Manual zu WaveTools (v 0.9 beta)

Bernhard Ömer

7. November 1994

1 Einführung

Bei WaveTools handelt es sich um eine Sammlung kleinerer Programme zur Bearbeitung und graphischen Darstellung von Tondateien im Wave-Format:

winf zeigt Header- und statistische Informationen über Wave-Files an.

wcat dient zum Aneinanderhängen und zur Konvertierung von Wave-Files.

wcut dient zum Ausschneiden von Sequenzen aus Wave-Files.

wflt führt diverse Transformationen und Filterungen durch.

wfct ist ein Funktionsgenerator.

wmix dient zur Überlagerung mehrerer Wave-Files.

wview dient zur interaktiven, graphischen Visualisierung.

wplot erzeugt ein PostScript-File mit dem Graphen eines Wave-Files.

Alle diese Programme wurden unter Linux 0.99.14 mit dem GNU C-Compiler Version 2.4.5 entwickelt, sollten sich aber, mit Ausnahme von `wview`, das die Linux SVGA-Library verwendet, mehr oder weniger problemlos auch auf andere Systeme portieren lassen. So existiert mittlerweile eine DOS Version, die mit dem Watcom Compiler erzeugt wurde.

1.1 Das Wave-Dateiformat

Das Wave-Format gethört zu den RIFF Multi-Media-Formaten und liegt zur Zeit nur in einer unkomprimierten Version vor, die sich durch ihre einfache sequentielle Anordnung der Daten sehr gut zur automatischen Weiterverarbeitung eignet. Obwohl auch Aufzeichnungen in Stereo vorgesehen sind ist im folgenden nur das Mono-Format beschrieben, da nur dieses von WaveTools in ihrer derzeitigen Version unterstützt wird.

$$
\begin{array}{lcl}
\langle \textit{wave-file} \rangle & = & \texttt{'RIFF'}\ \#\text{wave-block-length}_{32}\ \langle \textit{wave-block} \rangle \\
\langle \textit{wave-block} \rangle & = & \texttt{'WAVE'}\ [\ \langle \textit{info-list} \rangle\ ]\ \langle \textit{format-block} \rangle\ \langle \textit{data-block} \rangle \\
\langle \textit{info-list} \rangle_{opt} & = & \texttt{'LIST'}\ \#\text{info-block-length}_{32}\ \langle \textit{info-block} \rangle \\
\langle \textit{format-block} \rangle & = & \texttt{'fmt '}\ \#\text{format-data-length}_{32}\ \langle \textit{format-data} \rangle \\
\langle \textit{format-data} \rangle & = & \#1_{16}\ \#1_{16}\ \#\text{sample-rate}_{32}\ \#\text{bytes-per-second}_{32} \\
 & & \#\text{bytes-per-sample}_{16}\ \#\text{bits-per-sample}_{16} \\
\langle \textit{data-block} \rangle & = & \texttt{'DATA'}\ \#\text{samples-length}_{32}\ \langle \textit{samples} \rangle \\
\langle \textit{samples} \rangle & = & \{\ \#\text{sample-value}_{8\times bytes}\ \}
\end{array}
$$

Es bleibt noch zu bemerken, daß aus unerfindlichen Gründen die gesamplelten Werte bei einer Wortlänge von 1 Byte `unsigned`, bei größeren Wortlängen aber `signed` sind.

1.2 Konzepte und Konventionen

Das Grundkonzept von WaveTools ist es, durch modularen Aufbau und freie Kombinierbarkeit von kleineren Grundprogrammen nach dem Bausteinprinzip komplexere Transformationen durchzuführen. So erzeugt etwa die Befehlsfolge

```
(wfct 800Hz 3s; wfct -i50% 2Hz 3s) | wmix -m - - | \
wcat -r11025Hz -b8 -oalarm.wav warning.wav +500ms -
```

ein 3 Sekunden langes Sinussignal von 800 Hz, multipliziert es mit Rechteckimpulsen von 2 Hz und einer Breite von 250 ms, und fügt das so erzeugte Alarmsignal nach einer Pause von einer halben Sekunde an den Inhalt der Datei `warning.wav` an, die zuvor noch auf ein Format von 8 bit und eine Sampling-Frequenz von 11025 Hz konvertiert wurde, und schreibt alles in die Datei `alarm.wav`.

