



Compressed Compute-and-Forward mit korrelierten Audiosignalen

Lucas Weber

Geboren am: 10.01.1995

Matrikelnummer: 3969535

Immatrikulationsjahr: 2013

Raphael Hildebrand

Geboren am: 24.05.1995

Matrikelnummer: 3948419

Immatrikulationsjahr: 2013

Florian Roth

Geboren am: 13.01.1994

Matrikelnummer: 3966248

Immatrikulationsjahr: 2013

Orell Garten

Geboren am: 09.07.1993

Matrikelnummer: 3948375

Immatrikulationsjahr: 2013

13.07.2016

Betreuer

Dipl.-Ing. Carsten Hermann

Betreuender Hochschullehrer

Prof. Dr.-Ing. Dr. h.c. Frank H. P. Fitzek

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel *Compressed Compute-and-Forward mit korrelierten Audiosignalen* selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Es waren keine weiteren Personen an der geistigen Herstellung der vorliegenden Arbeit beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Dresden, 13.07.2016

Lucas Weber

Raphael Hildebrand

Florian Roth

Orell Garten

Inhaltsverzeichnis

1	Aufgabenstellung	4
2	Motivation	4
3	Theoretische Vorbetrachtung	5
3.1	Kreuzkorrelationsfunktion	5
3.2	Maßzahlen	7
4	Verfügbare Technik	10
4.1	Software	10
4.2	Hardware	10
5	Programm	11
5.1	Aufbau des Programms	11
5.2	Beispielparameter	12
5.3	Erweiterung der Analysemöglichkeiten	13
5.4	Probleme	14
6	Signalauswahl	15
6.1	Beispielsignale	15
6.2	Probleme bei der Signalauswahl	17
6.3	Fazit	18
7	Zusammenfassung	19

1 Aufgabenstellung

Es ist ein Programm zu entwerfen, welches Audiodateien einliest und dateiweise die beiden Stereo-Kanäle miteinander korreliert. Es sollen Maßzahlen entworfen und berechnet werden, die wesentliche charakteristische Eigenschaften der Korrelationsfunktion, insbesondere den Anteil dominanter Komponenten und deren Abklingverhalten, widerspiegeln. Dafür sind Audiosignale aufzunehmen, bezüglich der verwendeten Maße zu klassifizieren und entsprechend ihrer Klassifizierung systematisch abzuspeichern.

2 Motivation

Im Jahr 2022 werden 500 Milliarden internetfähige Geräte erwartet, die mit einander kommunizieren sollen. Das führt zu extrem hohen Datenmengen, die in kürzestmöglicher Zeit von A nach B transportiert werden müssen. Große Herausforderungen bestehen darin, dass man sehr kurze Verzögerungszeiten und eine hohe Widerstandsfähigkeit garantieren muss. Idealerweise benötigen die Geräte wenig Energie. Ein Ansatz zur Lösung dieses Problems ist die Netzwerkcodierung.

In bestimmten Szenarien ist eine große Anzahl an Geräten mit Sensorik zu Erfassung der Umgebung mit hohen Anforderungen an die Netzwerkkapazität zur Übertragung der erfassten Daten verbunden. Es stellt sich die Frage, wie viele Sensoren für eine ausreichend genaue Abbildung benötigt werden. Da die Quellen teilweise korrelierte Datenströme erzeugen, lässt sich die zu übertragende Gesamtdatenmenge reduzieren, was durch eine geeignete Kombination von Netzwerkcodierung mit Methoden des Compressed Sensing erreicht werden soll. In dieser Arbeit werden erste Ansätze entwickelt um diese Korrelationsstrukturen möglichst einfach sichtbar zu machen und die Auswahl von Signalen für den Test entsprechender Systeme sinnvoll auszuwählen.

3 Theoretische Vorbetrachtung

3.1 Kreuzkorrelationsfunktion

verfasst von Lucas Weber

Die Basis für die Bemessung der aufgenommenen Audiosignale bildet die sogenannte Kreuzkorrelationsfunktion (KKF). Wie in der Aufgabenstellung schon beschrieben werden an ihr die Bemessungsparameter festgelegt. Aufgrund der verschiedenen Blocklängen und der Masse an Daten, die korreliert werden sollen muss die Berechnung der KKF effizient und zeitsparend implementiert werden. Im Folgenden Abschnitt wird die KKF kurz theoretisch eingeführt und das das mathematische Konzept erklärt, auf dem die effiziente Berechnung der KKF beruht.

Zuerst haben wir für die KKF eine Funktion genutzt, die zur Berechnung Summen verwendete. Dabei ergab sich, dass die Berechnung zu langsam war. In der nun vorliegenden Octave-Version wird die KKF im Frequenzbereich berechnet.

Berechnungsvorschrift

Die KKF ist als aus zwei verschiedenen Funktionen gebildeter Erwartungswert definiert. Hier werden die Formeln allgemein für die Korrelation der Prozesse \mathbf{X} und \mathbf{Y} angegeben.[ISV, S. 84]

$$\psi_{\mathbf{XY}}(t_1, t_2) = E\{\mathbf{X}(t_1) \cdot \mathbf{Y}(t_2)\} \quad (0.1)$$

$$[\text{ISV, S. 84 Formel 2.200}] \quad (0.2)$$

Dadurch dass sich der Raum, das Wiedergabemedium und die Aufnahmeart während der Aufnahme nicht geändert haben, können die Prozesse als schwach stationär angenommen werden. Außerdem wird angenommen, dass die Prozesse ergodisch sind. Durch diese Voraussetzungen hängt die KKF nicht mehr von einer Verschiebung zu bestimmten Zeitpunkten ($t_1 - t_2$) ab sondern nur noch von ihrer Verschiebung τ . Reale, aufgenommene Audiosignale $s(t)$, die hier mit durch die KKF verrechnet werden, sind in jedem Fall Energiesignale, da sie rein reell sind, einen begrenzten Wertebereich haben und nach einer bestimmten Zeit enden.[vgl. ISV, S. 85 f]

$$\text{Signalenergie} := \int_{-\infty}^{\infty} s^2(t) dt < \infty \quad (0.3)$$

Da die aufgenommenen Audiosignale zeit-diskrete Energiesignale sind wird hier auch nur die zeit-diskrete Kreuzkorrelation beschrieben. Für zeit-diskrete Energiesignale ergibt sich die folgende Berechnungsvorschrift, wobei $x(n)$ und $y(n)$ diskrete Realisierungen der Prozesse \mathbf{X} und \mathbf{Y} sind.[ISV, S. 88 f]

$$\boxed{\psi_{\mathbf{XY}}^E(k) = \sum_{n=-\infty}^{\infty} x(n) \cdot y(n+k)} \quad (0.4)$$

$$[\text{ISV, S. 89 Formel 2.218}] \quad (0.5)$$

Kreuzkorrelation im Frequenzbereich und Faltungssatz

Am Anfang unserer Arbeit haben wir uns mit der Berechnung der Kreuzkorrelation beschäftigt. Da die Bildung der Summe der Multiplikation von $x(n)$ und $y(n+k)$ sehr rechenaufwändig ist, haben wir nach schnelleren Möglichkeiten gesucht die KKF der beiden Signale zu berechnen. Eine geeignet Möglichkeit ist die Berechnung der KKF im Frequenzbereich. Für die Berechnung der KKF im Frequenzbereich macht man sich die Ähnlichkeit der KKF zur Faltung und den Faltungssatz zunutze.

