

Hauptseminar Kommunikationssysteme  
Compressed Compute-and-Forward mit korrelierten Audiosignalen

Lucas Weber, Raphael Hildebrand, Florian Roth, Orell Garten

13.07.2016

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>1</b>
<b>2</b>	<b>Motivation</b>	<b>1</b>
<b>3</b>	<b>Theoretische Vorbetrachtung</b>	<b>1</b>
3.1	Kreuzkorrelationsfunktion . . . . .	1
3.1.1	Berechnungsvorschrift . . . . .	1
3.1.2	Kreuzkorrelation im Frequenzbereich und Faltungssatz . . . . .	2
3.2	Maßzahlen . . . . .	3
3.3	Gauß-Regression . . . . .	3
<b>4</b>	<b>Verfügbare Technik</b>	<b>3</b>
4.1	Software . . . . .	3
4.2	Hardware . . . . .	3
<b>5</b>	<b>Programm</b>	<b>3</b>
5.1	Aufbau des Programms . . . . .	3
5.1.1	Programmablaufplan . . . . .	4
5.2	Mögliche Einstellungen . . . . .	4
5.3	Probleme . . . . .	5
<b>6</b>	<b>Signalauswahl</b>	<b>5</b>
6.1	Beispielsignal . . . . .	5
6.2	Probleme bei der Signalauswahl . . . . .	5
<b>7</b>	<b>Zusammenfassung</b>	<b>5</b>

# 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 zur 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.

## 3 Theoretische Vorbetrachtung

### 3.1 Kreuzkorrelationsfunktion

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.

#### 3.1.1 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.

$$\psi_{\mathbf{X}\mathbf{Y}}(t_1, t_2) = E\{\mathbf{X}(t_1) \cdot \mathbf{Y}(t_2)\} \quad (1)$$

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.

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

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  $y(n)$  Realisierungen der Prozesse  $\mathbf{X}$  und  $\mathbf{Y}$  sind.

$$\boxed{\psi_{\mathbf{XY}}^E(k) = \sum_{n=-\infty}^{\infty} x(n) \cdot y(n+k)} \quad (3)$$

vgl. [ISV] S. 84 folgende

### 3.1.2 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 geeignete 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:

$$\psi_{\mathbf{XY}}^E(\tau) = x(-\tau) * y(\tau) \quad (4)$$

#### Zeit-diskreter Fall:

$$\psi_{\mathbf{XY}}^E(k) = x(-k) * y(k) \quad (5)$$

[ISV] Formel (2.217) S.89

**Faltungssatz in Verbindung mit KKF** Wir schreiben zuerst die KKF als Faltung.

$$\psi(t) = x(-t) * y(t) = \int_{-\infty}^{\infty} x(-\tau) \cdot y(t-\tau) d\tau \quad (6)$$

$$(7)$$

$$\underline{\Psi}(\omega) = \int_{-\infty}^{\infty} x(-t) * y(t) \cdot e^{-j\omega t} dt \quad (8)$$

$$= \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} x(-\tau) \cdot y(t-\tau) d\tau \right] \cdot e^{-j\omega(t-\tau+\tau)} dt \quad (9)$$

$$= \int_{-\infty}^{\infty} x(-\tau) \cdot e^{j\omega(-\tau)} \underbrace{\left[ \int_{-\infty}^{\infty} y(t-\tau) e^{-j\omega(t-\tau)} dt \right]}_{*} d\tau \quad (10)$$

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

$$\boxed{\underline{\Psi}(\omega) = \underline{X}^*(\omega) \cdot \underline{Y}(\omega)} \quad (11)$$

schreiben.

vgl. [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. Durch die IFFT von  $\underline{\Psi}(\omega)$  ergibt sich also die periodische KKF  $\tilde{\psi}(t)$ . vgl. [ISV] S.135

## 3.2 Maßzahlen

## 3.3 Gauß-Regression

# 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.

## 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 Signale vor der Aufnahme Einfluss auf die Korrelation haben. Für die Digitalisierung der Signale stand uns ein hochwertiges Audio-USB-Interface zur Verfügung (Scarlett 2i2 von Focusrite) zur Verfügung. Die Aufnahmen wurden in .wav gespeichert und sind somit verlustfrei. Außerdem konnten wir ein Stativ mit einer Mikrofonschiene verwenden, wodurch die Mikrofone konstanten Abstand hatten. Zur Erzeugung eines reproduzierbaren Klangsignales wurde eine portable Bluetooth-Anlage (Soundlink III von Bose) verwendet.

# 5 Programm

## 5.1 Aufbau des Programms

Das Programm ist in 3 große 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.

**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 3.3 "Gauß-Regression" 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  $\text{ripple}$ ,  $\sigma$ , Gleichanteil und Zeitverschiebung des Maximums aus dem Ursprung werden danach an eine Funktion übergeben, die diese Daten in einer Excel-Tabelle speichert.

**Speicherung** Die Speicherung der Daten erfolgt in einer Excel-Datei. Dabei wird zu erst der Dateiname des Samples und alles dazugehörigen Werte gespeichert. Außerdem wird noch ein Link zum Graphen der Korrelationsfunktion angegeben, damit man sich diese bei der Auswahl der Test-Signale anschauen kann.

