

**Technische Universität Berlin**



# Labor Kommunikationstechnik

Prof. Dr. Stefan Weinzierl

Sprachsignalanalyse – Arbeiten mit PRAAT

Dozent: Daniel Pape

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1. Einführung / Vorbemerkung</b>                               | <b>3</b>  |
| <b>2. Nutzung und Aufbau</b>                                      | <b>4</b>  |
| <b>Download</b>   | <b>4</b>  |
| <b>Programmteile und Plattformen</b>                              | <b>4</b>  |
| Objekte, Fenster, Editoren, Tiers, Matrizen                       | 4         |
| Start-Up-Script („buttons.ini“), PRAATCON                         | 5         |
| Objekte   | 6         |
| „Picture“ Fenster   | 8         |
| „Manipulation“  | 8         |
| Hilfe   | 9         |
| <b>3. Skripte</b>   | <b>11</b> |
| <b>Ausführung von Skripten</b>                                    | <b>11</b> |
| <b>Paste History</b>  | <b>12</b> |
| <b>Variablen (numerisch und Text)</b>                             | <b>13</b> |
| <b>Die Befehler „selected“ und „editor“</b>                       | <b>14</b> |
| <b>Übergabe von Parametern (Substitution von festen Namen)</b>    | <b>15</b> |
| <b>Eingaben des Benutzers und Pause</b>                           | <b>15</b> |
| <b>4. Signalverarbeitung</b>                                      | <b>17</b> |
| <b>Resample</b>   | <b>17</b> |
| <b>Filter und Präempase/Deemphase</b>                             | <b>17</b> |
| <b>Modify: Multiply, Add, Subtract...</b>                         | <b>18</b> |
| <b>Pitchshifting und Daueränderung (PSOLA und LPC-Resynthese)</b> | <b>19</b> |
| <b>5. Analsen mit PRAAT</b>                                       | <b>20</b> |
| <b>FFT (Spektrogramme und Spektren)</b>                           | <b>20</b> |
| <b>LPC (Spektrogramme und Spektren)</b>                           | <b>23</b> |
| <b>Pitch-Analyse (Grundfrequenzanalyse)</b>                       | <b>24</b> |
| <b>Auditorische Analysen</b>                                      | <b>24</b> |
| <b>Dauernmessung</b>  | <b>25</b> |
| <b>6. Literatur</b>   | <b>26</b> |

## **1. Einführung / Vorbemerkung**

Im folgenden soll ein Überblick über einige Möglichkeiten gegeben werden, welche die Software PRAAT bietet. Für weitere Informationen und Möglichkeiten wird auf die umfangreiche Hilfefunktion des Programms sowie diverse Online-Tutorials (siehe Kapitel Literatur) verwiesen. Dieses Tutorial versteht sich als Einführung in die Arbeit mit PRAAT sowie das Schreiben von Skripten. Vorausgesetzt werden grundlegende Kenntnisse in der Signalverarbeitung sowie Audiotechnik und einfache Programmierkenntnisse.

PRAAT ist niederländisch und bedeutet „ich spreche“, somit ist das hauptsächliche Aufgabenfeld der Software eingegrenzt – die Sprachsignalverarbeitung. Aber neben den umfangreichen Möglichkeiten, die sich für Phonetik, Sprachverarbeitung und –synthese bieten, ist das Programmpaket auch für die digitale Signalverarbeitung und Audiotechnik interessant. Neben diesen Funktionen sind einige einfachere Statistik- und mathematische Funktionen implementiert. Ein Hauptvorteil gegenüber anderen Softwarepaketen ist die intuitive Bedienung und die sehr guten und einfachen Möglichkeiten der eingebauten Skriptsprache, die sich im Wesentlichen an C und anderen einfachen Sprachen orientiert und durch die Modularität des Programms wie ein Baukastensystem benutzt werden kann.

## 2. Nutzung und Aufbau

### **Download**

PRAAT kann unter der Adresse:

<http://www.praat.org>

kostenlos und ohne Registrierung heruntergeladen werden. Das gepackte Programm ist ungefähr 1.5MB groß, so dass ein Download auch mit einem 56k Modem kein Problem darstellt.

### **Programmteile und Plattformen**

PRAAT ist in verschiedenen Versionen für Macintosh, Windows und Linux/Unix erhältlich. Weiterhin ist auf der Website das Hilfsprogramm „Praatcon.exe“ zu finden, das eine Verbindung zu anderen Programmen aufbaut. So ermöglicht es die Ausführung von Skripten ohne ein laufendes PRAAT-Programm. Eine Anwendung sei hier aufgezeigt: Über einen einfachen Aufruf können aus Matlab über das Praatcon Programm beispielsweise die umfangreichen Synthesemöglichkeiten von PRAAT genutzt und direkt in die Matlab-Umgebung eingebunden werden.

Beim ersten Aufrufen erscheinen das PRAAT-Objektfenster und das Bildfenster (welches für grafische Darstellungen wie Grundfrequenzkonturen oder Formantstrukturen benutzt wird). Im Objektfenster befinden sich erwartungsgemäß alle vom Nutzer oder vom Programm erzeugten Objekte sowie die umfangreichen Möglichkeiten, die das Programm für die Arbeit mit diesen Objekten bietet. Für verschiedene Objekte (genauer genommen wenn diese durch Anklicken ausgewählt werden) erscheinen unterschiedliche Buttons. So wird bei Auswahl eines Soundobjektes (Erzeugen eines Sinustones mit Rauschen beispielsweise: New → Sound → Create Sound...) ein Play Button sichtbar, der bei Auswahl einer Matrix (New → Matrix → Create Matrix...) nicht sichtbar ist.

### **Objekte, Fenster, Editoren, Tiers, Matrizen**

Wie beschrieben ist PRAAT objektorientiert aufgebaut. Somit sind alle Editoren, alle objektbezogenen Fenster und alle Manipulationsmöglichkeiten nur sichtbar, wenn das *zu bearbeitende Objekt* ausgewählt wurde. Ein typischer Fehler ist, dass man beispielsweise ein Skript aufrufen oder eine Operation auf einem Soundobjekt ausführen will, aber eine Fehlermeldung erhält. Der Grund ist dann oft, dass gerade ein anderes Objekt selektiert ist und PRAAT mit den auszuführenden Befehlen (zum Beispiel Filter) auf dem selektierten Objekt

(beispielsweise eine Matrix) nichts anzufangen weiß (da eine Matrix nicht mit einem Audiofilter bearbeitet werden kann).

Viele Objekte (Sound, Spektrogramm...) können in Editoren betrachtet und manipuliert werden, womit ein visuelles Arbeiten ermöglicht wird. Jeder Editor wird in einem neuen Fenster geöffnet. Einige Objekte (wie Formanten oder Pitch) können über sogenannte „Tiers“ bearbeitet werden, die im Grunde genommen die jeweilige Information (also Formant- oder Pitchinformation) zeitdynamisch darstellen.

Ein Grundprinzip von PRAAT ist, dass komplexe Information „heruntergebrochen“ werden kann. So ist es möglich, beispielsweise einen Grundfrequenzverlauf eines Signals (Pitch Tier) in eine Matrix zu überführen und danach in eine einfache Tabelle, die alle Pitch-Werte enthält, beispielsweise um sie in einem anderen Programm einzulesen.

### **Start-Up-Script („buttons.ini“), PRAATCON**

Eine interessante Option für Installationen oder für einen stationären Betrieb sind die sogenannten Start-Up-Scripts. Diese werden aufgerufen, sobald das PRAAT Programm in den Speicher geladen wird (also PRAAT gestartet wird). Sie dienen dazu, beispielsweise Pfade für die Arbeit zu definieren oder andere Skripte aufzurufen, ohne dass eine Benutzereingabe erforderlich wäre. Somit ist es möglich, mehrere aufeinanderfolgende, unterschiedliche gefilterte Soundobjekte zu manipulieren und abzuspielen, ohne dass eine Benutzereingabe in einem laufenden Programm vonnöten wäre.

Start-Up-Scripts werden auch oft benutzt, um schon definierte Skripte zu laden und sie sofort für die weitere Arbeit verfügbar zu haben. So ist es möglich, über zwei unterschiedliche Start-Up-Scripts schnell die Arbeitsumgebung und die benötigten Skripte und Objekte verschiedener Nutzer zu laden (die vielleicht an unterschiedlichen Projekten arbeiten). Grundsätzlich kann man alle Befehle und Skriptbausteine in das Start-Up-Script einzubauen.

Eine weitere Möglichkeit für die Bedienung von PRAAT ohne eine Interaktion mit dem Benutzer ist durch das Programm PRAATCON gegeben, welches ebenfalls auf der PRAAT-Homepage kostenlos heruntergeladen werden kann. Dieses erlaubt alle PRAAT-Befehle von einem anderen Programm aus aufzurufen, ohne ein aktuell laufendes PRAAT auszuführen. So kann also beispielsweise die Signalverarbeitung oder Segmentierung von Audiofiles durch PRAAT von Matlab aus erfolgen, ohne dass PRAAT gestartet werden müsste.

