler Signalbuffer/treiber
	Backplaneboard	Versorgung digitaler Signalbuffer/treiber

Tabelle 5.5: Benötigte Versorgungsspannungen

Für einen guten SNR bzw.THD+N Wert der Mikrofonboards sollte die Spannungsversorgung der analogen Schaltungsteile möglichst stabil und störungsfrei sein. Dies ist für den Einsatzbereich unter den für die Versorgung der analogen Schaltungsteile ist eine stabile und störungsfreie Spannung notwendig.

Spannung	gemessener Strom
+27 V	26 mA
±15 V	60 mA pro Strang
+5 V	50 mA
+3.3 V	13 mA

Tabelle 5.6: Gemessene Stromaufnahme am 2-kanaligen Mikrofonboardprototyp für verschiedene Versorgungsspannungen

Die tatsächliche Stromaufnahme des aktuellen Mikrofonboards muss bei der Vermessung ermittelt werden. Dann kann die Dimensionierung des Netzteiles für das Gesamtsystem durchgeführt werden.

5.6 FPGA-Board

Das Entwickeln der digitalen Hardware auf dem FPGA ist nicht Teil dieser Arbeit, dennoch soll hier ein kurzer Überblick über die Funktionsweise gegeben werden. Die digitale Hardware wurde von der Firma xFace entwickelt. Wie bereits in Abschnitt 5.2.4 erwähnt wurde das Hardwarekonzept und das Datenübertragungsprotokoll gemeinsam von JR und xFace entwickelt.

5.6.1 Digitale Hardware

Mit der digitalen Hardware können, wie in Kapitel 5.1 bereits beschrieben bis zu 64 digitale Audiokanäle bei einer Wortbreite von 24 Bit und Abtastraten von 48- 96- und 192 kHz erfasst werden. Die Daten werden anschließend via Gigabit Ethernet an den Analyse PC gesendet. Hierfür wird das in 5.2.4 beschriebene xFaceStream Protokoll verwendet, welches einen hohen Durchsatz bei geringen Latenzen ermöglicht. Jedes Datenpaket erhält einen 64 Bit Zeitstempel aus dem die genaue Samplezeit ermittelt werden kann.

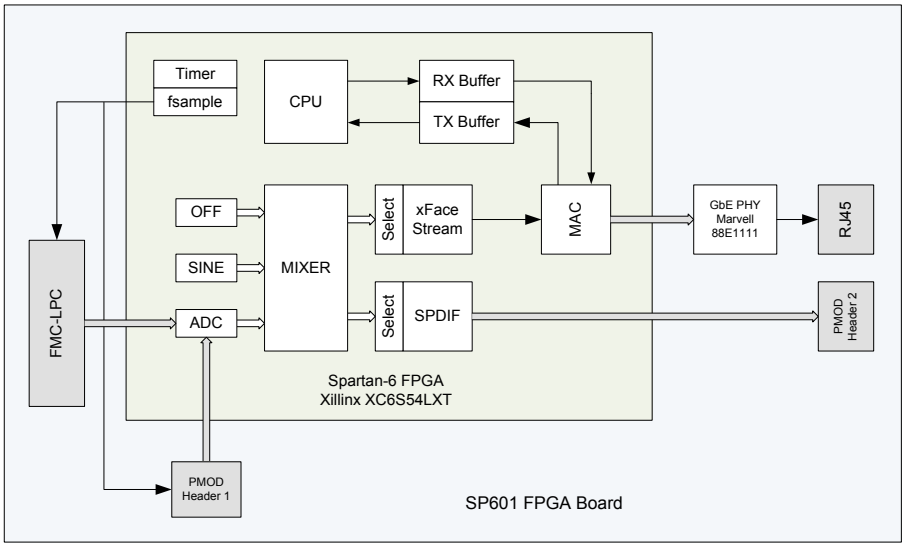


Abbildung 5.13: Schematische Darstellung der digitalen Hardware auf dem SP601 Board

Die digitale Hardware wurde auf einem Spartan-6 LX45T FPGA implementiert. Für diese Prototypapplikation wurde hierfür das SP601 Evaluation Board verwendet. Für genauere Angaben und Beschreibungen zu diesem Board wird hier auf das Dokument [18] verwiesen. Die Konfiguration des FPGA's wurde aus Verilog HDL Code synthetisiert. Bei der Synthese wird der Verilog Code in eine strukturelle Schaltung umgewandelt und anschließend mit dem Xilinx Map Tool den Spartan-6 Elementen zugeordnet. Mit Hilfe des Xilinx impact tools wird die digitale Schaltung nun per USB-Schnittstelle auf das ROM des SP601 Development Board gespeichert.

Abbildung 5.13 zeigt den schematischen Aufbau der digitalen Hardware am FPGA bzw. Evaluation Board. Alle Blöcke mit weißem Hintergrund wurden von xFace entwickelt.

Blöcke mit grauem Hintergrund stellen die benötigten physikalischen Schnittstellen des Boards dar. Im Folgenden werden einige der Blöcke im FPGA genauer erläutert.

CPU

Die Steuerungs- und Kontrollebene wird bei diesem System von einem On-Chip 32 Bit RISC Prozessor mit einem BRAM basierten RAM/ROM und einer AMBA-Peripherieschnittstelle übernommen.

xFaceStream

xFaceStream ist eine On-Chip Schnittstelle für kontinuierliche Datenströme. Sie fasst den kontinuierlichen Strom von 32 Bit Audiosamples zu Blöcken zusammen und überträgt diese Blöcke dann an die MAC-Einheit. Die Blockgröße kann hier zwischen 1 und 256 Samples variieren und ist abhängig von der Anzahl der Mikrofonkanäle.

Mixer

Mit dem Mixermodul können verschiedene Quelldatenströme auf einen Ausgangsdatenstrom geroutet werden. Alle Eingangsdatenströme müssen hierfür eine identische Kanalanzahl, Abtastrate und Phase haben. Welcher Quelldatenstrom an den Ausgang des Mixers geroutet wird, kann mittels Steuerungsprogramm auch zur Laufzeit verändert werden. Die Anzahl der Mixerkanäle kann bei der Programmsynthese mit Parametern festgelegt werden.

Datenquellen

OFF: Die Datenquelle liefert keine Daten, der entsprechende Ausgangsdatenstrom ist immer 0 und wird bei der weiteren Verarbeitung der Daten nicht mehr berücksichtigt.

SINE: Mit dem Onboard Sinus-Datenstromgenerator ist es möglich 24 Bit Sinusförmige Signale zu generieren. Dabei handelt es sich um ein 16 Bit Sinussignal, das auf 24 Bit erweitert wurde. Das LS-Byte enthält hier keine Information. Die Frequenz jedes Sinusgenerators kann über die Kontrollsoftware entsprechend eingestellt werden.

ADC: Wie bereits in den Spezifikationen in Abschnitt 5.2.4 beschrieben wurde, ist eine I^2S-Schnittstelle für den Datentransfer zwischen A/D-Wandler und digitaler Hardware vorgesehen. Dabei handelt es sich um eine Punkt-zu-Punkt Verbindung mit 4 Signalleitungen. Bei dieser Implementierung agiert das FPGA System als Master und der A/D-Wandler als Slave, was bedeutet, dass das FPGA die Taktsignale für den A/D-Wandler zur Verfügung stellt.