KKF als Faltung

Zeit-kontinuierlicher Fall:

$$\psi_{\mathbf{XY}}^E(\tau) = x(-\tau) * y(\tau) \quad (0.6)$$

$$[\text{ISV, S. 89 Formel 2.217}] \quad (0.7)$$

Zeit-diskreter Fall:

$$\psi_{\mathbf{XY}}^E(k) = x(-k) * y(k) \quad (0.8)$$

Faltungssatz in Verbindung mit KKF Wir schreiben zuerst die die KKF als Faltung. Danach transformieren wir die Faltung in den Frequenzbereich. Durch geschicktes Erweitern und Substitution findet man einen Ausdruck, um die KKF im Frequenzbereich zu berechnen.

$$\psi(t) = x(-t) * y(t) = \int_{-\infty}^{\infty} x(-\tau) \cdot y(t - \tau) d\tau \quad (0.9)$$

$$(0.10)$$

$$\underline{\Psi}(\omega) = \int_{-\infty}^{\infty} x(-t) * y(t) \cdot e^{-j\omega t} dt \quad (0.11)$$

$$= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(-\tau) \cdot y(t - \tau) d\tau \right] \cdot e^{-j\omega(t-\tau+\tau)} dt \quad (0.12)$$

$$= \int_{-\infty}^{\infty} x(-\tau) \cdot e^{j\omega(-\tau)} \underbrace{\left[\int_{-\infty}^{\infty} y(t - \tau) e^{-j\omega(t-\tau)} dt \right]}_{*} d\tau \quad (0.13)$$

Durch Substitution von $(t - \tau)$ durch t' ergibt sich $*$ zur Fourier-transformierten $\underline{Y}(\omega)$ von $y(t)$. Der restliche Ausdruck wird durch das positive Vorzeichen in der e-Funktion zur komplex konjugierten Transformierten $\underline{X}^*(\omega)$ der Funktion $x(t)$.

Die KKF lässt sich im Frequenzbereich also als

$$\boxed{\underline{\Psi}(\omega) = \underline{X}^*(\omega) \cdot \underline{Y}(\omega)} \quad (0.14)$$

schreiben. [ISV, S.180]

Wenn man nun die KKF im Frequenzbereich zeitsparend durchführen will, muss man die FFT für $x(k)$ und $y(k)$ (dabei $k \in \mathbb{N}_0^+$) der Länge N durchführen. Dabei muss man beachten, dass bei der FFT ein Linienspektrum ergibt. Die FFT beruht vor allem auch auf der Annahme, dass sich die N diskreten Werte periodisch wiederholen [ISV, S.135]. Durch die IFFT von $\underline{\Psi}(\omega)$ ergibt sich also die periodische KKF $\tilde{\psi}(t)$.

Zero Padding

Die periodische KKF $\tilde{\psi}(t)$ ist in unserem Projekt zudem die bessere Wahl. Berechnet man die KKF im Frequenzbereich wird durch die implementierten Funktionen von Octave, so wie von Python, sogenanntes Zero-Padding durchgeführt. Vorstellen kann man sich das als auffüllen der Daten mit N Werten $= 0$ an jeweils den Rändern einer der beiden Funktionen, da durch die Verschiebung eine der Funktionen über den Rand der anderen hinausragt. Wie man in Abbildung 0.1 erkennen kann ist die Funktion x_2 soweit verschoben, dass sie über den rechten Rand der Funktion x_1 hinausragt. Wenn man

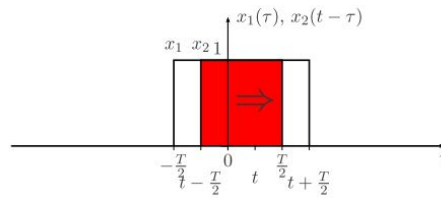


Abbildung 0.1: Anschauliches Beispiel zur Faltung [NT, S. 28]

die beiden Rechtecksignale als Bereiche sieht in denen echte Funktionswerte liegen, würde nun auf der positiven x-Achse ab $T/2$ mit Nullen aufgefüllt werden. Dadurch bekommt die KKF automatisch ein Abklingverhalten an ihren Rändern, das nur durch die begrenzte Anzahl an Werten verursacht wird. Die Korrelation nimmt an den Rändern nicht zwingend ab. Durch dieses Abklingverhalten würden unsere Ergebnisse also abgefälscht werden. Die periodische KKF ist also aussagekräftiger, da unser reales Signal in jedem Fall an den Rändern der korrelierten Blöcke nicht auf Null abklingt. Wenn man die periodische KKF im Zeitbereich berechnen möchte, müsste man an den Rändern der Funktionen nochmal die Funktion x_1 anhängen.

3.2 Maßzahlen

verfasst von Florian Roth

Im Anschluss an diese Arbeit gilt es zu untersuchen, in wie fern die aufgenommenen Signale geeignet sind um mittels Netzwerkcodierung und Compressed Sensing unter Einsparung von Datenverkehr versendet werden können.

Um das für eine große Anzahl an Signalen überhaupt möglich zu machen, ist es erforderlich die untersuchten Signale anhand von bestimmten Eigenschaften zu klassifizieren. Damit ein einfacher Vergleich mehrerer Signale schnell möglich ist, bietet es sich an diese Eigenschaften als Zahlenwert auszudrücken.

Die beschriebenen Eigenschaften sind entweder physikalischer Natur oder versuchen die Form der Kreuzkorrelation zu charakterisieren. Dabei wurde darauf geachtet, wenn möglich normierte Größen zu verwenden um die Vergleichbarkeit zwischen Signalen unterschiedlicher Abtastrate und Zeitdauer aufrecht zu erhalten.

Beim Entwerfen solcher Maßzahlen besteht die Schwierigkeit darin, möglichst viel aussagekräftige Information dahingehend zu vereinfachen, dass eine Überführung in eine Zahl überhaupt möglich ist. Gleichzeitig darf durch die Vereinfachung nicht die Aussagefähigkeit der Maßzahl zerstört werden, also die Möglichkeit auf eine Eigenschaft des Signals anhand des Zahlenwertes zurück zu schließen. Aufgrund dieser Anforderungen sind im Verlauf dieser Arbeit mehrere Maßzahlen entstanden, von denen einige im weiteren Prozess wieder verworfen wurden.