## Objekte

### Selektion vor Ausführung

Wie bereits beschrieben ist es notwendig die jeweiligen Objekte anzuwählen, bevor eine Bearbeitung erfolgen soll. Dies geschieht durch einfaches Anklicken. In Abbildung 1 ist ein Screenshot gezeigt, der das Objekt ‚JDLARYAU057‘ ausgewählt hat, im rechten Teil der Abbildung ist der zugehörige Editor gezeigt, der durch den Button „Edit“ geöffnet wird. Im Skriptbereich dieses Tutorials wird dann beschrieben, wie Objekte automatisch und über Variablen ausgewählt werden können.

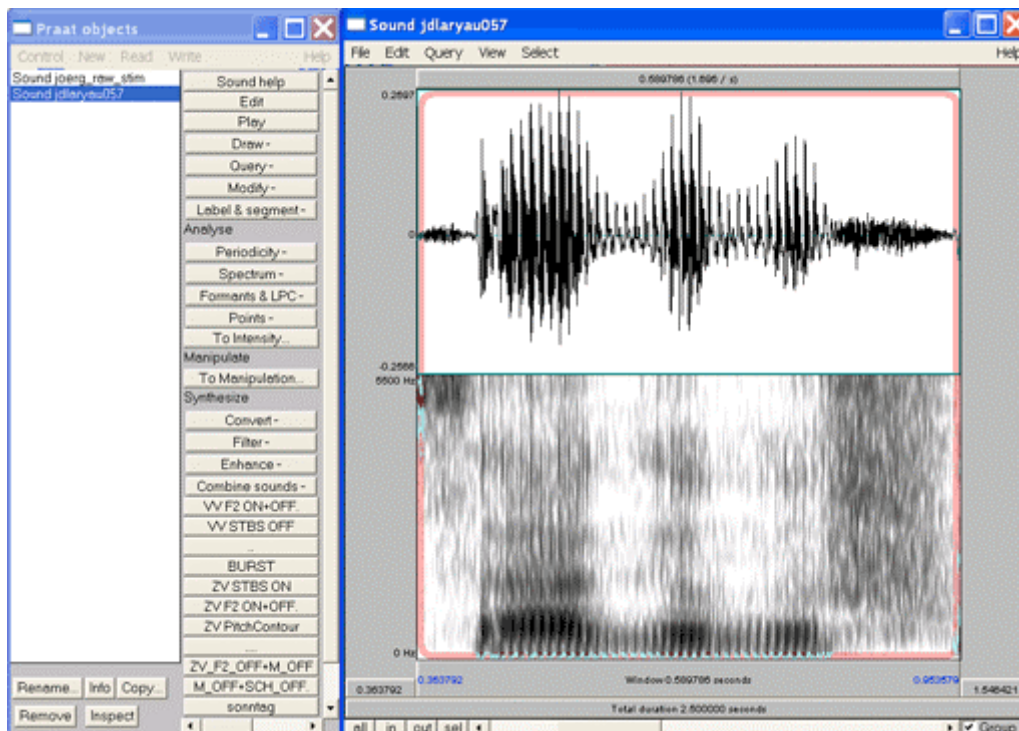


Abbildung 1: Links ist das PRAAT-Objektfenster zu sehen, mit einem ausgewählten Objekt. Rechts ist der zugehörige Editor (im oberen Teil das Zeitsignal, im unteren Teil das synchronisierte Spektrogramm) dargestellt.

## Editoren

Viele Objekte können in einem Editor visualisiert und bearbeitet werden (siehe rechten Teil der Abbildung 1). Diese sind, unter anderen:

- Sound
- Pitch (Grundfrequenz)
- Spektrum (unabhängig von der Methode, beispielsweise LPC oder FFT)

Diese Editoren können sinnvoll benutzt werden, um einen Teil des Signals auszuwählen, zu bearbeiten, das Signal auf Digital Null zu setzen, einzufügen... kurz, zu editieren. Wichtig für die spätere Arbeit mit Skripten sind die einzelnen Menüpunkte, die im Editor-Fenster (siehe Abbildung 2) zu sehen sind. Diese Menüpunkte können direkt in Skripten angesprochen werden (mit dem Befehl, der im Menü zu sehen ist). Zu beachten ist, dass pro Objekt mehrere identische Editoren geöffnet werden können, was bei umfangreichem Arbeiten ein Problem darstellen kann.

Im Soundeditor kann für den unteren Teil des Editors per Einstellung gewählt werden, welche Analysen für das Audiosignal angezeigt werden sollen: Zur Verfügung stehen Grundfrequenzkontur (Pitch), Formantkontur (LPC), Spektrogramm (FFT) und Intensitätskontur. In Abbildung 2 ist ein Spektrogramm mit überlagerter Formantkontur zu sehen.

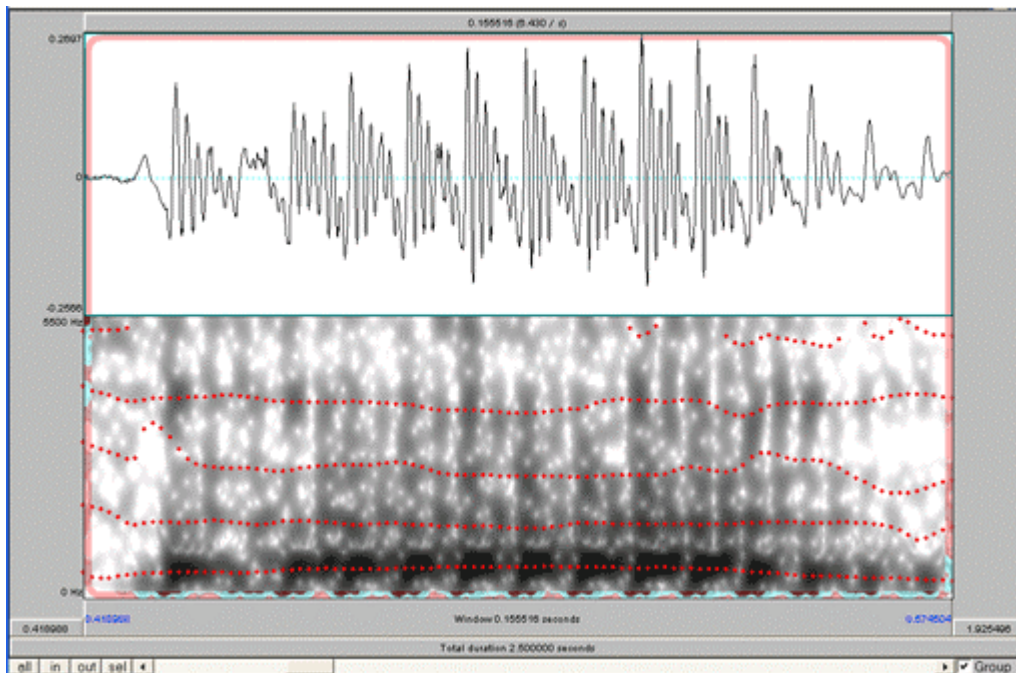


Abbildung 2: Im Soundeditor ist ein Vokal dargestellt, im oberen Teil das Zeitsignal, im unteren Teil das synchronisierte Spektrogramm, überlagert von einer Formantkontur.

### Gruppieren von Editoren

Editoren können mit dem Group-Häkchen (ganz unten rechts im Editor) synchronisiert werden, somit kann beispielsweise ein Sound-Editor und ein zugehöriger Pitch-Editor (der die Grundfrequenzwerte dieses Sounds enthält) „gruppiert“ werden. Bei jedem Zoom in das Soundsignal oder das Pitchsignal wird der andere Editor diesen Zoom ebenfalls nachvollziehen, bei einer Auswahl eines Teilbereiches wird der gleiche Teilbereich im anderen Editor ebenfalls ausgewählt werden.

### Tiers (numerisch und Editor)

Tiers sind bestimmte Werte, beispielsweise Formantwerte oder Grundfrequenzwerte, über den Zeitbereich abgetragen. Ein Formant-Tier, dessen Quantisierung vorher festgelegt wird, besteht für jeden einzelnen Zeitpunkt aus einem Wert und der Bandbreite des ersten Formanten, des zweiten Formanten und so weiter. Im Prinzip ist dieses Tier-Prinzip mit den Werten für digitales Audio vergleichbar, nur dass die einzelnen Samples in diesem Fall Container sind, in die alles mögliche gefüllt werden kann. Tiers können zur Visualisierung im Praat-Picture-Fenster gezeichnet werden.

## Matrix

Die Tiers können in Matrizen heruntergebrochen werden, um beispielsweise bestimmte Wertepaare in eine Tabelle zur Bearbeitung in einem anderen Programm bereitzustellen.