SPDIF

Die SPDIF Schnittstelle ist eine standardisierte digitale Audioschnittstelle. Mit diesem Modul können 2 Mixerkanäle (Kanal 0 und Kanal 1) ausgewählt und im SPDIF-Format ausgegeben werden. Physikalisch wird das SPDIF-Signal am PMOD Header 1 des FPGA

Boards unter Verwendung von TTL-Pegeln ausgegeben. Mit Hilfe der im Kapitel B.1 gebauten Hardware kann das TTL-Pegel SPDIF-Signal in ein AES/EBU-Signal umgewandelt werden. Das Timing für das SPDIF-Signal wird vom Takt des Eingangssignals abgeleitet. Diese Schnittstelle wurde prinzipiell nur für die Charakterisierung der einzelnen Kanäle der Mikrofonboards implementiert.

MAC

Das MAC Modul ist für den Empfang und Versand von Datenpaketen an das/vom Hostsystem zuständig. Eine detaillierte Beschreibung dieses Moduls kann in der xFace Dokumentation nachgelesen werden [19].

Timer fsample

Die Generierung der verschiedenen Taktsignale für die Taktung von A/D-Wandler und SPDIF-Schnittstelle wird von einem 12.288 MHz Signal eines Audioquartzes, mit Phase Locked Loop (PLL) mit Frequenzdivision -und Multiplikation abgeleitet. Die Taktung der CPU wird von einem 200 MHz OnBoard-Oszillator des SP601 Evaluation Board wiederum mit PLL und Frequenzmultiplikation und -division abgeleitet. Die genaue Funktionsweise des Timers und der Taktsignalgenerierung kann in der xFace Dokumentation nachgelesen werden [19].

5.6.2 xFace Steuerungssoftware

Mit der in ANSI-C geschriebenen Steuerungssoftware am FPGA können die Datenpfade im FPGA-System gesteuert werden. Sie ist unabhängig von der Datentransferebene und ist für die Systemkonfiguration, Systemmonitoring und den geregelten Betrieb der Datentransferebene zuständig. Zentrales Element der Steuerungssoftware ist ein 32 Bit RISC Prozessor mit Microblaze Befehlssatz.

xFaceStream Kontrollprotokoll

Für die UDP-Kommunikation mit der xFace Steuerungssoftware wurde das xFaceStream Kontrollprotokoll entwickelt und implementiert. In einem Datenpaket wird eine konstante Anzahl von 16 Anfragen transportiert. Die Antworten auf eine Anfrage haben dieselbe Größe. Mit jeder Anfrage kann ein 32 Bit Wert gelesen oder modifiziert werden.
Das Protokoll hat den in Tabelle 5.7 beschriebenen Aufbau.

Rückantworten vom FPGA-System haben denselben Aufbau wie das Kontrollprotokoll. Die Antworten haben als Befehlstyp jeweils CMD_TYPE_NONE und das 32 Bit Feld enthält die gewünschten Daten. Wird eine Schreiboperation durchgeführt, so enthält die Rückantwort den modifizierten Wert.

Byte	Header	Datenfelder
$0-13$ (14)	Ethernet Header	-
$14-33$ (20)	IP Header	-
$34-41$ (8)	UDP Header	2 Byte Quellport (0xFACE) 2 Byte Zielport (0xFACE) 2 Byte Länge (1042) 2 Byte Checksumme
$42-47$ (6)	xFaceStream Befehl Header	1 Byte Port (0x01) 1 Byte Format (0x07: 128-Bit Werte, unsigned) 4 Byte Reserve
$48-303$ (256)	16 xFaceStream Befehl	Der Befehlsaufbau ist in Tabelle 5.8 beschrieben.

Tabelle 5.7: xFaceStream Kontrollprotokoll

Byte	Datenfeld	Werte
$0-1$ (2)	Version	0xFA00
$2-3$ (2)	Befehl und Befehlstyp	0x0000: nop, 0x0010: status, 0x0020: read/write CMD_TYPE_NONE (0), CMD_TYPE_READ (1), CMD_TYPE_WRITE (2), CMD_TYPE_ADD (3), CMD_TYPE_SUB (4), CMD_TYPE_AND (5), CMD_TYPE_OR (6), CMD_TYPE_XOR (7), CMD_TYPE_SHL (8), CMD_TYPE_SHR (9)
$4-7$ (4)	Adresse	NC (0), VERSION(1), CHANNELS(2), FSAMPLE (3), MACOWN_H (4), MACOWN_L (5), IPOWN (6), MACHOST_H (7), MACHOST_L (8), IPHOST (9), UDPHOST_AUDIO (A), UDPHOST_CONF (B), STREAM_EN (C), STREAM_CNT (D), TIMEUP (E), TIME_H (F), TIME_L (10), MEMORY (11), SINE (12), MIXER (13), SPDIF (14), ENPROM (15)
$8-11$ (4)	Offset	Wird in Zusammenhang mit Adressen verwendet
$12-15$ (4)	Wert	-

Tabelle 5.8: xFaceStream Befehlsaufbau

5.6.3 Schnittstellen

Damit das FPGA-Board mit seiner Umgebung kommunizieren kann sind mehrere physikalische Schnittstellen vorgesehen. Diese sind in Abbildung 5.13 grau hinterlegt dargestellt. Auf die Netzwerkschnittstelle wird nicht näher eingegangen.

FMC-LPC Stecker

Mit dieser Schnittstelle werden Backplaneboard und FPGA-Board miteinander verbunden und so die physikalische Schnittstelle zwischen den A/D-Wandler IC's auf dem Mikrofonboard und dem FPGA hergestellt. Die A/D-Wandler arbeiten dabei im *Slave*-Modus, was bedeutet, dass alle für den Betrieb notwendigen Taktsignale vom FPGA vorgegeben werden. Auf dem SP601 Board steht hier der FMC Stecker zur Verfügung, über den diese Datenverbindung hergestellt wird. Die genaue Belegung des Steckers ist im Dokument [18] des Hersteller zu finden. Am VITA 57.1 FMC-LPC Stecker (Low Pin Count) werden die I^2S Taktsignale für die A/D-Wandler ausgegeben (SCK, BCK, WS) und die seriellen Datenströme der A/D-Wandler empfangen ($SD[0]...SD[15]$). Das Pinout der FMC-LPC Steckverbindung ist in Anhang **??** zu finden.

PMOD Header 1

An PMOD Header 1 kann für Testzwecke am Mikrofonboardprototypen eine A/D-Wandler Schnittstelle (I^2S-Schnittstelle) ausgegeben werden. Das zugehörige Pinout ist in Anhang **??** zu finden.

PMOD Header 2

Wie in 5.6.1 beschrieben wird ein PMOD Header 2 für die Ausgabe des SPDIF Signals verwendet. Das Pinout ist in Anhang **??** zu finden.

Kapitel 6

Software

Nachdem im vorigen Kapitel die entwickelte Hardware für die Datenakquisition vorgestellt wurde, beschäftigt sich dieses Kapitel nun mit der für das Projekt notwendigen Software, welche die Anbindung und die Steuerung der Hardware, sowie die Implementierung des Algorithmus beinhaltet.

6.1 Verwendete Tools, Framework

In diesem Kapitel wird die verwendete Entwicklungsumgebung und die verwendeten Tools vorgestellt, die für die Implementierung des Algorithmus gefordert bzw. notwendig waren. Die verwendeten Tools sollen auf Windows und POSIX Betriebssystemen verfügbar sein.