Die Bedeutung und Berechnung der finalen Maßzahlen wird hier kurz vorgestellt.

ripple

Die Maßzahl *ripple* trifft eine Aussage über die Energieverteilung im Signal. Zur Berechnung wird die Energie, die die obersten fünf Perzentile der berechneten Werte enthalten, ins Verhältnis zur Gesamtenergie gesetzt.

$$ripple = \frac{\sum_{\text{der obersten 5 Perzentile}} x_i^2}{\sum_{i=1}^N x_i^2}$$

ripple kann Werte zwischen 0 und 1 annehmen, wobei *ripple* = 0 bedeutet, dass es sich um eine konstante Funktion handelt, während *ripple* = 1 auf einen einzelnen Impuls schließen lässt.

sigma

Ziel war es eine Maßzahl zu finden, die eine Aussage über die Verteilung der größten Werte der Kreuzkorrelation liefert. In einer ersten Version des Programmes wurde die in Octave integrierte Funktion *xcorr* verwendet, die zero-padding nutzt, wenn sie die Signale im Zeitbereich zueinander verschiebt. Dadurch fiel die KKF zu den Seiten schnell ab und die Hüllkurve erinnerte stark an eine Glockenkurve. Ein Maß für die „Breite“ einer Glockenkurve stellt das *sigma* im Exponenten der *e*-Funktion dar.

Auch nach dem Umstellen auf eine andere KKF-Berechnung, bei der durch periodisches Aneinandersetzen der Signale kein Abfall auf Null zu den Rändern stattfindet, konnte in einem Großteil der Korrelationshüllkurven weiterhin eine Glockenkurvenform vorgefunden werden, allerdings nun mit Gleichanteil. Um die gewünschte Maßzahl *sigma* aus den numerisch vorliegenden Werten zu erhalten, muss ein mathematischer Ausdruck für die Hüllkurve gefunden werden. Die Hüllkurve *envelope* wiederum erhält man durch Amplituden-Demodulation der KKF. Dabei wird das Signal gleichgerichtet und auf ein Tiefpassfilter gegeben. Der Filter wird in dieser Arbeit im Frequenzbereich realisiert mit einer von der zeitlichen Länge des Signals t_{Signal} abhängigen Grenzfrequenz $f_c = \frac{20}{t_{Signal}}$. Bei dem Zahlenwert 20 handelt es sich um einen experimentell bestimmten Wert.

$$envelope = IDFT\{DFT\{|\psi_{XY}(n)|\} \cdot H_{TP}\}$$

Für die Hüllkurve *envelope* wird nun mittels Methode der kleinsten Quadrate eine Regressionsrechnung auf die vermutete Glockenkurvenfunktion vorgenommen. Die Funktionsvorschrift lautet dabei

$$y = a \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} + b$$

Mit den zu bestimmenden Konstanten a , b , μ und σ . Mit dem erhaltenen Wert für σ lässt sich keine Aussage über den Anteil der Fläche der Hüllkurve in den Grenzen $\mu - \sigma$ bis $\mu + \sigma$ treffen. Das liegt an dem Gleichanteil und keiner Verknüpfung mit der Amplitude a wie bei der Gauss'schen Glockenkurve. Sehr wohl stellt σ allerdings ein Maß für die Breite des peaks rund um den Nullpunkt der Korrelation dar. Die Angabe erfolgt in Sekunden.

timeDiff

Der Parameter *timeDiff* gibt den zeitlichen Versatz der Stereokanäle zueinander wieder. Dazu wird der Indexwert i des Maximums der Kreuzkorrelation bestimmt und anhand der Samplingrate r die Zeitdifferenz zum Nullpunkt ($\lambda = 0$) berechnet.

$$\Delta t = \frac{\Delta i}{r}$$

exp und area

Da die Regression auf eine Glockenkurve nicht für alle Signale sinnvoll ist, war eine Maßzahl gefordert, die zumindest den Abfall der Gesamtheit aller Amplituden angibt. Dazu wurden die diskreten Werte der Korrelation betragsmäßig absteigend geordnet. Anschließend wurde wieder eine Regressionsrechnung mit Hilfe der Methode der kleinsten Quadrate angewandt. Dabei handelt es sich nun um eine einfache abfallende *e*-Funktion

$$y = e^{-c \cdot x}$$

wobei der ausgegebene Parameter *exp* der ermittelten Konstante c entspricht, normiert auf die Länge des Signals. Bei Vergleich der Regressionskurve mit der ursprünglichen Funktion fällt auf, dass diese nur schlecht angenähert wird. Auf ein modifizieren der Funktion, beispielsweise durch Addition eines Gleichanteils, wird jedoch verzichtet, da die Verrechnung der zusätzlichen Parameter zu einer aussagekräftigen Maßzahl nicht glückte. Ein Maß für den geforderten Abfall stellt auch die einfache

e-Funktion dar.

Um einen alternativen Parameter zu bestimmen wird die Fläche unter der Kurve berechnet und auf die Länge des Signals normiert.

$$area = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} |\psi_{XY}(n)|$$

Eine Normierung auf den Maximalwert erfolgt schon beim Berechnen der Kreuzkorrelation. Dadurch kann *area* nur Werte zwischen 0 und 1 annehmen. Je kleiner *area*, desto steiler fallen die Amplituden ab.

Trotz der erfolgreich ermittelten Maßzahl *area*, berechnet das Programm weiterhin den Parameter *exp*, denn dieser lässt sich leicht modifizieren und kann somit noch an Aussagekraft hinzu gewinnen.

4 Verfügbare Technik

4.1 Software

Softwareseitig haben wir Octave benutzt. Als freie Alternative zu Matlab vereint Octave gute Performance, syntaktische Gleichheit zu Matlab, sowie kostenfreie Benutzung unter einem Dach. Im Vergleich mit Python haben wir festgestellt, dass die Geschwindigkeit aufwendiger Rechnungen, wie der Korrelation, bei Python schlechter ist. Somit haben wir uns für Octave entschieden. Um unseres selbstgeschriebenes Programm zu verifizieren haben wir eine Autokorrelation durchgeführt und diese mit der von Audacity berechneten AKF verglichen. Wir sind dabei zu dem Ergebnis gekommen, dass unser Programm funktioniert. Zur Aufnahme der Audiosignale haben wir ebenfalls die Software Audacity benutzt.