## Statistik

PRAAT bietet auch eine Vielzahl von statistischen Werkzeugen, um die entstandenen Daten weiterzubearbeiten. Hier wird auf die „Search PRAAT manual“ Funktion verwiesen.

## „Picture“ Fenster

In diesem Bereich können Daten gezeichnet werden. In Abbildung 3 ist ein Formant-Plot zu sehen, der über den Befehl „Draw“ aus einem Formant-Objekt erstellt wurde. Dieses Zeichnen ist, neben Erstellung von Grafiken für die Publikation, eine einfache Möglichkeit, um Daten wie Formanten oder Intensitäten zu visualisieren.

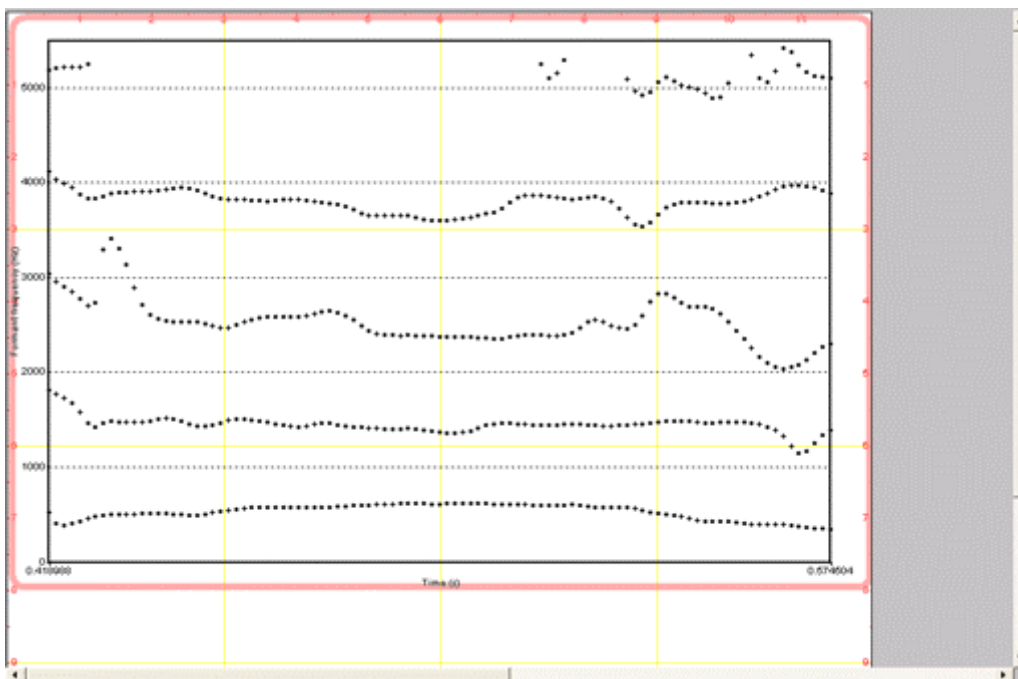


Abbildung 3: Das PRAAT Picture Window. Zu sehen ein Formant-Plot eines Vokals. Die zur Verfügung stehende Zeichenfläche kann beliebig aufgeteilt werden. Zum Export stehen einige Bildformate zur Verfügung.

## „Manipulation“

Eine sehr wichtige Funktion, um einen Sound zu manipulieren (also beispielsweise Länge oder Grundfrequenz zu ändern), ist die „To Manipulation...“ Funktion. Diese ist im Objektfenster zu sehen, wenn man einen Sound auswählt. Nach Eingabe der Parameter erscheint ein Manipulation-Objekt, das dann mit der Funktion „Edit“ bearbeitet werden kann. In diesem Editor (siehe Abbildung 4) können unabhängig



voneinander visuell die Länge und die Grundfrequenzkontur verändert werden. Das Ergebnis kann sofort mithilfe der PSOLA-Resynthese (Grundfrequenzänderung und Längenänderung) oder der LPC-Resynthese (Grundfrequenzänderung) synthetisiert werden. Die entstandene Resynthese kann dann komplett als neues Objekt berechnet werden.

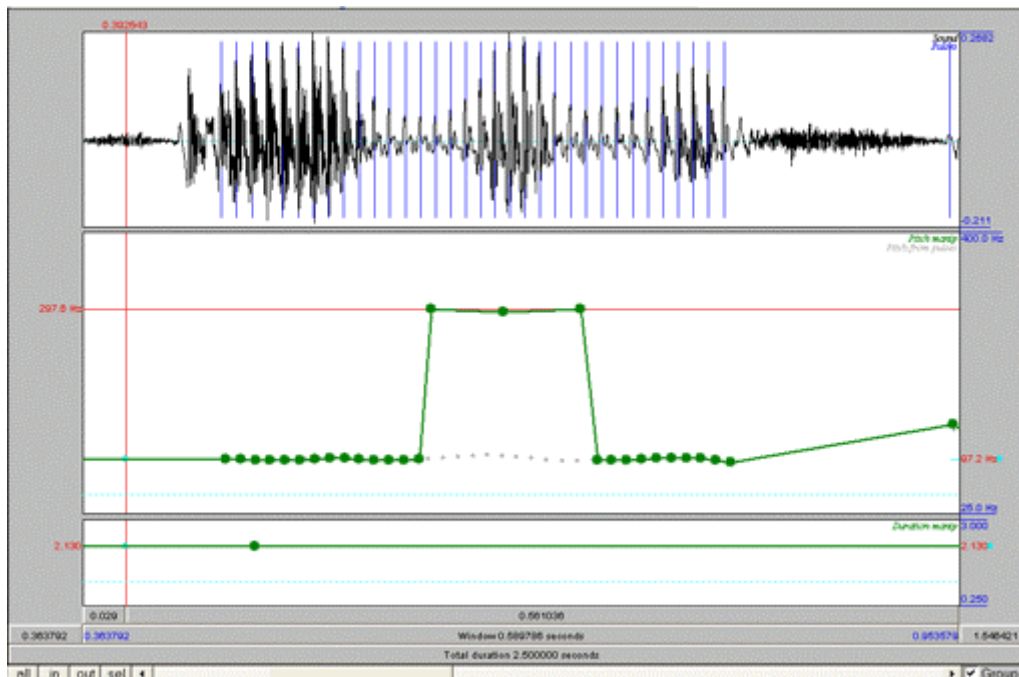


Abbildung 4: Manipulation-Editor. Oben das Oszillogramm mit den berechneten „pulses“, in der Mitte die Pitch Kontur (mit bereits manipulierter Kontur, es wurde ein Plateau bei 300Hz erzeugt) und unten die Längenänderung des Signals.

## Hilfe

Unter dem Menüpunkt „Help“ sind eine Vielzahl von Tutorials und Hilfefunktionen zu finden, die den Einstieg und das schnelle Arbeiten mit PRAAT fördern. Abbildung 5 zeigt, welche Tutorials erscheinen, sobald „Tutorials“ im Menüpunkt „Search PRAAT manual“ eingegeben wird. Es gibt eine schrittweise Einführung (Start mit „how to get a sound“), die durch alle wichtigen Bereiche von PRAAT führt.

Weiterhin stehen speziellere Tutorials zur Verfügung, um die Bearbeitung phonetischer Teilbereiche wie artikulatorische Synthese oder Quelle-Filter-Synthese in PRAAT zu beschreiben.

Es soll an dieser Stelle noch einmal betont werden, dass ALLE relevanten Informationen zur Arbeit mit PRAAT und dem Schreiben von Scripten aus der Online-Hilfe entnommen werden können. Alle Parameter und Funktionen sind dort ausreichend beschrieben.

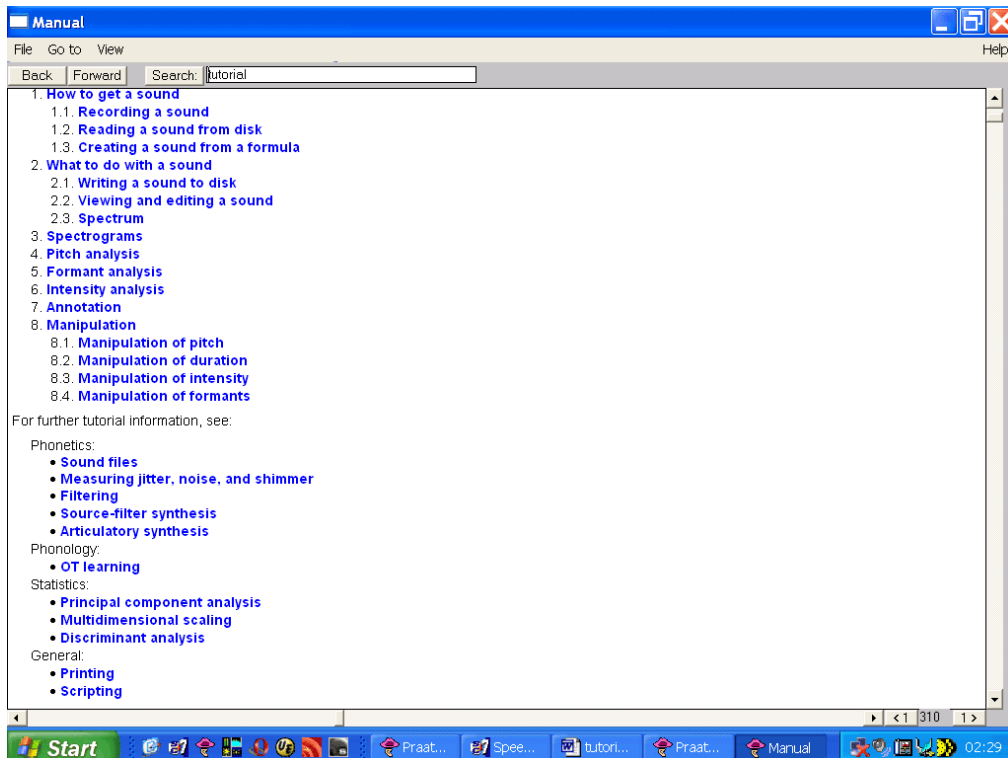


Abbildung 5: Über den Menüpunkt „Search PRAAT manual“ wurde der Begriff „tutorial“ eingegeben. Neben einführenden Tutorials gibt es auch speziellere wie Quelle-Filter-Synthese und Filterung.