1.2.1 Kombination von Programmen

Um die Kombination zu vereinfachen, genügen alle Programme den folgenden Konventionen:

- Ist kein Eingabefile angegeben, so wird der Input von der Standard-Eingabe `stdin` gelesen.
- Sind mehrere Eingabefiles möglich, so ersetzt ein Bindestrich `-` ein File, das von `stdin` gelesen werden soll.
- Alle Programme schreiben erzeugte Files in die Standard-Ausgabe `stdout` um ein "Pipen" des Outputs zu ermöglichen.
- Soll ein Ausgabefile erzeugt werden, so muß dies explizit durch die Option `-o`⟨*filename*⟩ angegeben werden.

- Alle Fehler– oder Statusmeldungen werden in die Fehler-Ausgabe `stderr` geschrieben und daher nicht weitergepiped.

DOS-User müssen ihre Beigeisterung über diese Möglichkeiten leider etwas zügeln, da DOS Binärdateien nicht richtig buffern kann. Zum Auslesen des File-Headers reicht es gerade noch, aber ansonsten seien DOS-User an das Anlegen temporärer Dateien oder an ein vernünftiges Betriebssytem verwiesen.

1.2.2 Optionen und Parameter

Programme wie `wflt` oder `wfct` verfügen über eine breite Palette von optionalen Parametern und Switches um die Standard-Syntax möglichst einfach zu halten (Mit Ausnahme von `wfct` können alle Programme auch ohne Parameter aufgerufen werden.). Optionen und Parameter genügen folgenden Konventionen:

- Die Option `-h` (help) führt zur Ausgabe eine `USAGE`-Meldung in der Syntax und Optionen erklärt werden.
- Die Option `-v` (verbose) führt zur vermehrten Ausgabe von Statusmeldungen.
- Physikalische Größen wie Zeit oder Frequenz erfordern immer die Angabe einer entsprechenden Einheit, die ohne Leerzeichen an den Zahlenwert anschließen muß.
- Bei Eingabefiles ist die Angabe der Extention `.wav` optional.

Wieder schlechte Nachrichten für DOS-User: Während nach den UNIX Konventionen Optionen an jeder Stelle in der Parameterliste auftreten können und zwischen Option und Argument Leerzeichen erlaubt sind, so hat man sich unter DOS streng an die Befehlssyntax zu halten, also die Optionen vor allen anderen Parametern und keine Leerzeichen vor optionalen Argumenten.

1.2.3 Einheiten

Zur Angabe physikalischer Größen stehen folgende Einheiten zur Verfügung:

Dimensionslose Einheiten: **%** (Prozent), **B** oder **b** (Bel) = $\log_{10}$

Phasen und Winkel: **deg** (Grad), **rad** oder **r** (Radianten)

Zeit: **s** (Sekunde)

Frequenz: **Hz**, **hz** oder **h** (Hertz), **f** (Basis-Sampling-Frequenz) = 11025 Hz

Länge: **m** (Meter), **pt** (Punkt) = 0.3516mm (dient zur Angabe der Schriftgröße bei `wplot`)

Allen diesen Einheiten können die dekadischen Multiplikatoren $\mathbf{m} = \frac{1}{1000}$, $\mathbf{c} = \frac{1}{100}$, $\mathbf{d} = \frac{1}{10}$, $\mathbf{k} = 1000$ und für Informatiker $\mathbf{K} = 1024$ vorangestellt werden.

1.2.4 Interne Darstellung von Wave-Files

Unabhängig von der tatsächlichen Auflösung (sample depth) werden die digitalisierten Werte der Wave-Files intern als Fließkommazahlen aus dem Intervall $[-1, 1)$ dargestellt und nur beim Einlesen bzw. der Ausgabe auf die jeweilige Integer-Dastellung umgerechnet. Alle Amplitudenangaben der Programme `winf`, `wview`, `wplot` und `wfct` beziehen sich auf diese interne Darstellung und sind daher für jedes Format gültig. Ist die Option `-v` gesetzt, werden etwaige Überläufe vom Programm gemeldet.

Die Programme `wflt`, `wmix` und `wview` müssen Wave-Files in den Speicher einlesen bevor sie mit der Ausgabe beginnen können. Bei langen Dateien kann es daher bei Systemen mit wenig Speicherplatz zu Problemen kommen. In diesem Fall muß die Eingabe mit `wcut` auf mehrere Dateien aufgeteilt werden, die separat bearbeitet und nachher mit `wcat` wieder zusammengefügt werden können.

2 Verarbeitung von Wave-Files

Zur Erzeugung und/oder Verarbeitung von Wave Dateien dienen die Programme `wcat`, `wcut`, `wmix`, `wflt` und `wfct`.

2.1 Verkettung (`wcat`)

Syntax: `wcat` [*options*] [⟨*wave-file*⟩ | `+`⟨*pause*⟩] ...

