

# ASSP und Vertiefung Grundfrequenzanalyse

Florian Schiel/Lasse Bombien

Seminar Werkzeuge der Sprachverarbeitung  
(Sommersemester 2012)

## Vorlesung

### ASSP Tools – Analyseprogramme

**acfana** Berechnung der Autokorrelationsfunktion

**f0\_ksv** Grundfrequenzanalyse (Schäfer-Vincent Verfahren)

**f0\_mhs** Grundfrequenzanalyse (Harmonic-Sieve Verfahren)

**forest** Formantanalyse

**rfcana** Linear Prediction Analyse

- Reflexionskoeffizienten  
oder Flächenfunktion  
oder “log area ratios”  
oder LPC Filter Coeffizienten

**rmsana** “Root Mean Square” Amplitude in dB: Energie des Signals.

**spectrum** Verschiedene Spektralanalysen:

- Diskrete Fourier Transformation  
oder LPC geglättetes Spektrum  
oder Cepstral geglättetes Spektrum  
oder Cepstrum

**zcrana** Berechnung der Nulldurchgangsrate

Der Output wird normalerweise im SSFF Format gespeichert. Man kann aber die Ausgabe als Text erzwingen (und das machen wir heute auch so!).

## ASSP Tools –Verarbeitungsprogramme

**affilter** Filterung eines Signals

- Hochpass
- Tiefpass
- Bandpass
- Bandsperfilter

**diffsig** Ableitung eines Signals (z.B. Geschwindigkeit der Änderung)

Die Ausgabe ist normalerweise im gleichen Format wie die Eingabe, es sei dann man bestimmt etwas anderes.

### Optionen

```
cip1 % zcrana -X
```

```
[...]
```

```
Options:
```

```
-h/--help  print this text
-b=<time>  set begin of analysis interval to <time> seconds
            (default: begin of file)
-e=<time>  set end of analysis interval to <time> seconds
            (default: end of file)
-s=<dur>   set analysis window shift to <dur> ms
            (default: 5.0)
-S=<dur>   set analysis window size to <dur> ms
            (default: 25.0)
-oA        output in plain ASCII
```

```
[...]
```

```
-of=<file> set name of output file to <file>
            (only in single-file mode; overrules -od option)
```

```
[...]
```

### Bedeutung der Optionen

Eine wichtige Option zum Erhalten von Hilfe ist die Option `-X`. Sie listet alle vorhandenen Optionen der Programme mitsamt des Hilfetextes auf. Das Listing oben enthält Optionen, die in nahezu allen Programmen eine Rolle spielen und über deren Bedeutung man Bescheid wissen sollte. `-b` und `-e` bestimmen den Beginn und das Ende des zu analysierenden Abschnitts in Sekunden. Werden sie nicht explizit gesetzt, wird als *default* der Anfang und das Ende der Eingabe angenommen. Wird nur der Beginn spezifiziert, endet die Analyse am Signalende. Genauso beginnt die Analyse immer am Signalanfang, wenn nur das Ende spezifiziert ist.

Viele Analysen verarbeiten ein Signal nicht „am Stück“, sondern verwenden ein „Fenster“, um einen kurzen Signalabschnitt nach dem anderen zu verarbeiten. Die Analyse jedes Fensters liefert genau einen Datensatz, der unterschiedlichen Umfang hat (1 Wert für F0 Analyse, aber normalerweise 4 Formantwerte usw.). Die Länge dieses Fensters kann in vielen Fällen mit der Option `-s` angegeben werden. Mit der Länge des Fensters legt man unter anderem fest, wie empfindlich die Analyse auf Veränderungen im Signal reagiert. Um kurzfristige Änderungen berücksichtigen zu können, benötigt man ein kleines Fenster, für langsamere Änderungen genügt ein längeres Fenster. Wenn die Verarbeitung des Fensters abgeschlossen ist, wird das Fenster weitergeschoben, um die Verarbeitung über dem nächsten Abschnitt zu beginnen. Die Option `-s` bestimmt dabei, wie weit das Fenster verschoben wird. Meistens ist die „Fensterverschiebung“ kleiner als die Fensterlänge, damit die zu analysierenden Signalabschnitte überlappen. Die Fensterverschiebung hat eine weitere, ganz konkrete Bedeutung, da sie die Datenrate der Analyse bestimmt: Wird das Fenster immer um 5 ms verschoben, so erhält man für die Analyse von 1 Sekunde Sprachdaten genau 200 Datensätze.