6.1.1 MATLAB

Der dieser Arbeit zu Grunde liegende Algorithmus wurde in MATLAB entwickelt. MATLAB ist eine Hochsprache und interaktive Entwicklungsumgebung für Algorithmenentwicklung, die Visualisierung und Analyse von Daten sowie für numerische Berechnungen und ist ein kommerzielles Produkt von *The MathWorks*, Inc. Ergänzende Toolboxen erweitern die MATLAB-Umgebung um eine Vielzahl von Funktionalitäten die in bestimmten Anwendungsbereichen gefordert sind. Diverse Parametersätze werden bei der Implementierung des Algorithmus offline mit MATLAB berechnet und die Ergebnisse in Dateien gespeichert. Zur Laufzeit des Algorithmus können dann verschiedene Parametersätze geladen werden.

6.1.2 Simulink

Der Algorithmus sollte gemäß Vorgabe in Simulink implementiert werden. Simulink ist eine Umgebung für Mehrdomänensimulation und das Model-Based Design dynamischer Systeme und Embedded Systems. Es bietet eine grafische Entwicklungsumgebung mit individuell anpassbaren Blockbibliotheken mit denen Signalverarbeitungssysteme entworfen, simuliert, implementiert und getestet werden können. Simulink ist in MATLAB integriert und ermöglicht so direkten Zugriff auf eine breite Palette von Werkzeugen zur Algorithmenentwicklung, die Analyse und Visualisierung von Simulationen. Einige weitere Hauptmerkmale sind:

- Verwaltung komplexer Entwürfe durch hierarchische Unterteilung von Modellen in Teilkomponenten
- Embedded MATLAB Funktionsblöcke zu Integration MATLAB-Algorithmen in Simulink-Modelle
- Drei verschiedene Simulationsmodi, mit denen sich Simulationen interpretiert oder mit der Geschwindigkeit von kompiliertem C-Code ausführen lassen.
- Grafischer Debugger und Profiler zur Untersuchung der Simulationsergebnisse sowie der Diagnose von Performance des Modells.
- Zugriff auf alle in MATLAB vorhandenen Funktionen zur Analyse und Visualisierung von Simulationsergebnissen sowie zur Definition von Signal-, Parameter- und Testdaten.
- Tools für Modellanalysen und -diagnosen die die Modellkonsistenz sicherstellen und Modellierungsfehler aufzeigen.

Zur Erweiterung der Funktionalität bietet Simulink die Möglichkeit selbstgeschriebenen C/C++-Code über S-Function Blöcke in das Modell einzubinden.

6.1.3 MATLAB Coder

Der MATLAB Coder generiert eigenständigen, leicht lesbaren und plattformunabhängigen C-Code aus MATLAB Code. Er unterstützt eine Teilmenge der MATLAB-Sprache. Mit dem Embedded MATLAB Block kann MATLAB Code in ein Simulink Modell eingebunden werden. Bei der Simulation wird dabei, abhängig vom Simulationsmodus, der MATLAB Code mit dem MATLAB Coder übersetzt und für die Simulation ins Modell eingebunden.

6.1.4 Simulink Coder

In Simulink erstellte Modelle können für die Codegenerierung konfiguriert und vorbereitet werden. Mit Hilfe des Simulink Coders lässt sich aus einem Modell C/C++-Code und/oder ausführbare Programme für Echtzeitsimulation erzeugen. Unterstützt werden dabei Singletasking-, Multitasking- und Mehrkern-Ausführung mit oder ohne RTOS. Der generierte Code ist auch außerhalb der MATLAB/Simulink Umgebung einsetzbar und lauffähig.

6.1.5 Python

Python ist eine objektorientierte, interpretierte höhere Programmiersprache, die mehrere Programmierparadigmen wie objektorientierte und funktionale Programmierung unterstützt. Bei diesem Projekt ist die von xFace gelieferte Kontroll- und Steuerungssoftware des FPGA in Python geschrieben. Insbesonders wurde hier auf die wxPython Library zurückgegriffen.

6.2 Simulink Implementierung

Gemäß den Vorgaben soll der Algorithmus bzw. die gesamte Software als Simulink-Modell implementiert werden. Mit Hilfe des Simulink Coders kann aus dem Modell heraus dann ein Programm erstellt werden, welches ohne MATLAB/Simulink-Framework nativ auf einem Computer lauffähig ist.

Wie bereits im Kapitel 6.1.2 erläutert kann für die Verwaltung von komplexen Entwürfen eine hierarchische Unterteilung von Modellen in Teilkomponenten erfolgen. Abbildung 6.1 zeigt den fertigen Entwurf des Simulink Modells und dessen hierarchische Unterteilung in einzelne Teilkomponenten.

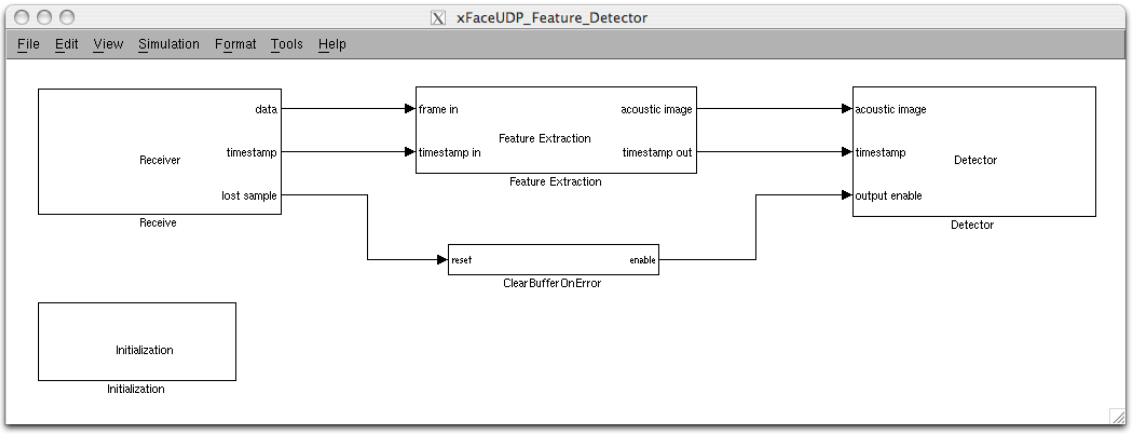


Abbildung 6.1: Fertiges Simulink Modell

Das Simulink Modell besteht aus fünf hierarchisch verschachtelten Blöcken, die im Folgenden genauer erklärt werden. Auf die in der ursprünglichen Implementierung des Algorithmus verwendeten graphischen Ausgabe des akustischen Bildes zusammen mit den Detektionsergebnissen wird aus Performancegründen verzichtet. Die Ausgabe der Detektionsergebnisse erfolgt in dieser Implementierungsversion in eine Datei.

Der in der MATLAB Implementierung für Berechnungen verwendete Datentyp *double* (Fließkommadarstellung mit 64 Bit Wortbreite) wird im Simulinkmodell durch den Datentyp *single* (Fließkommadarstellung mit 32 Bit Wortbreite) ersetzt, da bei der Verwendung des Simulink Coders nicht alle verwendeten Blöcke eine Implementierung für Double-Precision besitzen und somit mit diesen Blöcken kein C-Code erzeugt werden könnte.