### **3. Skripte**

Skripte werden in einem eigenen Fenster (script window) dargestellt. Nach einer Eingewöhnung empfiehlt sich aber, externe Editoren (beispielsweise UltraEdit) zu benutzen, um umfangreichere Skripte zu editieren, da PRAAT keinerlei Zusatzfunktionen wie Zeilennummern oder Syntax-Hervorhebung anbietet. Wird ein externer Editor benutzt, dann ist es sinnvoll das Skript über einen Button aufzurufen (siehe Punkt Ausführung von Skripten). Gleich zu Beginn soll auch auf das Scripting Tutorial in der PRAAT Hilfe verwiesen werden, was einen einfachen Start in die Möglichkeiten verschafft. Folgende Abschnitte sollen dazu dienen, einen einfachen Überblick zu bekommen und bekannte, kaum dokumentierte Fehler und Fallstricke zu umgehen.

Auf der Website dieses KT-Labors ist außerdem ein Beispielskript zum Download bereitgestellt, das den Aufbau und die Funktionsweise von PRAAT-Skripten dokumentiert. Alle dort benutzten Funktionen und Parameter sind in diesem Skript ausführlich beschrieben und auskommentiert.

#### ***Ausführung von Skripten***

Ist ein Skript erstellt worden, kann es über zwei verschiedene Arten aufgerufen werden:

##### ***1. Ausführung im Skriptfenster:***

Unter dem Menüpunkt „Run“ finden sich die Befehle „Run“ und „Run selection“. Der erste Befehl lässt das ganze Skript durchlaufen (oder bis zum Abbruch der Ausführung des Skriptes aufgrund einer Fehlermeldung), der zweite Punkt führt nur die Befehle im selektierten Text aus (der natürlich vorher selektiert sein sollte). „Run selection“ ist sehr sinnvoll zum Auffinden von Fehlern, da ein Skript damit Stück für Stück auf fehlerhafte Syntax oder andere Programmierfehler durchsucht werden kann.

Leider verhält es sich so, dass die jeweiligen Gründe für Fehlermeldungen, und damit der Abbruch des Skriptlaufes, nicht immer eindeutig zu klassifizieren sind. Oft kommt es vor, dass nur die Zeile des Abbruchs genannt wird, aber nicht der genaue Grund, oder es kommen mehrere Gründe für einen Fehler in Frage. Deshalb ist die Fehlersuche in PRAAT-Skripten oft mühsam und zeitraubend. Dagegen ist natürlich anzuführen, dass es sich entwicklungsstechnisch um eine Skriptsprache handelt und keine komplette Programmierumgebung, so dass Limitierungen ganz natürlich sind.

##### ***2. Ausführung über Buttons:***

Die andere Möglichkeit ein Skript aufzurufen besteht darin, ein Skript-File mit einem Button in PRAAT zu verknüpfen. Dies geschieht entweder global

im PRAAT Objektmenü (im Skriptfenster: File --> „Add to fixed Menu“) oder zugehörig zu einem bestimmten Objekt (im Skriptfenster: File --> „Add to dynamic Menu“; Class 1 ordnet das Objekt zu, Command gibt dem Button einen frei wählbaren Namen). Wird das jeweilige Skript einem bestimmten Objekttyp zugeordnet, dann erscheint der Button nur, wenn ein solches Objekt angewählt wird.

Ein Beispiel: In Abbildung 1 ist links unten zu erkennen, dass ich einige neue Buttons erstellt habe (zum Beispiel „ZV PitchContour“). Dieses Skript „ZV Pitch Contour“ kann nur ausgeführt werden (und gleichzeitig ist damit der Button auch NUR dann zu sehen), wenn ein Soundobjekt ausgewählt wurde, wie in Abbildung 1 zu sehen durch Auswahl des Objektes „jdlaryau057“.

Mit der Möglichkeit, Skripte über Buttons zu starten können diese in einem externen Editor wie „Ultraedit“ oder „Emacs“ bearbeitet werden und dann die Ausführung über den Button gestartet werden.

## ***Paste History***

Eine sehr nützliche Funktion, um ohne viel Programmier- und Syntaxkenntnisse das Schreiben von Skripten in PRAAT zu beginnen, ist die „Paste History“ Funktion im Skript-Fenster. Man sucht sich „per Hand“ ein Vorgehen, um eine bestimmte Aufgabe in PRAAT zu lösen, also beispielsweise:

*„Ich wähle meinen Vokal aus. Daraufhin erzeuge ich eine Grundfrequenzkontur mit dem Befehl „To Pitch...“. Aus diesem erzeuge ich ein neues Anregungssignal mit dem Befehl „To Sound (Pulses)“. Dann erzeuge ich den Filter des Objektes über ein LPC-Objekt des Vokals mit dem Befehl „To LPC (burg)...“. Diesen erzeugten Filter wähle ich zusammen mit dem erzeugten Sound-Objekt aus (mit STRG-Taste) und wähle „Filter...“. Der erzeugte Sound ist die LPC-Resynthese meines Ausgangssignals.“*

Um nun ein lauffähiges Skript zu generieren, dass auf JEDEM jeweils ausgewählten Sound eine Resynthese ausführt, wird zuerst im neu erzeugten Skript-Fenster „Clear History“ ausgeführt, wodurch eventuell im Speicher befindliche PRAAT-Befehle gelöscht werden. Nun wird wie oben beschrieben der ganze gewünschte Ablauf „per Hand“ in PRAAT ausgeführt. Danach können per „Paste History“ alle verwendeten PRAAT Befehle in das Skript-Fenster eingefügt werden. Nun müssen nur noch festen Namen durch Variablen ersetzt werden (siehe unten) und das Skript kann für jedes beliebige Sound-Objekt benutzt werden.

## **Variablen (numerisch und Text)**

Variablen werden in PRAAT grundsätzlich klein geschrieben, während Befehle grundsätzlich mit Grossbuchstaben beginnen.

Wie in vielen Programmiersprachen gibt es unterschiedliche Variablentypen in PRAAT-Skripten: numerische und String-Variablen. Eine numerische Variable (vereinfacht dargestellt: mit dieser kann gerechnet werden, sie kann aber nicht in ein Textfile geschrieben werden) wird einfach zugewiesen:

```
zahl1 = 125  
zahl2 = 1.56874
```

Eine Stringvariable wird durch ein String-Zeichen („\$“) zugewiesen:

```
name1$ = "Günther"  
name2$ = "125"
```

Versucht man nun:

```
name2$+name2$
```

zu rechnen, erhält man eine Fehlermeldung (da wie beschrieben mit Strings keine Rechenoperationen ausgeführt werden können), der Befehl:

```
zahl2+zahl2
```

führt jedoch zum gewünschten Ergebnis.

Möchte man eine Stringvariable in eine numerische Variable umwandeln, kann man die Befehlsfolge „Hochkomma+Anführungsstriche“ benutzen:

```
zahl3 = '"name2"'
```

Der umgekehrte Weg, eine numerische in eine Stringvariable umzuwandeln, vollführt man am elegantesten (der oben beschriebene Weg funktioniert allerdings auch in diesem Fall) mit dem Befehl „fixed\$“, der neben der Umwandlung auch noch eine Rundung vornimmt, was sinnvoll für eine Textausgabe beispielsweise von Zahlen des Typs „125.23454756847378499485848484Hz“ ist, wenn eigentlich 125Hz als Formatwert ausreichend ist:

```
name3$ = fixed$ (zahl3, 1)
```

womit auf eine Nachkommastelle gerundet wird.