$$\frac{\frac{1000\text{ms}}{5\text{ms}}}{\text{s}} = 200 \frac{1}{\text{s}} = 200\text{Hz}$$

Es gibt einige Optionen, die Ausgabe beeinflussen. Alle fangen mit `-o` an. `-oA` sorgt dafür, dass die Ausgabe im ASCII Format gespeichert wird, also als Text und nicht im binären SSFF Format. Mit `-of=filename` kann der Name der Ausgabedatei direkt angegeben werden.

Für viele der Analyseprogramme gibt es spezielle Optionen, die man sich durch den „nackten“ Aufruf des Programms auflisten lassen kann, z.B.: `cip1 % forest`.

Durch anhängen der Option `-X` erhält man zusätzliche Information zu den erweiterten Optionen. Im Folgenden wenden wir uns der Grundfrequenzanalyse zu und betrachten einige der spezielleren Optionen.

## Grundfrequenzanalyse

```
cip1 % f0_ksv -X
[...]
-g=<code> set gender-specific F0 ranges; <code> may be:
          "f[emale]" (80.0 - 640.0 Hz)
          "m[ale]" (50.0 - 400.0 Hz)
          "u[nknown]" (default; 50.0 - 600.0 Hz)
-M=<freq> set maximum F0 value to <freq> Hz
-m=<freq> set minimum F0 value to <freq> Hz
          (absolute minimum: 10.0)
-oA      output in XASSP ASCII format
[...]
-of=<file> set name of output file to <file>
          (only in single-file mode; overrules -od option)

cip1 % f0_ksv primadanke.wav
```

```
cip1 % f0_ksv -g=f -oA affaere.wav
```

```
cip1 % less affaere.f0
```

### Geschlecht und Grundfrequenz

Es ist kein Geheimnis, dass sich die Geschlechter u.a. in der durchschnittlichen Grundfrequenz unterscheiden. Dieser Unterschied hat auch Auswirkungen auf einige Signalanalysen, darunter die Grundfrequenzanalyse selbst. Auf der Suche nach Periodizität können die Algorithmen ausreißern und dem kann man entgegenwirken, indem man den Frequenzbereich eingrenzt, in dem gesucht werden soll. Die Genderoption `-g` definiert drei Standardbereiche:

- `-g=f` „female“ definiert den Bereich von 80 Hz – 640 Hz
- `-g=m` „male“ definiert einen Bereich der etwas niedriger anfängt und auch nicht so weit hinauf geht: 50 Hz – 400 Hz
- `-g=u` „undefined“ bedeutet, dass das Programm die Standardbereiche für beide Geschlechter abdeckt.

In einigen Fällen kann es notwendig sein, diesen Bereich für bestimmte Sprecher genauer einzustellen. Dies kann über die Optionen `-m` für das untere Ende und `-M` für das obere Ende erreicht werden.