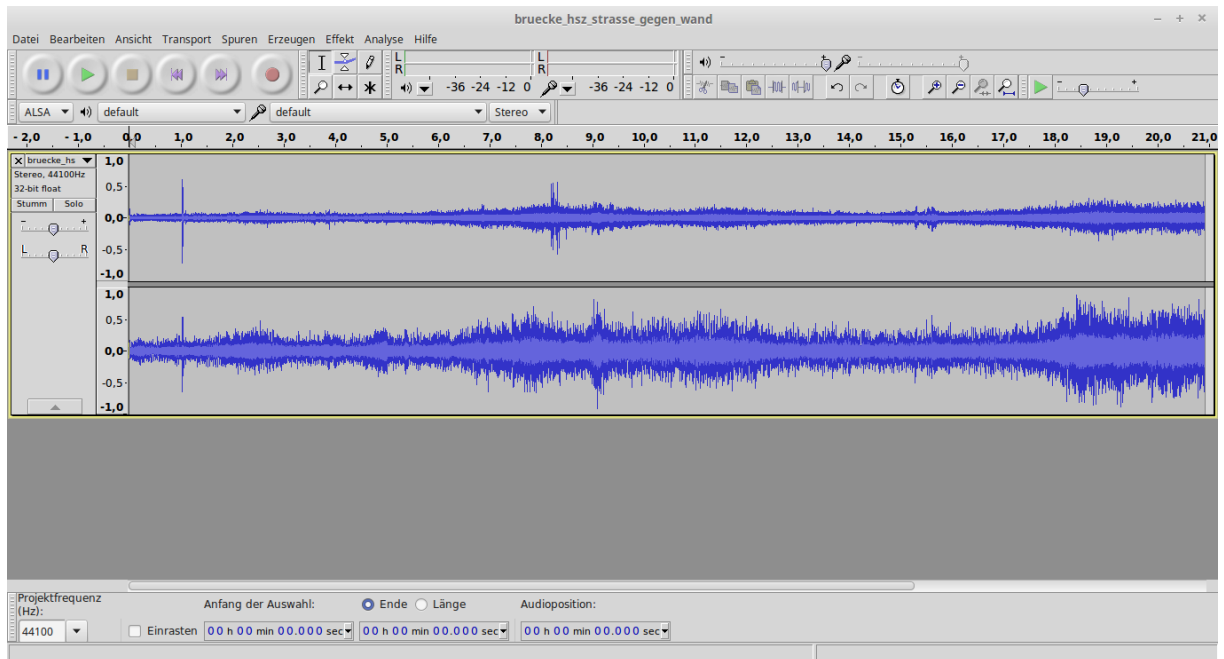


Abbildung 0.2: Screenshot des Aufnahmeprogramms Audacity

4.2 Hardware

Uns standen zwei hochwertige Kondensatormikrofone (M5 Matched Pair Compac 1/2" Cardioid Condenser Microphones von Rode) zur Verfügung. Diese Mikrofone sind für dieses Projekt besonders geeignet, da durch die Abstimmung (matched pair) nur die Unterschiede im Signal vor der Aufnahme Einfluss auf die Korrelation haben. Da es sich um Mikrofone mit Nierencharakteristik handelt, ist jedoch zu beachten, dass es sich bei den Aufnahmen um Gerichtete handelt und nicht wie bei Mikrofonen mit Kugelcharakteristik von einer Abbildung des gesamten gemessenen Raumes ausgegangen werden kann. Für die Digitalisierung der Signale stand uns ein hochwertiges Audio-USB-Interface (Scarlett 2i2 von Focusrite) zur Verfügung. Die Aufnahmen wurden in .wav gespeichert und sind somit verlustfrei. Außerdem konnten wir ein Stativ mit einer Mikrofonchiene verwenden, wodurch die Mikrofone konstanten Abstand hatten. Zur Erzeugung eines reproduzierbaren Klangsignales wurde eine portable Bluetooth-Anlage (Soundlink III von Bose) verwendet.

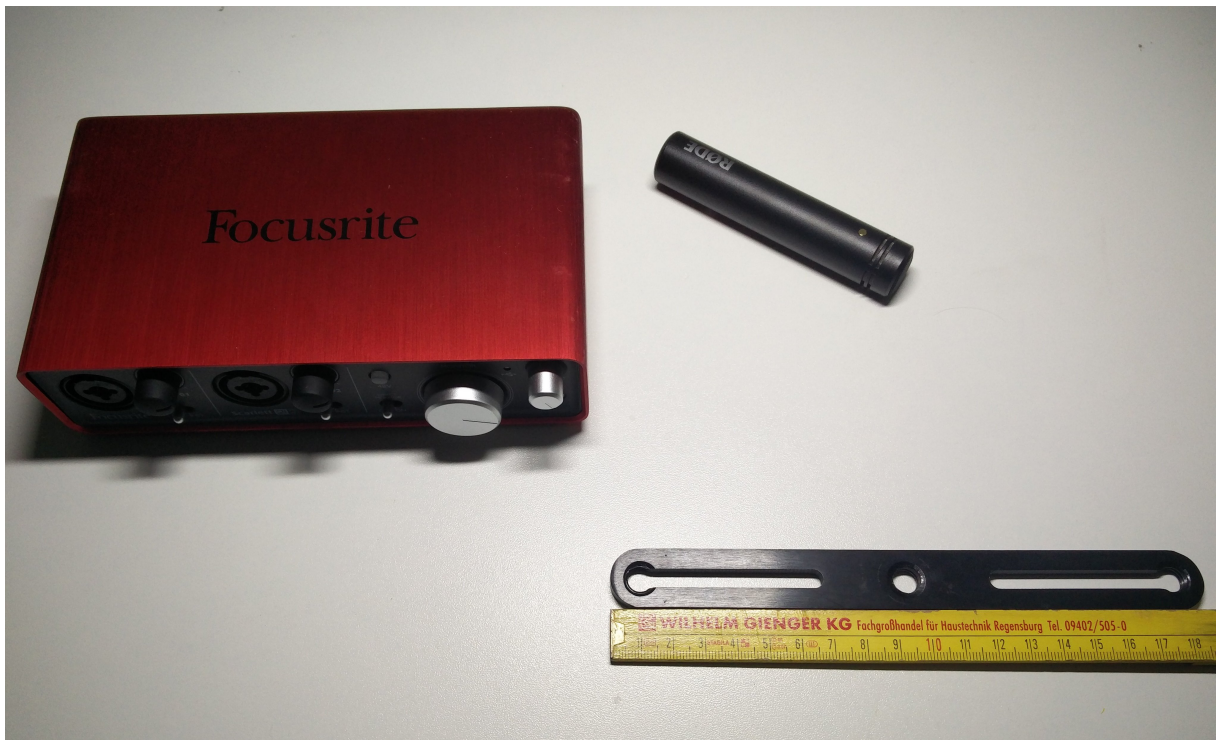


Abbildung 0.3: Verfügbare Technik

5 Programm

5.1 Aufbau des Programms

Das Programm ist in 3 funktionale Teile gegliedert. Dazu zählt das Einlesen der Audiodateien, die benötigte Signalverarbeitung inklusiver Berechnung der gewünschten Parameter und das Speichern der gewonnenen Werte in Form einer Excel-Datei. Die einzelnen Module werden in der Datei *main.m* zusammengeführt. Außerdem werden da die gewünschten Einstellungen überprüft und entsprechende Parameter an die verschiedenen Funktionen weitergegeben.