Options: `-r`⟨*sample rate*⟩ : Sampling-Frequenz
`-b`⟨*sample depth*⟩ : Auflösung in bit
`-s`⟨*speed*⟩ : Abspielgeschwindigkeit
`-o`⟨*output file*⟩, `-h`, `-v`

`wcat` dient zum Aneinanderhängen mehrerer Wave-Files und zur Formatkonvertierung. Wird mit den Optionen `-r` oder `-b` ein bestimmtes Format vorgegeben, so werden alle Files (und natürlich auch etwaige Pausen) wenn nötig auf dieses Format konvertiert. Ansonsten bestimmt das erste angegebene File das Ausgabeformat.

```
wcat -r1f -b8 -oVoc8.wav Voc16.wav
```

konvertiert z.B. die Datei `Voc16.wav` auf eine Sampling-Frequenz von 11025 Hz und eine Auflösung von 8 bit und erzeugt die Datei `Voc8.wav`.

Wird auf eine höhere Sampling-Frequenz konvertiert, so wird zwischen den einzelnen Abtastpunkten linear interpoliert, wodurch das Digitalisierungsrauschen gesenkt wird. Bei niedrigeren Frequenzen wird über das Abtastintervall gemittelt um höherfrequente Anteile herauszufiltern. Wird auf eine niedrigere Auflösung konvertiert, so werden die Werte gerundet.

Pausen zwischen Files können mit `+`⟨*Dauer*⟩ eingefügt werden. Mit der Option `-s`⟨*speed*⟩ kann die Wiedergabegeschwindigkeit verändert werden, `-s1` bzw. `-s100%` stehen dabei für die Originalgeschwindigkeit. Mit `-s200%` würde demnach eine Datei doppelt so schnell abgepielt. Die Länge der eingefügten Pausen ändert sich allerdings nicht.

```
wcat -s150% -oMickeyMouse.wav Mickey +500ms Mouse
```

spielt die Dateien `Mickey.wav` und `Mouse.wav` mit $1\frac{1}{2}$-facher Geschwindigkeit ab, verbindet sie mit einer Pause von einer halben Sekunde und schreibt das Ergebnis in `MickeyMouse.wav`.

2.2 Ausschneiden (`wcut`)

Syntax: `wcut` [*options*] [⟨*wave-file*⟩]

Options: `-s`⟨*start time*⟩, `-e`⟨*end time*⟩, `-l`⟨*length*⟩
`-o`⟨*output file*⟩, `-h`, `-v`

`wcut` dient zum Ausschneiden einer Sequenz aus einem Wave-File. Von den drei optionalen Zeitangaben dürfen höchstens zwei verwendet werden. Die Angabe kann dabei in Sekunden aber auch als Nummer des jeweiligen Sample-Wertes (ohne Einheit) angegeben werden. Die Aufrufe

```
wcut -s400ms -e800ms -oAm.wav WhoAmI.wav      und
wcut -s4410 -l4410 -oAm.wav WhoAmI.wav
```

sind äquivalent wenn man annimmt, das das File `WhoAmI.wav` mit 11025 Hz gesampled wurde. Werden weniger als zwei Parameter angegeben, so gelten folgende Defaults:

⟨*start time*⟩ = 0 s
⟨*end time*⟩ = Länge des Files

2.3 Erzeugen (`wfct`)

Syntax: `wfct` [*options*] ⟨*frequency*⟩ ⟨*duration*⟩

Options: `-s`⟨*sample rate*⟩ : Sampling-Frequenz (Standard: 11025 Hz)
`-b`⟨*sample depth*⟩ : Auflösung (Standard: 8 bit)
`-a`⟨*amplitude*⟩ : Amplitude (Standard: 1 = 100 %)
`-p`⟨*phase*⟩ : Phasenverschiebung (Standarg 0 rad/deg/%)
`-r` : Rechteckschwingung
`-t` : Dreieckschwingung
`-w` : Sägezahnschwingung
`-n` : Rauschen
`-i`⟨*width*⟩ : Rechteckimpulse (Breite in s oder %)
`-o`⟨*output file*⟩, `-h`, `-v`

Der Funktionsgenerator `wfct` erzeugt Wave-Files periodischer Funktionen mit wählbarer Frequenz und Dauer (beide Angaben müssen als reguläre Parameter übergeben werden.). Ist keine bestimmte Funktion angegeben, so wird eine Sinusschwingung erzeugt. Die Aufrufe

```
wfct -oSinus.wav 440Hz 10ms     und
wfct -t -oDreieck.wav 300Hz 10ms
```

erzeugen die in Abb. 1 und 2 dargestellten Dateien `Sinus.wav` und `Dreieck.wav`.

