Grundlagen der Spektralanalyse

In der Audiotechnik ist es oft uninteressant, wie der exakte zeitliche Amplitudenverlauf eines Signals ist. Wichtiger ist hingegen, wie dessen Spektrum aussieht, d.h. die Abhängigkeit der Amplitude von der Frequenz. Beispiele dieser Amplitudenspektren zeigt Abbildung 1 einer aus neun Teiltönen angenäherten Sägezahnschwingung (oben) bzw. Rechteckschwingung (unten).

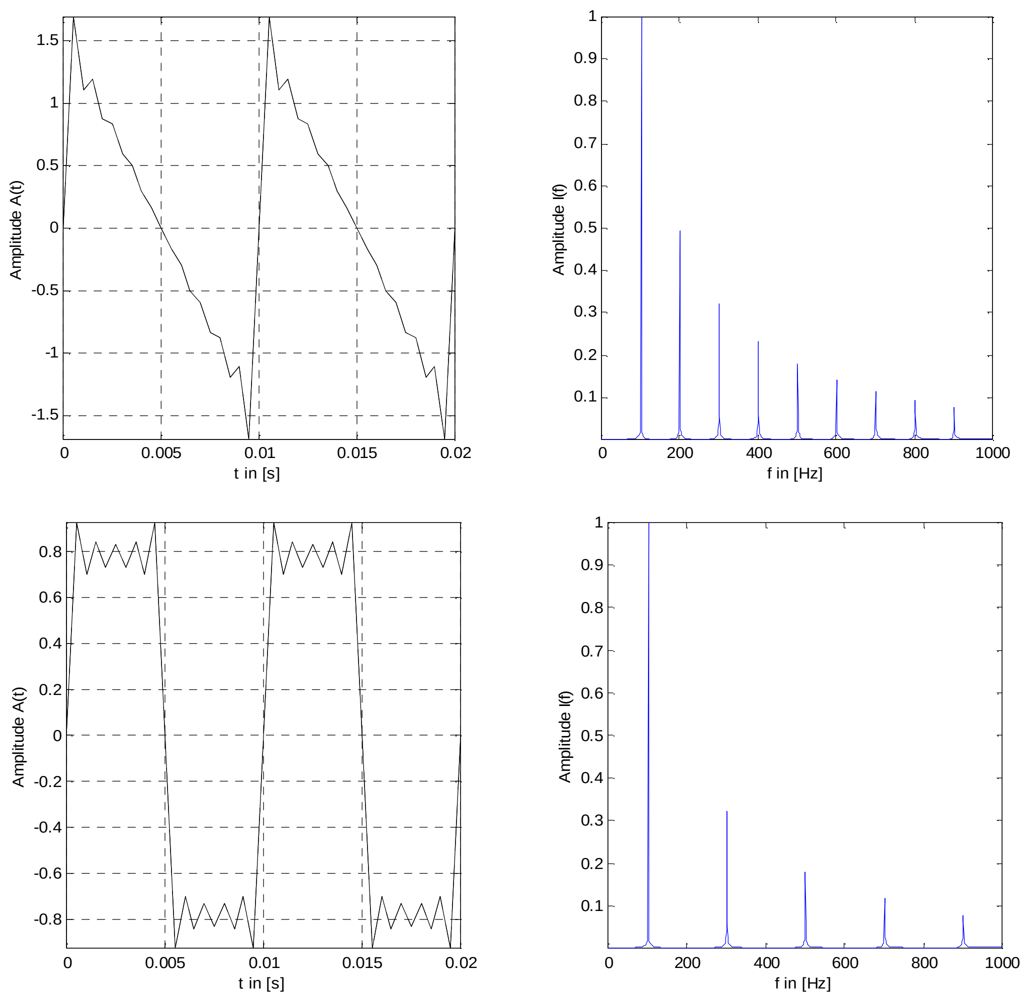

Abbildung 1: Zeitfunktionen von Sägezahn und Rechteckschwingung (links) und dazu gehörige Amplitudenspektren (rechts).