Bei der Simulation des Modells wird ein Fixed-Step Solver verwendet, der nur diskrete Zeitschritte zulässt. Die dem Solver zugrunde liegende Sampletime wurde mit 1 festgelegt. Damit das Modell in als eigenständiges Programm ohne zeitliche Begrenzung läuft, wird als Startzeit der Simulation 0 und als Endzeit *inf* in den Simulationsparametern eingestellt.

6.2.1 Initialization

Mit diesem Block wird der Parametersatz für alle im Modell benötigten Parameter geladen. Dabei wird die Callback Funktionalität von Simulink genutzt um bei der Initialisierung des Blocks mit dem Callback *InitFcn()* die Datei *Prm.mat* in den Modelworkspace zu laden. Auf im Modelworkspace befindliche Daten kann im gesamten Modell von jeder hierarchischen Ebene aus zugegriffen werden.

Für die komfortable Erstellung der *Prm.mat* Datei wird das MATLAB Skript *Parametrization.m* verwendet, das mit einem Doppelklick auf den Block geöffnet wird. Mit diesem Skript werden alle für den Algorithmus benötigten Parameter eingestellt, Konstanten berechnet und Datenstrukturen erzeugt. Einige der einzustellenden Parameter sind:

- Abtastrate
- Umgebungstemperatur
- Buffergröße und Bufferüberlappung des Daten Empfangsbuffers
- Geometrische Abmessungen und Anzahl der Analysepunkte auf der Analysefläche
- Geometrische Positionen der Mikrofone im Mikrofonarray sowie Position und Lage des Mikrofonarrays
- Detektionsparameter wie Detektionsschwellwerte, Spurbeschreibungen und Detektionsbufferparameter

Die Einheiten und Wertebereiche der einzustellenden Parameter sind in der Datei *Prm.mat* dokumentiert.

6.2.2 Receiver

In diesem Block ist alles zusammengefasst, was mit dem Empfang der Daten vom Mikrofonarray und deren Bufferung für die Weiterverarbeitung durch den Algorithmus zu tun hat. An den Ausgängen des Blocks werden die zu verarbeitenden Datenvektoren, der dazugehörige Zeitstempelvektor und ein Sampleverlustindikator ausgegeben. Abbildung 6.2 zeigt den Inhalt des Receiver Blocks. Die hierarchischen Ebenen sind durch die übereinanderliegenden Fenster dargestellt.

xFaceReceiveUDP

Mit diesem Block wird die Schnittstelle zwischen dem Mikrofonarray und dem Simulink Modell realisiert. Die verwendete Schnittstelle wurde sowohl physikalisch als auch datentechnisch in Kapitel 5.2.4 spezifiziert. Hinter diesem Block steckt eine von mir in der Programmiersprache C geschriebene S-Function, mit welcher das Übertragungsprotokoll umgesetzt wird. Die S-Function beinhaltet einen UDP-Receiver, der die per UDP Protokoll über Ethernet gesendeten Daten empfängt, auf Gültigkeit überprüft und anschließend

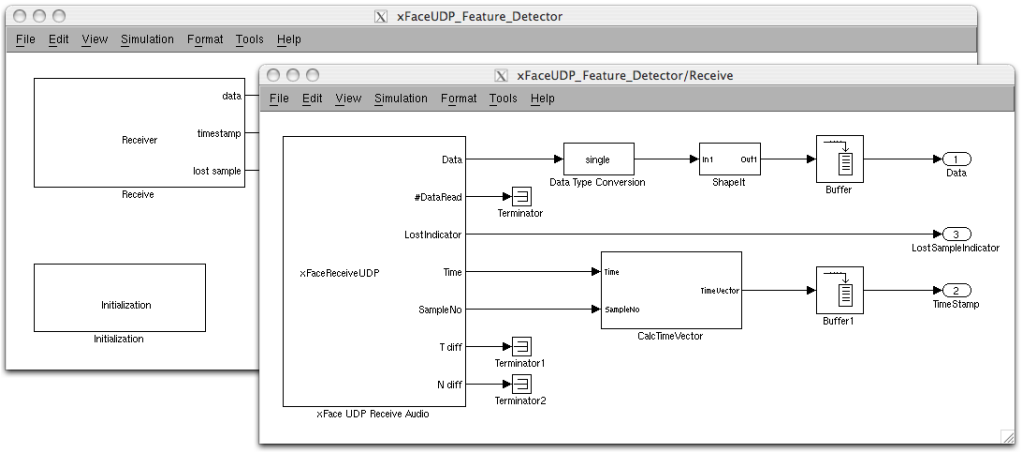


Abbildung 6.2: Simulink Block Receive

an den Ausgang des Blocks zur Verarbeitung weiterleitet.

Der Block besitzt einige Parameter, die über die zugehörige Parametermaske eingestellt werden können:
Anzahl Mikrofonkanäle: Anzahl der übertragenen Mikrofonkanäle, hängt direkt von der im FPGA verwendeten Software ab und nicht von der tatsächlich verwendeten Anzahl an Mikrofonen im Mikrofonarray. Dieser Wert hat direkten Einfluss auf die Größe des Datagramms.
Samplingrate: Wird über einen globalen Parameter im Simulink Modell festgelegt.
UDP Portnummer: Beschreibt die Portnummer des Empfangsports.
Datensegmentgröße [Byte]: Die Datensegmentgröße ist fix mit 1034 festgelegt und darf momentan nicht verändert werden.
Timeout [ms]: Mit diesem Parameter kann das Datenempfangs-Timeout festgelegt werden. Der Wert 0 deaktiviert diese Funktionalität. Ein Wert größer 0 setzt das Timeout auf einen entsprechenden Wert. Findet innerhalb des Timeouts kein Datenempfang statt, wird der Empfang abgebrochen und ein Fehler ausgegeben.
Faktor für Ethernet Empfangsbuffervergrößerung: Abhängig vom Betriebssystem kann der Ethernet Empfangsbuffer auf Betriebssystemebene verschiedene Größen haben. Bei der Verwendung von hohen Abtastraten, d.h. einer sehr hohen Datenrate kann es bei einem zu klein dimensionierten Empfangsbuffer zu Bufferüberläufen und dadurch zu Paketverlusten kommen. Mit diesem Faktor kann die Größe des Buffers vervielfacht werden.
Samplezeit: Bestimmt die Samplezeit, mit welcher der Block von der Simulink Engine zur Ausführung aufgerufen wird.

Die Größe der empfangenen Datagramme wird auf die mittels Parameter festgelegte Datensegmentgröße überprüft. Jedes Datagramm beinhaltet einen Zeitstempel, mit dem ein Paketverlust bei der Übertragung oder ein Ethernetempfangsbufferüberlauf detektiert werden kann. Wird nun ein Paketverlust detektiert, so wird das am *LostIndicator*-Ausgang des Blocks für die Dauer eines Simulationszeitschrittes mit einem logischen High-Signal angezeigt und der restliche Teil des Algorithmus muss entsprechend darauf reagieren, in-

dem alle Buffer ab dem Zeitpunkt der Sampleverlustdetektion neu gefüllt werden müssen.

Nach einem erfolgreichen Datenempfang wird das Datagramm in die einzelnen Datenteile aufgespalten und die Daten an die entsprechenden Ausgänge weitergegeben.