Eine weitere Nebenbemerkung, die lange Fehlersuche vermeidet: Bei der Namensvergabe für Variablen sollte beachtet werden, dass keine Namen verwendet werden, die geschützt sein könnten: So ist es nicht klug, Namen wie „formantwert“ oder „pitch1“ zu verwenden, da diese Begriffe

in PRAAT teilweise geschützt sind: Die Verwendung MUSS nicht in einer Fehlermeldung resultieren (die oft ohne nähere Fehlererklärung auftritt, was die Suche sehr erschwert), aber sie kann dazu führen. Mir ist allerdings keine Liste bekannt, in denen die geschützten Befehle aufgelistet sind, grundsätzlich spart man sich aber durch die Vermeidung solcher Namen eine unnötige Fehlersuche.

### **Die Befehle „selected“ und „editor“**

Die Struktur der Skripte in PRAAT ist so angelegt, dass man in einen Editor „hineinspringen“ muss, sofern die gewünschte Operation (beispielsweise „Extract selection“) sich nur in einem Editor anwählen lässt. Dies bedeutet, dass ein Skript immer erst im PRAAT-Objektfenster nach einem Befehl sucht (dies könnten z.B. „Resample“ oder „Edit“ sein). Ist dieser dort nicht zu finden, erhält man eine Fehlermeldung. Um nun in einen Editor zu springen, benutzt man den Befehl „editor“, gefolgt vom Namen des Objektes, in dessen Editor gesprungen werden soll. Den Namen des Objektes erhält man am elegantesten mit der Befehlskombination „selected“ und „selected\$“. Eine Variablenzuweisung könnte folgendermaßen aussehen (vorausgesetzt, das Objekt ist bereits ausgewählt):

```
name_objekt = selected ("Sound")
name_objekt$ = selected$ ("Sound")
```

In der Klammer sollte immer der Objekttyp stehen, um den es geht, also Sound, Formant, Pitch etc. Mit der ersten Zeile wird eine numerische Variable erzeugt (z.B. Object127), mit der zweiten wird einer Stringvariable der Name zugewiesen (z.B. name\_object ist gleich „Vokal2“).

Ist nun der Name zugewiesen, kann über die Kombination aus „editor“ und der Substitution des Variablennamens in den Editor gesprungen werden:

```
editor 'name_object$'
```

Nun können alle gewünschten Befehle im Editor ausgeführt werden. Zum Zurückspringen ist der Befehl „endeditor“ ohne Parameter auszuführen.

Eine weitere Nebenbemerkung, die viel Kopfzerbrechen ersparen kann: Wird ein Skriptablauf innerhalb eines Editors unterbrochen, steckt das Skript auch bei der nächsten Neuausführung in diesem Editor fest, und merkwürdige Fehlermeldungen können resultieren. Aus diesem Grund ist es sinnvoll, jedes Skript immer mit dem Befehl „endeditor“ zu beginnen. Somit ist sichergestellt, dass PRAAT immer zuerst im Objekt-Fenster mit der Abarbeitung des Skriptes beginnt.

## **Übergabe von Parametern (Substitution von festen Namen)**

Liegt die Struktur eines Skriptes über die „Paste history“ Funktion vor, dann kann man beginnen, die festen Parameter, die PRAAT für die Objekte vergeben hat, zu verändern (also beispielsweise das Soundobjekt „Sound127“ soll in einem Skript in „Vokal\_a“ umbenannt werden zum übersichtlichen Arbeiten). Dies geschieht mithilfe des Hochkommazzeichens. Ein paar Zeilen, die das eben erwähnte Beispiel ausführen, wären diese:

```
vokalname$ = "vokal_a"  
Select Sound vokal$  
Rename... 'vokalname$'
```

Genauso kann mit der Substituierung der Editorennamen verfahren werden. Zu beachten ist hierbei, dass es aus Kompatibilitätsgründen am sinnvollsten ist, die Globalisierung des Editornamens nach folgendem Schema vorzunehmen:

```
vokal_name$ = selected$("Sound")  
editor Sound 'vokal_name$'
```

(womit z.B. im Skript folgende vorherige Zeile ersetzt werden würde: „editor Sound Testvokal“). Durch diesen Weg ist nun nicht mehr nur die Bearbeitung im Editor des Objektes „Testvokal“ möglich, sondern für jedes Soundobjekt, das gerade ausgewählt ist, bevor das kleine Skript gestartet wird.

## **Eingaben des Benutzers und Pause**

Folgender Abschnitt ist für Benutzer gedacht, die sich etwas mit der Architektur mehrerer Skripte und den Kommunikationsmöglichkeiten untereinander beschäftigen möchten. Er ist für eine Einführung nicht unbedingt erforderlich.

Ein Skript kann angehalten werden, um dem Benutzer etwas Zeit zu geben, eine bestimmte Einstellung zu tätigen. Dies kann beispielsweise sinnvoll sein, wenn der Zeitpunkt in einem Soundfile abgefragt werden soll (z.B. um den Beginn eines Vokals zu messen). Zu diesem Zweck setzt man in das Skript vor die Abfrage „Get cursor“ einen Haltepunkt („Pause“), dem man auch einen sinnvollen Namen geben kann (beispielsweise „pause Setze Cursor auf den Vokalbeginn“). Der Benutzer sieht dann ein Fenster mit besagtem Titel und wird mit der Wahl des Skriptabbruchs oder der Skriptfortsetzung konfrontiert.

Etwas umständlich und eingeschränkt ist es, Eingaben während eines laufenden Skriptes vorzunehmen, um beispielsweise eine Auswahl unter

mehreren Möglichkeiten (wie im Internet beispielsweise über ein Formular mit ‚Radio-Buttons‘) oder eine Zahleneingabe für die Weiterverarbeitung vorzunehmen. Für diesen Fall ist in PRAAT der Befehl „form“ vorgesehen, der IMMER zuerst am Beginn des Skriptes ausgeführt wird, wenn er vorhanden ist, unabhängig davon, an welcher Stelle des Skriptes der Befehl „form“ steht. Daraus kann auch das Problem abgeleitet werden, da „form“ die einzige komfortable Möglichkeit der Kommunikation mit dem Benutzer bietet. Durch die erzwungene Ausführung zu Beginn des Skriptdurchlaufs müssen alle Eingaben bereits festgelegt werden, ein nachträglicher Eingriff durch den Benutzer ist nur durch folgende Umgehung durchführbar:

Es können andere Skripte aus Skripten mit Argumenten aufgerufen werden. In diesem Fall dient der Befehl „form“ im aufzurufenden Skript zur Argumentübergabe. Zahl und Art der Argumente im Aufruf und in der „form“ Zeile des aufzurufenden Skriptes müssen dann übereinstimmen. Der „form“ Befehl im aufzurufenden Skript wird dann nicht mehr als Benutzereingabe in einem eigenen Fenster geöffnet, sondern dient nur zur Parameterübergabe.

Für eine weitere Einführung in die Parametrisierung und den Aufbau des „form“ Befehls wird auf „Scripting 6.1. Arguments to the script“ in der PRAAT Online-Hilfe verwiesen.



## 4. Signalverarbeitung

PRAAT kann schnell und einfach benutzt werden, um bestimmte Filter oder Effekte wie z.B. pitch-shifting intuitiv und visuell über eine beliebige Grundfrequenzkontur (Beispiel: Erzeugung eines dynamischen Vibrato) zu generieren.

So wird PRAAT oft benutzt, um ein Audiosignal zu bearbeiten, wenn externe Geräte zu umständlich zu verkabeln sind oder etwas schnell getestet werden soll, bevor es über externe Geräte oder Plug-Ins realisiert wird.

Im folgenden sollen einige Signalverarbeitungsmöglichkeiten für PRAAT kurz vorgestellt werden. Für nähere Ausführungen können die gewünschten Begriffe in die „Search PRAAT manual“ Hilfe eingegeben werden. Alle beschriebenen Algorithmen befinden sich in der obersten Hierarchieebene, also im PRAAT-Objektfenster.

### ***Resample***

Oft ist es wünschenswert, ein Signal unterabzutasten, dass beispielsweise mit einer Abtastrate von 44100Hz vorliegt. Der Grund dafür liegt darin, dass historisch viele Algorithmen für eine Abtastrate von 16000Hz optimiert sind. Zur Analyse von Vokalen beispielsweise hat sich eine Abtastrate zwischen 8000Hz und 16000Hz durchgesetzt, da in diesem Bereich einige Formant-Algorithmen die besten Ergebnisse liefern (eine genauere Erklärung ist in der Literatur zur Signalverarbeitung nachzulesen).

### ***Filter und Präempase/Deemphase***

PRAAT bietet einige einfache Möglichkeiten der Filterung, dazu gehören einige Arten von Bandpass- und Bandstopfilter (die mit der entsprechenden Grenzfrequenz natürlich auch als Tiefpass oder Hochpass zu benutzen sind). Eine weitere Möglichkeit für fortgeschrittene Nutzer ist die Eingabe der Filterkoeffizienten, um so eine vom Nutzer gewünschte Filtercharakteristik zu erhalten (analog zu Matlab).