Wie bereits erklärt, lässt sich jede mit T_0 periodische Funktion als Überlagerung von ggf. phasenverschobenen Sinusschwingungen auffassen:

$$x_p(t) = \sum_{i=0}^{\infty} c_i \sin(2\pi \cdot i \cdot f_0 \cdot t + \varphi_i) \quad \textbf{(Formel 1)}$$

Die c_i sind hierbei die Spitzenamplituden der enthaltenen Teilschwingungen, φ_i beschreibt die Phasenverschiebungen der einzelnen Schwingungen. Es lässt sich zeigen, dass man jede Teilschwingung auch genauso gut als Überlagerung von

Sinus- und Kosinusschwingungen gleicher Frequenz <u>ohne</u> Phasenverschiebung, aber mit entsprechenden Vorfaktoren *a* und *b* schreiben kann:
:

$$c_i \sin(2\pi \cdot i \cdot f_0 \cdot t + \varphi_i) = a_i \sin(2\pi \cdot i \cdot f_0 \cdot t) + b_i \cos(2\pi \cdot i \cdot f_0 \cdot t)$$ **(Formel 2)**

Es gilt dabei:

$$c_i = \sqrt{a_i^2 + b_i^2}$$ **(Formel 3)**

Die so genannten Fourierkoeffizienten a_i und b_i, die ausdrücken, wie stark jede Sinus-, bzw. Kosinusschwingung im Signal enthalten ist, lassen sich mit folgenden Integralen berechnen:

$$a_i = \int_0^{T_0} x_p(t) \cdot \sin(2\pi\, i\, f_0 t)dt \qquad b_i = \int_0^{T_0} x_p(t) \cdot \cos(2\pi\, i\, f_0 t)dt$$ **(Formel 4)**

Man kann sich die Integration wie eine Art "Suchfunktion" vorstellen, mit der $x_p(t)$ auf das Vorkommen der Teilschwingungen $\sin(2\pi\, i\, f_0\, t)$ bzw. $\cos(2\pi\, i\, f_0\, t)$ getestet wird. Sind diese nicht enthalten, ergibt sich für a_i bzw. b_i jeweils 0.

In der Praxis lassen sich die Integrale aber nur für wenige Funktionen $x_p(t)$ direkt berechnen. Hinzu kommt, dass die meisten Audiosignale als Messreihe vorliegen und damit die mathematische Funktion für $x_p(t)$ nicht bekannt ist. Außerdem ergibt sich bei der Verarbeitung auf dem Rechner, dass die Audiosignale als diskrete Abtastwerte vorliegen und nicht als kontinuierliche Funktionsverläufe. Bei der Abtastung mit der Abtastfrequenz f_a besteht zwischen einer allgemeinen Zeitfunktion *x(t)*und der aus ihr resultierenden Abtastwertfolge *x(k)* folgender Zusammenhang:

$$x(k) = x(t)\Big|_{t=k\cdot T_a} \quad , \; T_a = \frac{1}{f_a}$$

Damit ergibt sich z.B. für die einfache Sinusfunktion folgende Abtastwertfolge:

$$x(k) = \sin(2\pi\; f_0 t)\Big|_{t=k\cdot T_a} = \sin(2\pi\; f_0 k T_a) = \sin(2\pi \frac{f_0}{f_a} k)$$

Die Abtastung wird in Abb. 2 an Hand einer beliebigen periodischen Funktion *x(t)* verdeutlicht.