Einlesen der Audiodaten Die Audiodateien liegen im WAV-Format als Stereoaufnahme vor. Zunächst wird eine Liste mit allen Dateien in einem bestimmten Ordner erstellt, damit die Dateien nacheinander eingelesen werden können. Im nächsten Schritt werden die beiden Kanäle voneinander getrennt, um diese dann in die Signalverarbeitung zu übergeben.

Signalverarbeitung Das Kernstück der Signalverarbeitung ist eine periodische Korrelationsfunktion, die den linken und rechten Kanal miteinander korreliert. Die dabei entstandene Korrelationsfunktion wird dann weiter untersucht. Als nächstes wird eine Art Einhüllende berechnet, die ein Maß für die Steilheit der Kurve ist. Wie bereits im Abschnitt 3.2 beschrieben, wird dann mit Hilfe der Methode der kleinsten Quadrate eine Gauß-Glocke so angepasst, dass sie den Verlauf der Hüllkurve der KKF möglichst gut abbildet. Die Parameter ripple, σ , Gleichanteil und Zeitverschiebung des Maximums aus dem Ursprung werden danach an eine Funktion übergeben, die diese Daten in einer Excel-Tabelle speichert. Der Signalverarbeitungsteil hat auch die Möglichkeit aus einer Datei verschieden viele Abschnitte mit unterschiedlichen Längen zu korrelieren. Damit kann man aus relativ wenigen aufgenommen Signalen relativ viele Signal-Abschnitte gewinnen, die der Korrelation zugeführt werden.

Speicherung Die Speicherung der Daten erfolgt in einer Excel-Datei. Nach jedem Korrelationsdurchgang werden die berechneten Werte einer Matrix hinzugefügt. Dabei wird zu erst der Dateiname des

Samples und alles dazugehörigen Werte gespeichert. Desweiteren wird die Zeitdauer des Ausschnitts, sowie die Sampling-Rate gespeichert.

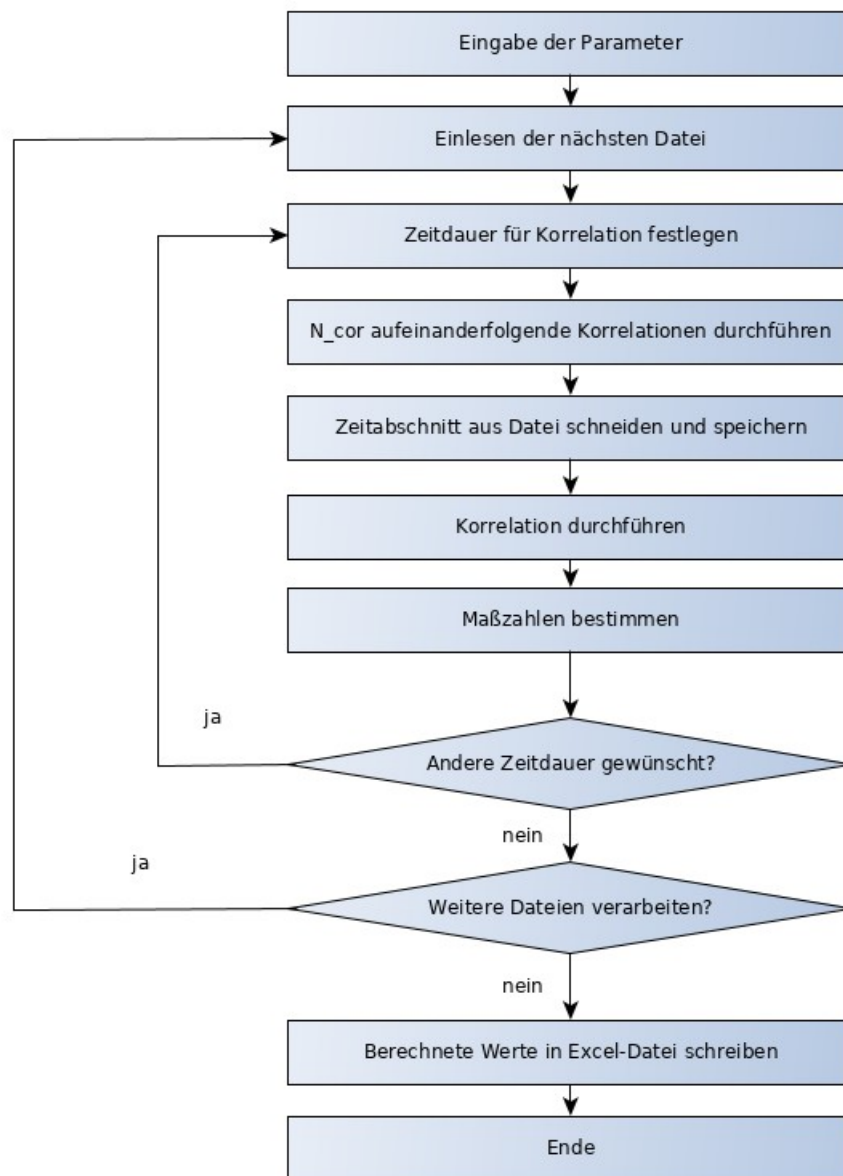


Abbildung 0.4: Programmablaufplan

5.2 Beispielparameter

Im Folgenden betrachten wir einige Zeilen Code, die verdeutlichen sollen, welche Möglichkeiten man zur Einstellung hat.

```
path = 'RECORDS\lucas';
excel_path = 'RESULTS\data.xlsx';
% results (display/save_param/save_all)
output = 'display';

% calculate correlation with (xcorr/freqMult)
calc = 'freqMult';
```

```

% plot in (samples/seconds)
x_axes = 'seconds';

% set priority (time/length)
priority = 'time';

% start of correlation in audio file in seconds
% time -> and durations
t_start = 8;
t_dur = [0.5 2];

% amount of correlations
Ncor_init = [4 1];

% length -> length of correlation in samples
Lcor = 8192;

```

In **Zeile 1 und Zeile 2** werden die Ein und Ausgabepfade definiert. Dabei spielt es für die Variable *path* keine Rolle, ob es sich um einen Ordner oder eine Datei handelt. Dies wird automatisch erkannt und im Programm entsprechend berücksichtigt.

