

Hauptseminar Kommunikationssysteme  
Compressed Compute-and-Forward mit korrelierten  
Audiosignalen

Florian Roth, Raphael Hildebrand, Lucas Weber, Orell Garten

13.07.2016

**Abstract**

abstract

# Contents

<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	Korrelationsfunktion . . . . .	1
3.2	Parameter . . . . .	1
3.3	Gauß-Regression . . . . .	1
<b>4</b>	<b>Verfügbare Technik</b>	<b>1</b>
4.1	Software . . . . .	1
4.2	Hardware . . . . .	2
<b>5</b>	<b>Programm</b>	<b>2</b>
5.1	Aufbau des Programms . . . . .	2
5.2	Mögliche Einstellungen . . . . .	2
5.3	Probleme . . . . .	3
<b>6</b>	<b>Signalauswahl</b>	<b>3</b>
6.1	Beispielsignal . . . . .	3

# 1 Aufgabenstellung

Es ist ein Programm zu entwerfen, welches Audiodateien einliest und nacheinander die beiden Stereo-Kanäle miteinander korreliert. Des weiteren sollen Parameter berechnet werden, welche die entstehende Korrelationsfunktion möglichst genau beschreiben. Dafür sind Audiosignale aufzunehmen, mit der geschriebenen Software zu verarbeiten 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.

Ein Beispiel für eine zukünftige Anwendung ist ein riesiges Sensornetzwerk. Es stellt sich dabei die Frage wie viele Sensoren man benötigt, um sinnvolle Aussagen treffen zu können. Netzwerkcodierung in Verbindung mit Compressed Sensing ermöglicht es hierbei die Datenmenge im Netzwerk mittels Korrelationsverfahren zu reduzieren und damit den Durchsatz zu erhöhen.

# 3 Theoretische Vorbetrachtung

## 3.1 Korrelationsfunktion

## 3.2 Parameter

## 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

Es wurden zwei hochwertige Kondensatormikrofone vom Telekom-Lehrstuhl der TU Dresden zur Verfügung gestellt. Außerdem konnten wir ein Stativ, eine Stereoschiene sowie eine USB-Soundkarte verwenden.

## 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 "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.

### 5.2 Mögliche Einstellungen

Im Code gibt es diverse Einstellungen, die das Verhalten des Programms je nach Wunsch des Anwenders verändern:

- *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
- *priority* - Unterscheidung, ob angegebene Blocklänge oder Zeitdauer priorisiert wird
- *x\_axes* - Unterscheidung, ob Korrelation gegen Samples oder Zeit aufgetragen wird
- *Lcor* - Blocklänge der Korrelation
- *Ncor* - Anzahl verschiedener Blöcke pro Signal
- *t\_start*, *t\_end* - Start- und Stopzeit

### 5.3 Probleme

## 6 Signalauswahl

### 6.1 Beispielsignal