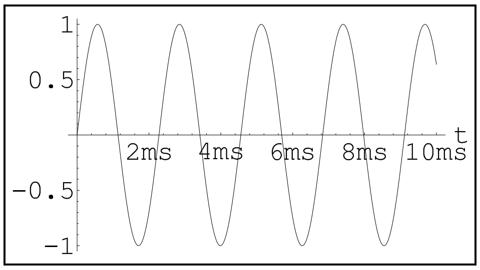

Abbildung 1: `Sinus.wav`

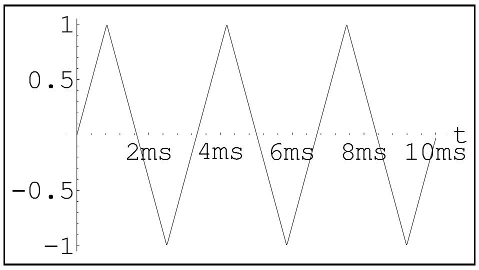

Abbildung 2: `Dreieck.wav`

Das Format der Ausgebedatei kann durch die Optionen `-s` und `-b` festgelegt werden, ansonsten werden die Standardwerte (Sampling-Frequenz = 11025 Hz, Auflösung = 8 bit) verwendet. Werden andere Standardwerte gewünscht, so sind die Einträge in der Datei `defaults.h` entsprechend zu ändern und `wfct` anschließend mit `make wfct` neu zu compilieren.

Normalerweise werden Funktionen mit einer Maximalamplitude von 1 und einer Phasenverschiebung von 0° erzeugt. Mit den Optionen `-a` und `-p` können andere Werte gewählt werden. Die Phase kann dabei in % der Periode oder als Winkel angegeben werden. Die Ergebnisse der beiden folgenden Aufrufe sind in Abb. 3 und 4 dargestellt.

```
wfct -r -p90deg -oRechteck.wav 500hz 10ms
wfct -i2ms -p60% -a70% -oImpuls.wav 100Hz 10ms
```

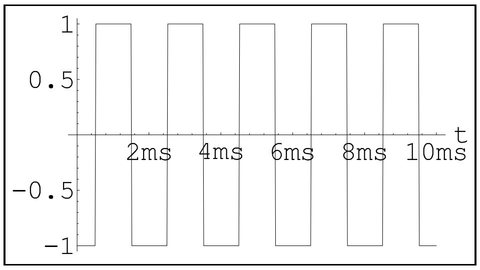

Abbildung 3: `Rechteck.wav`

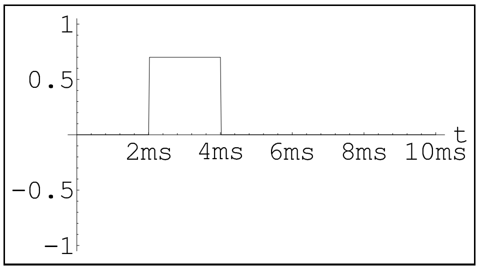

Abbildung 4: `Impuls.wav`

2.4 Mischen (`wmix`)

Syntax: `wmix` [*options*] sexpwave-file ...

Options: `-n` : Normiere das erzeugte File (Setze Maximum auf 1)
`-s` : Multipliziere jedes der n Files mit $\frac{1}{n}$
`-m` : Multiplikation statt Addition der Eingabefiles
`-o`⟨*output file*⟩, `-h`, `-v`

`wmix` dient zum Mischen – genauer gesagt zum Addieren bzw. (mit der Option `-m`) zum Multiplizieren – mehrerer Wave-Files. Wird `wmix` ohne Optionen aufgerufen, so werden alle angegebenen Wave-Files addiert und – falls der Dynamikbereich übersteuert wurde – normiert d.h. die Ausgabedatei wird geeignet multipliziert, sodaß die maximale Amplitude noch darstellbar ist. Mit der Option `-n` kann dies in jedem Fall erzwungen werden. Eine andere Art Übersteuerungen von vornherein zu vermeiden ist die Option `-s` (scale), bei der die Ausgabedatei durch die Anzahl der eingelesenen Dateien dividiert wird. Diese Option ist beim Multiplizieren von Wave-Files mit `-m` unnötig, da hierbei ohnehin keine Überläufe auftreten lönnen. (Es sei daran erinnert, daß in der internen Darstellung alle Amplituden auf das Intervall $[-1, 1)$ abgebildet werden.) Die Aufrufe

```
(wfct 410Hz 1s; wfct 390Hz 1s) | wmix -s - -     und
(wfct 400Hz 1s; wfct -p90deg 10Hz 1s) | wmix -m - -
```

sind, wenn man von Rundungsfehlern absieht, äquivalent und schreiben die gleichen Dateien nach `stdin`, da gilt:

$$\frac{1}{2}(\sin\alpha + \sin\beta) = \sin\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2} \quad \text{und} \quad \sin\left(x + \frac{\pi}{2}\right) = \cos x$$

Die ersten 100 ms der erzeugten Datei sind in Abb. 5 dargestellt.