Eine oft in der Sprachanalyse benutzte Spezialform eines Filters ist die Präemphase/Deemphase: Durch die akustische Dämpfung höherer Frequenzen während der Sprachproduktion werden diese im akustischen Signal sehr schwach dargestellt, was das Auffinden von hohen Formanten beeinträchtigt. Durch die Vorschaltung eines Prä-Emphase-Filters, welcher hohe Frequenzen ab einer einstellbaren Grenzfrequenz verstärkt, wird dieser Effekt gemindert und auch höhere Formanten sind gut zu erkennen. Die Umkehrung des Effekts wird De-Emphase genannt und bewirkt analog

eine Abschwächung der hohen Frequenzen ab einer einstellbaren Grenzfrequenz.

### ***Modify: Multiply, Add, Subtract...***

Über den Button "Modify" können einige Basisoperationen auf dem Audiosignal durchgeführt werden:

*Änderung der Maximalamplitude:* Unglücklicherweise ist in PRAAT kein „gain“ Regler vorhanden, wie in anderen Audioprogrammen üblich. Möchte man die Maximalamplitude eines Soundfiles ändern, muss man daher auf andere Wege zurückgreifen. Der erste besteht natürlich darin, über „Add“ oder „subtract“ einen bestimmten Wert auf jedes Sample des Objektes zu addieren bzw. zu subtrahieren. Ein anderer Weg besteht darin, sich im Editor anzusehen, wie aktuell die Maximalamplitude des Files ist. Die digitale Maximalaussteuerung ist in PRAAT mit 1.0 (bzw. -1.0) festgelegt. Beispielsweise kann nun zur Maximalaussteuerung des Soundfiles jedes Sample mit demjenigen numerischen Wert addiert werden, der für das Sample mit der maximalen Aussteuerung eine Amplitude von 1.0 erreicht, so dass der höchste Wert des Files mit 1.0 resultiert. Der zweite Weg besteht darin, zur Maximalaussteuerung den Befehl „Scale“ im gleichen Menü zu benutzen.

*DC Offset:* Eine Überlagerung von Gleichstrom auf dem Signal (sogenannter DC Offset) kann einfach mit dem Befehl „Subtract mean“ entfernt werden (für eine nähere Erklärung wird ebenfalls auf die Literatur zur Signalverarbeitung verwiesen).

*Reverse:* Die Samples des Soundfiles werden in umgekehrter Reihenfolge geschichtet (entspricht rückwärtigem Abspielen).

*Faltung von zwei Soundfiles:* Über diesen Befehl können zwei Soundfiles miteinander gefaltet werden (zur genauen Erklärung der Faltung siehe ebenfalls die Signalverarbeitungsliteratur).

*Konkatenierung:* Mit dieser Funktion ist die serielle Verkoppelung zweier Soundobjekte möglich (entspricht einem „Aneinanderhängen“ der Soundfiles).

*„Formula“:* Hier ist für fortgeschrittene Nutzer über die Eingabe einer Formel möglich, Samples mit jeder nur möglichen Bearbeitung zu versehen, sofern diese über eine Formel basierend auf Sampleebene zu realisieren ist.

### ***Pitchshifting und Daueränderung (PSOLA und LPC-Resynthese)***

Die einfachste Möglichkeit einer Tonhöhenänderung besteht in dem Befehl „Override sample rate“, welcher schlicht alle Samples mit einer anderen Abtastrate abspielt (bei höher gewählter Abtastrate bekannt als „MickeyMouse-Effekt“), entsprechend einem schnelleren oder langsameren Abspielen eines Tonbandes.

Eine weitere Möglichkeit besteht in der Nutzung des Manipulation-Editors, um grafisch PSOLA oder LPC-Resynthese mit beliebigen entstehenden Grundfrequenzkonturen und Längenänderungen zu ermöglichen. Eine reine Längenänderung ist ebenfalls über den Button „Enhance“ und die Option „Lengthen (PSOLA)“ zu erreichen.

## 5. Analysen mit PRAAT

Im folgenden sollen einige oft benutzte Sprachanalysetechniken besprochen werden. Wie im vorherigen Abschnitt wird bei Interesse an der Funktionsweise auf die Literatur der Signalverarbeitung verwiesen, hier soll die Anwendung der Analysen im Vordergrund stehen.

### ***FFT (Spektrogramme und Spektren)***

Es werden in der Sprachverarbeitung meist zwei unterschiedliche Fensterbreiten benutzt, um das Unschärfeproblem zumindest abzuschwächen. Die folgenden Erklärungen gelten sowohl für Spektrogramme als auch für Spektren. Es gilt natürlich, dass Spektrogramme aus diskreten Spektren, über den Zeitbereich gereiht, bestehen. So kann zu jedem angegebenen Zeitpunkt aus einem Spektrogramm ein Spektrum an eben diesem Zeitpunkt extrahiert werden.

In einer Breitbandanalyse werden die einzelnen Harmonischen „geglättet“, so dass die Formanten sichtbar werden. Die Bandbreite beträgt meist um die 200Hz (in PRAAT muss die zugehörige Fensterlänge angegeben werden: 200Hz entsprechen rund 0.005s, in Abhängigkeit von der Fensterfunktion), eine Prä-Emphase hebt die hohen Frequenzen an, um der Dämpfung durch Vokaltrakt und Lippen entgegenzuwirken. Die zeitliche Auflösung ist sehr gut, die einzelnen Pulse der glottalen Anregung sind gut zu erkennen und zu analysieren, während die spektrale Auflösung naturgemäß schlecht ist (weshalb die Harmonischen „verschmiert“ werden, so dass die generelle Formantstruktur übrig bleibt). Abbildung 6 zeigt diese Eigenschaften eines Breitbandspektrogrammes deutlich, Abbildung 7 zeigt die Extraktion eines Spektrums (über die Funktion „To spectrum (Slice)“) aus einem Zeitpunkt in der Mitte des Spektrogramms in Abbildung 6.

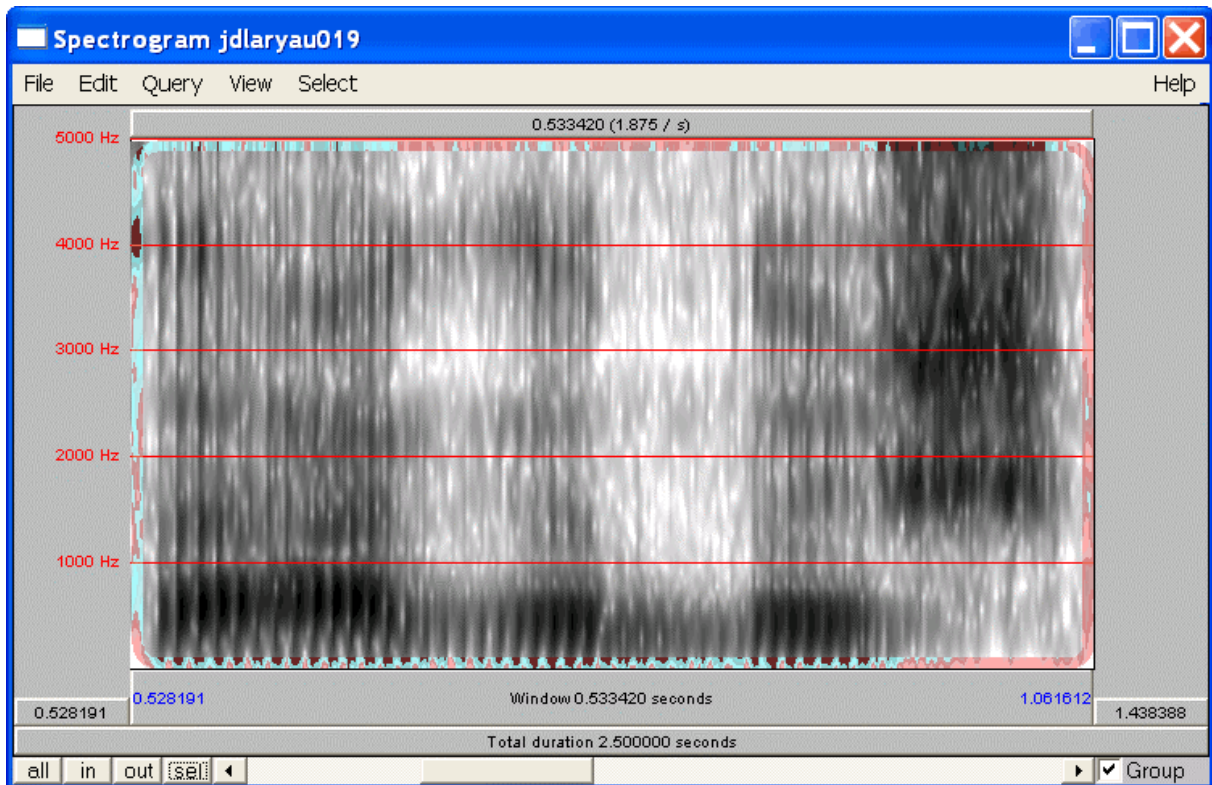


Abbildung 6: Breitbandspektrogramm eines kompletten Satzes. Die einzelnen glottalen Pulse sind aufgrund der hohen zeitlichen Auflösung gut zu erkennen, während die einzelnen Harmonischen „verschmiert“ werden, so dass nur noch die generelle Formantstruktur erkennbar ist.