In **Zeile 4** kann man einstellen, wie die Ergebnisse präsentiert werden. Die Option *display* zeigt die entstehenden Graphen auf dem Bildschirm an. Nur bei geringer Anzahl an Korrelationen zu empfehlen, da der Rechenaufwand sehr groß ist. *save_param* sorgt dafür, dass alle Parameter in einer Excel-Tabelle gespeichert werden, ohne dass man die Möglichkeit hat, sich die Graphen dazu anzuschauen. Die letzte Möglichkeit ist die Option *save_all*, welche dazu dient, die Diagramme zusätzlich zu der Exceldatei abzuspeichern.

In **Zeile 7** wird angegeben, welche Methode zur Berechnung verwendet wird. *xcorr* ist hierbei die Korrelation im Zeitbereich und benutzt die Octave-Funktion *xcorr*, welche zero padding betreibt. Nutzt man die Option *freqMult* wird das Ergebnis im Frequenzbereich berechnet, ohne dass es zu zero padding kommt. Die Berechnung im Frequenzbereich ist bei großen Blocklängen zu empfehlen, da es wesentlich schneller ist.

In **Zeile 10** kann eingestellt werden, ob die x-Achse der Diagramme in Sekunden oder Samples skaliert wird. Je nach Anwendungsfall zu entscheiden.

Zeile 13 gibt an, ob die folgenden Zeiteinstellungen oder die angegebene Blocklänge zu korrelieren ist. *time* ist hierbei offensichtlich die Priorisierung der Zeitangaben, während *length* die Blocklänge bevorzugt.

Zeile 17 gibt an ab wann die Verarbeitung starten soll. Im Beispiel wird alles was vor Sekunde 8 kommt, vernachlässigt.

In **Zeile 18** wird ein Array deklariert, welches Zeitdauern enthält. In diesem Beispiel werden Zeitdauern von 0.5 und 2 Sekunden benutzt.

Zeile 21 gibt nun in Verbindung mit **Zeile 18** an, wie oft die jeweilige Zeitdauer nacheinander korreliert werden soll. In unserem Beispiel heißt das, dass 4 hintereinanderliegende Blöcke mit einer Länge von 0.5 Sekunden verarbeitet werden und danach ein Block mit einer Länge von 2 Sekunden. Das Array welches durch *N_cor* definiert wird, muss immer genauso groß sein, wie das Array *t_dur*.

Zeile 24 gibt an wie lang die Blocklänge für die Korrelation sein soll. Wird nur mit der Option *length* der priority-Einstellung verwendet. Wird die Berechnung im Frequenzbereich durchgeführt, ist es sinnvoll, 2-er Potenzen zu benutzen, da die FFT nur mit Blocklängen umgehen kann, die eine Potenz von 2 sind. Ist dies nicht der Fall, wird der Rest der Werte mit 0 aufgefüllt.

5.3 Erweiterung der Analysemöglichkeiten

Um das Octave-Script um weitere Analysemethoden zu erweitern, ist vor allem die Funktion *analysis* interessant, die die folgende Signatur hat:

```
function [ripple, sigma, ex, area, lagDiff, timeDiff, envelope, reg_gauss,
        sorted, reg_exp] = analysis(correlation, lags, rate, x_axes)
```

Es gibt 4 Eingabeparameter. **correlation** ist ein Array, welches die Werte der diskreten Korrelation enthält. Die restlichen Parameter sind normale Werte. Alle zu berechnenden Maßzahlen werden dem Aufrufer der Funktion zurückgegeben. Um sinnvolle Plots erstellen zu können, werden unter anderem die Hüllkurve und die sortierten Werte zurückgegeben. Innerhalb der Funktion *analysis* werden nun einige Funktionen zur eigentlichen Analyse der Daten aufgerufen. Dazu zählen die Funktionen *calc_ripple*, *calc_reg* und *calc_envelope*. Sollen weitere Maßzahlen berechnet werden, müssen hier neue Funktionen hinzugefügt werden. Außerdem müssen die Funktionen *buildXLSMatrix* und *writeExcel* erweitert werden, damit die entsprechenden Kennwerte auch entsprechend abgespeichert werden.

5.4 Probleme

Zu Beginn unserer Arbeit wurde die Korrelation zuerst in Python implementiert. Allerdings nahm die Berechnungsdauer mit zunehmender Blocklänge stark zu. Da die .wav Dateien mit 44100Hz abgetastet werden werden die zu berechnenden Summen schnell sehr groß. Wie in der Theorie und im Abschnitt zur Software schon erklärt, wird nun Octave und die Fouriertransformation genutzt.

Darauf aufbauend war noch die Problemstellung der Bemessung der korrelierten Signale zu lösen. Dabei mussten Maßzahlen entstehen, die viele verschiedene Faktoren beachteten und möglichst gute Aussagen über „Peakyness“ und Streuung der KKF trafen. Die Idee war einen Fit mit einer Funktion wie der Gauß-Kurve, die als Parameter schon eine Verteilung beinhaltet. Allerdings ist die KKF nach der Berechnung sehr „eckig“ und weist viele Nullstellen auf. Um den Fit nicht durch diese Eigenschaften verzerren zu lassen, wird die KKF durch einen Tiefpass gefiltert. Auf die geglättete Kurve angewendet, ist die angewendete Regression aussagekräftiger.

6 Signalauswahl

Wir haben für die Aufnahme der Signale viel verschiedene Situationen ausgesucht, um ein möglichst breites Spektrum an Raum-Effekten zu erhalten. Aufnahmeorte waren beispielsweise der Platz vor dem HSZ, die Wiese zwischen Physik- und Mathematikgebäude, sowie der Treftzbau. Außerdem wurde in einer Wohnung gemessen, um Effekte von schallabsorbierenden Stoffen wie Teppich oder Bett zu erhalten. Soweit möglich, haben wir die natürliche Geräuschkulisse am jeweiligen Ort eingefangen. Zusätzlich dazu wurde ein definiertes Signal mittels eines Lautsprechers erzeugt, um Direktschall zu nutzen. Bei diesen Aufnahmen sollten die Effekte des Raumes am deutlichsten hervortreten.