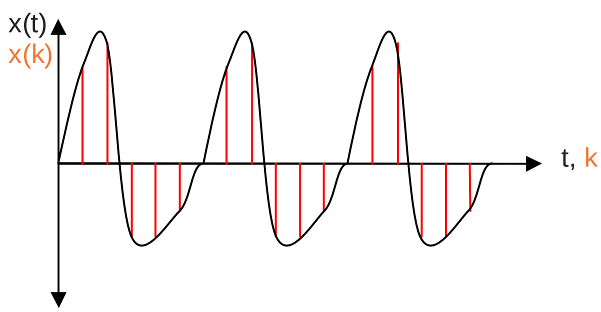

Abbildung 2: Kontinuierliche periodische Funktion *x(t)* und die daraus folgende Abtastwertfolge *x(k).*

Um nun das Spektrum der abgetasteten Funktion *x(k)* zu berechnen, entnimmt man aus der Funktion eine Folge von *N* Abtastwerten (siehe Abb. 3). Die Anzahl *N* bezeichnet man dabei auch als "Analysefensterlänge". Da sich der Klang in einem Audiosignal normalerweise ständig ändert und damit auch das Spektrum variiert, hat es wenig Sinn, z.B. sämtliche Abtastwerte in einer Audiodatei für die Spektralanalyse zu verwenden. Man würde so nur herausfinden, welche Frequenzen im Mittel in einem Musikstück von beispielsweise drei Minuten Länge auftreten, nicht aber wie sich das Spektrum im Laufe der Zeit ändert.

GRAM bietet typische Fensterlängen *(FFT Size)* von 512, 1024, 2048 usw. Abtastwerten an. Die reale zeitliche Länge des Fensters hängt daher von der Abtastfrequenz f_a ab. Bei einem f_a von 5500 Hz entsprächen zum Beispiel 512 Abtastwerte gerade einer Dauer von 93 ms. Da man in der Regel die Periodendauer der abgetasteten Funktion *x(k)* nicht kennt oder diese möglicherweise auch gar nicht periodisch ist, wendet man folgenden Trick an: Man tut so, als würde sich der Signalabschnitt im Analysefenster alle *N* Abtastwerte periodisch fortsetzen (siehe Abb. 4). Es wird deutlich, dass man auf diese Weise nicht das Spektrum der ursprünglichen Funktion *x(k)*, sondern das einer dieser ähnlichen, mit N periodischen Funktion *x'(k)* berechnet.

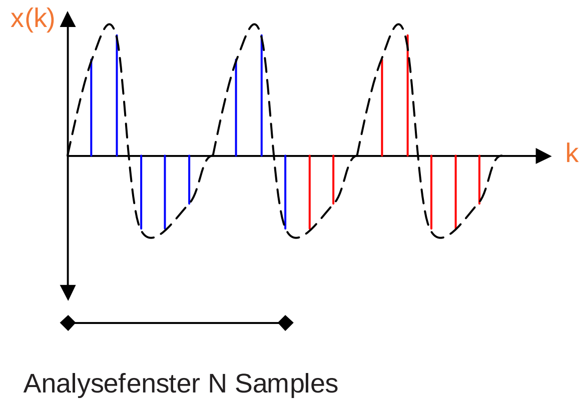

Abbildung 3: Entnahme eines Analysefensters von N Abtastwerten (blau) aus der abgetasteten Schwingung (rot) für die Berechnung des Spektrums.

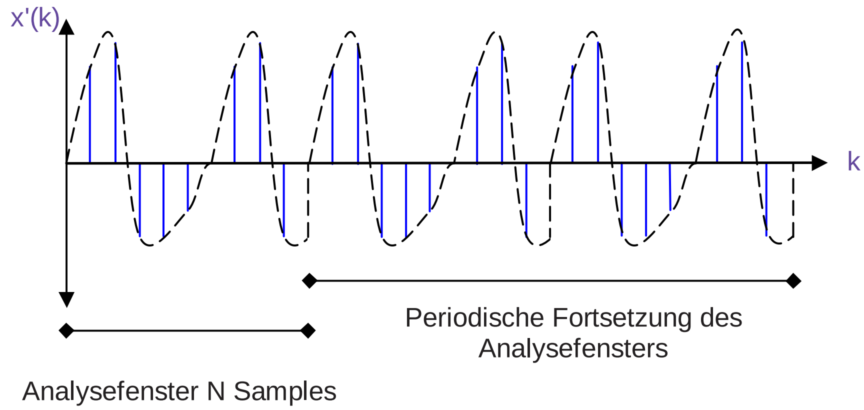

Abbildung 4: Periodische Fortsetzung des Analysefensters und die dabei entstehende, dem ursprünglichen *x(k)* ähnliche Funktion *x'(k).*