ShapeIt

In diesem Block werden die Daten der standardmäßig 32 vom FPGA übertragenen Mikrofonsignale auf die reale Anzahl an Mikrofonkanälen zusammengekürzt und für die Speicherung der Daten im Buffer in eine entsprechende Form gebracht.

CalcTimeVector

Der aus Sekundenzeit und Samplenummer bestehende Zeitstempel wird mit Hilfe dieses Blocks in Sekundenzeit und Nanosekunde innerhalb der Sekunde umgerechnet. Diese Information wird für den Zeitstempel beim Auslesen des Datenbuffers benötigt.

Buffer, Buffer1

Mit dem Buffer Block wird das für den Algorithmus benötigte Framing der Daten erzeugt. Die Buffergröße und die Bufferüberlappung wird, wie in Kapitel 6.2.1 beschrieben über Parameter festgelegt. Der Block Buffer1 enthält die zu den Daten zugehörigen Zeitstempel und muss mit denselben Werten wie der Buffer Block parametriert werden.

6.2.3 Feature Extraction

Mit diesem Block wurde der als *Generierung der akustischen Bilder* in Kapitel 3.2.1 bezeichneten Teil des Analysealgorithmus als Simulink Teilmodell implementiert.
Aufgrund der prinzipiell vektorbasierten Verarbeitung in Simulink wurde für die Entwicklung des Modells zuerst der Originalalgorithmus in MATLAB auf vektorbasierte Verarbeitung umgebaut und anschließend als Simulink Modell realisiert. Während des Umbaus musste immer wieder die Identität mit dem Original überprüft und sichergestellt werden.

Die für die Berechnung des akustischen Bildes benötigten Parameter werden wiederum mit dem in 6.2.1 erklärten MATLAB Skript erzeugt und bei der Modellinitialisierung geladen.

Eingangsdaten sind für diesen Block frameweise aus den unter 6.2.2 erläutertem Buffern ausgelesenen Mikrofonsignaldaten aller Mikrofone und ebenso frameweise ausgelesene Zeitstempeldaten. Beide Frames, also Datenframe und Zeitstempelframe müssen dieselbe Framelänge und dieselbe Frameüberlappung besitzen. Aus den Mikrofonsignaldaten wird das akustische Bild der Analysefläche berechnet, der älteste Zeitstempel aus dem Zeitstempelframe wird als Zeitstempel für das zu erzeugende akustische Bild übernommen. Die berechneten Daten werden anschließend an den Ausgängen des Blocks zur weiteren Verarbeitung ausgegeben.

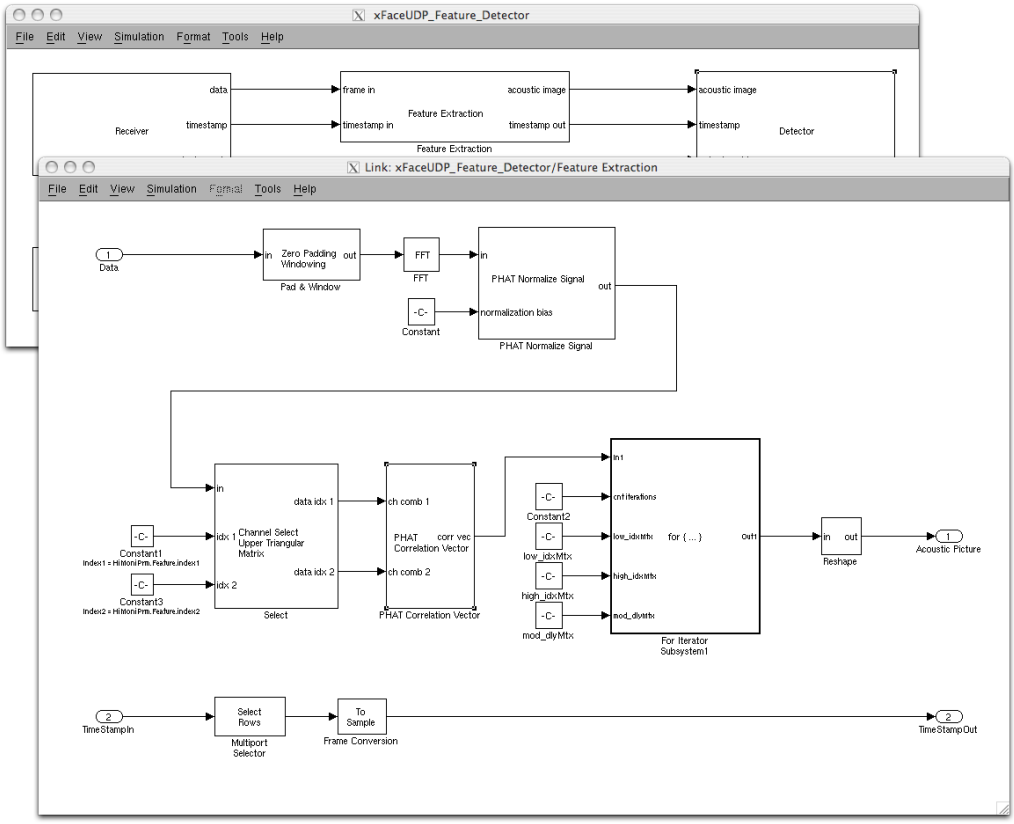


Abbildung 6.3: Simulink Block Feature Extraction

6.2.4 Detector

Im Detector Block sind alle für die Detektion der Fahrzeuge und anschließend Ausgabe der Detektionsdaten benötigten Modellteile zusammengefasst. Die Eingangssignale des Blocks bestehen aus den Daten des akustischen Bildes, dem dazugehörigen Zeitstempel und dem OutputEnable Signal.

Buffer

Mit dem Buffer Block wird der Mittelungsbuffer realisiert. Die Dimension und die Überlappung des Buffer werden wiederum in den Modellparametern festgelegt.

Buffer1

Damit der richtige Zeitstempel für die Detektion verwendet wird, muss auch der Zeitstempel mit einem Buffer mit denselben Parametern wie der Datenbuffer gebuffert werden. Mit dem nachfolgenden *Multiport Selector* Block wird dann der zur Detektion gehörende Zeitstempel aus dem Buffer selektiert.