6.1 Beispielsignale

Signal 1 - trefftz_wiese_m

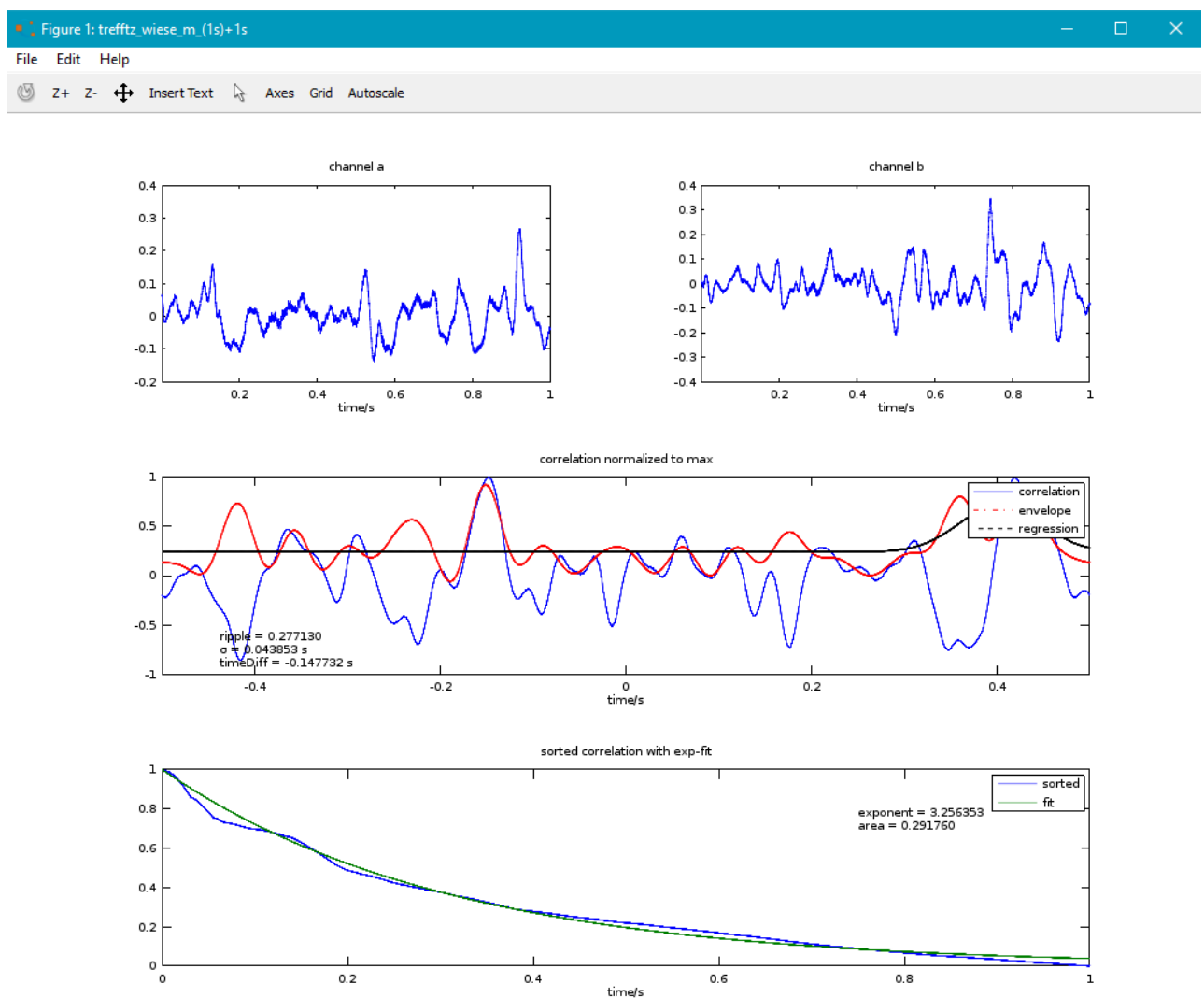


Abbildung 0.5: Signal 1

Aufnahmesituation Dieses Signal wurde auf der Wiese zwischen dem Gebäude der Mathematik- und Physikfakultät aufgenommen. Dies stellt eine relativ große Freifläche mit wenigen Hindernissen mit ungehinderter Schallausbreitung dar, wobei jedoch auch die umliegenden Gebäude einen Einfluss auf das Ausbreitungsverhalten haben können. Es wurde eine zusätzliche Primärschallquelle genutzt.



Abbildung 0.6: Aufnahmesituation auf der Wiese des Trefftzbaus

Signalbeschreibung Wie in Abbildung 0.5 zu sehen ist, ändern sich beide Kanäle relativ langsam. Sowohl Kanal A als auch Kanal B sind klar definiert und im Vergleich zum Rauschen relativ groß. Man erkennt jedoch bereits beim einfachen Betrachten, dass sich beide Seiten nur sehr geringfügig ähnlich sehen.

Beschreibung der KKF Die Kreuzkorrelationsfunktion schwankt sehr stark über den gesamten Zeitbereich. Deswegen ist auch die Hüllkurve stark schwankend. Da die Regression über die Hüllkurve berechnet wird, wird diese dem Signal auch nicht gerecht.

Auswertung der Maßzahlen Der ripple-Faktor von Signal 1 ist mit 0.2 vergleichsweise niedrig. Dass nur ein geringer Anteil der Energie des Signals in den obersten 5 Prozent der Werte vorhanden ist, lässt sich auch gut an der KKF erkennen. Es gibt nur sehr wenige Peaks. Die Aussagekräftigkeit von σ ist hier sehr gering. Wie gut zu erkennen ist, hat die Kurve keine Ähnlichkeit zu einer Gauß-Glocke. Auch die anderen Werte sind für dieses Signal schwierig zu bewerten, da die Ähnlichkeit generell sehr klein ist.

Signal 2 - trefftz_fahrstuhl_m

Aufnahmesituation Diese Aufnahme fand im Fahrstuhl des Trefftzbaus statt. Das heißt, der Raum war relativ klein und ist mit dicken Glaswänden versehen. Um ein Signal zu erhalten wurde eine Primärschallquelle in Form eines hochwertigen Lautsprechers genutzt über den ein Beispielsignal wiedergegeben wurde.

Signalbeschreibung In Abbildung 0.7 erkennt man sehr gut, dass sich beide Kanäle sehr schnell ändern und einen ähnlichen Verlauf haben. Lediglich die Lautstärke des Signals ist unterschiedlich. Das stört jedoch nicht für die Berechnung der Maßzahlen, da diese nicht von der Amplitude abhängig sind.