### 5.1.1 Programmablaufplan

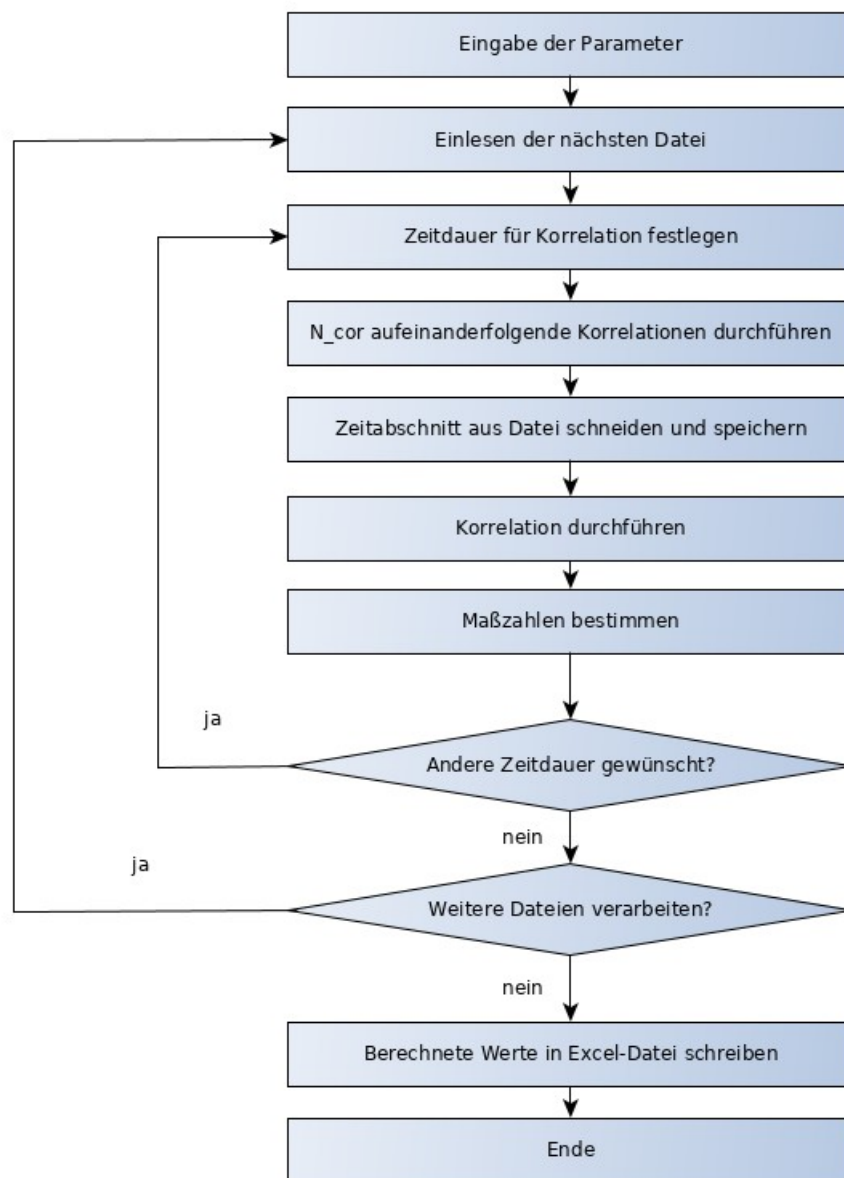


Abbildung 1: Programmablaufplan

## 5.2 Mögliche Einstellungen

Im Code gibt es diverse Einstellungen, die das Verhalten des Programms je nach Wunsch des Anwenders verändern. Diese sind am Beginn der main.m-Datei festgelegt und werden im folgenden beschrieben:

- *path* - Pfad zur Sammlung der WAV-Dateien
- *excel\_path* - Pfad unter dem Excel-Datei mit Lösungen gespeichert wird
- *output* - Unterscheidung, ob Ergebnisse gespeichert oder angezeigt werden
- *calc* - Wechsel zwischen Berechnung im Zeit- und Frequenzbereich möglich.
- *x\_axes* - Unterscheidung, ob Korrelation gegen Samples oder Zeit aufgetragen wird
- *priority* - Unterscheidung, ob angegebene Blocklänge oder Zeitdauer priorisiert wird

- $t\_start$  - Startzeitpunkt der Korrelation
- $t\_dur$  - Array  $A1$  mit Menge an Zeitdauern die korrelierten werden sollen, Korrelation beginnt immer bei  $t\_start$
- $Ncor\_init$  - Array, mit identischer Länge zu  $A1$ . Gibt an wie oft korrespondierender Eintrag in  $A1$  hintereinander korreliert wird.
- $Lcor$  - Blocklänge der Korrelation

### 5.3 Probleme

## 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 Trefftzbau. 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 Beispielsignal

### 6.2 Probleme bei der Signalauswahl

## 7 Zusammenfassung

## Literatur

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