### Kurzer Exkurs: AWK

- AWK ist ein Scripttool zur Verarbeitung von Textdateien. Der Inhalt einer Datei wird zeilenweise behandelt. Dabei ist jedes „Wort“ einer eigenen Variable zugewiesen.
- AWK ist besonders gut geeignet, um Tabellen mit fester Anzahl von Spalten zu bearbeiten.
- AWK ist gut geeignet, um mit regulären Ausdrücken zu arbeiten und für jedes Muster eine bestimmte Aktion durchzuführen.
- Die simpelste Aufruf lautet: `cip1 % gawk <Program> <Input-File>`
- Ein Programm hat immer die Form: `<Muster> <Aktion>`
- Beispiel: `cip1 % gawk '/^ORT/ { print $3 }' g121axx0_000_OLV.par`
- Das Muster darf fehlen, dann wird jede Zeile bearbeitet: `cip1 % gawk '{ print $3 }' g121axx0_000_OLV.par`
- Komplexe Programme können als Script gespeichert werden, damit sie nicht immer wieder eingetippt werden müssen. Außerdem will man häufig viele Dateien auf einmal verarbeiten: `cip1 % cat file1 file2 file3 ... | gawk -f <ProgrammDatei>`

## Ein simples awk-Skript für F0 Daten (f0average.awk)

```
BEGIN {
    count = 0
    f0sum = 0
    total = 0
}
/^XASSP.*/ {
    next
}
/^ / {
    if ($2 > 0) {
        count++
        f0sum = f0sum + $2
    }
    total++
}
END {
    print count " of " total " values voiced"
    print "Mean F0: " f0sum/count
}
```

### Erklärung des Skripts

Das Skript enthält 4 Programmblöcke. Zwei davon sind in gewisser Weise speziell. Sie beginnen mit den Schlüsselwörtern `BEGIN` und `END`. Die enthaltenen Anweisungen werden dementsprechend ausgeführt bevor der Eingabestrom verarbeitet wird, bzw. nachdem der Eingabestrom verarbeitet wird. Die übrigen Blöcke werden auf jede Zeile im Eingabestrom angewendet.

Der `BEGIN` Block definiert drei Variablen und setzt ihren Wert auf Null. Es handelt sich um `count`, einen Zähler für alle F0-Werte  $> 0$ , `f0sum`, die Summe aller F0-Werte  $> 0$ , sowie `total`, einen Zähler für *alle* F0-Werte.

Der nächste Block wird ausgeführt, falls am Anfang der ZEILE die Zeichenkette „XASSP“ erscheint. In diesem Fall wird die Verarbeitung der Zeile abgebrochen und zur nächsten Zeile gesprungen. Auf diese Weise kann das Skript die erste Zeile in der Ausgabedatei von `f0_ksv` ignorieren, die ja auch keine Werte enthält.

Der wiederum nächste Block passt auf alle Zeilen, die mit einem Leerzeichen beginnen. Dies trifft in den F0 Dateien auf alle Zeilen mit regulären Werten zu. Davon gibt es zwei (Zeitpunkt und F0-Wert), die dementsprechend über die Variablen `$1` und `$2` angesprochen werden können. Das Program testet nun, ob der F0-Wert in `$2` größer ist als 0. Ist das der Fall, wird `count` um 1 erhöht und `f0sum` um den Wert von `$1`. In jedem Fall wird `total` um 1 erhöht. Sind alle Zeilen abgearbeitet, gibt das Skript die Werte von `count` und `total` aus und berechnet den Quotienten aus `f0sum` und `count`, also den Mittelwert aller F0-Werte  $> 0$ .

- Anwendung:

```
cip1 % awk -f f0average.awk affaere.f0  
cip1 % cat affaere.f0 | awk -f f0average
```

### gemeinsame Übung

#### Übung

- Berechnen Sie die Grundfrequenz für die Datei `primadanke.wav`, sodass Sie sie mit dem AWK Script verarbeiten können.
- Berechnen Sie die Grundfrequenz für die Datei `affaere.wav` mit allen Möglichkeiten der "Gender" Option. Verwenden sie die Option "-of" oder "-ox" zum Abspeichern der drei verschiedenen Ergebnisse.
- Wenden sie das AWK Script auf die drei Versionen an. Gibt es einen Unterschied?
- Versuchen Sie, die Optionen "-m" und "-M" für die beiden Dateien `affaere.wav` und `primadanke.wav` möglichst optimal einzustellen.