Beschreibung der KKF Für dieses Signal ist auch an der Kreuzkorrelationsfunktion klar zu sehen, dass es ein Maximum in der Mitte gibt. Das heißt, die Signale sind sich sehr ähnlich. Zu den Seiten

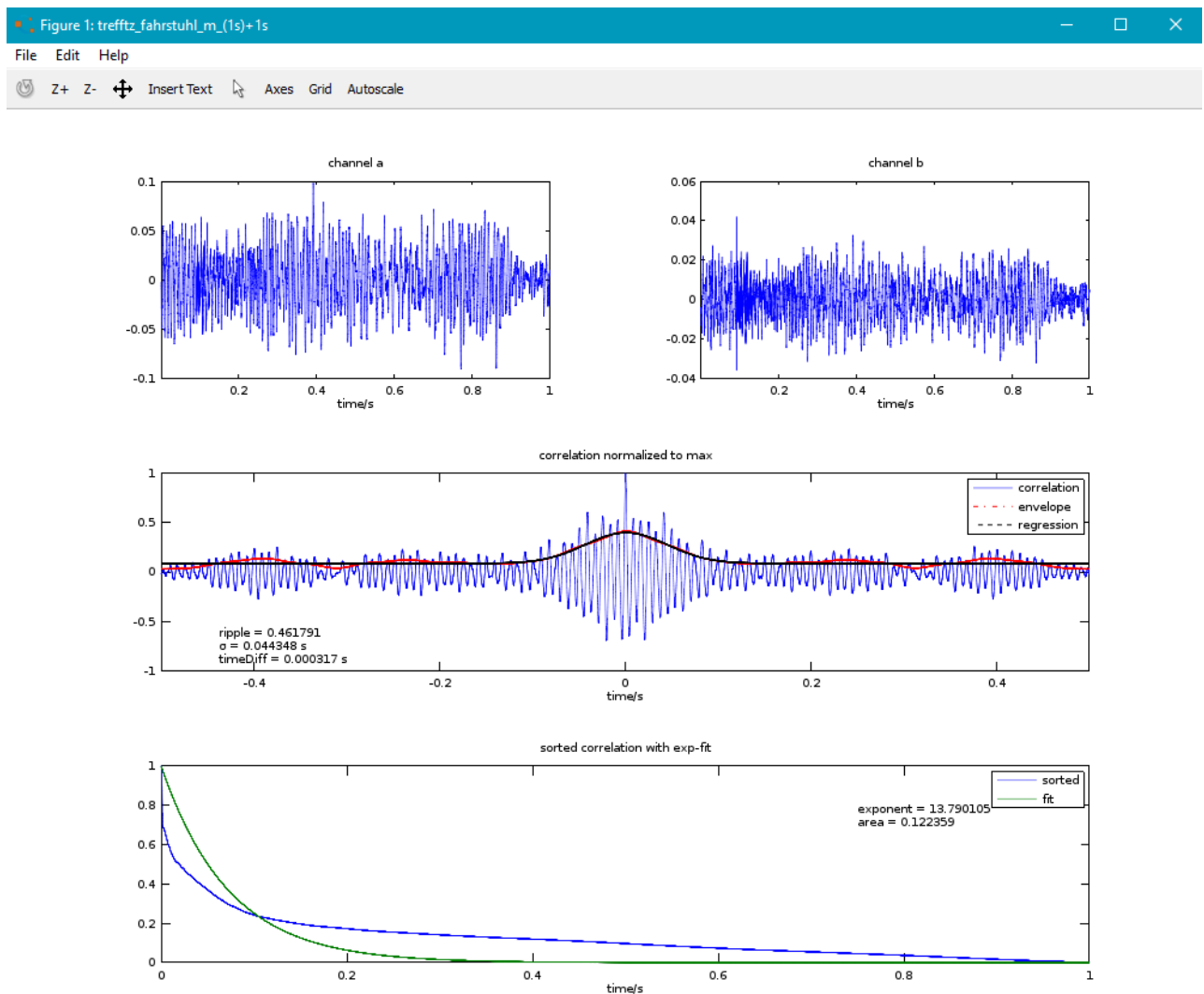


Abbildung 0.7: Signal 2

nimmt die KKF langsam ab. Für solch einen Verlauf ist die Aussage der Hüllkurve und der Regression sehr gut, da diese Kreuzkorrelation gut mit einer Gauß-Kurve approximiert werden kann.

Auswertung der Maßzahlen Der ripple-Faktor liegt mit 0.43 recht hoch, das heißt ein großer Teil der Signal-Energie liegt innerhalb des fünften Percentils. Die Aussage des Gauß-Fits ist hier wesentlich besser als bei Signal 1. Dies kann man gut in der Abbildung 0.7 erkennen, denn der Verlauf der Hüllkurve kommt der Form einer Gauß-Glocke recht nah. Die Zeitverschiebung des Maximums der KKF ist minimal. Daran ist zu erkennen, dass das Signal an beiden Mikrofonen nur mit sehr kleiner Verzögerung angekommen ist, was im begrenzten Raum des Fahrstuhl auch Sinn ergibt.

6.2 Probleme bei der Signalauswahl

Bei der Signalauswahl ergab sich das Problem, dass man möglichst viele verschiedene Raumsituationen erfassen musste, um eine große Menge verschiedener Daten zu bekommen. Dabei war es jedoch nur schwer möglich vor Ort zu entscheiden, ob die entsprechende Aufnahme sinnvolle Ergebnisse liefert. In den meisten Situationen war der Lautstärkepegel im Raum zu gering um 20s aufzunehmen, ohne dass der Großteil der Aufnahme aus Rauschen bestand. Aus diesem Grund haben wir ein zusätzliches Signal erzeugt. Dadurch gibt es jedoch in den meisten Aufnahmen eine Primärquelle, die das Spektrum



Abbildung 0.8: Aufnahmesituation im Fahrstuhl des Trefftzbaus

maßgeblich bestimmt.

6.3 Fazit

Abschließend lässt sich feststellen, dass der Gauß-Fit erst ab einem ripple-Faktor von 0.3 sinnvolle Ergebnisse liefert. Bei Werten die kleiner als 0.3 sind, ist die Hüllkurve einer Gauß-Kurve zu unähnlich. Es ist jedoch festzustellen, dass die Regression der Exponentialfunktion mit kleiner werdendem ripple besser über der nach Größe sortierten Amplituden liegt. Es ist empfehlenswert, Signale in Räumen aufzunehmen, die ausreichend klein sind, damit die Raumeffekte Auswirkungen auf das Signal haben. Sonst nimmt man größtenteils rauschen auf, welches sehr geringe Aussagen zulässt.

7 Zusammenfassung

Es wurde ein Octave-Skript entwickelt mit welchem sich die Kreuzkorrelationsfunktion der beiden Stereo-Kanäle einer Audioaufnahme berechnen lässt. Auf Basis der KKF wurden einige einfache Maßzahlen zur Charakterisierung der Aufnahmen entwickelt. Damit lassen sich für bestimmte Anwendungen Signale zu Testzwecken auswählen. In Zukunft kann die Software auf bestimmte Anwendungsfälle angepasst werden, in dem neue Regressionsmodelle implementiert werden, die der gewünschten Nutzung der Signale besser gerecht werden.

Literaturverzeichnis

[ISV] Rüdiger Hoffmann, Matthias Wolff: Intelligente Signalverarbeitung 1. Springer Verlag Berlin Heidelberg 2014

[NT] Prof. Dr.-Ing. Dr. h.c. Gerhard P. Fettweis: Einführung in die Nachrichtentechnik. Technische Universität Dresden, Fakultät Elektrotechnik, Vodafone Stiftungslehrstuhl Mobile Nachrichtensysteme, D-01062 Dresden Sommersemester 2015