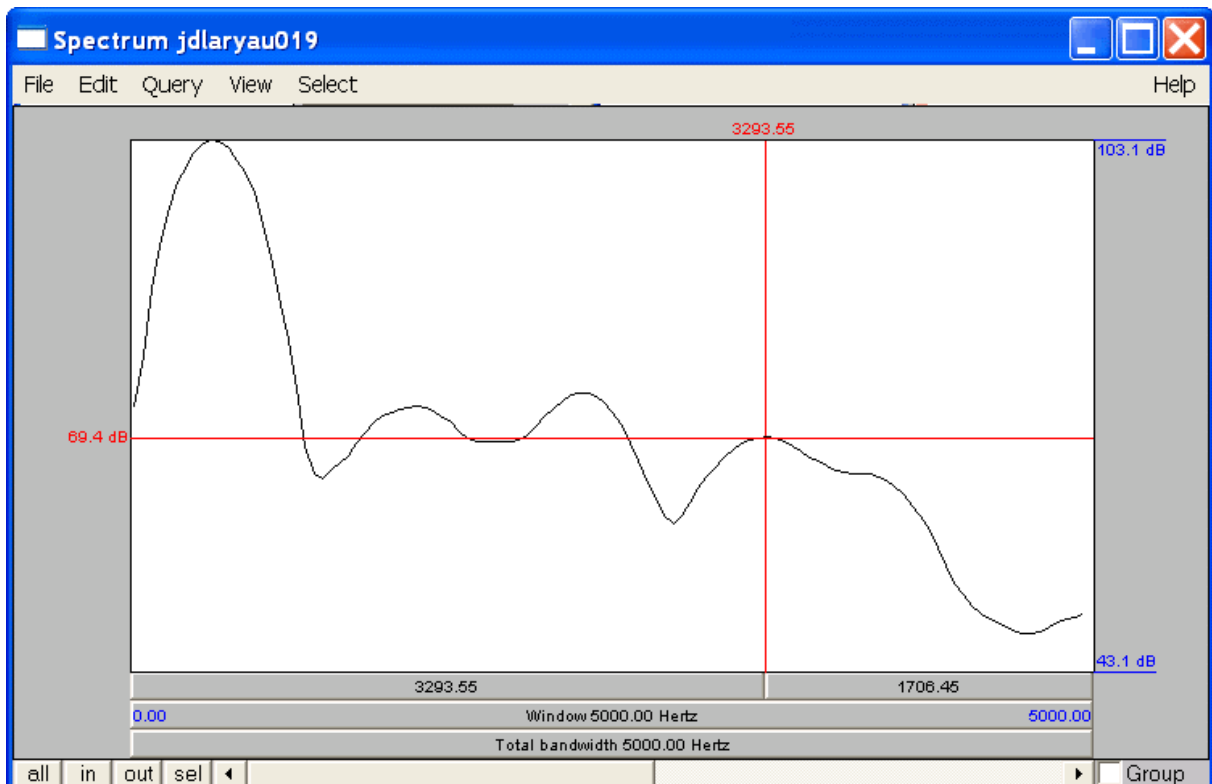


Abbildung 7: Breitbandspektrum, herausgeschnitten aus der Mitte der vorherigen Abbildung. Es gelten die gleichen Ausführungen wie in der vorherigen Abbildung.

In einer Schmalbandanalyse werden die einzelnen Harmonischen gut dargestellt, die Formanten sind nur schwer zu erkennen, da sie von den Harmonischen überlagert werden. Die Bandbreite beträgt meist um die 20Hz (entspricht einer Fensterlänge von ungefähr 0.05s). Die zeitliche Auflösung ist sehr schlecht, die einzelnen glottalen Pulse sind aufgrund der großen Fensterlänge nicht mehr erkennbar. Die spektrale Auflösung ist dagegen sehr gut. Abbildung 8 zeigt diese Eigenschaften eines Schmalbandspektrogrammes, Abbildung 9 zeigt die Extraktion eines Spektrums aus diesem Spektrogramm (über die Funktion „To spectrum (Slice)“) aus einem Zeitpunkt in der Mitte des Spektrogramms.

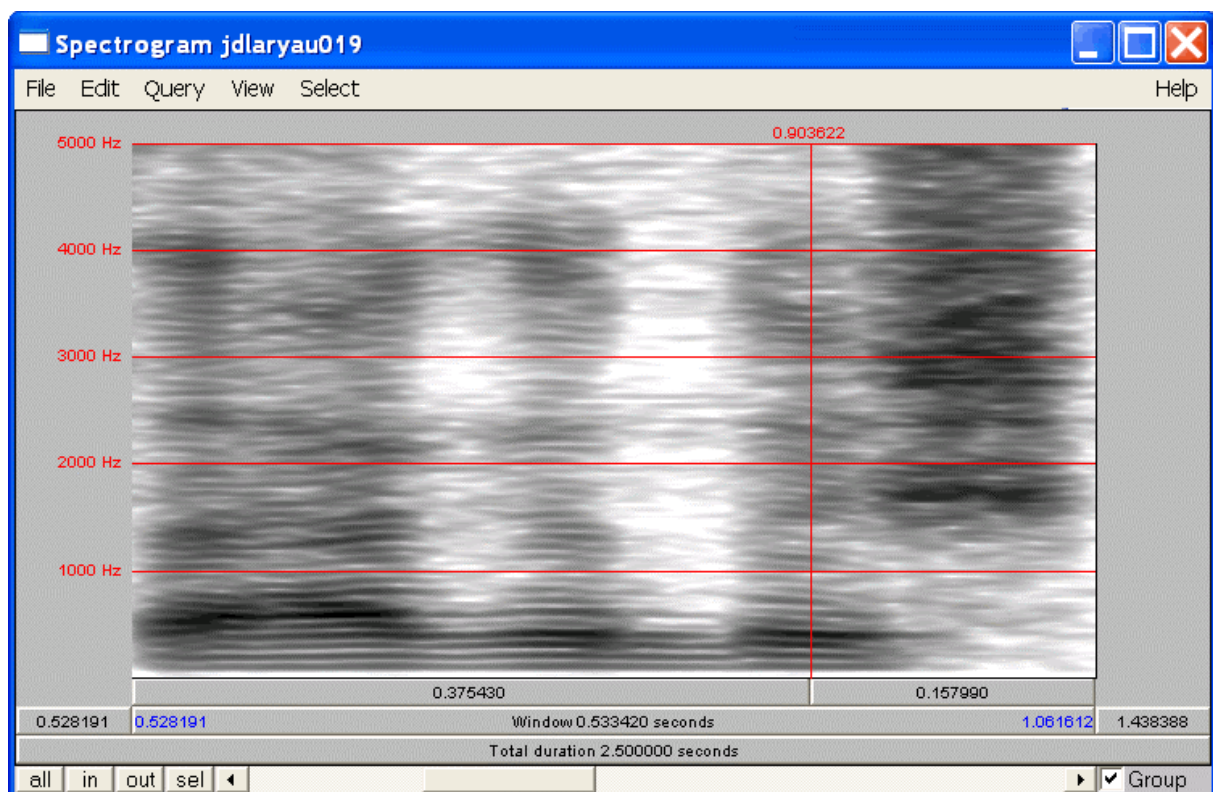


Abbildung 8: Schmalbandspektrogramm eines kompletten Satzes. Die einzelnen glottalen Pulse sind aufgrund der schlechten zeitlichen Auflösung nicht mehr zu erkennen, dafür sind die einzelnen Harmonischen gut dargestellt und leicht zu verfolgen. Der Nachteil ist, dass die Formantstruktur nur noch schlecht erkennbar ist.