Andererseits kann man durch die periodische Fortsetzung eine ähnliche Art der Berechnung anstellen wie für die Fourierkoeffizienten a_i und b_i der kontinuierlichen periodischen Zeitfunktion x(t). Dabei ergibt sich eine Reihe von Vereinfachungen:

Aus einem Integral im Kontinuierlichen (links) wird eine Summation im Zeitdiskreten (rechts):

$$X = \int_0^{T_0} x_p(t)dt \qquad X' = \sum_{k=0}^{N-1} x(k)$$

Das kann man sich folgendermaßen erklären: Das Integral beschreibt bekanntlich bei der kontinuierlichen Funktion x(t) den Flächeninhalt unter der Kurve, wobei positive Abschnitte der Funktion zur Fläche beitragen (+) und negative davon abgezogen werden (Abbildung 5 oben).

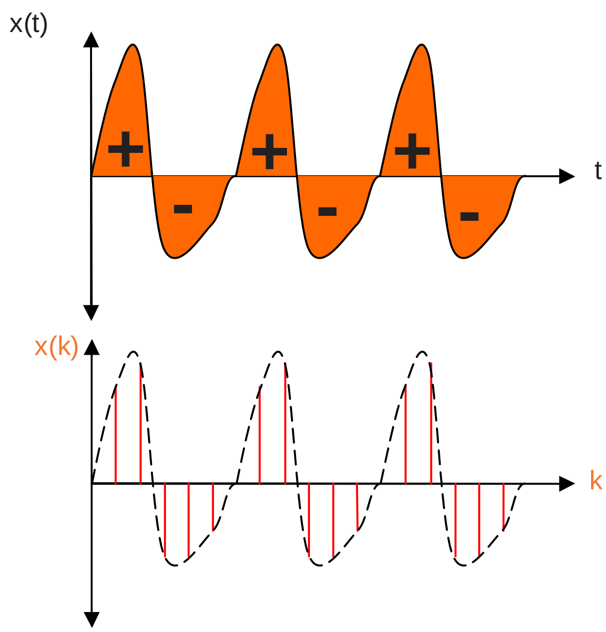

Abbildung 5: Der Flächeninhalt unter der kontinuierlichen *x(t)* (oben) entspricht bei der diskreten Funktion *x(k)* der Summe der Abtastwerte.

Hingegen kann man bei der zeitdiskreten Funktion *x(k)* (Abb. 5, unten) genau genommen gar nicht von einem Flächeninhalt sprechen, denn sie existiert nur zu einzelnen Zeitpunkten. Aus der Integration wird daher einfach eine Summation über die Abtastwerte der Funktion, die natürlich gemäß ihrem Vorzeichen addiert werden.

Wenn wir nun die Fourierkoeffizienten a_i und b_i für die N Abtastwerte von *x(k)* berechnen wollen, vereinfacht sich die ursprüngliche Formel 4

$$a_i = \int_0^{T_0} x_p(t) \cdot \sin(2\pi\, i\; f_0 t)dt \qquad b_i = \int_0^{T_0} x_p(t) \cdot \cos(2\pi\, i\; f_0 t)dt$$

schrittweise folgendermaßen:

Aus $x_p(t)$, $\sin(2\pi i f_0 t)$ und $\cos(2\pi i f_0 t)$ werden durch die Abtastung zeitdiskrete Funktionen:

$$a_i = \int_0^{T_0} x(k) \cdot \sin(2\pi\, i\, \frac{f_0}{f_a} k)dt \qquad b_i = \int_0^{T_0} x(k) \cdot \cos(2\pi\, i\, \frac{f_0}{f_a} k)dt$$

x(k) existiert wie auch die anderen Funktionen nur zu festen Zeitpunkten im Abstand von kT_a. Daher wird aus der Integration eine Summation. Die Periodendauer T_0 der ursprünglichen Schwingung $x_p(t)$ wird dabei durch die Anzahl der Abtastwerte N_0 ausgedrückt, die gerade während einer Periode T_0 gewonnen werden.