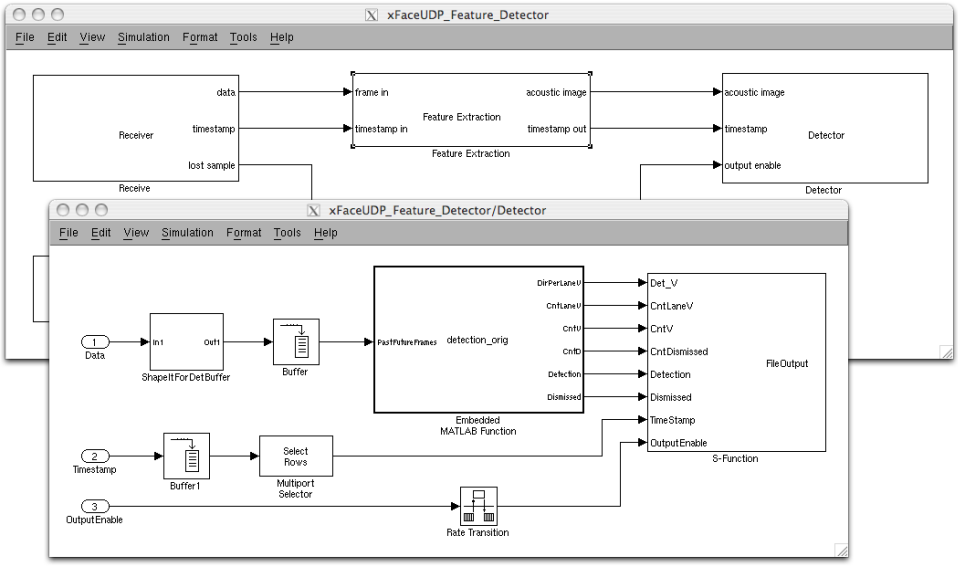


Abbildung 6.4: Simulink Block Detector

Embedded MATLAB Function - *detection_ orig()*

Nach mehreren erfolglosen Versuchen die Detektionsfunktion des Originalalgorithmus als Simulinkmodell zu implementieren wurde dieser Teil als Embedded MATLAB Function in einem entsprechenden Block realisiert. Damit dieser Block ohne MATLAB Framework simuliert werden kann, ist das MATLAB Coder Tool notwendig, um aus MATLAB-Code C-Code erzeugen zu können, der dann in das Modell eingebunden wird.

Prinzipiell wurde der Algorithmus wie in 3.2.1 beschrieben vom Original übernommen, es mussten jedoch einige Anpassungen wegen des eingeschränkten Befehlssatzes in Embedded MATLAB vorgenommen werden. Folgende Ausgangssignale werden erzeugt:
DirPerLaneV: Bei einer Fahrzeugdetektion werden hier die Fahrtrichtungen der Fahrzeuge in Vektorform ausgegeben.
CntLaneV: Gesamtanzahl an detektierten Fahrzeugen pro Spur
CntV: Gesamtanzahl an detektierten Fahrzeugen
CntD: Gesamtanzahl an Fehldetektionen
Detection: Bei einer erfolgreichen Detektion wird an diesem Ausgang für einen Simulationszeitschritt ein logischer High-Pegel ausgegeben
Dismissed: Ist während der Detektion ein Fehler aufgetreten oder kann das detektierte Fahrzeug keiner Fahrtrichtung zugeordnet werden, so wird die Detektion als Fehldetektion gewertet. Am Ausgang wird dann für die Dauer eines Simulationszeitschrittes ein logischer High-Pegel ausgegeben.

FileOutput

Wie bereits vorher beschrieben erfolgt die Ausgabe der Detektionsergebnisse in eine Datei. Simulink stellt hierfür den *ToFile*-Block mit dem *mat*-Dateien generiert werden können zur Verfügung. Dieser Block hat allerdings den Nachteil, dass die Daten erst nach Beenden der

Simulation in die Datei geschrieben werden. Da in diesem Fall die Simulationszeit aber mit ∞ angegeben ist, wird die Datei also quasi erst beim Beenden des Programms geschrieben. Gefordert ist die Dateiausgabe allerdings unmittelbar nach einer Detektion. Deshalb wurde die Dateiausgabe als eigene S-Function in C geschrieben.
Die Log-Daten werden in die Datei *output.dat* im Klartext ausgegeben. Bei Start und Ende des Programms wird die aktuelle Rechnerzeit in die Datei ausgegeben. Bei Detektionen wird der vom Mikrofonarray gelieferte Zeitstempel und die detektierte Fahrtrichtung auf den einzelnen Fahrspuren ausgegeben, bei Fehldetektion der Zeitstempel mit einem Hinweis, dass es sich um eine Fehldetektion handelt.
Die Daten werden sofort in die Datei ausgegeben, sodass die Funktionsweise des Programms überwacht werden kann.
Mit dem *OutputEnable*-Eingangssignal kann für den Fall eines Paketverlustes im beim Datenempfang vom Mikrofonarray die Ausgabe der Detektionsergebnisses in die Datei ausgesetzt werden.

6.2.5 ClearBufferOnError

Mit diesem Block wird das Protokollieren der Detektionsdaten gesteuert. Wird ein Fehler vom *Receiver* Block am Eingang gemeldet, so wird die Protokollierung für eine bestimmte Anzahl an Simulationszyklen ausgesetzt. Die Anzahl wird aus den im Datenfluss verwendeten Buffergrößen und Bufferüberlappungen berechnet und bei der Initialisierung des Modells geladen.

6.3 Verwendung des Simulink Coders

Zur Erstellung eines nativ lauffähigen Programms muss nun der Simulink Coder auf das zuvor erstellte Simulink Modell angewandt werden.

Folgende Parameter sind dabei im Simulink Coder einzustellen

- Entsprechendes System Target File muss gewählte werden
- Die Programmiersprache muss festgelegt werden
- Etwaige Optimierungsparameter für den Build-Process müssen noch angegeben werden.

Das aus diesem Prozess entstehende ausführbare Programm kann dann in der Shell gestartet werden und liefert dieselben Daten wie das Simulink Modell.

6.4 xFace Steuerungs- und Kontrollsoftware

Die von xFace gelieferte Steuerungs- und Kontrollsoftware wurde in Python geschrieben und dient zur Konfiguration, Steuerung und Kontrolle des Programms im FPGA.

6.4.1 xFaceStream

Mit dem Shell Programm *xfacestream.py* wurde das in Tabelle 5.7 beschriebene Kontrollprotokoll implementiert. Damit ist es möglich, auf das in der CPU des FPGA Boards laufende Programm Einfluss zu nehmen. Dabei können Datenpfade im FPGA gesteuert, Daten abgefragt und verändert werden und Konfigurationen an der digitalen Hardware vorgenommen werden. Die möglichen Befehle können der Hilfe des Programms entnommen werden.

6.4.2 Micarray Control

Das Programm *micarray_ ctrl.py* biete eine Graphische Oberfläche zur Statuskontrolle von an den Rechner angeschlossenen FPGA-Boards. Über den Menüpunkt *Action - Discover* kann nach FPGA-Boards durch Eingabe deren IP Adresse im Netzwerk gesucht werden. Wird mit der angegebenen IP-Adresse ein FPGA-Board gefunden, so werden nun im 250 ms Zyklus Statusinformationen der gefundenen Boards abgefragt und im GUI angezeigt.

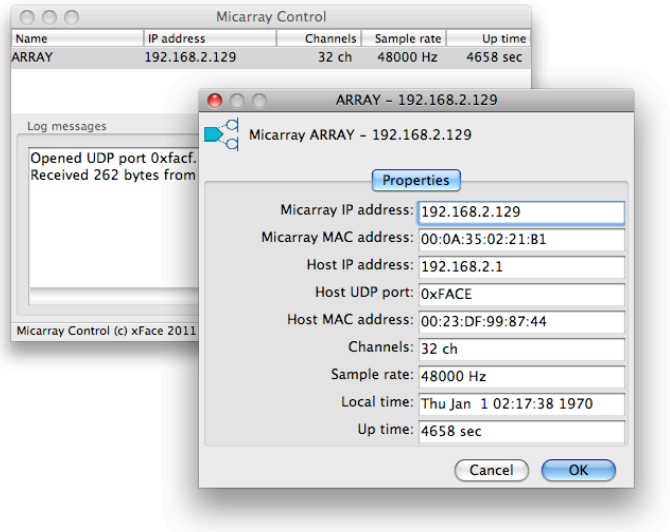


Abbildung 6.5: xFace Micarray Control Software

Kapitel 7

Systemtests

In diesem Kapitel wird nochmal zusammengefasst, welche Tests mit den verschiedenen Systemen durchgeführt wurden.