2.5 Filtern (`wflt`)

Syntax: `wflt` [*options*] [⟨*wave-file*⟩]

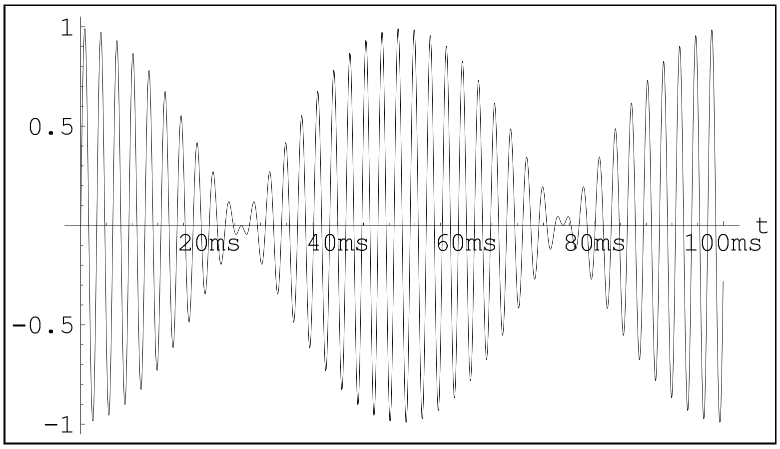

Abbildung 5: **Schwebung**

Options: `-m`⟨*width*⟩ : Mittelwertfilter
`-l`⟨*frequency*⟩ : Tiefpaßfilter
`-f`⟨*frequency*⟩ : Hochpaßfilter
`-b`⟨*frequency*⟩ : Bandpaßfilter
`-w`⟨*frequency*⟩ : Breite des Bandpasses (nur mit Opt. `-b`)
`-g`⟨*gain*⟩ : Verstärkung
`-a`⟨*bias*⟩ : Verschiebung des Nullpunktes
`-c` : Zentrieren der erzeugeten Datei
`-n` : Normalisieren der erzeugten Datei
`-r` : Datei verkehrt abspielen
`-i` : Datei invertieren
`-o`⟨*output file*⟩, `-h`, `-v`

`wflt` ermöglicht eine breite Palette von beliebig kombinierbaren Transformationen, die in der oben angeführten Reihenfolge angewendet werden. Alle Frequenzangaben sowie die Angabe der Breite des Mittelwertfilters können direkt als Frequenz in Hz, als Periodendauer in s, oder als Anzahl der Sample-Werte, die einer Periode entsprechen (keine Einheit), angegeben werden. Für eine Datei mit einer Sample-Frequenz von 11025 Hz sind demnach die Frequenz-Parameter `2205Hz`, `0.4531ms` und `5` äquivalent.

In den folgenden Funktionsbeschreibungen stellt x immer die Datei vor, und y die Datei nach der jeweiligen Transformation dar. Indizes beziehen sich auf die jeweilige Fileposition (x_i entspricht demnach dem i-ten Sample-Wert). Der Differenz-Operator Δ sei definiert als $\Delta x = x_i - x_{i-1}$. Weiters seien f_s bzw. $\omega_s = 2\pi f_s$ und $T_s = \frac{1}{f_s}$ Sampling-Frequenz bzw. Kreisfrequenz und Tastperiode.

2.5.1 Frequenzfilter

Mit Ausnahme des Mittelwertfilters sind alle implementierten Frequenzfilter (Optionen `-l`, `-f` und `-b`) den jeweiligen elektronischen Grundschaltungen (RC, RL und RCL-Glied) nachempfunden und als Differenzengleichungen mittels Rekursion implementiert. Um die Genauigkeit zu erhöhen und um bei Filterfrequenzen größer als $\frac{f_s}{2\pi}$ ein Überschwingen zu vermeiden, wird $\Delta t = \frac{4}{f_s}$ als Schrittweite verwendet und zwischen zwei Sample-Werten linear interpoliert. Nichtsdestoweniger bleiben Filterfrequenzen größer als $\frac{1}{2}f_s$ sinnlos, da diese nicht mehr darstellbar sind (Digitalisierungsrauschen).

