

Digitale Signalverarbeitung

Florian Schiel/Lasse Bombien

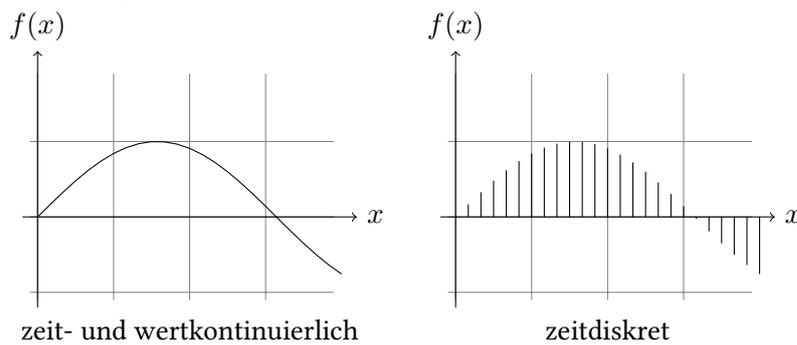
Seminar Werkzeuge der Sprachverarbeitung
(Sommersemester 2012)

VORTRAG

5 Digitale Signale

Für die Verarbeitung am Computer müssen in der Natur vorkommende Signale zeit- und wertdiskretisiert werden

Abtastung



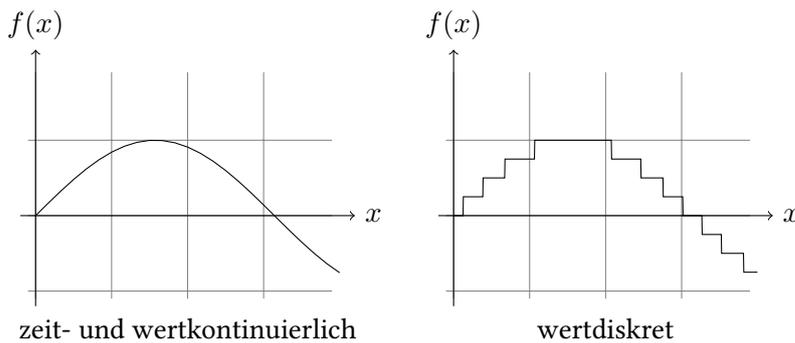
Die Abtastrate oder sampling rate bestimmt, wieviele Werte pro Sekunde gemessen werden, und damit auch indirekt den Abstand zweier Abtastwerte auf der Zeitachse. Je schneller sich ein Signal mit der Zeit ändert, desto höher muss logischerweise auch die Abtastrate sein.

Sprache: 8000 - 48000 Hz

Musik: 44100 Hz

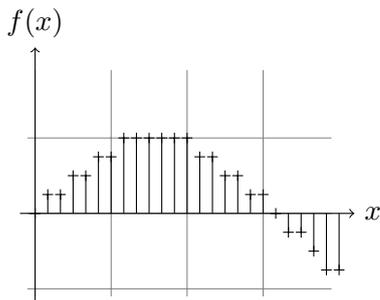
EMA: 200 Hz

Quantisierung



Die Wortbreite oder Bitauflösung bestimmt, mit welcher Genauigkeit das Signal beschrieben wird. Allgemein üblich ist heutzutage eine Bitauflösung von 16 bit. D.h. das Signal kann Werte zwischen -32768 und +32767 annehmen. 16 Bit entspricht schon HiFi; für ein Telefon-Signal z.B. genügen 8 Bit (+/- 128). Ein Abtastwert wird auch häufig 'Wort' genannt; daher der Begriff „Wortbreite“. Ein Signal - meistens handelt es sich in unserem Institut um ein Sprachsignal - wird z.B. mit einem Mikrophon aufgenommen, d.h. die akustischen Schwingungen in elektrische umgewandelt, und anschliessend in äquidistante Abtastwerte digitalisiert, die in einem File (Audio-, Sound-File) abgespeichert werden. Da es viel effektiver ist, Werte als sog. binäre Daten abzuspeichern, anstatt als lesbaren Text, können die meisten Audio-Files nicht mit dem Editor angeschaut werden. (Es gibt aber UNIX Befehle, die das machen, z.B. „od“.)