7.1 Analysealgorithmus

Die Implementierung des Analysealgorithmus in Simulink wurde mit denselben Testdaten wie die originale MATLAB Implementierung getestet und lieferte auch dieselben Ergebnisse. Eine wirkliche Performanceanalyse kann jedoch erst erfolgen, wenn das Gesatmsystem im Zuge einer Langzeitmessung getestet wird.

7.2 Eigenbau Hardware

7.2.1 Mikrofonboard

Messungen an den bisher entwickelten Prototypen wurden wie in Kapitel 5.3.5 beschrieben durchgeführt. Das aktuelle Mikrofonboard ist zum Zeitpunkt der Erstellung dieser Arbeit gerade in Produktion es können daher gegenwärtig noch keine genaueren Angaben dazu gemacht werden. Durch das 4-lagige Platinenlayout wird aber eine deutliche Erhöhung des THD+N Wertes gegenüber den Vorgängervarianten erwartet.

7.2.2 Backplaneboard

Das Backplaneboard wird erst gefertigt, wenn die Mikrofonboards den Anforderungen entsprechen. Gegenwärtig kann hierzu also noch keine Angabe gemacht werden.

7.3 Datenübertragungssystem

Die Datenübertragungssicherheit zwischen dem xFace FPGA Board und dem Analyse PC ist für den Einsatz ein sehr wesentlicher Parameter und wurde daher umfangreich getestet.

Auf dem xFace FPGA Board wurde für diesen Zweck der in 5.6.1 beschriebene Sinus-Datenstromgenerator implementiert, mit dem sehr einfach Testdaten von vielen Audiokanälen mit der entsprechenden Abtastrate generiert werden können.

Am Analyse PC wurde ein neues Simulink Modell mit dem in 6.2.2 beschriebenen *xFaceReceiveUDP* Block erstellt, daraus ein ausführbares Programm generiert und damit die Datenübertragung getestet. Variiert wurde dabei die Abtastrate (48 kHz, 96 kHz und 192 kHz) und die Anzahl der zu übertragenden Audiokanäle (32 und 64).

Getestet wurde auf WindowsXP und Mac OSX10.5.8 auf den in Anhang A angeführten Computern. Gleiche Programmeinstellungen gelten für diesen Vergleichstest als Voraussetzung.

WindowsXP

Unter WindowsXP traten mit Standardeinstellungen bereits bei 48 kHz Abtastrate, 32 zu übertragenden Audiokanälen und ohne sonstige Belastung des Computers bereits gelegentlich Paketverluste auf. Nach Optimierung der Systemeinstellungen für Audiorecording war das schließlich nicht mehr der Fall. Bei höheren Abtastraten bzw. der Erhöhung der Anzahl der zu übertrangenden Audiokanäle traten allerdings sofort wieder Datenpaketverluste auf.

OSX 10.5.8

Auf dem UNIX-basierten Betriebssystem konnte die Abtastrate auf bis zu 192 kHz erhöht werden und gleichzeitig die zu übertrangende Kanalanzahl auf 64 Audiokanäle erhöht werden, ohne dass Datenpaketverluste detektiert wurden. Dieser deutliche Unterschied ist auf die unterschiedliche Funktionsweise des Taskschedulers zurückzuführen.

Kapitel 8

Zusammenfassung und Ausblick

Das im Rahmen dieser Arbeit zu implementierende Gesamtsystem zur Fahrtrichtungsdetektion von Fahrzeugen wurde in zwei Teilen entwickelt:
Im ersten Teil wurde die MATLAB Version des bereits existierenden Fahrtrichtungsdetektionsalgorithmus analysiert, optimiert und als Simulink Modell implementiert. Aus dem Modell wurde ein ausführbares Programm generiert und damit die Echtzeitfähigkeit der Algorithmusimplementierung gezeigt.
Im zweiten Teil wurde ein Konzept für ein prototypisches Gesamtsystem entwickelt und realisiert. Dabei sollte die National Instruments Datenakquisitionshardware durch billigere Hardware ersetzt werden. Eine Marktrecherche lieferte einen Überblick über erhältliche Akquisitionssysteme, mit dem Resultat, dass entweder nicht alle technischen Anforderungen erfüllt wurden, oder der Preis des Systems zu hoch war. Mit der Firma xFace wurde daraufhin gemeinsam ein Konzept für ein eigenes Datenakquisitionssystem erarbeitet, bei dem die Audiodaten im Mikrofonarray digitalisiert und anschließend via Ethernet an das Analysesystem übertragen werden. Für die Mikrofonboards des Akquisitionssystems wurden Prototypen entwickelt und vermessen. Die aktuelle Version ist zum Zeitpunkt der Erstellung dieser Arbeit gerade in Produktion.
Die von xFace gelieferte digitale Hardware nimmt die digitalisierten Audiodaten der Mikrofonbaords am FPGA entgegen und sendet sie via Ethernet an das Analysesystem. Für die Datenübertragung wird dabei das UDP Protokoll verwendet.
Die Datenschnittstelle zum Simulink Modell des Fahrtrichtungsdetektionsalgorithmus wurde als S-Function ausgeführt und unter verschiedenen Betriebssystemen getestet. Dabei konnte auf dem UNIX basierten Betriebssystem OS X eine wesentlich bessere Stabilität beobachtet werden als unter Windows XP.
Alle Teile des Datenakquisitionssystems, außer den Mikrofonboards, sind fertig aufgebaut und getestet.

Mit fertig aufgebauten Mikrofonboards kann das Datenakquisitionssystem getestet und charakterisiert werden. Damit kann anschließend das gesamte Fahrtrichtungserkennungssystem installiert und Praxistests durchgeführt werden.

Das Analysesystem könnte in einer Ausbaustufe mit einer GUI zur Datelleung des Videobildes samt dazughörigen akustischen Bild des beobachteten Straßenabschnitts, sowie einer grafischen Darstellung der Detektionsergebnisse ausgestattet werden.

Anhang A

Technische Daten Analyse PC

A.1 Siemens Fujitsu Mini PC

Modellname	ESPRIMO Q5020
Prozessortyp	Intel Core 2 Duo
Pozessorgeschwindigkeit	2 GHz
Anzahl der Prozessoren	1
L2-Cache	2 MB
Speicher	2 GB
Betriebssystem	Microsoft Windows XP Professional SP3

Tabelle A.1: Technische Daten Siemens Fujitsu PC

A.2 Apple MacBook

Modellname	MacBook5,2
Prozessortyp	Intel Core 2 Duo
Pozessorgeschwindigkeit	2 GHz
Anzahl der Prozessoren	1
L2-Cache	3 MB
Speicher	2 GB
Betriebssystem	Mac OSX 10.5.8

Tabelle A.2: Technische Daten Apple Laptop

Anhang B

Zusätzliche Testhardware

B.1 SPDIF - AES/EBU Signalformatwandler

Für die Charakterisierung der Mikrofonboardprototypen ist das AudioPrecision 2700 Messsystem der TU Graz vorgesehen. Das Messsystem besitzt digitale Audioeingänge im AES/EBU Format, die auf dem AES3-Standard [20] basieren. Das SPDIF Format wird vom selben Standard abgeleitet und stellt das Consumer-Format der Schnittstelle dar.
Elektrisch verwendet die AES/EBU Schnittstelle Pegel und Signalformat nach der Norm EIA-422. Dabei handelt es sich um eine differentielle Datenübertragung, die in [21] spezifiziert ist. Die SPDIF Schnittstelle des xFace FPGA liefert ein unsymmetrisches Digitalsignal im TTL Pegel. Damit nun die in Abschnitt 5.6.1 beschriebene SPDIF Schnittstelle für diese Messungen genutzt werden kann muss das elektrische Übertragungsformat entsprechend angepasst werden.