In Tabelle 2.5.1 sind die mathematischen Definitionen der einzelnen Filter sowie ihre Diskretisierungen angegeben. $f_0 = \frac{1}{t_0} = \frac{f_s}{n} = \frac{\omega_0}{2\pi}$ steht dabei für die jeweilige Filterfrequenz. Im Falle des Bandpasses (`-b`) gibt $w = \frac{\Gamma}{2\pi}$ die Breite des Bandpasses an; ist diese nicht mit der Option `-w` angegeben so wird sie standardmäßig auf $w = \frac{1}{2}f_0$ gesetzt. Der Wert der Normierungskonstante $N = T_s\Gamma\sqrt{4\omega_0^2 - \Gamma^2}$ ist so gewählt, daß die maximale Verstärkung einer Sinus-Schwingung genau 1 ergibt. Anfangswerte (y_0) und undefinierte Randwerte (z.B. x_{-1}) sind auf 0 gesetzt.

Mittelwertfilter	$y(t) = \frac{1}{t}\int_{t-\frac{t_0}{2}}^{t+\frac{t_0}{2}} x(\tau)\,d\tau$	$\Longleftrightarrow$	$y_i = \frac{1}{n}\sum_{j=i-\frac{n}{2}}^{i+\frac{n}{2}} x_j$
Tiefpaß	$\frac{dy}{dt} = \omega(x - y)$	$\Longleftrightarrow$	$y_i = y_{i-1} + T_s\omega_0(x_i - y_i)$
Hochpaß	$\omega_0 y = \frac{dx}{dt} - \frac{dy}{dt}$	$\Longleftrightarrow$	$y_i = y_{i-1} + \Delta\,x - T_s\omega_0$
Bandpaß	$\frac{d^2y}{dt^2} + 2\Gamma\,\frac{dy}{dt} + \omega_0^2 y = x$	$\Longleftrightarrow$	$\Delta^2\,y = -2T_s\Gamma\,\Delta\,y - T_s^2\omega_0^2 y + N\,x$

Abbildung 6: **Frequenzfilter**

Die folgenden Aufrufe wenden jeden der vier Filter auf die Datei `Impuls.wav` aus Abb. 4 an. Abb. 7 – 10 zeigen die Ausgaben der Aufrufe.

```
wflt -m1kHz -oMeanvalue.wav Impuls.wav
wflt -l200Hz -oLowpass.wav Impuls.wav
wflt -f200Hz -oHighpass.wav Impuls.wav
wflt -b200Hz -oBandpass.wav Impuls.wav
```

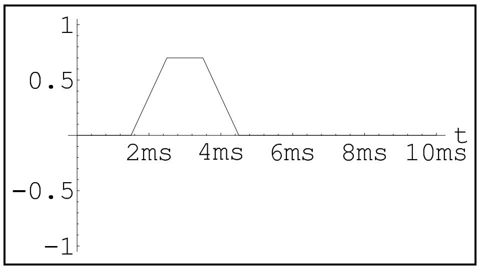

Abbildung 7: **Mittelwertfilter 1 kHz**

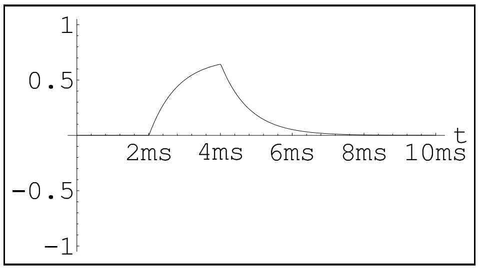

Abbildung 8: **Tiefpaß 200 Hz**

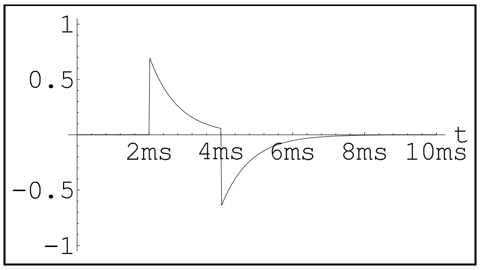

Abbildung 9: **Hochpaß 200 Hz**

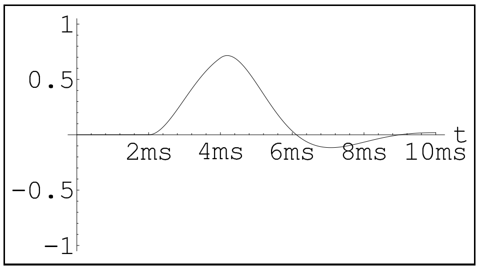

Abbildung 10: **Bandpaß 200 Hz**

Wie anhand der Abbildungen leicht zu erkennen ist, verschiebt der Tiefpaß die Phase nach hinten, der Hochpaß nach vor während sie der Mittelwertfilter unverändert läßt.