Zeit- und wertdiskretisiert



zeit- und wertdiskret

Es gibt eine endliche Anzahl von Datenpunkten (keine unendliche). Die Präzision (der Wertebereich) ist auf ein endliches Maß reduziert. Beispiel: Minute Sprache bei 16 kHz und 16 bit: $60 \text{ s} \cdot 16 \text{ kHz} \cdot 2 \text{ B} = 1920 \text{ kB} = 1,92 \text{ MB}$ 16 bit Wertebereich: -32.768-32.767

Byteorder

Da Computer grundsätzlich ihre Daten in Blöcken von je 8 Bit (= 1 Byte) anordnen, gibt es bei Bitauflösungen größer 8 Bit außerdem das Problem der „Byte-Reihenfolge“ (Byte-order). Auf Maschinen mit Intel- und Digital-Prozessoren kommt das *untere* Byte zuerst im File, danach das *obere* Byte. Bei Motorola-, Sun- und vielen anderen Prozessoren ist es genau umgekehrt. Beispiel:

		unteres	oberes Byte
INTEL	('small endian', '01', 'low-high') :	10110100	00010110

MOTOROLA ('big endian', '10', 'high-low') : 00010110 10110100

Meistens brauchen wir uns um die Bytereihenfolge nicht zu kümmern.

6 Sound eXchange - sox

Sound eXchange -sox

Das open source sox Programm ist für alle Betriebssysteme erhältlich. Wir haben es schon als universellen Dateiformat-Konverter kennengelernt. sox kann aber sehr viel mehr: z.B. Filtern, re-sampling, Kanäle neu mischen oder zusammenfassen, alle möglichen Effekte anwenden etc.

Die Syntax des sox Kommandos ist ausführlich in der man page beschrieben. Im Allgemeinen sieht sie so aus:

```
sox [generell opt] [input opt] input [output opt] output [effect [effect opt]]
```

Gemeinsame Übung

Beispiele:

Wir wollen ein NIST audio file rückwärts abspielen und in einem Microsoft wav file speichern:

```
sox -V -t sph signal.nis signal.wav reverse
```

-V allgemeine Option 'verbose', die bewirkt, dass sox Informationen ausgibt

-t sph input Option 'type' definiert den Formattyp für das Inputfile als NIST SHERE (sox erkennt die Extension 'nis' nicht!)

signal.nis : input File

signal.wav : output File (sox erkennt an der Extension 'wav', dass es sich um ein Microsoft wav File handelt und konvertiert automatisch)

reverse: Effekt reverse bewirkt Umdrehen des Signals

Wir wollen die Abtastrate von 22050Hz auf 16000 Hz senken (down sampling):

```
sox -V -t sph signal.nis -r 16000 signal.wav
```

-r 16000 : output Option Abtastrate bei 16000Hz

Wir wollen ein Stereo Sound-File ohne Header (raw sound file) in ein Mono Microsoft wav File umwandeln. Dazu müssen wir die Abtastrate, die Anzahl der Bits pro Abtastwert (Wortlänge) und die Kodierung der Abtastwerte (coding) kennen:

```
sox -V -t raw -r 22050 -2 -s -c 2 signal.raw -c 1 signal.wav
```

-t raw input Option Formattyp ist raw

-r 20050 input Option Abtastrate ist 22050Hz
-2 input Option zwei Byte pro Abtastwert
-s input Option signed integer, also -37676 ... +37677
-c 2 input Option 2 Kanäle (Stereo)
 signal.raw : input File
-c 1 output Option 1 Kanal (Mono)
 signal.wav : output File