$$a_i = \sum_{k=0}^{N_0-1} x(k) \cdot \sin(2\pi\, i \frac{f_0}{f_a} k) \qquad b_i = \sum_{k=0}^{N_0-1} x(k) \cdot \cos(2\pi\, i \frac{f_0}{f_a} k)$$

Nun kann das Verhältnis f_0/f_a auch durch T_a/T_0 ausgedrückt werden bzw. durch $1/N_0$. Anders ausgedrückt: Eine Abtastperiode T_a dauert gerade einen Abtastwert lang, eine Periode T_0 genau N_0 Abtastwerte.

$$a_i = \sum_{k=0}^{N_0-1} x(k) \cdot \sin(2\pi\, i \frac{1}{N_0} k) \qquad b_i = \sum_{k=0}^{N_0-1} x(k) \cdot \cos(2\pi\, i \frac{1}{N_0} k)$$

Bisher gehen wir davon aus, dass für die Spektralanalyse eine Fensterlänge N_0 verwendet wird, die exakt der ursprünglichen Periodendauer T_0 entspricht. Diese ist aber, wie bereits zuvor erklärt, selten bekannt. Wir verwenden daher typischerweise eine Fensterlänge *N*, die nicht N_0 entspricht (Abb. 3), und gehen *fälschlicherweise* von einer periodischen Fortsetzung des Fensterinhalts aus (Abb. 4). Damit wird aus *x(k)* die diesem ähnliche Funktion *x'(k)*. Das berechnete Spektrum entspricht also nicht exakt dem tatsächlichen, stellt aber in der Praxis einen brauchbaren Kompromiss dar:

$$a_i = \sum_{k=0}^{N-1} x'(k) \cdot \sin(2\pi\, i \frac{1}{N} k) \qquad b_i = \sum_{k=0}^{N-1} x'(k) \cdot \cos(2\pi\, i \frac{1}{N} k)$$

Die letzte Formel bezeichnet man auch als *Diskrete Fourier-Transformation*. Diese lässt sich besonders effizient berechnen, wenn *N* eine Potenz von 2 ist und heißt dann Fast Fourier Transform (FFT). Diese wird auch in *GRAM* verwendet.

Der Index *i* bei den Fourierkoeffizienten a_i und b_i drückt den Index des in *x(k)* enthaltenen Teiltons aus, dessen Amplitude berechnet werden soll. Für die Frequenz dieses Teiltons f_i gilt der folgende Zusammenhang:

$$f_i = i \cdot \frac{f_a}{N}$$

Das bedeutet also, angenommen, man tastet ein Signal mit 10000 Hz ab und wählt ein *N* von 100, dann würde man die Fourierkoeffizienten für die Frequenzen 0, 100, 200, 300 Hz usw. bekommen. Verdoppelt man *N* auf 200, wären die Frequenzen 0, 50, 100, 150, 200 Hz usw. Das bedeutet, je länger das Analysefenster ist, desto höher die Frequenzauflösung. Diesen Zusammenhang sieht man auch im Einstelldialog von *GRAM.* Wählt man ein f_a (*Sample Rate)* von 22 kHz und ein *N (FFT Size)* von 1024, dann ist die Frequenzauflösung *(Freq Resolution)* gerade 21.5 Hz, das heißt, man bekommt das Spektrum für die Frequenzen 0, 21.5, 34, 55.5 Hz usw. angezeigt. Die Dauer des Analysefensters ist dabei:

$$T = \frac{1024}{22050 Hz} = 46{,}4\mathrm{ms}$$

Der Nachteil bei einem besonders langen Fenster mit hoher Frequenzauflösung ist allerdings, dass über eine große Menge von Abtastwerten gemittelt wird, während derer sich das Spektrum bereits verändert haben kann. Daher ist es normalerweise günstiger, Fensterlängen von etwa 50-200 ms zu verwenden.