Die Frequenzgänge, also die Verhältnisse zwischen Eingangs- und Ausgangsamplituden einer angelegten Sinusschwingung der Kreisfrequenz ω, von Tief-, Hoch- und Bandpaß sind in Tabelle 2.5.1 angegeben.

Tiefpaß	$1 : \frac{1}{\sqrt{1+\frac{\omega^2}{\omega_0^2}}}$
Hochpaß	$1 : \frac{1}{\sqrt{1+\frac{\omega_0^2}{\omega^2}}}$
Bandpaß	$1 : \Gamma\sqrt{\frac{4\omega_0^2-\Gamma^2}{(\omega_0^2-\omega^2)^2+4\Gamma^2\omega^2}}$

Abbildung 11: **Frequenzgänge**

Für den Fall einer Filterfrequenz von 1000 Hz sind die Frequenzgänge in den Abb. 12 – 14 graphisch dargestellt.

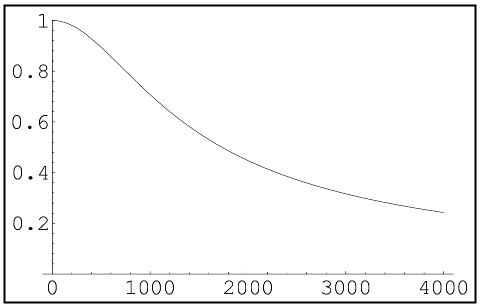

Abbildung 12: **Frequenzgang TP**

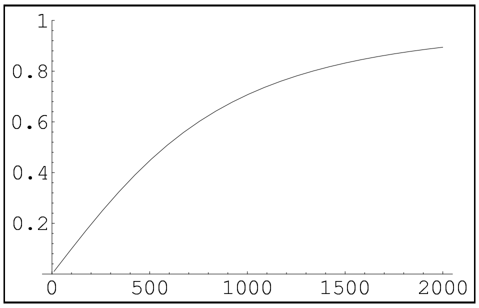

Abbildung 13: **Frequenzgang HP**

2.5.2 Lineare Transformationen

Mittels der Optionen `-g`$\langle gain \rangle$ und `-a`$\langle bias \rangle$ kann eine einfache lineare Transformation der Form $y = gain \times x + bias$ durchgeführt werden. Wird die Option

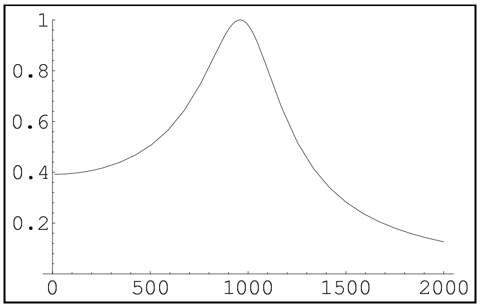

Abbildung 14: **Frequenzgang BP**

`-c` gewählt, so wird ⟨*bias*⟩ automatisch so gewählt, daß der Durchschnittswert $\bar{a} = 0$. Die Option `-y` normiert die erzeugte Datei sodaß $\max y = -\min y = 1$. Werden die Optionen `-c` und `-n` gemeinsam verwendet, so wird der Output bei maximaler Aussteuerung zentriert, also $\bar{a} = 0 \wedge (\max y = 1 \vee \min y = -1)$.

Die Option `-r` vertauscht die Reihenfolge der y_i, spielt also die Datei verkehrt ab. Die Option `-i` negiert alle Werte ($y = -x$).

3 Auswertung und Visualisierung von Wave-Files

Im Gegensatz zu den bisher erwähnten Programmen erzeugen `winf`, `wview` und `wplot` keine Wave-Files, sondern informieren über deren Inhalt.

3.1 Textuelle Information (`winf`)

Syntax: `winf` [*options*] [⟨*wave-file*⟩] ...

Options: `-s` : Statistik, `-h`, `-v`

Ohne Optionen aufgerufen, gibt `winf` die Header-Information der angegebenen Dateien aus (Länge, Sampling-Frequenz und Auflösung). Der Aufruf

```
winf Sinus Schwebung
```

liefert die folgenden Informationen:

```
Sinus    : time 0:00.01 s (  110 values), sampling 11025 Hz 8 bits
Schwebung: time 0:01.00 s (11025 values), sampling 11025 Hz 8 bits
```

Die Option `-s` führt dazu, daß die ganze Datei (nicht nur der Header) gelesen wird und einige statistische Informationen ausgegeben werden.