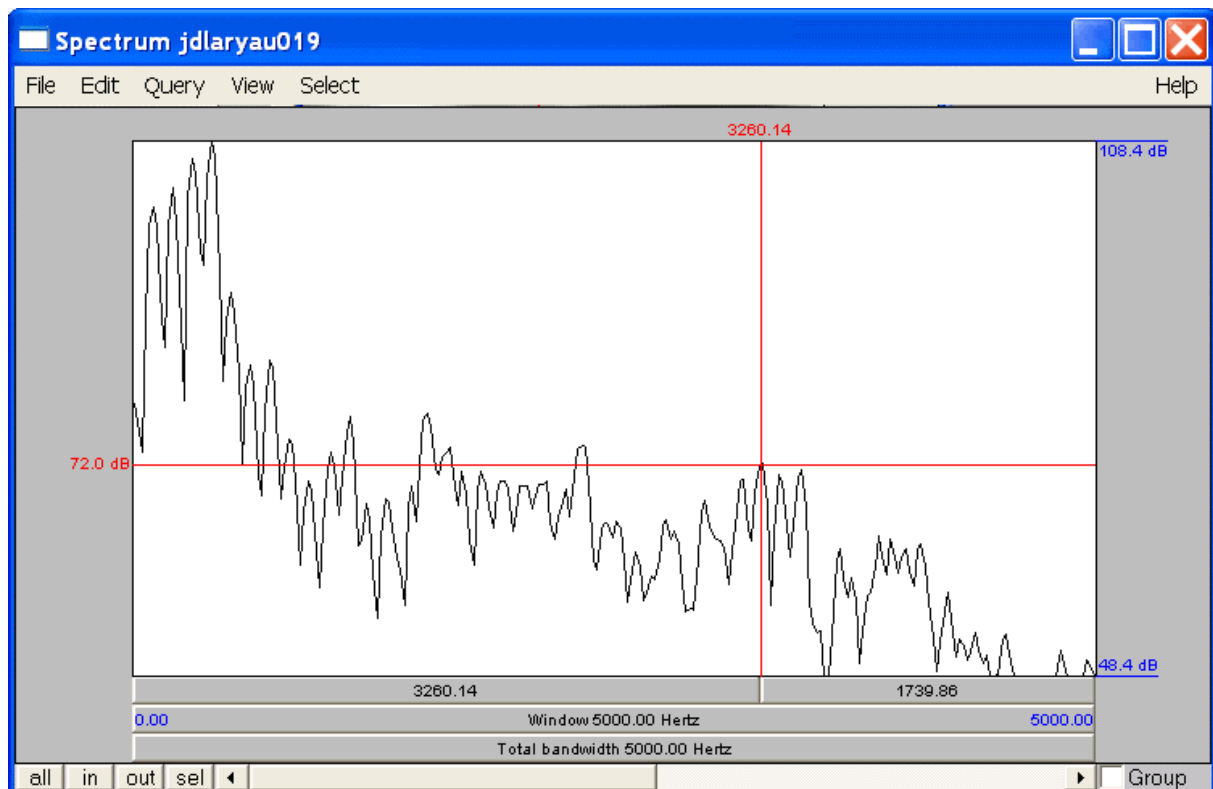


Abbildung 9: Schmalbandspektrum, herausgeschnitten aus der Mitte der vorherigen Abbildung. Es gelten die gleichen Ausführungen wie in der vorherigen Abbildung.

### **LPC (Spektrogramme und Spektren)**

Spektrogramme und Spektren können auch mit der LPC-Methode erstellt werden. Der Vorteil einer Formantmessung über LPC ist, dass die Formantwerte sehr genau sind, allerdings muss man einige Vorüberlegungen treffen, die eine gewisse Erfahrung erfordern. Durch die naturgemäß schlechte spektrale Auflösung von Breitbandspektren können Formantwerte nur mit einer sehr hohen Ungenauigkeit (die in der Ordnung der Fensterbreite liegt) gemessen werden. Allerdings hat die Methode den Vorteil der hohen historischen Verbreitung: Formantwerte wurden seit mehreren Dekaden nach dieser Methode gemessen, damit sind Messungen mit Daten in der Literatur sehr gut vergleichbar. Nichtsdestotrotz ist unbedingt zu beachten, dass Formantmessungen mit der FFT-Methode besagte Ungenauigkeit aufweisen. Werden einige Voraussetzungen beachtet (die PRAAT mit dem Befehl „To formants...“ selbsttätig einstellt), so sind Formantmessungen mit der LPC-Methode um ein Vielfaches genauer: Dem Algorithmus muss mitgeteilt werden, wieviele Pole (Formanten) bei einer bestimmten Abtastrate zu erwarten sind. Ist diese Zahl zu gering, so werden Formanten übersehen, die eigentlich vorhanden sind. Ist die Zahl zu hoch, so werden auch Formanten gefunden, die gar keine sind, also zu viele Formanten gefunden (siehe Vokaltraktmodelle und Resonanztheorie für eine genaue Begriffsklärung des Begriffs „Formant“). Damit ist diese Einstellung im Algorithmus sehr kritisch.

Ansonsten gelten alle Einstellungen und Extraktionen wie im Abschnitt „FFT“ beschrieben.

### ***Pitch-Analyse (Grundfrequenzanalyse)***

Vorangestellt werden soll, dass der Begriff Pitch strenggenommen auditorisch/psychoakustisch belegt ist und auf die Tonhöhenempfindung infolge eines akustischen Reizes hinweist. Trotzdem wird in der Literatur oft irrtümlich der Begriff Pitch mit Grundfrequenz gleichgesetzt, so auch in PRAAT. Im folgenden sollen einige Möglichkeiten der Grundfrequenzanalyse und deren Probleme in PRAAT besprochen werden.

Neben der Formantanalyse (oder auch Spektralanalyse) ist die Grundfrequenzanalyse eines der klassischen Probleme der Signalverarbeitung, sowohl in der musikalischen und akustischen Literatur als auch in der Sprachverarbeitung. Durch die immense Wichtigkeit der beiden Bereiche für die technische Übertragung von Sprache über kodierte Kanäle (z.B. Mobilfunk) und für die Sprachsynthese gibt es zu diesen Bereichen eine Fülle von Literatur und Forschung. Im folgenden soll nur auf die Grundfrequenzanalyse mittels Autokorrelation eingegangen werden, welche in einem Vergleich verschiedenster Methoden überdurchschnittlich gegenüber anderen Methoden bezüglich Genauigkeit und Fehlerrobustheit abgeschnitten hat (Vergleich verschiedener F0-Algorithmen von de Cheveigné).

Alle Grundfrequenzanalysemethoden und ihre Parameter sind über den Periodicity-Button im PRAAT-Objektfenster anwählbar (sofern im Vorfeld ein Sound-Objekt ausgewählt wurde).

Ein klassisches Problem von Grundfrequenzanalysemethoden ist das Auffinden der Stellen, an denen die Grundfrequenz gerade eben noch oder eben nicht mehr messbar ist (stimmhafte vs. stimmlose Stellen im Sprachsignal). Da die Extraktion an diesen Stellen besonders anfällig ist, tritt hier auch die Mehrzahl der Fehler auf. In der Arbeit mit PRAAT ist zu empfehlen, mögliche Fehler zu kontrollieren und gegebenenfalls die stimmlosen Signalstellen im Pitch-Editor wirklich zu löschen (mit dem Befehl „Remove Pitch Points“ nach Anwählen eines Signalabschnittes). Eine weitere Möglichkeit besteht in der Anpassung der Parameter für die Grundfrequenzanalyse („Voicing threshold“ und „voiced/unvoiced cost“), welche die Entscheidung des Algorithmus beeinflussen, ob ein Signalabschnitt als stimmhaft (also Pitch-Werte aufweist) einzustufen ist.

### ***Auditorische Analysen***

PRAAT bietet einige Möglichkeiten, psychoakustische Messungen vorzunehmen, so zum Beispiel auditorische Filter. Es wird hier auf das



PRAAT Manual verwiesen, da dies den Rahmen des Tutorials sprengen würde.

### ***Dauernmessung***

Selbsterklärend ist die Dauernmessung mit PRAAT, mit der Vokallängen, VOT („voice onset time“) und ähnliches bestimmt werden können. Dazu existieren einige Annotationshilfen, „TextGrid“ genannt. Diese können zu einem Soundfile synchronisiert werden und gestatten auf beliebig vielen Ebenen entweder das Labeln von Zeitpunkten oder Intervallen. Im Anschluss können diese TextGrids nach Stichworten durchsucht oder in ein Textfile extrahiert werden.

## 6. Literatur

Johnson, K. (2003). "Acoustic and auditory phonetics", 2. Auflage, Blackwell Publishing, ISBN 1405101229

Neppert, J. (1999). "Elemente einer akustischen Phonetik", 4. Auflage, Hamburg:Buske, ISBN 3875481542.

Oppenheim, A. and R. Schaefer. "Digital Signal Processing", Prentice Hall, ISBN 0132146355

Paulus, Erwin (1998). "Sprachsignalverarbeitung: Analyse, Erkennung, Synthese", Heidelberg, Berlin: Spektrum Akademischer Verlag.

Pompino-Marschall, B. (1995). "Einführung in die Phonetik". Berlin, New York: de Gruyter, ISBN 3110136864.

Reetz, Henning (2003). "Artikulatorische und akustische Phonetik", 2.Auflage, Trier: Wissenschaftlicher Verlag Trier, ISBN 3884766171.

Stevens, K. (1998). "Acoustics Phonetics", Cambridge, London: The MIT Press, ISBN 026219404X.

Wood, Sidney (2005). "Beginners guide to PRAAT", zu erreichen unter: <http://www.praat.org>