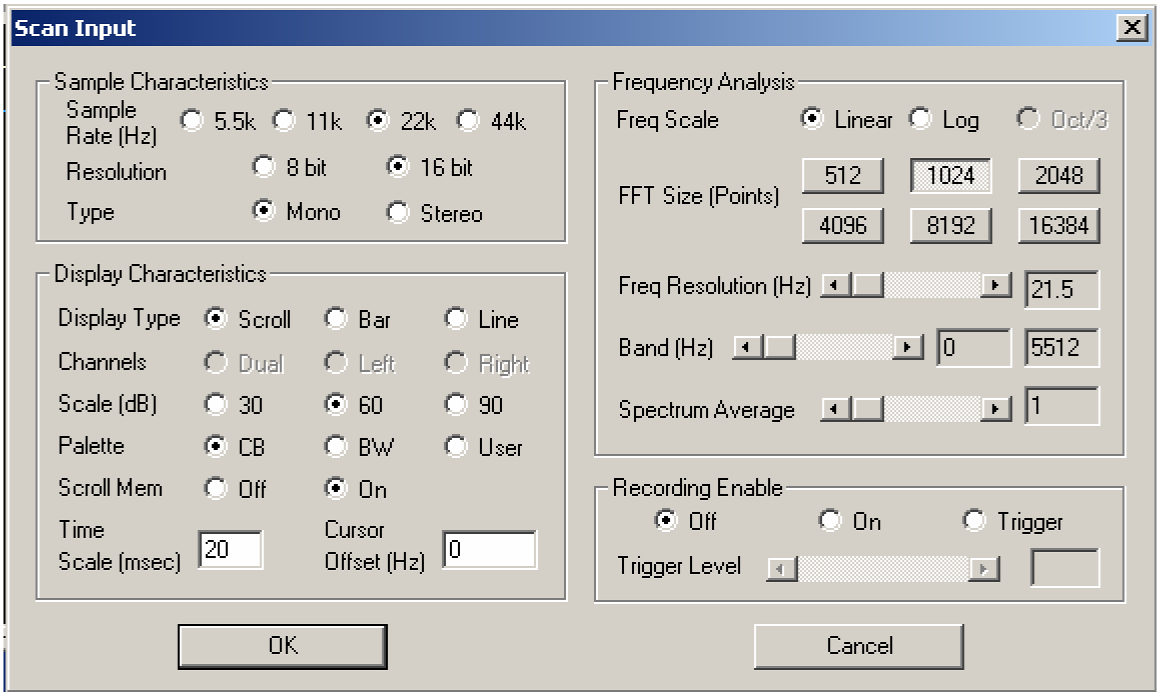

Abbildung 6: Einstelldialog in GRAM: Die Frequenzauflösung (Freq Resolution) ergibt sich aus Abtastfrequenz f_a (Sample Rate) und Fensterlänge N (FFT Size).

Wie bereits bekannt, kann bei einer Abtastung mit f_a nur der Frequenzbereich bis unterhalb $f_a/2$ ohne Auftreten von Aliasing genutzt werden. Das bedeutet, dass auch nur die Teiltöne bis zu dieser Frequenz von Interesse sind:

$$f_i = i \cdot \frac{f_a}{N} \quad , \quad i = 0 \ldots \frac{N}{2} - 1$$

Abschließend die Diskrete Fourier-Transformation als C-Source-Code:

```
for (i=0; i < N/2; i++) //fuer alle N/2 Teiltoene (Frequenzen) i
      {
      a[i]=b[i]=0;
      for (k=0; k < N; k++)//fuer alle N Samples im Analysefenster
            {
            a[i]+=x[k]*sin( (2*pi*i*k)/N );
            b[i]+=x[k]*cos( (2*pi*i*k)/N );
            }
      }
```