`winf Dreieck.wav` liefert

```
Dreieck.wav: time 0:00.01 s (110 values), sampling 11025 Hz 8 bits
  Amplitude (x[n])   : min = -1.00000 mid = -0.00426 max = +0.99219
  Step (x[n+1]-x[n]): min = -0.43750 mid = +0.10518 max = +0.43750
  Zero Values        : n =       5  n/sec =  501.1
  Extreme Values     : n =       6  n/sec =  601.4
  Standard Deviation (Volume): s = 0.91054 = 128.8 %
```

`min`, `mid` und `max` in der ersten Zeile liefern den Minimal-, Mittel- und Maximalwert der Amplitude (x). In der zweiten Zeile stehen die gleichen Angaben für die Amplitudenänderung (Δx), wobei bei `mid` über den Bertrag von Δx gemittelt wurde. Die nächsten beiden Zeilen enhalten die Anzahl der Nulldurchgänge und Extremata, sowohl absolut als auch pro Sekunde. Die letzte Zeile enthält noch die Standardabweichung ($\sigma = \sqrt{\frac{1}{n}\sum(x-\bar{x})^2}$) sowie die auf Sinus geeicht Lautstärke ($\sqrt{2}\sigma$) in Prozent.

3.2 Interaktive Graphische Darstellung (`wview`)

Syntax: `wview` [*options*] [⟨*wave-file*⟩]

Options: `-h`, `-v`

`wview` zeigt Wave-Files mittels der Linux SVGA-Library interakiv am Bildschirm an. Der Bildschirm besteht aus drei Teilen, der Kopfzeile mit den Header-Informationen (siehe `winf`), dem Zoom-Fenster mit Raster und Skalierung sowie dem Positions-Fenster, in dem die ganze Datei dargestellt und der gerade gezoomte Teil weiß hervorgehoben ist.

Die Bedienung von `wview` ist denkbar einfach: Die Tasten `Left` und `Right`, `Home` und `End` dienen zum Verschieben des Zoom-Fensters, `Up`, `Down`, `PageUp` und `PageDown` zum Zoomen (`Up` erhöht dabei die Auflösung, `Down` senkt sie). Mit der Taste `Q` oder mit `Ctrl+C` kann `wview` wieder verassen werden (Letzteres ist besonders wichtig, da die VGA-Library beim "Pipen" des Inputs keine anderen Tastendrücke ans Programm meldet.).

3.3 Plotten (`wplot`)

Syntax: `wplot` [*options*] [⟨*wave-file*⟩]

Options: `-s`⟨*start time*⟩, `-e`⟨*end time*⟩, `-l`⟨*length*⟩
`-t` : Echte Zeit-Skala
`-w`⟨*width*⟩ : Breite der Graphik
`-f`⟨*size*⟩ : Schriftgröß in Punkten
`-o`⟨*output file*⟩, `-h`, `-v`

`wplot` ermöglicht durch die Erzeugung von PostScript-Files die Ausgabe von Wave-Dateien am Drucker und das Einbinden von Graphen in Dokumente (wie auch in dieses Manual). Die erzeugten PostScript-Files sind vom selben Format wie die 2-D-Plots von Mathematica (PS-Adobe-2.0), mit dem Unterschied, daß bei von `wplot` erzeugten Files der BoundingBox-Kommentar, der zur korrekten Einbindung von Graphiken in Dokumente benötigt wird, mit der tatsächlichen Größe der Graphik übereinstimmt und nicht mit dem US-Letter Format.

Die Optionen `-s`⟨*start time*⟩, `-e`⟨*end time*⟩ und `-l`⟨*length*⟩ legen den zu plottenden Ausschnitt des Wave-Files fest und funktionieren genauso wie bei `wcut`. Normalerweise beginnt unabhängig von der gewählten Startzeit die Skalierung der Zeit-Achse immer mit 0 s. Soll die tatsächliche File-Position angegeben werden, kann dies mit Option `-t` erreicht werden.

Die Optionen `-w`⟨*width*⟩ und `-f`⟨*size*⟩ legen Breite und Schriftgröße der erzeugten Graphik fest; Abb. 5 wurde zum Beispiel mit dem Aufruf

```
wplot -w10cm -f10pt -l100ms -oSchwebung.ps Schwebung.wav
```

erzeugt, wobei die Angabe der `-w` und `-f` Parameter in diesem Fall überflüssig ist, da sie genau den in `defaufts.h` definierten Standardwerten entsprechen, die aber wie bei `wfct` leicht verändert werden können.