Auftretende Signalfrequenzen

Da die gesamte Signalaufbereitung für eine maximale Abtastrate von $f_S = 192\,\text{kHz}$ ausgelegt ist, muss auch der Signalwandler für entsprechende Datenraten ausgelegt sein.

Bei einer Abtastrate von $f_S = 192\,\text{kHz}$ und einer Wortbreite von $w = 32\,\text{Bit}$ (max. 24 Bit Audiodaten + Overhead) ergibt sich durch die Übertragung von zwei Kanälen in einem Frame eine Bitrate von

$$b = 2 \cdot f_s \cdot w = 2 \cdot 192000 \cdot 32 = 12.288\,{}^{\text{MBit}}/_{\text{s}}$$

Als Kanalkodierung kommt beim AES3-Format der Biphase-Mark-Code zum Einsatz. Die maximale zu übertragende Signalfrequenz ergibt sich, wenn das Nutzsignal aus logisch "1" besteht.

$$f_{Signal_{max}} = 2 \cdot b = 24.576\,\text{MHz} \tag{B.1}$$

Schaltung und Platinenlayout

Als physikalische Schnittstelle mit dem FPGA wurde der PMOD Header 2 am FPGA Board festgelegt. Hierbei handelt es sich um einen 6-Pin Header, auf den die Signalwandlerschaltung aufgesteckt werden kann. Ausgangsseitig wird ein gewöhnlicher XLR-Stecker

verwendet. Das Pinout des PMOD Headers ist in Anhang **??** aufgelistet.

Das FPGA kann am PMOD Header 3.3 V ausgeben und übernimmt damit die Spannungsversorgung der Signalwandlerschaltung. Es wird der SN65HVD379 Treiberbaustein von Texas Instruments eingesetzt, der den Anforderungen an Versorgungsspannung und maximal zu übertragende Signalfrequenz genügt. Der IC wandelt ein unsymmetrisches TTL Signal am Eingang in ein differentielles Signal mit der EIA-422 Norm entsprechenden Pegeln um.
Da bei dieser Schaltung der Prototyp gleichzeitig die Endversion ist, wurden einige Platzhalter für eventuelle nachträgliche Änderungen bzw. Anpassungen der Schaltung eingebaut. Die Widerstände R_2 - R_5 werden mit $0\,\Omega$ bestückt, mit R_6 und R_7 können nachträglich Pull-Down Widerstände in die Schaltung eingebaut werden. Sie werden vorerst nicht bestückt. Es kann ein 1-lagiges Platinenlayout verwendet werden.

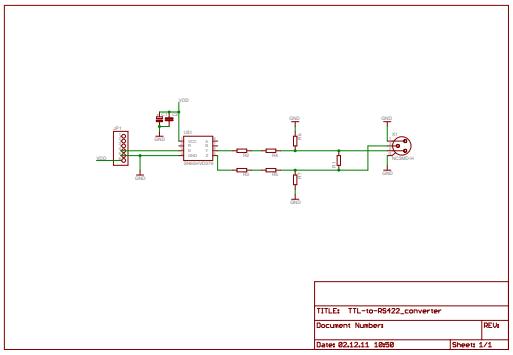


Abbildung B.1: Schaltung des SPDIF - AES/EBU Signalformatwandler

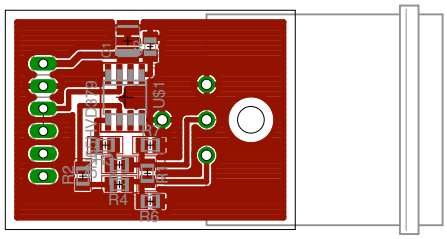


Abbildung B.2: Platinenlayout des SPDIF - AES/EBU Signalformatwandler

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] P. Dollfuß, "Driving direction detection with microphone arrays," Master's thesis, Graz University of Technology, May 2010.

[2] M. Stadtschnitzer, H. Rainer, P. Dollfuß, and F. Graf, "Final report - Hi-Moni: Highway-Monitoring - acoustic detection of wrongway drivers," Jänner 2010.

[3] L. J. Ziomek, *Fundamentals of acoustic field theory and space-time signal processing*. CRC Press, 1995. [Online]. Available: http://books.google.at/books?id=RRiNXMCRVJQC

[4] I. Mccowan, "Microphone Arrays : A Tutorial," *PoLAR*, no. April, pp. 1–36, 2001.

[5] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*, ser. Digital signal processing. Springer, 2001. [Online]. Available: http://books.google.at/books?id=nND6ObXSNoEC

[6] A. Oppenheim, R. Schafer, and J. Buck, *Zeitdiskrete Signalverarbeitung*. Pearson Studium, 2004. [Online]. Available: http://books.google.at/books?id=-LxkOQAACAAJ

[7] MMF, "M28 Modul - Bedienungsanleitung," pp. 1–4.

[8] STMicroelectronics, "LM234 Datasheet," 2007. [Online]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000458.pdf

[9] National Semiconductor, "LM317L Datasheet," 2010. [Online]. Available: http://www.national.com/profile/snip.cgi/openDS=LM317L

[10] Texas Instruments, "OPA1611 Datasheet," 2011. [Online]. Available: http://www.ti.com/lit/gpn/opa1611

[11] ——, "PCM4202 Datasheet," 2004. [Online]. Available: http://www.ti.com/lit/gpn/pcm4202

[12] ——, "Gain Scaling and Audio Performance of the PCM1804," 2004. [Online]. Available: http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=slea003

[13] ——, "PCM4202EVM User's Guide," 2004. [Online]. Available: http://www.ti.com/lit/ug/sbau103/sbau103.pdf

[14] ——, “OPA1632 Datasheet,” 2010. [Online]. Available: http://www.ti.com/lit/gpn/opa1632

[15] Philips, “I2S Bus Specification,” pp. 1–7, 1996. [Online]. Available: http://www.nxp.com/acrobat_download/various/I2SBUS.pdf

[16] FMC, VITA57.1 Spezfikationen. [Online]. Available: http://www.vita.com/fmc.html

[17] Pericom, “PI49FCT32807 1:10 CMOS Clock Driver Datasheet,” 2008. [Online]. Available: http://www.pericom.com/products/timing/clock/PI49FCT32807/

[18] Xilinx, “SP601 Hardware User Guide,” pp. 1–52, 2011.

[19] J. Wolkerstorfer, “Micarray Project Documentation,” 2011.

[20] “Aes3 standard for digital audio - digital input-output interfacing - serial transmission format for twochannel linearly represented digital audio data.” Audio Engineering Society, Tech. Rep., 1992 (R2003).

[21] “Electrical characteristics of balanced voltage digital interface circuits,” ANSI/TIA/EIA-422- B-1994, Telecommunications Industrys Association, Tech. Rep., 1994 (R2000) (R2005).