Wir wollen ein Microsoft wav File mit zwei Kanälen so auftrennen, dass jeder Kanal in einem separaten File gespeichert ist (split):

```
sox -V stereo.wav -c 1 kanal_links.wav mixer 1,0  
sox -V stereo.wav -c 1 kanal_rechts.wav mixer 0,1
```

- c 1 output Option 1 Kanal
- mixer : Effekt Mischer
- 1,0 Option zu Effekt 'mixer': erster (linker) Kanal bleibt gleich, zweiter Kanal (rechts) wird unterdrückt (mit Null multipliziert)
- 0,1 Option zu Effekt 'mixer': erster (linker) Kanal wird unterdrückt (mit Null multipliziert), zweiter Kanal (rechts) bleibt unverändert

7 Kleine Übung mit Audacity

Audacity – ein Soundeditor

Das Open Source Programm Audacity¹ (verfügbar auf vielen Plattformen) können komplexe Audioprojekte erstellt und verarbeitet werden. Für uns ist Audacity besonders interessant als Aufnahmeprogramm und zum Inspizieren sehr großer Audiofiles.

Erste Aufnahme

- Überlegen Sie sich eine kurze Äußerung und lernen Sie, diese rückwärts zu sprechen.
- Verwenden Sie Audacity, um diese Äußerung aufzunehmen und als WAV Datei zu exportieren.
- Verwenden Sie SOX, um die Aufnahme zu reversieren und spielen Sie sie ab.

Vortrag

¹<http://audacity.sourceforge.net/>

8 ASSP

ASSP Library

Eine Funktionsbibliothek zur Handhabung, Manipulation und Analyse von Sprachsignalen (Advanced Speech Signal Processing).

- Lesen und Schreiben von Sprachsignalen (und abgeleiteten Daten) in unterschiedlichen Formaten.
- Manipulation solcher Signale/Daten: ausschneiden, zusammenfügen, Kanäle trennen oder zusammenführen etc.
- Konvertierung von Audioformaten
- Verarbeitung und Analyse von Sprachsignalen: Hauptfunktionen liegen als Kommandozeilenprogramme vor.

ASSP Tools

Kommandozeilenprogramme

Es gibt 8 Analyseprogramme und 2 Verarbeitungsprogramme.

Analyseprogramme

Analyseprogramme leiten parametrische Daten ab. Das Ergebnis ist kein Signal mehr, es hat einen anderen Datentyp. Beispiel: Formantanalyse `forest`.

Verarbeitungsprogramme

Andere Programme verändern das Signal oder leiten es ab, ohne den Datentyp zu ändern. Beispiel: Signalfilterung `affilter`.

Sonstige

`afconvert` ist ein kleines SOX,

`ffinfo` gibt Informationen über eine Datei aus.

Übung

`ffinfo`

„File Format Info“. Aufruf: `ffinfo <Datei> [<Datei>]`
Hilfe zum Programm

```
cip1 % ffinfo  
cip1 % ffinfo -h
```

```
cip1 % ffinfo primadanke.wav
```

afconvert

„Audio Format Converter“. Aufruf:

```
afconvert <opts> <file> <opts> <file> <opts>
```

Hilfe zum Programm

```
cip1 % afconvert
```

```
cip1 % afconvert -h
```

Ein paar Beispielanwendungen

```
cip1 % aplay primadanke.wav
```

```
cip1 % afconvert primadanke.wav -Rs=20000 -of=primadanke20.wav
```

```
cip1 % aplay primadanke20.wav
```

(Weitere Abtastraten ausprobieren)

Zerschneiden

```
cip1 % afconvert primadanke.wav -e=0.501360 -of=p1.wav
```

```
cip1 % afconvert primadanke.wav -b=0.501360 -of=p2.wav
```

Formatumwandlungen

```
cip1 % aplay msdjc002.ssd #klappt nicht...
```

```
cip1 % ffinfo msdjc002.ssd
```

```
cip1 % afconvert msdjc002.ssd -F=WAV
```

```
cip1 % ffinfo msdjc002.wav
```

```
cip1 % aplay msdjc002.wav
```